L O A D R E C O R D E R

AADIL AHMED WAJA

Thesis submitted in part fulfilment of the requirements for the Master's Diploma in Technology in the School of Electrical Engineering at the Cape Technikon.

CAPE TOWN

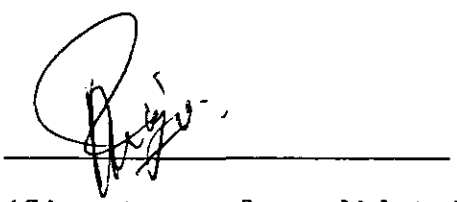
NOVEMBER 1987

DECLARATION

I declare that the contents of this thesis represents my own work and the opinions contained here are my own. It has not been submitted before for any examination at this or any other institute.

AADIL AHMED WAJA

(Name of candidate)



(Signature of candidate)

DEDICATION

This humble effort is dedicated to All Mighty Allah who gave me life, health, wisdom and guidance to complete this thesis. It is also dedicated to my dear father (Ismail) and mother (Zaytoon) for their never ending source of encouragement and interest shown in my work. To my brothers (Waleed and Ziyaad) and sisters (Hawa, Wedaad and Aalayya) for their patience and support I enjoyed throughout my studies. To my grandmothers and the rest of the family who have played a loving supportive role.

CONTENTS

CHAPTER 2. L O A D R E C O R D E R. PAGE

CHAPTER 3. E M U L A T O R. PAGE

A N N E X U R E S.

LIST OF DIAGRAMS.

LIST OF TABLES.

S U M M A R Y

SUMMARY

This thesis describes the development of a computerized Load Recorder. The load recorder is used by the Cape Town City Council to assist in the tariff investigation of electricity consumers. This investigation assists the consumers in selecting the best cost effective electricity tariff. It also motivates the consumers to maintain a constant load which in turn assists the Council in supplying the required electricity.

The load recorder replaces a manual and time consuming method used in conducting the tariff investigation. The development of the load recorder involved the design of hardware and software. It was designed in a compact enclosure to hook up to the electricity meters of the consumer for a period of 7 days. The software was designed in the 68Ø5 assembly language to log data and record the electricity load every 15 or 3Ø minutes for this period. At the end of this period the data is downloaded into the HP85 personal computer.

A basic program was designed for the HP85 to analyse and compute the downloaded data. A graphical representation and analysis is printed by the HP85 computer. The given graph of the results represents the electricity used for this period. The tariff rates are analysed and calculated to determine the best cost effective tariff.

A CMOS micro computer intergrated circuit was chosen due to the determined specification of the load recorder. In order to make the development of the load recorder possible an aid (tool) had to be designed and built for the chosen microprocessor. This development aid, the emulator, is included as part of this thesis.

The Motorola exorciser only supported a cross assembler for the chosen microprocessor family. The emulator was designed and built to enable testing and debugging on the Exorciser. The development on the emulator involved a detailed analysis of the Exorciser development system. The emulator was designed using hardware and software. The hardware emulator board was designed as a standard Motorola size card which plugs into the Exorciser. The software was designed for the 68Ø9 exorciser and for the 68Ø5 emulator. The emulator was soak tested and debugged during the development of the load recorder.

The emulator opened further avenues for future microcomputer design projects especially where a confined area and compactness is an important factor.

The design and development of the emulator and the load recorder was conducted in the Computer Section of the City Electrical Engineer's Department of Cape Town.

OPSOMMING.

Hierdie tesis beskryf die ontwikkeling van ´n gerekenariseerde LAS-OPNEMER. Die LAS-OPNEMER word deur die Stadsraad Kaapstad gebruik om behulpsaam te wees met tarief-ondersoeke van elektrisiteitsverbruikers. Hierdie ondersoek verleen hulp aan die verbruiker om die mees koste effektiewe elektrisiteitstarief te bepaal. Dit motiveer ook die verbruiker om ´n konstante las te handhaaf wat op sy beurt die Raad ondersteun om die nodige elektrisiteit te voorsien.

Die LAS-OPNEMER vervang ´n meganiese en tydrowende metode om tariefondersoek te lei. Die ontwikkeling van die LAS-OPNEMER het die ontwerp van apparatuur en programmatuur behels. Dit was ontwerp in ´n kompakte houer om aan die elektrisiteitsmeter van die verbruiker te koppel vir ´n periode van 7 dae. Die programmatuur is ontwerp in die 68Ø5 masjien taal om gedurende hierdie periode elke 15 of 3Ø minute die elektrisiteitslas data te versamel en te stoor. Aan die einde van die periode word die data oorgeplaas in die HP85 rekenaar.

´n BASIC program is ontwerp vir die HP85 om die oorgeplaasde data te analiseer en te bereken. ´n Grafiese voorstelling en analise word deur die HP85 rekenaar gedruk. Die gegewe grafiek van die resultate verteenwoordig die elektrisiteit wat gebruik is gedurende die periode. Die tarief skale word geanaliseer en bereken om die mees koste effektiewe tarief vas te stel.

´n CMOS mikro-rekenaar geintegreerde stroombaan is gebruik weens die vasgestelde spesifikasie van die LAS-OPNEMER. Ten einde die ontwikkeling van die LAS-OPNEMER moontlik te maak, moes ´n hulpmiddel vir die gekose mikroverwerker ontwerp en gebou word. Hierdie ontwikkelingshulpmiddel, die nabootser, word ingesluit as deel van hierdie tesis.

Die Motorola Exorciser het slegs ´n kruissamesteller vir die gekose mikroverwerker familie ondersteun. Die nabootser is ontwerp en gebou om toetsing en ontfouting in die Exorciser moontlik te maak. Die nabootser ontwikkeling het ´n uitgebreide analise van die Exorciser ontwikkeling-stelsel behels. Die nabootser is ontwerp deur apparatuur en programmatuur te gebruik. Die apparatuur nabootser bord is ontwerp as ´n standaard Motorola stroombaankaart wat in die Exorciser kaartrak pas. Die programmatuur is ontwerp vir die 68Ø9 Exorciser en vir die 68Ø5 nabootser. Die nabootser is getoets en ontfout tydens die ontwikkeling van die LAS-OPNEMER.

Die nabootser het verdere moontlikhede voorsien vir toekomstige mikrorekenaar ontwerp projekte, veral waar beperkte area en beknoptheid ´n belangrike faktor is.

Die ontwerp en ontwikkeling van die nabootser en LAS-OPNEMER is gedoen in die Rekenaarafdeling van die Stad Kaapstad se Departement van die Elektriese Ingenieur.

C H A P T E R 1

I N T R O D U C T I O N

1. INTRODUCTION

1.1. General.

A consumer's electricity consumption is usually measured with Ferranti type rotating disc KWh meter, or if independent of power factor the KVAh meter. These meters are either single or three phase meters depending on 2, 2.5 or 3 measuring elements. The power or rate of energy consumed is proportional to the speed of the rotating disk. The total energy consumed is represented by the total number of revolutions registered on a cyclometer type dial over a given meter period. The highest rate of energy consumption for a demand period (typically 15 or 30 minutes) is the maximum demand. This maximum demand is recorded by measuring the maximum number of revolutions in any demand period. Since each revolution represents an amount of energy consumed, the rotating disc can be fitted with an impulse type contact where each contact closure represents an amount of energy used. Alternatively an optical pickup could be used to detect the rotation from a given indicator on the disc.

1.2. Tariff Structure.

There are basically two types of electricity tariff structures, namely:

1.2.1. General Low Voltage Rate.

This is a one part tariff where the consumer is billed on the total amount of units used (energy consumed) over the whole period.

1.2.2. Large User Low or High voltage rate.

This is a two part tariff where the consumer is billed on the amount of energy consumed and the maximum demand recorded for the period.

A consumer has the option of choosing his tariff structure. The City Council conducts a tariff investigation to assist the consumers in selecting the best cost effective tariff.

1.3. Motivation.

The time consuming and laborious method of calculating the best cost effective tariff was a prime motivating factor for the need of a measuring instrument. Previously the max demand and load factors were determined by manually comparing an ordinary chart readings of a consumers electricity consumption. These values were then used at the end of the test period as a basis for calculating the best cost effective tariff.

In place of the above a computerized data capture recorder was required to assist in these tariff investigations. This unit was aptly named the LOAD RECORDER as it fulfilled the need of the consumer to record the electricity load during a required period. The Load Recorder had to fulfil the following general specifications in order to calculate the best cost effective tariff for the consumer:

i) It had to be equiped with sufficient data storage for a period of 7 days at intervals of 15 or 30 minutes.

ii) This data storage had to be non volatile.

iii) It had to be a battery operated unit completely

independent of the electricity which it recorded.

iv) It had to be portable and compact so that it does not interfere with the consumers activity.

1.4. Design approach.

1.4.1. Type of components.

The power consumption of the load recorder was a determining factor in the size of the load recorder. It was found that the high speed CMOS intergrated circuits were best suited for the load recorder. Only four C type dry cells were ample to cater for the maximum power consumption of this unit.

1.4.2. Choice of microprocessor.

The Motorola MC1468Ø5E2 single chip computer was found to be ideal for the load recorder because of the following factors:

1.4.2.1. It is a compact single chip microcomputer which reduces the overall chip count.

1.4.2.2. Its power consumption is very low, viz:

i) the operating power is 35 milli Watts and

ii) the standby power is 25 micro Watts.

1.4.2.3. Of the two development systems available, the Motorola Exorciser supported a cross assembler for the 68Ø5 microprocessor family, whereas, the Intel system did not have any support for the 8Ø48 and the 8Ø51 series of the single chip microcomputer intergrated circuits.

1.4.3. Need for emulation.

A development system is a computer used to aid in the design and development of a microprocessor application. The functions of a development system are as follows:

i) To link and assemble the application programs.

ii) To test and debug the target system hardware and software.

The process of testing and debugging is known as emulation. During emulation the development system replaces the micro of the target system. The Motorola exorciser did not have any emulation support for the MC1468Ø5E2 microcomputer. In order to make development possible an emulator was designed and built for the exorciser. This emulator is included as part of this thesis.

1.4.4. Emulator.

A complete working knowledge of the motorola exorciser was researched to assist in the design of the emulator. The emulator was developed on a printed circuit board and debugged in the exorciser. The overhead software was designed for the 68Ø9 exorciser as well as this emulator.

1.4.5. Load recorder.

The following procedure was used in the final design of the load recorder:

i) The hardware circuit diagram was designed.

ii) A wire wrap prototype was built.

iii) The software for the load recorder was designed using an algorythm and flowcharts.

iv) With the aid of the emulator the hardware and software was debugged.

v) The load recorder was then designed on a printed circuit board to fit into a confined enclosure.

1.4.6. HP85 personal computer.

This computer is used to give a graphical representation of the power consumption over the test period and thereafter to compute the best tariff structure. A basic program had to be designed for the HP85 to achieve the following functions:

i) Accepting the downloading of data from the load recorder using the RS232.

ii) Storing and computing the downloaded data.

iii) Printing the results as a graphical representation of the electricity consumed and calculating the tariff costs.

1.5. Aim of this study.

The aim of this research study was to become familiar with the following:

i) The design and development of microprocessor controlled applications.

ii) The operation of various microprocessors and peripheral intergrated circuits.

iii) The operation of the exorciser by developing an

emulator.

iv) The basic high level language by designing a program for the HP85 computer.

C H A P T E R 2

L O A D R E C O R D E R

2. LOAD RECORDER.

2.1. DESCRIPTION OF COMPONENTS.

2.1.1. Load Recorder.

The main data logging unit houses a central computer (MC1468Ø5E2); 8K data memory; real time clock; peripheral intergrated circuits; 16 key keypad; 16 by 4 line dot matrix display and 4 C type cells. The input/output communication to the load recorder is conducted via the plugs at the rear end of the unit, that is,

2.1.1.1. The large plug is used for RS232 communication with the HP85.

2.1.1.2. The medium 5-way plug is used for input pulses from the tariff meters.

2.1.1.3. The small sub-miniature earphone jack is the main on/off switch. The entire circuit is disconnected from the batteries when the plug is engaged, thus prolonging the battery life when the unit is shelved.

NOTE: This causes loss of data.

2.1.2. Interface board.

This board consists of screw terminals which are used to interface the three pulsing contacts from the meters to the load recorder. A cable with the 5-way lemo plug is attached to the interface board.

2.1.3. HP85 magnetic tape cartridge.

This tape contains the basic analysis program for the

HP85 personal computer.

NOTE: Downloaded data must not be stored on this cartridge.

2.2. INSTALLATION.

2.2.1. Precautions.

2.2.1.1. Handle the unit with care. Malfunctioning would occur if the unit is dropped.

2.2.1.2. Place the unit in a secure and suitable place for ease of operation.

2.2.1.3. Do not connect any voltage or current to the input pulses (KWh; KVAh and Reset) on the interface board. Connect only voltage free impulsing contacts.

2.2.1.4. Do not use any hard or sharp objects to operate the keypad.

2.2.2. Unit interfaces.

2.2.2.1. Input pulses.

Three voltage free impulsing contacts namely KWh, KVAh and the Reset are attached to the interface board as shown in fig (2.1.). The interface board is connected to the load recorder with the 5-way lemo plug.

2.2.2.2. RS232.

The downloading of data for analysis to the HP85 is achieved at any time during or after the test period, with the RS232 current loop cable from the HP85. This cable plugs into the large socket at the rear end of the load recorder.

2.2.2.3. **Main Power switch.**

When the sub-miniature earphone plug is inserted into the socket it disconnects the power from the load recorder. This operation is mainly used to economise on the batteries when the unit is not used over a long period.

NOTE: The recorded data is lost when the power is disconnected to the load recorder.

DIAGRAM SHOWING INTERFACE WIRING.

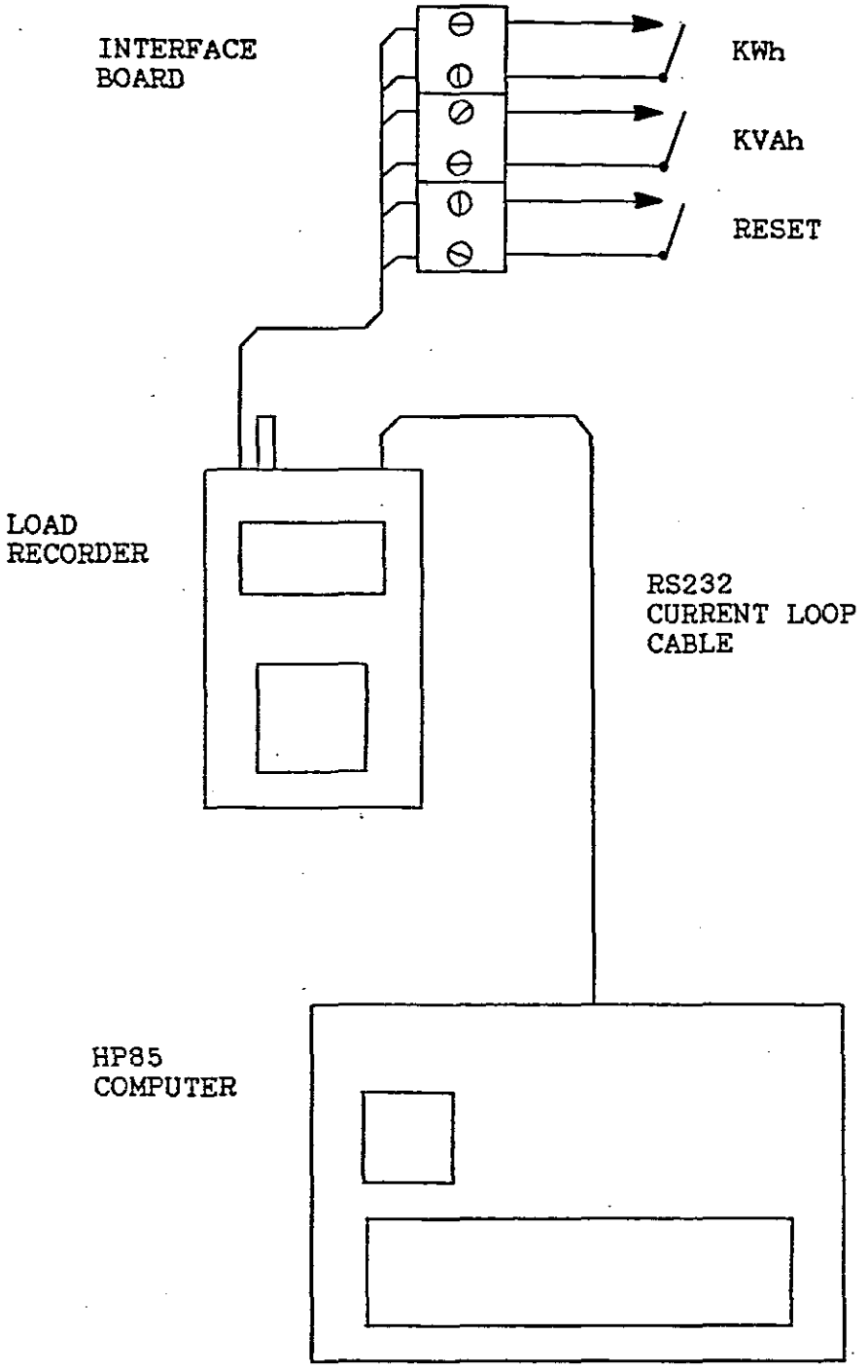


FIGURE 2.1

2.3. OPERATING INSTRUCTIONS.

2.3.1. LOAD RECORDER.

2.3.1.1. Terms and notations.

i) Keypad.

The function of each key is as follows:

KEYS	FUNCTIONS
Ø to 9	numbers Ø to 9
A	CURSER LEFT
B	CURSER UP/DOWN
C	CURSER RIGHT
D	MODE SELECT
E	ENTER
F	ON/OFF

TABLE 2.1 LOAD RECORDER KEYBOARD FUNCTION TABLE.

ii) Period Dur.

Duration in minutes of a demand period which could be changed from 1 minute to 6Ø minutes. The demand periods of 15 or 3Ø minutes are recommended.

iii) Period No.

The demand period is counted from the begining of the day i.e. ØØhØØ. Period Number displayed in Mode 1 is the present period being logged.

iv) Demand pulse factors.

Each demand pulse (contact closure) for a particular demand meter would represent a certain amount of energy. This is represented as KWh/pulse and KVAh/pulse. These factors are obtained by calculating the product of the following constants:

a) Current transformer ratio (CT),

b) Voltage transformer ratio (VT) and

c) Meter pulse constant.

Each consumer installation has a different pulse factor.

v) KWh

This unit represents the amount of true energy.

vi) KVAh

This unit represents the amount of apparent energy.

.3.1.2. ON/OFF Switching.

The load recorder executes start and initializing routines when the main sub-miniature earphone plug is pulled out. This main switch could be used to allow the load recorder to execute a cold restart at any point in time. After a cold restart the unit is in a power saving mode with the display and the keyboard switched off. Pressing the F key switches the display and keyboard on. This brings the unit into an edit and

display mode. Logging of data is unaffected when in the power saving mode. Power saving mode is activated automatically, i.e. after 5 minutes if no key is pressed, or if the F key is pressed again.

2.3.1.3. Display format.

Information is displayed on the load recorder in a page format. A screen of data represents a page which is known as a MODE. There are 7 different pages of information, i.e. MODE 1 to MODE 7.

MODE	INFO. DISPLAYED
1	date & time
2	start of test
3	pulse factors
4	present demand
5	max. demand
6	tot. demand
7	pulse logging

TABLE 2.2 LOAD RECORDER MODE TABLE.

After a cold start the unit displays MODE 1, i.e. date & time. To select a different mode press the D key. The load recorder now prompts for a display mode number. MODE 1 to 7 could now be entered, or stepped through sequentially using the B (curser UP/DOWN) key. To enter into a selected mode the E (ENTER) key is pressed.

Displayed data in MODES 1, 3 and 7 can be changed and edited while display MODES 2, 4, 5 and 6 are only used for displaying information. To edit and change data in MODES 1, 2 and 7 the E (ENTER) key is used. This enables the edit function for a particular mode and is indicated with a flashing curser. Variables could be changed using the curser movement and number keys. The E key is pressed again to finally enter the edited data.

NOTE: It is not possible to enter into a different mode when in the edit function.

The description of the various modes are as follows:

i) Display MODE 1.

Displays the following:

present date and time;

present period number;

period duration.

Date, time and period duration could be changed using the edit function. The period number cannot be changed because it is calculated by the load recorder from the begining of the day.

ii) **Display MODE 2.**

Displays the date, time and period number when the test started. The demand period will be 1 because it will always be the first demand period being logged.

iii) Display MODE 3.

This mode is used to display and change the pulse factors or pulse ratios. Each factor represents a quantity of KWh or KVAh per contact closure. These factors consist of a four digit multiplication part and a one exponent digit division part. These two parts could be changed using the edit function. Any value from 1 to 9999 could be entered for the multiplication part and 1 E Ø to 1 E 9 for the division part. Hence, pulse factors in the range of 1 E -9 to 9999 could be entered. KWh and KVAh have separate multiplication parts but a common division part. These pulse factors could be changed at any point in time (i.e. before, during or after a test) because the data is logged on the number of pulses and not on the quantity of energy.

iv) Display MODE 4.

Displays the present period demand values. These values are the product of the pulse count and the pulse factors.

v) Display MODE 5.

This mode displays the maximum KVA recorded since the commencement of the test. The time and date at which this was recorded is also displayed.
The value displayed is a quantity of power in KVA obtained by multiplying the maximum pulse count

per period with the pulse factor and dividing it by the period duration in hours.

This demand value is reset when a new test is started.

vi) Display MODE 6.

The total energy accumulated since the beginning of the test is displayed for KWh and KVAh. These values are the product of the pulse counts and the pulse factors.

NOTE: No values would be displayed in MODES 4 , 5 and 6 if the pulse factors are set to Ø in MODE 3.

vii) Display MODE 7.

The pulse logging mode which is used to control the activity of load recorder, i.e. data logging, could be started, stopped or continued at any time during the test. The unit enters into a stop mode after a cold restart. A particular option is entered with the edit function and executed after pressing the E key again. The options of the start, stop and continue are as follows:

a) Option 3 - START

It is used to start data logging at the begining of a test. The following occurs after execution:

-Mode 2, start of test is initialized to present time.

-Mode 4 , 5 and 6, variables are cleared to ensure the new maximum demand and the total accumulated values.

-input pulses are now read and logged accordingly.

This option could only be executed after the unit has been stopped.

b) Option 2 - STOP

The logging of input pulses are stopped after execution of option 2 but data and variables are all retained. This option is used when the load recorder is transported to the HP85 for data downloading and analysis, or for temporarily halting of data logging. This option could only be executed after the unit has either been started or continued.

c) Option 1 - CONTINUE

Option 1 is used to continue data logging from where it was halted while retaining the previous data. This option could only be executed after the load recorder has been stopped (option 2).

NOTE:

i) No option can be executed twice simultaneously, eg. the load recorder can not be started again once it has been started. It must first be stopped and then started.

ii) Date and time in MODE 1 must be edited correctly before starting the test in MODE 7.

2.3.2. HP85.

The downloading of data from the load recorder to the HP85 can be done during a test period or after the test period when the load recorder has been stopped.

Before switching the computer on, ensure that the RS232 interface is installed correctly at the HP85 end. Insert the tape cartridge with the load recorder program (LRCD.PRG) into the computer and then switch on. The program will then be loaded and run automatically.

The data to be analysed could be obtained from two sources, namely, the Load recorder or the Files on data disk.

2.3.2.1. Load recorder.

When the option L is entered in the program prompt, the HP85 then request for connection of the RS232 to the load recorder. Before pressing CONTINUE replace the tape cartridge (LRCD.PRG) with a data cartridge. The computer then downloads data from the load recorder into files on the tape. After completing the downloading, the files are computed and analysed.

NOTE: If the load recorder does not respond to the computer after a few attempts, then RESET the HP85 computer and try again.

2.3.2.2. Files on data disk:

When the option F is entered in the program prompt enter the data file name when requested, with the appropriate tape catridge in the computer.

The advantages of storing the data first on tape

are:

i) Sufficiant memory space is needed by the HP85 for analysis.

ii) Results could be obtained at any time without the need of another load recording.

For a detailed description of the HP85 software, flowcharts and listings refer to Appendix G, H and I.

2.4. THEORY OF OPERATION.

2.4.1. HARDWARE.

This circuit was designed using the Complementry Metal Oxide Selicon (CMOS) technology to ensure a low overall power consumption. Refer to the block diagram figure 2.2 on page 2-16 for a brief hardware description.

As shown in this diagram the load recorder consists of three main blocks, namely,

i) The micro computer unit (MCPU)

ii) The external Input/Output

iii) The memory

The MCPU is the heart of the load recorder which consists of a MC146805E2 single chip computer. The MCPU controls, the I/O via the input/output ports and the internal memory via the address/data buss. It also controls the address buss, and the control buss. The I/O is connected to the micro as memory mapped I/O.

The I/O devices which are connected to the ports are slow devices consisting of the following:

i) The RS232 signals

ii) The input pulses from meters

iii) The keypad signals

The memory block consists of the folowing:

i) The real time clock (RTC)

ii) The display

iii) The RAM

iv) The EPROM

The address decoding logic is used to address these devices in the memory map.

The memory capacity needed to store one week of data was calculated as follows:

i) max no. of memory locations needed for 1 demand period was 9 bytes.

ii) minimum demand period duration is 15 minutes

iii) no. of demand periods per day = 24 x 4 = 96

iv) no. of demand periods per week = 96 x 7 = 672

v) no. of memory locations needed per week = 9 x 672 = 6Ø48 bytes

Hence the 8k 6264 RAM chip was found to be most suitable for this application.

A 15 minute demand period stores data for 9 days and a 3Ø minute demand period stores data for 18 days.As the maximum addressing capability of the micro is 8k, a problem was encountered with memory space for the RTC, the Display and the Eprom. To solve this problem a 16k memory map was used in a 2 page mode of 8k each. These two pages are activated using port line PAØ. Page Ø is known as the main memory map and page 1 is known as the data memory map. The main memory map consisted of the internal Ram and I/O of the micro, the RTC, the Display and the Eprom. The data

memory map consisted only of the 8k Ram chip (6264).

The MC146818 is used as a RTC device because it relieves the micro from real time keeping and adds additional ram which is used as a scratch pad during program execution. The RTC generates interrupts to the micro at regular intervals which is programmed by the micro during initialization. The RTC is driven by a seperate clock oscillator because there is no system clock available from the micro during a standby power saving mode (STOP). Hence these interrupts enable the micro to exit the stop mode and service the required information. The above technique was used to reduce overall power consumption.

A stand alone display unit is connected directly on to the address/data buss and control buss of the micro. This unit contains all the necessary hardware to control the 16 character by 4 line display. The display unit is viewed by the micro as two memory locations. One for instructions and one for data.

A 27C64 CMOS Eprom is used to store the source code program for the load recorder. The complete 8k eprom is not available to the micro in the main memory map. The area HØØØØ to HØ1FF consists of the I/O, the RAM, the RTC, and the Display.

A 74HC138 intergrated circuit was used as the main address decoding logic. It is a 1 to 8 line decoder/demultiplexer. The outputs of this chip are used to select the various devices on their chip select pins.

For a detailed hardware description, address decoding and circuit diagram refer to Appendix A, B, and C.

BLOCK DIAGRAM OF THE LOAD RECORDER.

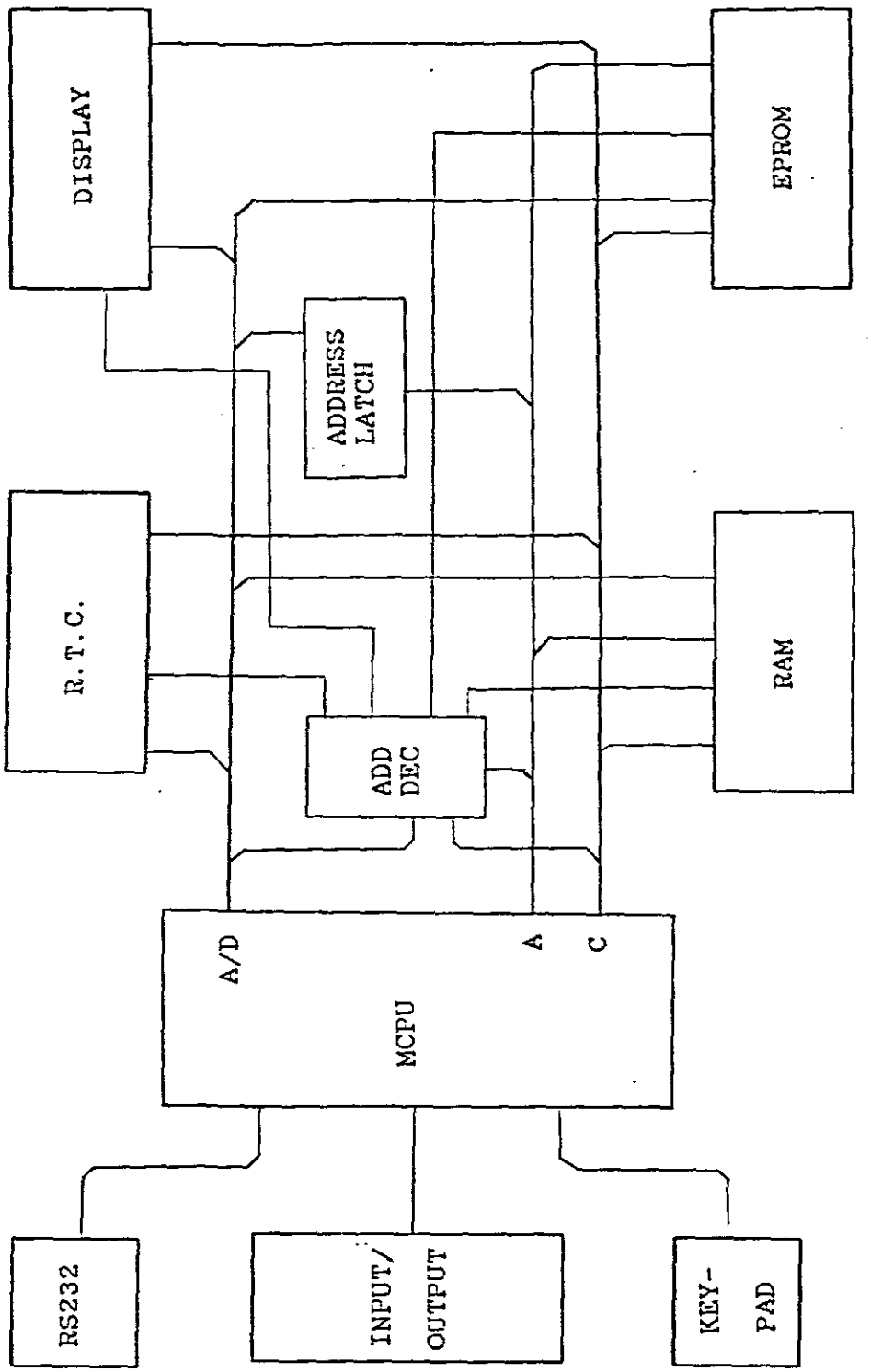


FIGURE 2.2

2.4.2. SOFTWARE

The operating system program is contained in the 8K Eprom situated on the main memory map. Due to memory mapped I/O the bottom 512 bytes of the eprom are unavailable to the micro. The 68Ø5 micro has a limited instruction set and architecture. It has a single 8 bit accumulator, an 8 bit index register, a 7 bit stack pointer which is fixed and cannot be accessed and a condition code register. To achieve the functions required for the load recorder many basic subroutines were designed, for example, a special routine was designed using a carry and add method to achieve decimal adjust. These limitations caused the program to occupy approximately 6k of machine code. The hardware was designed to achieve maximum program space in the main memory map.

The instruction set is similar to the 68ØØ but limited due to CPU registers. There are four bit manipulation instructions on the 68Ø5 which are not accomodated on the 68ØØ, namely, Bit set, Bit clear, Branch on set bit and Branch on clear bit. These instructions are only applicable for the first 125 bytes (i.e. internal I/O and Ram).

The restart and interrupt vectors are situated on the top of the main memory map in the Eprom from H1FF6 to H1FFF. The restart vector at memory locations H1FFE and H1FFF contains the begining address of the program (H4ØØ).

The program commences by initializing I/O, Ram, RTC and Display. It then enters a DO FOR EVER LOOP (the main program loop). This loop consists of a main STOP

instruction, a WAIT instruction and a background task. After initialization the micro enters this loop at the main Stop instruction. An exit from the stop mode can only occur with an interrupt. After servicing an interrupt in the interrupt routine the micro returns to the instructions proceeding the Stop instruction which is the background task. The following routines are executed in the background task:

i) Process input pulses

ii) Process data logging

iii) Check and debounce ON/OFF switch

iv) Control display

v) Execute keyboard functions

vi) Execute RS232 control routine

The interrupts could be serviced at any time during the background task. This task is not dependent upon time as compared to the interrupt routine. In this way continuous processing is achieved while servicing all interrupts simultaneously, that is, data logging, display and keyboard control, and the RS232 downloading can be done without affecting anything.

The background task is terminated by entering into the Stop instruction after all the necessary processing has been achieved. Each subroutine contained in this task has associated DO flags. The subroutine is only executed if the DO flag is set. These flags are enabled by the interrupt service routine. Hence only necessary background task routines are executed.

The micro executes the WAIT instruction in place of the

STOP instruction when the RS232 is activated because a faster response is required for RS232 communication. The RS232 operates at 1200 baud. The recovery time (+-2000 clock cycles) is longer from a STOP mode then a WAIT mode because the clock oscillator of the micro is totally shut down in the STOP mode. The clock oscillator continues to function during a WAIT mode, hence interrupts are serviced immediately.

The micro has two main interrupt signals, namely, the timer interrupt and the external interrupt. When an external interrupt is generated on the INT pin, the interrupt vector in memory locations H1FFA and H1FFB are used. The timer interrupt from the WAIT state vector at H1FF6 and H1FF7 are used when the internal timer generates an interrupt during a WAIT mode. Vectors at H1FF8 and H1FF9 are used when the timer interrupt occurs during a STOP mode or during the program execution. The interrupt signals are serviced according to the following priority:

i) RS232

ii) Keyboard

iii) RTC

On an external interrupt, the devices are polled according to their priority.

2.4.2.1. RS232

The on chip timer is only used during the RS232 communication. Two parallel port lines of the micro are used for the TX and RX signals. The serial information is achieved using the on chip timer and bit

minipulation instructions. During TX the timer is only used to transmit the information. During RX, the external interrupt line is used to detect a start bit and thereafter synchronization is kept by using the on chip timer.

2.4.2.2. Keyboard

A data available signal triggers the INT line when an active key exists. This signal is then polled and read by the micro in the interrupt routine and the binary value of the key being pressed is stored into a register. This key is then processed in the background task after returning from the interrupt routines.

2.4.2.3. RTC

RTC is programmed during initialization to generate interrups every 1Ø ms, 1 sec, and 1 min. These interrupts are identified by reading the contents of memory location H8C (register HC of RTC).

The 1Øms interrupt is programmed as a periodic interrupt PI (bit 6 of register H8C). This interrupt causes the execution of the INPT subroutine in the interrupt routine. This interrupt routine does the debouncing for the three input pulses. Switch closure is first detected at the begining of the subroutine by enabling the current loop, thereafter reading the input and then disabling the current loop. The Make signal is debounced for a duration of 7 counts (7Øms) and the break is debounced for 3 counts (3Øms). These signals

are then stored in the memory and is later processed in the background task.

The 1 sec interrupt is known as the update interrupt, i.e. during this time the RTC executes an update (bit 4 of register H8C). This interrupt is only used to detect a time change in the RTC. Once detected in the interrupt routine the display is refreshed (MODE 1) in the background task to update date and time.

The 1 min interrupt is programmed as alarm interrupt (bit 5 of register H8C). This interrupt is used to count the duration of the present demand period. This feature is mainly used as a backup reset for the external reset pulse. This is to ensure constant demand periods.

Each display Mode is executed by a different subroutine. These subroutines are entered using a DO CASE method, where the CASE is distinguished by the mode number. The mode number selected is stored in a register and the DO CASE MODE acts on this register. Refreshing and updating of the display is executed in the appropriate mode routines. These mode routines are only executed when the display and the keyboard are in the ON state which is controlled by the F key. The ONOFF subroutine in the background task is used to automatically switch the display and the keyboard OFF after 5 minutes if no key is pressed.

For detailed explanation of subroutines refer to the discription, the flowcharts and the software listing in Appendix D, E, and F.

2.5. MAINTENANCE

The only maintenance required on the load recorder is the batteries. Changing of batteries must only be done before commencing a new test to ensure continuous operation for more than one week. Low batteries is indicated when the visibility of the display deminishes. When this happens there is approximately 5 days of operation left.

Note: Changing of batteries invalidates data storage memory.

The procedure to change the batteries is as follows:

2.5.1. Switch the load recorder off by inserting sub-miniature earphone plug into the socket.

2.5.2. The load recorder housing is made up of a top and a bottom half. The keypad is attached to the top half with a ribbon cable connected to the P.C.B. on the bottom half.

2.5.3. Open the load recorder by lifting the top part which clips off.

2.5.4. The four batteries are housed in individual battery holders situated behind the display.

2.5.5. Remove the two small nuts and bolts on the display so that the display could be lifted away from the batteries.

2.5.6. After removing the display, the top battery and holder unplugs by lifting it upwards. Note the direction and place from where the holder is unplugged so that it could be replaced correctly.

2.5.7. Remove and replace the three batteries from their holders on the P.C.B.

2.5.8. Replace the forth battery and holder into position.

2.5.9. Fix display into position and mount into place using the small nuts and bolts.

2.5.10. Fit top cover into place and press firmly to secure clips.

2.5.11. Thereafter follow operating instructions.

2.6. SPECIFICATIONS.

System Components: Load Recorder unit;
HP85 cassette tape.

Supply Voltage: 4 by C type rechargable or dry cells. (approximately 20 days of operation when fully charged.)

Operating Temperature: 0 to 55 degrees celsius
(32 to 131 degrees fahrenheit)

Relative humidity: To 80 percent without condensation.

Dimensions: Load Recorder:

length	188mm
width	110mm
max. height	90mm
min. height	46mm

Mass: 853 grams.

Software features: Load Recorder:

(i) Records two input channels, namely, KWh and KVAh from voltage free contacts made available to it.

(ii) Pulses are recorded and time tagged after each demand period.

(iii) The demand period is initiated by a further reset pulse or by the program.

(iv) Information displayed during test period:

-current time and date and demand period No.

-time and date of commencement of test

-max. demand, tot. demand and present demand

-pulse ratios.

HP85 personal computer:

Load analysis program computes the downloaded data, producing required results.

C H A P T E R 3

E M U L A T O R

3. E M U L A T O R

3.1. DESCRIPTION OF THE COMPONENTS.

3.1.1. MAIN EMULATOR BOARD.

This emulator is designed on a standard Motarola Exorciser board that plugs into the mother board of the Exorciser. On this board is housed the master MC146805E2 micro, 8 k Ram which is shared between the Exorciser and the 6805 micro, a MC14684Ø programmable timer IC, a 2716 EPROM and all the necessary buffers and decoding logic. A ribbon cable connector at the end of the board is used to attach a POD buffer with a cable to an external user system.

3.1.2. MC 146805E2 POD

There are two flat ribbon cables attached to either end of the box. One of these ribbon cables is plugged into the socket of the micro MC146805E2 on the target system. The other ribbon cable with a header is attached to the emulator board.

3.2. INSTALLATION.

3.2.1. Precautions.

3.2.1.1. As the Emulator is a printed citcuit board which is not housed in an enclosure it should be handled with care.

3.2.1.2. Ensure that the emulator board is securely plugged into the Exorciser mother board before switching

on.

3.2.1.3. Ensure the the ribbon cables from the POD is correctly plugged into the emulator board and the target system.

3.2.2. EMULATOR INTERFACES.

3.2.2.1. Exorciser.

The edge connector is the standard Motorola buss connection which could be plugged into any socket in the mother board of the Exorciser. Before inserting the board into the buss ensure that the tracks of the PCB connector are free from dirt.

3.2.2.2. POD.

The POD is connected with one end to the emulator board and the other to the target user system. The power to the user system could be supplied by the emulator board or by an external power supply.

3.2.2.3. Clock frequncy.

During emulation the target system clock frequency is generated by the emulator board. The crystal on the target system is not used.

3.2.3. Target system power.

The +5V of the emulator board is connected permanently to pin 4Ø of the pod. This is used to supply the target system with power. If an external power supply is used then pin 4Ø of the micro on the target system

must be disconnected to avoid a power supply feedback.

3.2.4. Exorciser Hardware Requirements.

3.2.4.1. The 68Ø9 microprocessor control card with a modified Eprom firmware for the emulator board must be used in the exorciser.

3.2.4.2. The 64k Ram card is configured to the executive map for file editing and processing.

3.2.4.3. The 16k Ram card is configured to the user memory map from HCØØØ to HFFFF.

3.2.4.4. The Emulator board is inserted into the mother board.

3.3. OPERATING INSTRUCTIONS.

3.3.1. INTRODUCTION.

The method used by the emulator is known as the microprocessor emulation i.e. the micro of the target system is replaced by the emulator pod. Basically the system can be seen as two microprocessors sharing a common memory map (8k map of the MC1468Ø5E2). The 68Ø9 micro contained in the exorciser is used to edit, modify and update data in the target system memory map with the aid of the XBUG commands. In this way various types of hardware faults could be debugged. Software emulation (or program debugging) is achieved by transferring this 8k memory map to the 68Ø5 micro, which is contained on the emulator. Various commands on the exorciser are used to control the 68Ø5 program execution.

3.3.2. 6805 8k memory map.

This target system memory map is configured to the exorciser USER memory map from H0000 to H1FFF. This memory map consists of, the Ram, I/O and timer of the micro, the 8k emulator ram on the emulator board and the target system memory map. The 8k emulator ram can be configured into various structures depending on the mode of emulation and the target system hardware. This emulator ram area is divided into four parts of 2k each which can be enabled or disabled individually during initialization.

The hexidecimal value written to memory location HFFF0 of the USER memory map in the exorciser is used by the initialization routine to set up the emulator memory structure. Hexidecimal values from H10 to HFF disables the complete 8k memory on the emulator board. The following table represents the relation between the least significant nibble and the 8k memory for value H00 to H0F. Each bit represents a 2k block of memory.

Data bit	Memory block
D0	H200 to H7FF
D1	H800 to HFFF
D2	H1000 to H17FF
D3	H1800 to H1FFF (default)

TABLE 3.1 EMULATOR MEMORY ENABLE.

A logic 1 on the data bit enables the associated memory

block. The highest 2k block is always enabled during initialization for the 6805 emulation software.

3.3.3. Emulation modes.

3.3.3.1. Soft emulation.

The complete memory map consists of the ram contained on the emulator board without the target system hardware been connected. Software subroutines which do not require the hardware are executed and debugged in this mode. The hexidecimal value H0F is stored into the memory location HFFF0 before initialization to enable total emulation ram.

3.3.3.2. Soft and hard emulation.

The memory map of the emulator system consists of the software ram area contained on the emulator board and the hardware contained in the target system. The emulator ram area is enabled according to the design of the target system hardware. The default memory ram area initialized is the top 2k from H1800 to H1FFF. Any value from H01 to H0F could be stored in address HFFF0 before initialization.

3.3.3.3. Hard emulation.

The memory map of the emulated system is only contained in the target system hardware. The ram area of the emulator board is disabled. The target system must contain a memory of at least 256 bytes at the top of the memory map for the

emulator software. To disable the ram area completely the hexadecimal value of HFØ is stored in address HFFO. The emulator 68Ø5 micro runs only on the target system hardware.

3.3.4. Emulator board control and indication.

On the emulator board are two light emitting diodes (Red and Green LEDs) and one push button switch which has the following functions:

i) The green LED indicates that the program is running.

ii) The red LED indicates that the memory is transferred to the 68Ø5 micro.

iii) The push button switch (ABORT) stops the 68Ø5 program execution.

3.3.5. Emulator initialization.

The procedure used to initialize the emulator board is as follows:

i) Switch the power on the exorciser, only after installing the emulator and the target system.

ii) Type USER + (CR) to envoke the user memory map of the exorciser.

iii) Store mode select value in the memory location HFFFØ.

iv) Type BEGN + (CR) to initialize the emulator board.

The BEGN instruction is an added XBUG command used to initialize the emulator board. These commands are subroutines contained in the eprom on the emulator.

3.3.6. Hardware debugging.

After initialization the target system memory map is under the control of the exorciser. All the commands available on the exorciser could be used to edit, debug and modify the hardware of the target system.

The memory locations HØØ to H7F represents the image of the internal memory map of the MC146805E2 micro. This memory map is not directly accessable by the exorciser but only through the image. The following procedure is used to change the internal memory:

i) Modify the required byte to be changed in the image.

ii) Store the address of the byte to be changed in memory location HØC (Change register).

iii) Type CHNG + (CR) to change the internal byte.

The CHNG instruction transfers the memory to the 68Ø5 micro, to change the internal byte from the image. All internal memory is updated to the image before the memory is re-transferred to the exorciser. In this way the CHNG instruction could be used to update the image at any time without entering an address into the HØC location.

3.3.7. Program loading.

Program loading into the target system for the 68Ø5 is identical to the procedure used for a 68Ø9 system. Refer to the 68Ø9 exorterm users guide for

detail loading description.

NOTE: The initialization instruction BEGN must be executed immediately after the program is loaded.

3.3.8. Microprocessor register assignments.

The 68Ø5 microprocessor registers are assigned to the 68Ø9 registers. The table below represents these assignments.

68Ø9	FUNCTION	68Ø5
PC	Program counter	PC
AA	Accumulator A	A
XX	Index register	XX
BB	Accumulator B	Condition code
U	Stack pointer	S

TABLE 3.2 EMULATOR REGISTER ASSIGNMENT TABLE.

The 68Ø5 registers are modified by changing the appropriate 68Ø9 registers.

3.3.9. Break points

A maximum of eight break points could be set in the program. Basically the emulator software inserts SWI instructions into the program to achieve the break points. The procedure used to set a break point is as follows:

i) Initialize the Y register of the 68Ø9 with the

required break point address.

ii) Execute the BPST (Break Point SeT) instruction to record the address into the break point stack.

The procedure used to clear a particular break point is as follows:

i) Initialize the Y register of the 68Ø9 with the required break point address.

ii) Execute the BPCL (Break Point CLear) instruction to clear the address from the stack.

The instruction BPDS (Break Point DiSplay) is used to display the break point stack at any time.

The instruction BPAC (Break Point All Clear) is used to clear all the break points.

3.3.1Ø. Program execution.

The break points and the program counter must first be initialized before the 68Ø5 program is executed. The instruction CONP (CONtinue Program) is used to transfer the memory to the 68Ø5 for program execution. The program is halted when the 68Ø5 micro encounters a SWI break point transfering the memory back to the exorciser. The ABORT push button could also be used to stop the 68Ø5 program.

NOTE: The break point at the present program counter address will not be active after continuing the program because the SWI instruction is not inserted.

3.3.11. Emulator board instruction table.

The following table represents a summary of the emulator board instructions.

INSTRUCTIONS	FUNCTIONS
BEGN	BEGiN instruction for initialization
BPAC	Break Point All Clear
BPCL	Break Point CLear
BPDS	Break Point DiSplay
BPST	Break Point SeT
CHNG	CHaNGe internal memory map
CONP	CONtinue Program

TABLE 3.3 EMULATOR INSTRUCTION TABLE.

3.4. THEORY OF OPERATION

3.4.1. Hardware.

Figure 3.1 on page 3-12 illustrates the emulator hardware block diagram. The emulator board in the exorciser can be viewed as an 8k memory block addressed in the USER memory map from HØØØØ to H1FFF. This 8k memory block is the memory map of the MC146805E2 microprocessor which consists of the ram on the emulator and the hardware on the target system. The emulator board provides all the necessary logic to interface the 6805 memory map to the 6809 exorciser. The 8k memory map consists of the 8k ram chip on the emulator board and the POD which is connected to the target system. Buffer 1 and buffer 2 is used to

multiplex the 68Ø9 (exorciser) and the MC1468Ø5E2 to this 8k memory map. Buffer 2 consists of the octal enable buffers which enables the 68Ø5 micro busses to the 8k memory block. Buffer 1 consists of the octal buffers and control logic which converts and interfaces the non-multiplexed 68Ø9 buss to the multiplexed 68Ø5 buss. The multiplexing of both buffers is controlled by the 68Ø9 exorciser with the B signal which is the I/O available on the emulator board.

The emulator operates in two modes, namely:

3.4.1.1. Target system hardware debug.

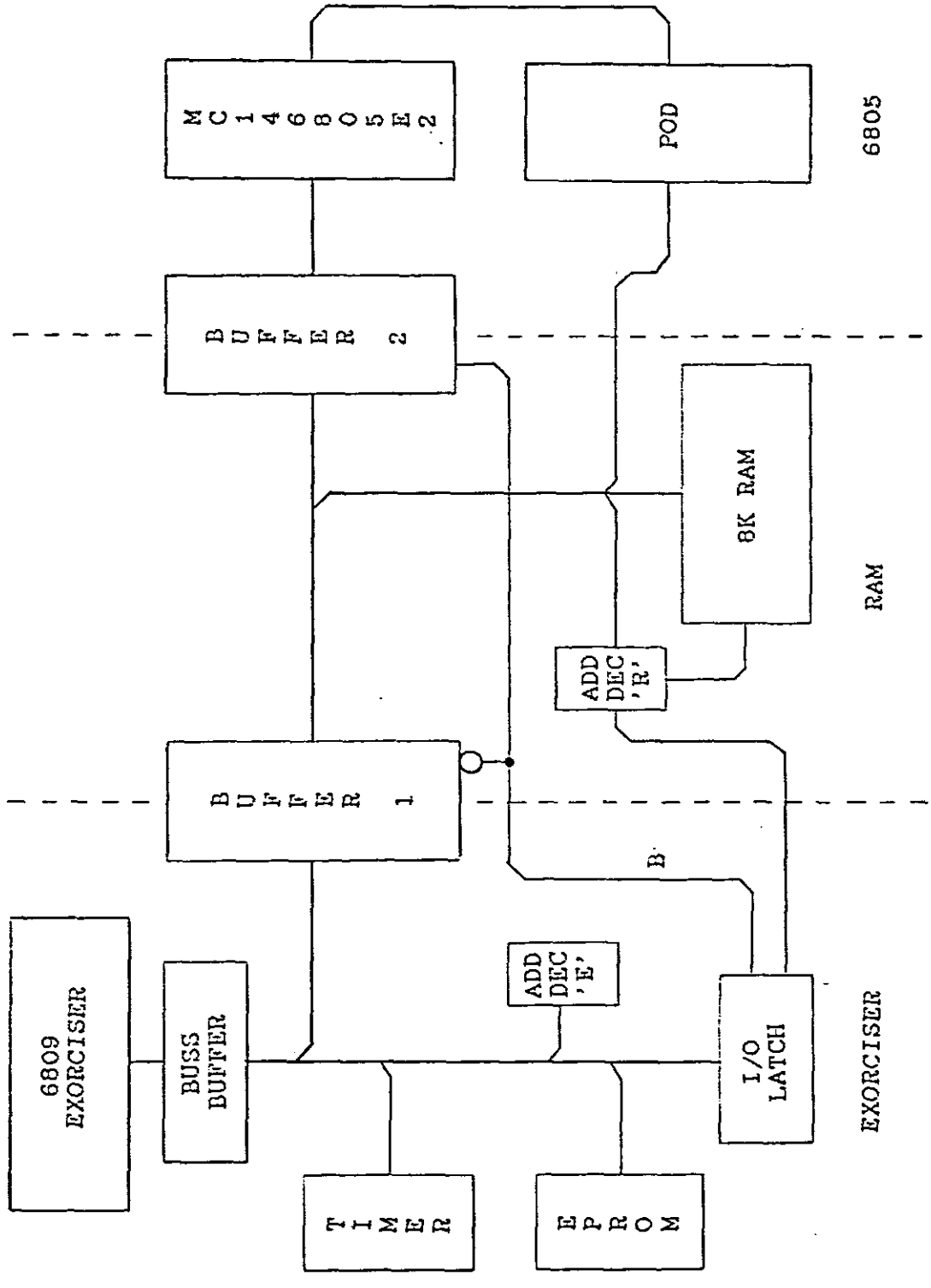
In this mode the target system hardware is debugged by enabling the 8k memory map to the 68Ø9 micro exorciser. To enable the 8k memory map to the 68Ø9, buffer 1 must be enabled and buffer 2 disabled. The 8k memory map of the target system would then be accessable by the exorciser in the user memory map from HØØØØ to H1FFF.

3.4.1.2. Target system software debugging.

The software target system routines to be debugged are loaded into the 68Ø5 memory map when buffer 1 is enabled. The memory map is then transferred to the 68Ø5 micro for execution. The control and timing of the memory map transfer is achieved by the emulator board software in conjunction with the timers and I/O available on the emulator board.

Refer to Appendix J, K and L for a detail hardware description, memory map and the circuit diagram.

FIGURE 3.1 BLOCK DIAGRAM OF THE EMULATOR BOARD



3.4.2. Software.

The eprom on the emulator board contains the software to control the multiplexing and timing of the 8k memory map for the target system software debugging. The eprom contains the 68Ø5 and 68Ø9 control routines. The 68Ø5 routines are loaded by the exorciser into the 68Ø5 memory during initialization.

The 68Ø9 routines are all added XBUG commands which are executed from the command line of the exorciser. The table of the added XBUG commands are located in the eprom at memory location HE14Ø. The exorciser XBUG eprom on the DEBUG board was modified to enable only the BEGN instruction from the table. The memory address HFB61 located in the XBUG eprom at HF2B7/8 was changed to HE14Ø (begining of table). Refer to Annexure 2 pages B-15 and 5-3Ø of the 68Ø9 EXORterm Development System User's guide for initialization and adding of XBUG commands. During the initialization ram routine of the exorciser, the memory address HFFØE - HFF11 is initialized for the BEGN instruction. This enables the BEGN instruction to be executed when the exorciser is switched on. The BEGN routine which is contained in the emulator eprom is used to initialize the entire table into the XBUG ram at memory locations HFFØE to HFF11 and to initialize the emulator board. This BEGN instruction also loads the 68Ø5 routines into the 8k memory map.

Refer to Appendix M, N and O for the detail software description, flowcharts and software listing.

CONCLUSION

4. CONCLUSION.

The City Council of Cape Town conducts a tariff investigation to assist the large consumers in selecting the best cost effective tariff. In this way the consumers are able to moniter their monthly electricity bill by selecting the best tariff.

The tariff investigation is advantageous to the City Council as well as the consumer. The City Council obtains the required electricity for the consumer from ESCOM at a two part tariff rate. This rate decreases when there is a lower maximum demand. To achieve this lower maximum demand, the City Council makes the consumers aware of their maximum demand. This awareness motivates the consumer to save on his monthly electricity bill.

The advent of the load recorder replaces the time consuming and laborious method used by the City Council for this tariff investigation. These tariff investigations are now carried out efficiently, accurately and with minimum delay. The first load recorder tariff investigation was carried out at the Cape Technikon for the period 11/11/87 to 20/11/87. The load recorder was hooked up to the electricity meters of the Technikon at 12h24 on the 11/11/87. The electricity supplied to the Technikon is at a high voltage level and billed on a two part tariff rate. At the end of this test period the data was downloaded from the load recorder into the HP85 computer. The results of this test was computed, analysed and then printed by the HP85. A copy of the results is shown in figure 4.1 and 4.2 on pages 4-6 and 4-7.

4.1. Results.

The information displayed on these results are as follows:

4.1.1. Place of the tariff investigation.

4.1.2. Date and time of test commencement.

4.1.3. The demand period of a 15 minute duration.

4.1.4. The maximum demand recorded in KW and KVA.

4.1.5. The total consumption recorded in KWh and KVAh.

4.1.6. The X-axis of the graph indicates the time.

The Y-axis of the graph indicates the KW, KVA and the power factor (cos ø).

4.1.7. This entire graphic representation is divided up into days, i.e. from DAY 1 to DAY 7 and DAY 1 to DAY 2 of the ensuing week.

4.1.8. The three plots on this graph, namely, KW, KVA and the power factor are indicated here. The top plot indicates the power factor, the middle plot indicates the KVA and the bottom plot indicates the KW.

4.1.9. Calculations.

These calculations are based on a seven day period. In this investigation the HP85 has based the following calculations from the begining of DAY 2 (Thursday), of week 1, to the end of DAY 1 (Wednesday), of week 2.

4.1.9.1. Combined maximum demand 328.9 KVA

4.1.9.2. Consumption for the week 18542.97Ø6 KWh.

4.1.9.3. Power factor Ø.92

This is the average power factor for this period. It is calculated by dividing the total accumulated KVAh into the total accumulated KWh.

4.1.9.4. Load factor 36.5 %

This is the relationship between the average demand to the maximum demand for this period. It is calculated as a percentage by dividing the average KVA by the maximum KVA.

4.1.9.5. Monthly expense for the General rate:

Service charge	R	2.20
First 1500 units	R	219.90
Exceeding units	R	8139.25
Total	R	8361.35

4.1.9.6. Monthly expense for the low voltage large user rate.

Service charge	R	30.80
Demand charge	R	4471.68
Energy charge	R	3604.75
Total	R	8107.23

NOTE: The High voltage large user rate is 7 % less than the above rate of R 8107.23. The Cape Technikon is presently on this user rate.

4.1.9.7. Power factor improvement.

Improving the power factor close to unity would reduce the maximum demand, thus saving on the demand charge. Hence, as this power factor at maximum demand is 0.97, there is no justification for a power factor improvement.

4.2. Improvements on the load recorder.

4.2.1. The load recorder receives the energy signals from voltage free impulsing contacts in the meters. An

optical pickup from an indicator on the disc would improve the installation of the load recorder at a tariff investigation. This optical pickup could be made as an external unit instead of connecting leads to the meters.

4.2.2. An IBM or compatible personal computer could be used to compute and analyse the data from the load recorder instead of the HP85. In this way the results could be improved by using graphic plotters. The various plots on the graph could be indicated in different colours. This data could be stored on floppy disks enabling a greater amount of information to be stored and retrieved faster. This information would also be easily accessable.

4.3. Emulator.

The emulator was designed for the exorciser, in order to make the development of the load recorder possible. The development of this emulator took up more time, greater effort and a greater portion of this thesis. It required more research and development than the load recorder.

The emulator improved the exorciser development system. Previously development was only possible for the 68ØØ and the 68Ø9 microprocessors. The emulator created other avenues for future 68Ø5 microcomputer applications. Computer projects requiring low power consumption, compactness and simpler hardware design could now be developed by the Computer Section of the City Council of Cape Town.

4.4. Improvements on the emulator.

With the present POD, development is only possible for the MC1468Ø5E2 microcomputer. However, different types of PODs could also be developed to connect into the emulator which would make development of other 68Ø5 microcomputers possible.

ELECTRICITY TARIFF INVESTIGATION
RESULTS FOR CAPE TECHNIKON

Start of test period:
87 / 11 / 11 12 h 24

Demand Period: 15 min

Max. demand:
320.5224KW
328.8000KVA

Tot. consumption:
26001.4425KWh
27952.6000KVAh

GRAPH SHOWING
ELECTRICITY
CONSUMPTION FOR
THIS PERIOD

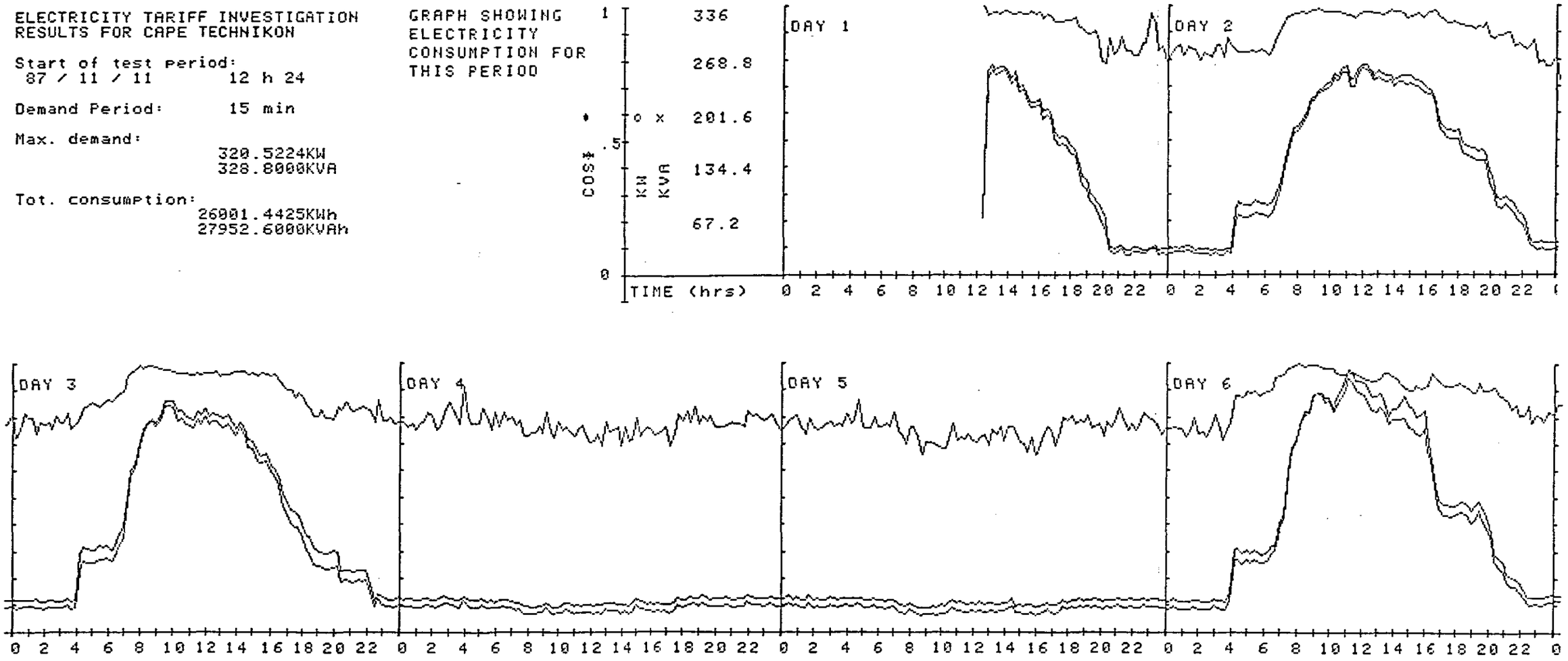


FIGURE 4.1 GRAPHIC REPRESENTATION OF THE RESULTS.

```
Combined maximum demand:
                    328.8000KVA

Consumption for the week:
                   18542.9706KW.h

Power factor:
                  0.92

Load factor:
                  36.5%

MONTHLY EXPENSE FOR:

*****************************

GENERAL RATE:

Service charge:
                        R      2.20

First 1500 units:
                        R    219.90

Exeeding units:
                        R   8139.25

TOTAL:
                        R   8361.35

*****************************

L.V. LARGE USER RATE:

Service charge:
                        R     30.80

Demand charge:
                        R   4471.68

Energy charge:
                        R   3604.75

TOTAL
                        R   8107.23

*****************************

POWER FACTOR IMPROVEMENT:

Power factor at maximum demand:
                  0.97

Improvement of power factor is
not justified.

*****************************
```

FIGURE 4.2 CALCULATED RESULTS.

A P P E N D I X A

L O A D R E C O R D E R
H A R D W A R E D E S C R I P T I O N

APPENDIX A

5. LOAD RECORDER HARDWARE DESCRIPTION.

5.1. MICROPROCESSOR.

The MC1468Ø5E2 is the heart of the load recorder. It is a single chip micro computer unit consisting of, a 68Ø5 based eight bit microprocessor, 125 bytes of on board RAM including stack area addressed at page Ø, two input/output ports (16 I/O lines) where each port could be assigned independently and an eight bit programmable timer with a prescaler. It has an internal clock oscillator/driver. A 5MHZ crystal is used for the oscillator which is divided down to 1MHZ operating frequency. Power on reset is achieved with a 47 kilo ohm resistor and a Ø.1 micro farad capacitor on the reset pin of the microprocessor.

The micro has two power saving modes called the Stop and the Wait modes. Their functions are:

5.1.1. STOP.

Activated after executing the stop instruction. This switches the micro into a halt mode with all clocks and frequencies switched off, reducing power to 25 micro Watts. The microprocessor is then enabled again with an interrupt pulse on the interrupt pin. This mode is always executed when the micro is idle (no program execution).

5.1.2. WAIT.

Activated after executing the Wait instruction. This switches the micro into a wait mode with system clock

still operating. It is a low power mode but it has a faster recovery response to interrupts. The wait mode is only used during RS232 communication with the HP85.

5.2. Input/Output.

I/O, RS232 and the keyboard signals are connected to the port lines of the micro computer. The input pulses KWh, KVAh and Reset are optically isolated for electrical separation and current looping. During pulse logging the opto isolators are pulsed with a 20mA current through the transistor BC237 by port pin PA4. This 100 micro second pulse occurs every 10 milli-seconds under program control. With all three input contacts closed a 60mA pulse would flow, giving a maximum average current for the input circuit:

$$\frac{60\text{mA} \times 0.1\text{ms}}{10\text{ms}} = 0.6\text{mA}$$

Hence a low overall power operation and a high current loop was achieved for the input pulsing contacts. This pulsing method eliminated interference caused by noise. The debouncing of input voltage free contacts is accomplished by the software routines in the program.

The RS232 and keyboard circuitry are only activated when required. A 16 keyboard decoder/debouncer (74C922) was used to decode the matrix keypad. The power to the decoder chip is switched by the microprocessor with the port pin PB5 to disable the keypad in the power saving mode. The F key is seperately wired to the port pin PA2 with a pullup resistor to ensure switch on when the keyboard decoder is switched off. This key toggles the load recorder display

and the keyboard ON and OFF. The matrix keypad is connected to the main board with a 1Ø way ribbon cable onto the printed circuit board plug PL2.

After a valid key has been debounced by the 74C922, it transmits a binary code and a data available (DA) signal. The binary code is connected to the port pins PBØ to PB3 and the DA signal is used to generate an interrupt signal to the micro requesting service.

The RS232 communication is achieved with a 2ØmA current loop. The received data is optically isolated to the micro and is used to generate an interrupt pulse requesting service. TX data is achieved with port PB7 via a transistor for the required current loop.

Real Time Clock (RTC), Display, Ram and Eprom are inter-connected to the microprocessor by the address/data multiplex buss (ADØ to AD7), high order address buss (A8 to A12) and control buss (DS, AS and read/write). These devices are enabled by the address decoding logic.

The load recorder has two addressable memory maps, viz, the Main memory map and Data storage memory map. Both memory maps occupy the maximum addressing space of the micro computer (8k bytes). The active memory map is toggled with the PAØ I/O line. Hence the addressable memory of the micro was increased from 8k to 16k. Memory maped I/O (internal), RTC, Display and Eprom are mapped onto the main memory map. The 8k Ram is mapped onto the data storage map. This map is only used for logging of data. The 8k storage capacity enables the load recorder to log data for 9 to 18 days.

5.3. Real Time Clock (RTC).

The RTC is an accurate clock IC that relieves the micro of the load recorder from keeping time and date. It also has 125 bytes of Ram which is accessed by the micro. The internal ram of the micro and the RTC ram is used as a scratch pad during program operation. An independent external low power crystal oscillator circuit was used to drive the RTC. After initialization the RTC generates 10 ms, 1 sec and 1 min interrupts to the micro. The 10ms interrupts are used to service impulsing contacts, the 1 sec interrupts is used to update time in the display and the 1 min interrupts is used to count duration of the present period. These interrupts are connected to the interrupt pin of the micro. The various interrupt causes are distinguished by reading the control registers in the RTC. During the intervals the micro is in the stop mode. The overall power consumption of the RTC and the micro is much less than the power consumption of the micro if a software real time clock was used. For this reason a seperate clock oscillator for the RTC was used because during the stop mode the micro clock oscillator is disabled.

5.4. DISPLAY.

An Optrix display module was connected to the printed circuit board with plug PL1. It is a 16 by 4 line dot matrix display. The module unit contains all the control, latching, decoding and driving circuitry for the display. The display is connected to the micro as memory map I/O directly onto the data buss and control buss. The display

to the micro is viewed as two memory locations in the main memory map. The memory location H1ØØ represents the display instruction and the memory location H18Ø represents the display data. The register select (RS) line on the display module is driven by address line A7 to decode the instruction and the data registers. The display has many features and is fully programmable. A display of data is activated by sending, the instructions to H1ØØ and the data to H18Ø. Data that is written to the display can also be read by reading memory location H18Ø. A preset potentiometer which controls the voltage Vee is used to vary the viewing angle of the display. The micro drives the display with the normal Read/Write memory cycles i.e. the data, R/W and RS is set up first and then strobed into the display module with an E pulse derived from the data strobe and address decoding.

5.5. EPROM.

Internal I/O and ram, RTC and Display is mapped from HØØØØ to HØ1FF in the main memory map. Memory locations HØ2ØØ to H1FFF is mapped onto a 27C64 Eprom. This eprom contains the operating system program, interrupt vectors and various look-up tables. The chip select signal CS is derived from the address decoding circuit. The data strobe and R/W signals from the micro are NANDed to achieve an Output Enable (OE) strobe, for data read from the 27C64 Eprom. An octal latch (74HC373) was used to demultiplex the address from the address/data lines for the Eprom. The address strobe AS was used to drive the latch enable (LE)

pin on the octal latch.

5.6. RAM

A low power 8k RAM chip (6264) was used for data logging. This Ram chip occupies the total second data storage map. Port pin (PAØ) is used to select the two memory maps.

PAØ = Ø selects Main memory map

PAØ = 1 selects Data storage map

The internal I/O and RAM are the only comman sections between the two memory maps. It is impossible to map the internal section totally out. This small loss of data memory is insignificant. The address demultiplexer for the Ram is achieved with the same 74HC373 latch used for the Eprom. The same OE signal which is used for the Eprom is used for the Ram and a further Write Enable (WE) signal is generated for memory write instruction.

$\overline{OE} = \overline{DS \cdot R/\overline{W}}$ memory read

$\overline{WE} = \overline{DS \cdot \overline{R}/W}$ memory write

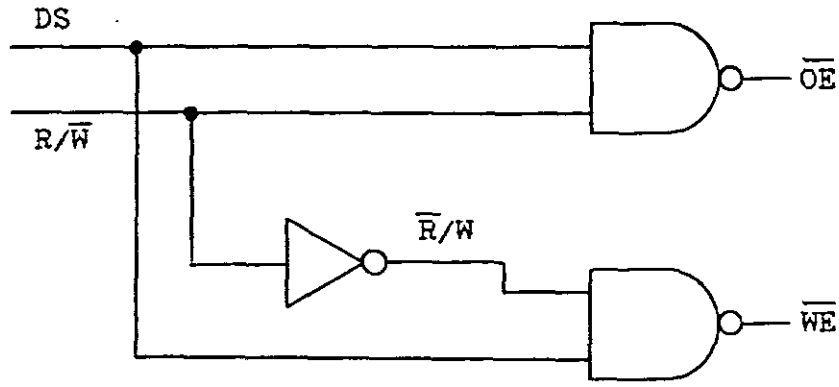


FIGURE 5.1 OE AND WE LOGIC OF LOAD RECORDER.

5.7. ADDRESS DECODING.

The address decode map is derived from the memory map from where the address decoding circuit was designed.
The two memory maps are selected using PAØ pin from the micro. This pin is connected to G2B of U3 and CS1 of U6 (RAM). The ram chip (U6) is enabled when PAØ is a logic 1 and U3 is enabled when PAØ is a logic Ø. An eight input OR/NOR gate U11 (74HC4Ø78) is used to disable both memory maps from HØØØØ to HØØ7F. Pin 13 of U11 (the NOR output) is a logic 1 when any address above H7F is selected and a logic Ø for addresses HØØ to H7F. This output is connected to CS2 of U6 and G2A of U3. Hence when pin 13 of U11 is a logic 1 both intergrated circuits (U3 and U6) are disabled. The logic truth table, below, illustrates the function of PAØ and pin 13 of U11.

PAØ	PIN 13	OUTPUT
Ø	Ø	U3 selected
Ø	1	internal HØØ to H7F
1	Ø	U6 selected
1	1	internal HØØ to H7F

TABLE 5.1 PAGE SELECT TRUTH TABLE OF LOAD RECORDER.

When power is first applied to the circuit, or after a hardware reset, all the I/O lines of the micro are programmed as inputs. This will cause the CS1 of U6 and G2B of U3 to float (not committed to any voltage) before the ports are initialized. During this period the 1 M resistor

forces a logic Ø to ensure that the main memory map is selected.

Pin 1 of the second eight input OR/NOR gate (U9) is used to generate a logic Ø for memory locations HØ to H1FF and a logic 1 for memory locations H2ØØ to H1FFF. This signal with the address line A8 and DS is used to control U3 at inputs A, B and C. The address lines are connected as follows:

i) A8 is connected to A.

ii) Pin 1 of U9 is connected to B.

iii) Data Strobe is connected to C.

The truth table below illustrates the chip select signals for the RTC, Display and Eprom. The outputs of U3 are active low.

C	B	A	OUTPUT
Ø	Ø	Ø	Ø
Ø	Ø	1	1
Ø	1	Ø	2
Ø	1	1	3
1	Ø	Ø	4
1	Ø	1	5
1	1	Ø	6
1	1	1	7

TABLE 5.2 MAIN MEMORY MAP ADDRESS DECODE

TABLE OF LOAD RECORDER.

5.7.1. EPROM

The eprom is selected when DS is a logic 1, and for

memory addresses H2ØØ to H1FFF, that is,

$$CS = DS \cdot B$$

$$CS = DS \cdot B \,(\overline{A8} + A8)$$

$$CS = DS \cdot B \cdot \overline{A8} + DS \cdot B \cdot A8$$

$$CS = C \cdot B \cdot \overline{A} + C \cdot B \cdot A$$

$$\overline{CS} = \overline{C \cdot B \cdot \overline{A}} \cdot \overline{C \cdot B \cdot A}$$

$$\overline{CS} = 6 \cdot 7$$

$$\overline{CS} = \overline{\overline{6 \cdot 7}}$$

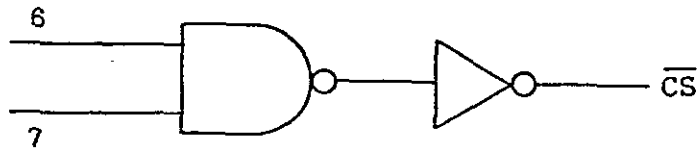


FIGURE 5.2 CHIP SELECT LOGIC FOR EPROM OF LOAD RECORDER.

5.7.2. RTC

The RTC is enabled when memory locations HØ to H1FF is selected AND address line A8 is a logic Ø, for both variations of DS. This enables it to be selected between HØØ and HFF. Since G2A input is disabled for internal memory the RTC is addressed from HØ8Ø to HØBF, that is,

$$CE = \overline{A8} \cdot \overline{B} \cdot \overline{DS} + \overline{A8} \cdot \overline{B} \cdot DS$$

$$\overline{CE} = \overline{\overline{A} \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot C}$$

$$\overline{CE} = Ø \cdot 4$$

$$\overline{CE} = \overline{\overline{Ø \cdot 4}}$$

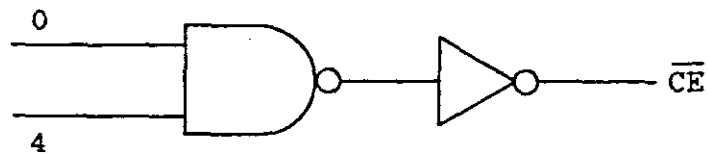


FIGURE 5.3 CHIP SELECT FOR RTC OF LOAD RECORDER.

5.7.3. Display

The display is selected in the memory region from H1ØØ to H1FF (A8 is a logic 1) and when DS is a logic 1, that is,

$$E = A8 \cdot \overline{B} \cdot C$$

$$E = A \cdot \overline{B} \cdot C$$

$$E = \overline{5}$$

The address line A7 is used for the register select (RS) line of the display module to select between the instruction and the data registers.

The data strobe (DS) control line is used in the address decoding logic to generate a strobe chip select signal for the display and the eprom. This strobe pulse decreases power consumption because the eprom is only selected when data is read into the micro.

A P P E N D I X B

L O A D R E C O R D E R
M E M O R Y M A P A N D
A D D R E S S D E C O D E M A P

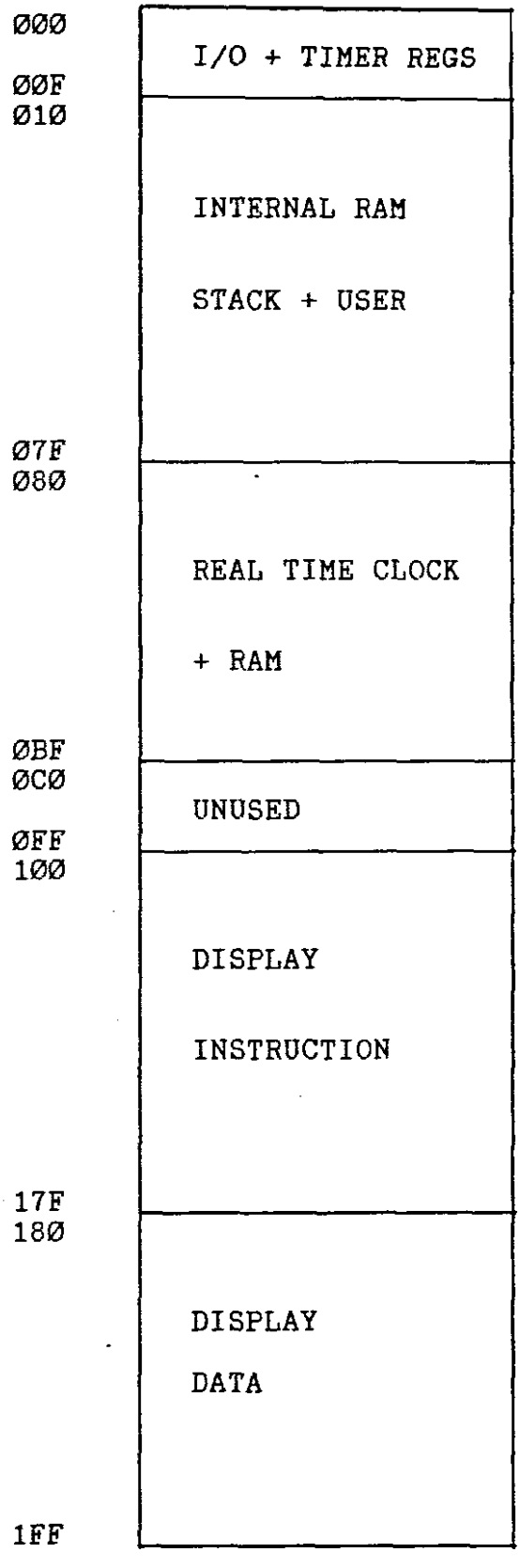


FIGURE 6.1 LOAD RECORDER HØØØ TO H1FF OF MAIN MEMORY MAP.

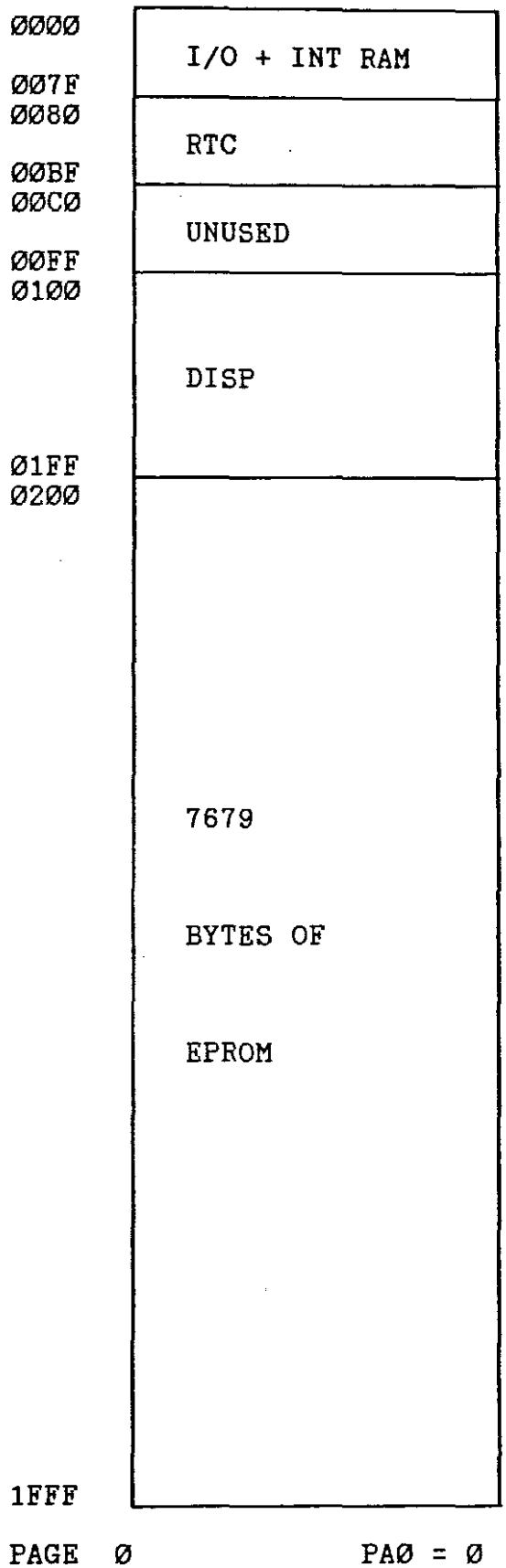


FIGURE 6.2 MAIN MEMORY MAP OF LOAD RECORDER

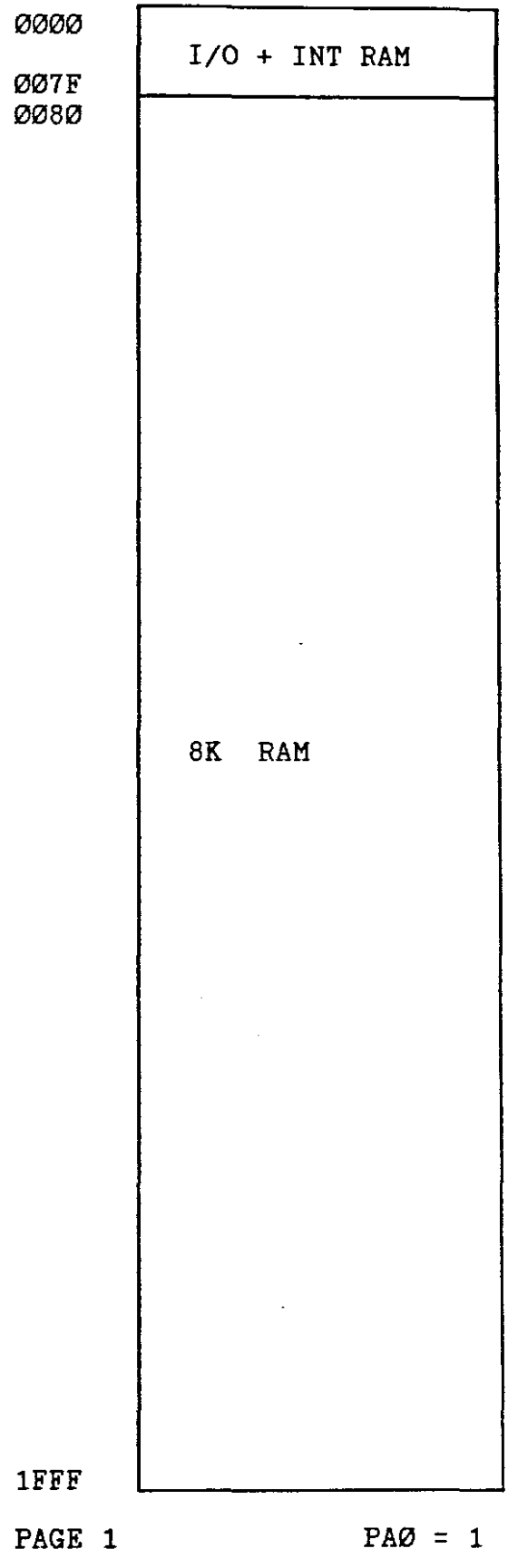


FIGURE 6.3 LOAD RECORDER DATA STORAGE MEMORY MAP.

DEVICE	PAØ	R/W	DS	A12	A11	A1Ø	A9	A8	A7	A6-AØ	IC NO.
RTC	Ø	.	.	Ø	Ø	Ø	Ø	Ø	1	.	IC 7
DISP	Ø	.	1	Ø	Ø	Ø	Ø	1	.	X	EXT
EPROM	Ø	1	1	*	*	*	*	.	.	.	IC 5
8K RAM	1	.	.	*	*	*	*	*	.	.	IC 6

Ø - Logic zero

1 - Logic one

. - Both logics

X - Don't care

* - Any combination of ones

FIGURE 6.4 LOAD RECORDER ADDRESS DECODE MAP

A P P E N D I X C

L O A D R E C O R D E R
C I R C U I T D I A G R A M

CIRCUIT DIAGRAM OF LOAD RECORDER

CITY OF CAPE TOWN ELECTRICITY DEPARTMENT	DRAWN			SCALE	REV.
D/O MANAGER.	COMPILED	S.AUSTIN	87.10.22	FIG.	1
	INFO.	A.A.WAJA	87.10.14	SK3892	
C.E.E.: F.L.U.DANIEL	CHECKED				

A P P E N D I X D

L O A D R E C O R D E R
S O F T W A R E D E S C R I P T I O N

APPENDIX D

8. LOAD RECORDER SOFTWARE DESCRIPTION

This section contains a detail description of each subroutine.

8.1. MAN H40Ø

Routine: Hardware reset (cold start).

Routines called: MRS; CONT; SSS; SPRES; ONOF; JJT; KKT; DBUS; RTCR

Description: Main DO FOR EVER routine. Initialization of RAM, I/O, Display and RTC is first executed and then the DO FOR EVER loop is entered. This loop consists of a main STOP instruction at label MAN7, a WAIT instruction between MAN27 and MAN19 and a background task which are all the instructions between the STOP and WAIT instructions. The STOP instruction is executed when the micro is idle and not during the RS232 communication. The WAIT instruction is executed when the micro is idle and during the RS232 communication. The background instructions consists of branch to subroutine instructions (BSR) which executes routines to process data on demand made by the interrupt routine.

Long branches are achieved by executing a jump (JMP) instruction.

8.2. ICON H53A

Routine: Increments and decimal adjust a counter.

Routines called: DAA

Description: Before this routine is activated the A accumulator contains the length of the counter in the number of bytes and the X register contains the address of the least significant byte minus 5.
This routine increments the counter from the least significant byte (LSB) and then branches to the DAA routine to decimal adjust. According to the result which is contained in the carry bit, the next byte is incremented.

8.3. CONT H54F

Routine: Increments the present, total and maximum demand counters for KWh and KVAh.

Routines called: ICON

Description: Bit No. 3 of register FRED indicates to this routine to increment KWh and bit No. 4 is used for KVAh. The total and present demand values are incremented. The present demand values are compared to the maximum demand and if greater the

maximum values are updated as well as the time when it occured.

This routine is only executed if the counting is enabled (logging mode) when bit 5 of CARR is set.

8.4. SPRES H5C9

Routine: Reset routine for end of demand period.

Routines called: SSS; IDEMP

Description: This routine is only executed when in the logging mode, bit 5 of CARR is set. A reset to the next period would occur if a reset pulse was detected at the input or the period had expired. When a reset occurs, routine SSS (records period into data memory) is called, the present demand values would be cleared and the demand period would be incremented with routine IDEMP.

8.5. IDEMP H5F8

Routine: Increments demand period No.

Routines called: DAAC

Description: This routine first checks the day change bit (6 of CARR) and if set the demand period No. is set to 1 or else it is incremented to the next value. This register (DEMP) is then decimal adjusted.

8.6. DAAC H6ØA

Routine: Decimal adjust for accumulator A only.

Description: Performs a decimal adjust only on the accumulator. This routine executed on the value H9A would result in the clearing of the accumulator.

8.7. SSS H61F

Routine: Stores stack of data in data memory.

Routines called: IACO

Description: A store enable bit is used (6, FRED) so that this routine could be executed by the background task to complete the data store.

Present demand values for KWh and KVAh, day number, and demand period No., are time tagged and stored into the 8k data memory independently of address locations. Memory locations SAL and SAH are used as an address pointer for present logging position in the data memory.

The format in which the data is stored into the data Ram is as follows:

.....:FF SD WW WW VV VV DP HH MM:FF.....

: :

9 locations max and 5 locations min

FF -Indicates start of text

SD -Digits of demand and Day No.

LSN - Day No.

MSN - XXYY

YY=ØØ - no KVAh reading (Ø)

YY=Ø1 - only one byte

YY=11 - two bytes

XX - (same for KWh)

WW WW - KWh readings, 1 or 2 bytes

VV VV - KVAh readings, 1 or 2 bytes

DP - Demand period No. of the day.

HH:MM - Time at which demand period ended.

8.8. IACO H694

Routine: Increments address pointer and stores in memory.

Routines called: IADD; WMMR

Description: This routine operates on the address pointer with address of high address contained in the index register (X). This pointer is incremented and checked

if the maximum ram address is reached. If it is reached the pointer is changed to HØØAØ (the rap around address). The byte contained in the accumulator is then stored in the data RAM using the routine WMMR.

8.9. APPS H6A9

Routine: Points address pointer to previous FF

Routines called: DADD; RMMR

Description: This routine changes the address pointer described by the index register to the previous start of text character FF in the data storage Ram.

8.1Ø. APNS H6B4

Routine: Points address pointer to next FF

Routines called: APPS; IADD; RMMR

Description: This routine changes the address pointer to point to the next start of text character FF in the data ram. If present logging address pointer is reached (SAH and SAL) then address pointer is decremented to the previous FF character.

8.11. RMMR H6DC

Routine: Read from data memory.

Routines called: MRS

Description: Reads one byte of data from data memory to which the address pointer points. The X register contains the address of the high address of the address pointer. Register XREG is initialized to HØ8 which indicates a memory read to the MRS routine being called. This routine returns with the result in the accumulator.

8.12. WMMR H6EC

Routine: Write to data memory.

Routines called: MRS

Description: The accumulator contains the data byte to be written. It writes one byte of data to data memory to which address pointer points. The X register contains the address of the high address of the address pointer. Registerr XREG is initialized to HØ which indicates a memory write to the MRS routine.

8.13. IADD H6F8

Routine: Increments address pointer.

Description: Increments the address pointer by 1. The X register contains the address of

the high address of the address pointer to be incremented. The carry bit is set when the pointer reaches H2ØØØ, indicating an overflow.

8.14. DADD H7ØA

Routine: Decrements the address pointer.

Description: Decrements the address pointer described by the index register. The carry bit is set if the pointer reaches HØØBØ (start of storage memory).

8.15. ONOF H726

Routine: ON/OFF switching of display.

Description: The input F key at port pin PA2 is debounced and when activated this routine toggles the display and keyboard ON or OFF. This is used to place the load recorder into a power saving mode.

8.16. JJT H75E

Routine: Jump to mode jump table.

Routines called: The jump table JMR at H1A4Ø.

Description: Due to the 8 bit index register this routine is used to execute the DO CASE mode. This routine is entered with the mode number in the index register. This value is then multiplied by 3 (JMP instruction is a 3 byte instruction) and

then a jump table is referenced which enables the mode number routine to be executed.

8.17. JKT H76E

Routine: Jump to keypad jump table.

Routines called: Jump table JMK at H184Ø.

Description: Register KKV contains the activated key. The offset is then computed and the JMP table is referenced to execute a DO CASE key pressed. Each key has a separate service routine.

8.18. DBUS H786

Routine: Display busy check.

Description: Display instruction register is read and stored in memory location DAR. Bit 7 set indicates that the display is busy. This routine continuously checks the busy flag and returns when not busy.

8.19. GTCD H793

Routine: Get 2 numbers from display.

Routines called: GACD

Description: Reads two numbers from display with the index register pointing to the most significant byte in the display. It returns with the two BCD numbers in the accumulator.

8.2Ø. GACD H7A5

Routine: Get a number from display.

Routines called: DBUS

Description: Reads the ASCII character at the display location contained in the index register. It returns with the result in the accumulator.

8.21. STCD H7B3

Routine: Stores two numbers into display.

Routines called: STACH

Description: Changes the byte in the accumulator to two ASCII characters and stores it into the display using the address contained in the index register.

8.22. STACH H7D2

Routine: Stores a single character into display.

Routines called: DBUS

Description: Changes the byte contained in the accumulator to ASCII and stores it into the display using the display address contained in the index register.

8.23. PXM H7E4

Routine: Prepares index for mode.

Description: Prepares the index register for the mode selected to accomplish a DO CASE function with the jump instruction.

8.24. IACR H7EF

Routine: Increments address and read.

Routines called: IADD; RMMR

Description: Increments the indexed address memory pointer and reads the data from the memory.

8.25. STAM H8Ø3

Routine: Stores character into memory buffer.

Routines called: IAC

Description: Converts the character in the accumulator to ASCII and stores it into the memory buffer block which is used for RS232 communication.

8.26. STNM; STNM1;
STNM2; STNM3 H815; H826; H84E; H866

Routine: Stores characters into memory buffer.

Routines called: STAM; IACO

Description: Converts more than one character to ASCII and stores it into the RS232 memory buffer.

8.27. CTBM H889

Routine: Clears data memory

Routines called: IACO

Description: Clears the data memory bytes indexed on the X register. The amount of memory bytes to be cleared is contained in the

accumulator.

8.28. KBN H896

Routine: BCD number

Routines called: STACH; PXM; JNT

Description: This routine is executed when a BCD number is pressed. An immediate return occurs if the load recorder is not in the edit mode. The BCD number is converted to ASCII and stored into display. The curser is then moved to the next variable to be changed for the specific mode by using a jump table for a DO CASE function.

8.29. KBA; KBB; KBE H8B2; H8BB; H8C4

Routine: Curser movement keys

Routines called: Jump tables JAT; JBT; and JNT

Description: Moves the curser on the display for Left, Right, and Up/Down keys. The jump tables are used to achieve curser movement for the selected mode.

8.30. KBC H8D2

Routine: Enter key

Routines called: MCX; DBUS; Jump table SMJT

Description: This routine is executed when the Enter key is pressed which enables the Edit mode or changes necessary values

depending on the mode presently in. For the Edit mode the curser is enabled on the display at the first variable for the mode selected, using table FAV. Values are changed by using a look-up jump table (SMJT) to execute the change subroutine for the mode selected.

8.31. KBD H8FC

Routine: Mode change key

Routines called: UDA

Description: The mode select prompt is displayed by calling routine UDA. This mode is known as mode A.

8.32. STCL H918

Routine: Stores in display omiting leading zeros.

Routines called: STACH; STCD; MTN4; GACD

Description: Stores two characters into display and supresses leading zeros for KWh and KVAh readings.

8.33. WVM H98E

Routine: Multiplication routine

Routines called: UMD; MUL

Description: This routine multiplies the pulse count to the pulse factors to achieve true values for the display. The memory area from HAØ to HAF and HBØ to HBF is used

for the multiplication process for KWh and KVAh respectively. The multiplying factors are placed into memory locations 6 to 7 and the pulse count is placed into locations 8 to D. The product would appear at memory locations Ø to 5. These locations are all relative to memory area HAØ and HBØ. The method in which multiplication is achieved is the shift and add method. Routines ROLB, ROR, and ROL is used to shift the whole data number. Routine ADDB is used to add the shifted data into the product locations and then after each addition a decimal adjust is performed by routine DAA for the complete number. These decimal adjusted products are updated in the display by the mode routines.

8.34. RS232 Description.

PB6 is used to receive serial data and PB7 is used to transmit the serial data. The bit minipulation instructions is used to control the transmitting and receiving of data. Data to be transmitted or received is buffered through register RXTX. The RS232 format used is 1 start, 7 data (ASCII), 1 even parity and 2 stop bits. A 12ØØ Baud rate is used, giving a bit time of 833 micro seconds. This bit time is obtained by utilizing the on chip timer which generates an interrupt at the end of each bit time.

The timing diagram of the RS232 is as follows:

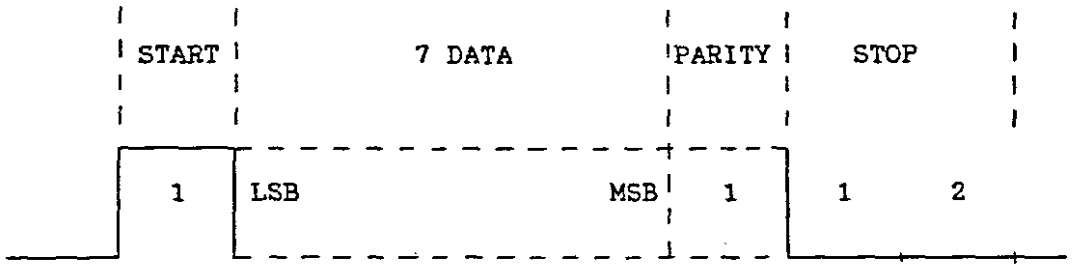


FIGURE 8.1 RS232 TIMING DIAGRAM

The timer interrupt routine controls the port lines to receive and transmit necessary data. Data communication is half dulplex. The TX and RX sequences are identical, hence only the transmitting sequence is described below.

To initiate a TX sequence the TXSR routine is executed with the ASCII data contained in the accumulator. This routine stores the data into buffer RXTX, and initializes the flags and timer for a 832 micro second interrupt. The first 832 micro second is used as an inter-digit pause. The timer interrupt routine RXR is executed when the timer, times out. Each time this routine is executed it updates the timer for the next bit and controls the PB7 according to the bit counter and the data. For the bit counter equal to Ø a single stop bit is set on the port. For the bit counter > Ø and < 8 the seven data bits from RXTX is transmitted using a rotate through carry instruction (ROR). Each time a logic 1 is transmitted the parity counter is incremented. At the end of a data transmission when the bit counter is 8 the parity counter is divided by 2 and if a carry exists, a logic 1 parity bit is used to achieve an even parity.

Transmission sequence is terminated by sending two stop bits for bit counter equal to 9 and 1Ø. The control is then handed over to the RS232 control routine.

This control routine (RTCR) performes the following functions:

i) Controls TX and RX

ii) Controls hand shaking using TXON and TXOFF signals

iii) Sends acknowledge instruction

iv) Processes faults

v) Processes instructions sent by HP85

This is one of the main background task routine which is being serviced each time an interrupt occurs.

The summery of the RS232 routines are:

RXR -Timer interrupt routine for bit transmission.

RTCR -Control routine

RXSR -Receive initiate

TXSR -Transmit initiate

8.35. RXSR HAFF

Routine: Receive data

Description: This routine is called from the main interrupt routine when the first start bit is detected. This routine is not executed if receiving of data is in progress (6,FLGN). The timer is synchronised to allow bit reading to occur at the middle of each received bit. Each bit is read by the timer interrupt routine RXR.

8.36. TXSR HBØD

Routine: Transmit data

Description: Enables a transmitting sequencing only if not presently busy transmitting or receiving. The on chip timer is programmed for the bit time, to allow the timer interrupt routine (RXR) to control the port for serial data transmission.

8.37. RXR HB2Ø

Routine: RS232 interrupt routine.

Description: The timer interrupt routine which is used to control the RS232 serial communication on two parallel ports. This routine is divided into two sections, namely, an RX and a TX section. The RXTX memory location is used as a buffer to minipulate, receive and transmit information. The byte to be transmitted is loaded into RXTX and it is rotated through the TX port, one bit at a time. The received information is rotated from the RX port into the RXTX register. The bit timing is programmed into the timer each time this routine is executed.

8.38. RTCR HBB2

Routine: RS232 control routine.

Routines called: TXSR; Jump table RXIN.

Description: This main RS232 control routine is called from the main DO FOR EVER loop. It controls the data communication with the HP85 computer. Transmitting of data is dependent on instructions received from the HP85 computer. The load recorder enters into a wait for instruction mode (from RS232) when activated after inserting RS232 cord. RS232 communication is half duplex because the same timer is used for both RX and TX. The XON and the XOFF signals are used for hand shaking between the two computers. A look-up jump table is used to execute an instruction received from the HP85 computer which downloads requested information.

8.39. BUFU HCØA

Routine: Updates RS232 buffer

Routines called: IACR; STAM; STNM1; IACO; CTBM

Description: This routine is called to update the RS232 buffer with a new stack of data which is to be downloaded. The information is read from data memory, converted to ASCII and then stored into

the buffer, ready for downloading. A sum check byte is calculated for each batch (a demand period) and also stored into this buffer.

8.40. IN00 HC9D

Routine: Start instruction

Routines called: BUFU; TXSR

Description: This routine is executed from the instruction jump table (RXIN). It resets the RS232 pointers and updates the memory buffer with the first demand period.

8.41. IN01 HCB2

Routine: Transmits information

Routines called: IACR; TXSR

Description: This routine is executed from the instruction jump table. It consists of seven operations depending on the value sent as an instruction (1 to 7). The buffer contains 7 types of information and is labelled according to a number. The seven operations are the sending of these labelled information accordingly i.e. instruction 2 received would instruct the load recorder to send information (or word) number two in buffer.

8.42. IN08; IN09; IN0A HCE3; HCF9; HD0A

Routine: Updates buffer with information

Routines called: APPS; BUFU; TXSR

Description: Updates the memory buffer with the present, the previous and the next information. These routines are called from instruction jump table.

8.43. IN0B HD1A

Routine: RS232 off.

Routines called: TXSR.

Description: This instruction switches the RS232 off (idle mode).

8.44. IN0C; IN0D HD23; HD7A

Routine: Starts variables

Routines called: IACR; STAM; TXSR; STNM1; IACO

Description: Updates the RS232 buffer with the start variables consisting of the start time, the date, the pulse factors and the divisors.

8.45. IN0E HD97;

Routine: Acknowledge.

Routines called: TXSR

Description: This routine is used to send an acknowledgement to the HP85 computer.

8.46. **MEMS; MEMR** HDA3; HDAB

Routine: Data memory read/write

Description: The internal Ram area is the only part of the memory which is common to both memory maps. These two position independent routines are copied into this area and accessed when the data memory map is enabled for data logging. One byte at a time could be read or written by first enabling the memory map, store/read in extended direct mode and then disable the memory map.

8.47. **MRS** HDB3

Routine: Data read/store control.

Routines called: MEMS; MEMR in ram.

Description: Controls the routine which is used to initialize the MEMS and MEMR routines in the ram area to execute a read or store from data memory. After initialization the routines in the ram area (MEMS, MEMR) H3Ø to H3F is called.

8.48. **MODx** x- mode number

Routine: Mode control routine.

Description: All the display modes have seprate mode control routines which are called from a jump table executed from the main DO FOR EVER routine.

8.49. VDx — x- mode number

Routine: Updates variables.

Description: Updates the variables in the display modes as selected.

8.5Ø. UDx — x- mode number

Routine: Updates display

Description: Updates the fixed data in the display modes.

8.51. MPx; MNx; MTNx — x- mode number

Routine: Moves curser.

Description: Moves the curser on the display in the edit mode to the required position on the display.

8.52. MINT — H14B4

Routine: Main interrupt routine.

Routines called: KINT; RINT; INPT

Description: This main interrupt routine is executed when an external interrupt occurs. All interrupt sources are polled to determine the interrupt cause. The particular interrupt routine is then executed. Routines RINT and KINT are used to service the RTC and the keyboard. The INPT routine is used to debounce the input pulses on the 1Øms interrupt.

A P P E N D I X E

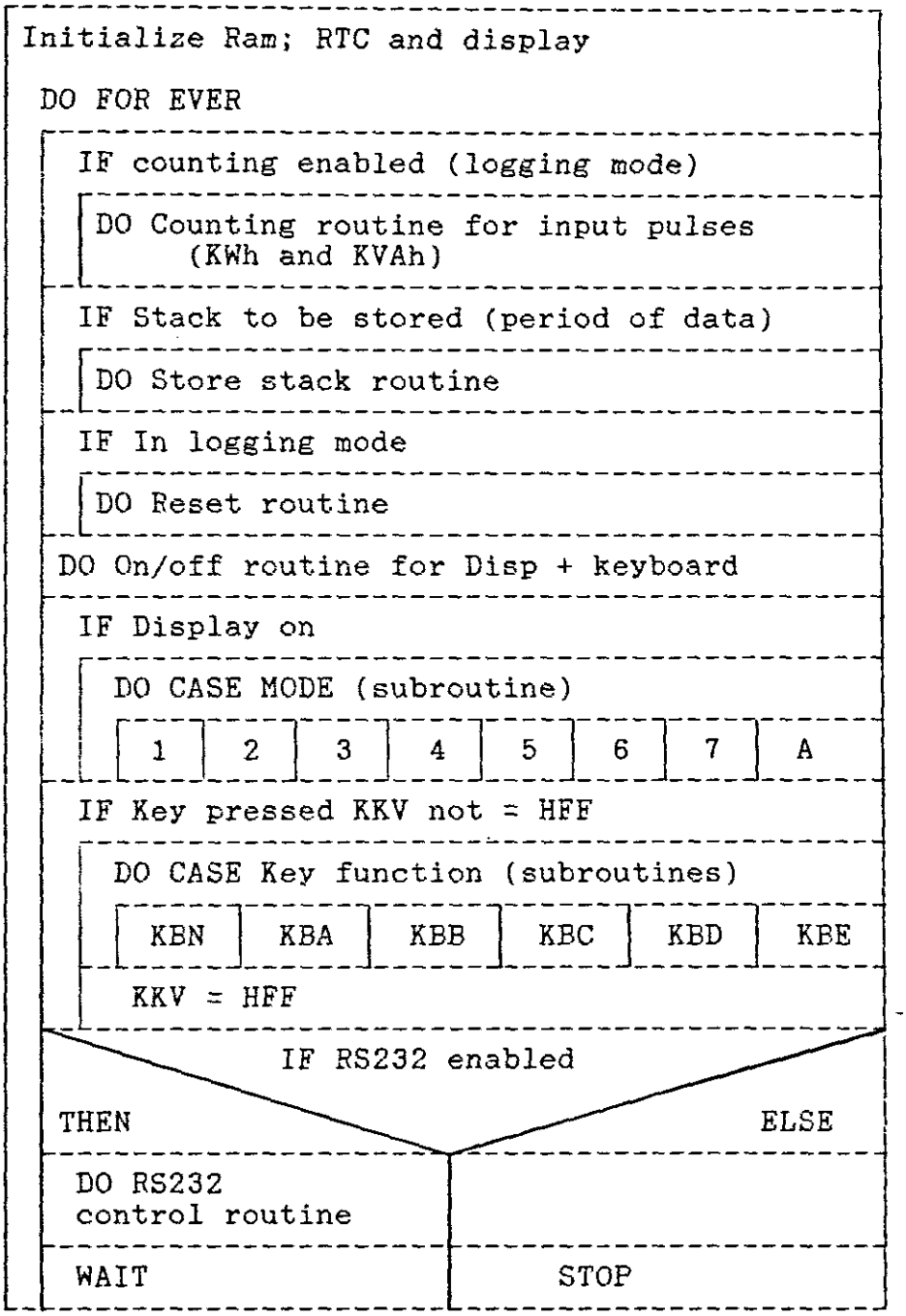
L O A D R E C O R D E R
F L O W C H A R T S

APPENDIX E

9. LOAD RECORDER FLOW CHARTS.

9.1. Main routine (background task).

MAN



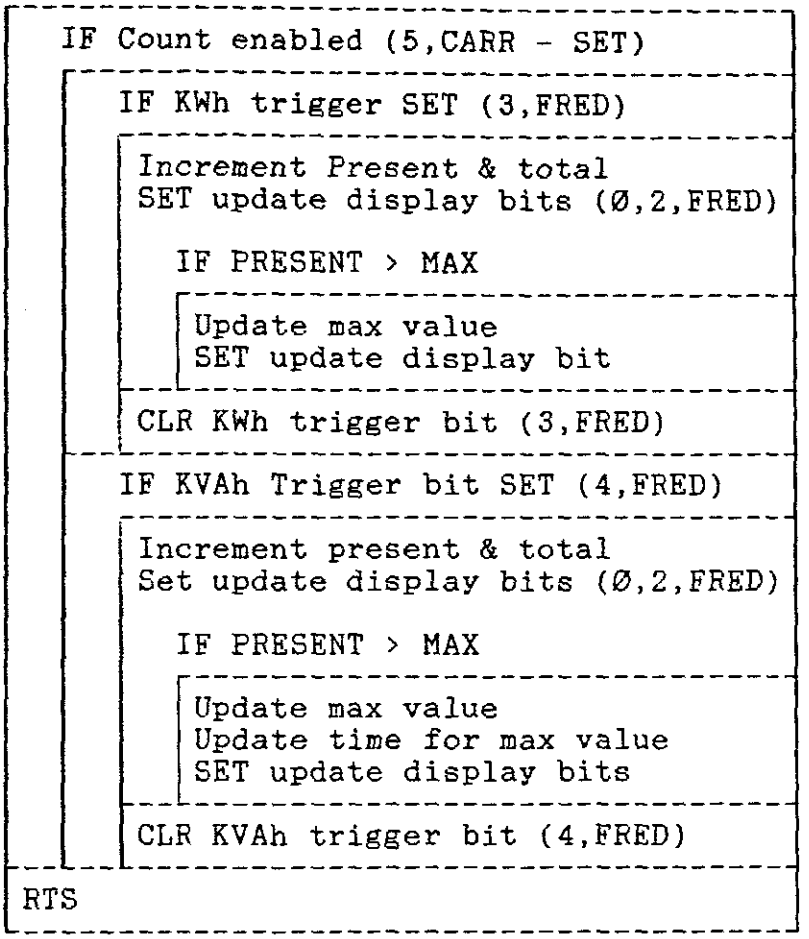
9.2. Subroutine Update Day and Time

SUDT

```
Check display busy (DBUS)
Clear whole display

  DO FOR X=Ø TO H1F
    LDA WITH MODF1,X

      IF A NOT = HAØ (space)
        DO routine STACH

  DO FOR X=H4Ø TO H 5F
    LDA WITH MODF1,X

      IF A NOT = HAØ
        DO Routine STACH

SVDT

 Load X with first var of disp
 Store into disp Y/M/D
 Load X with first var+ H4Ø
 Store into disp H:M:S

RTS
```

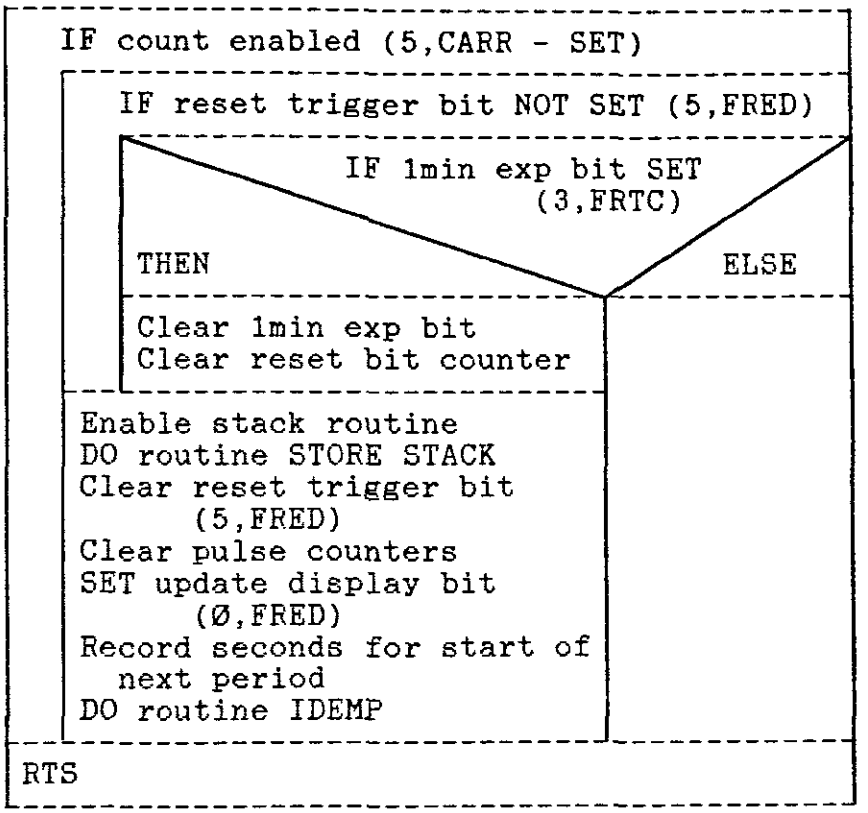
9.3. Count input pulses

CONT



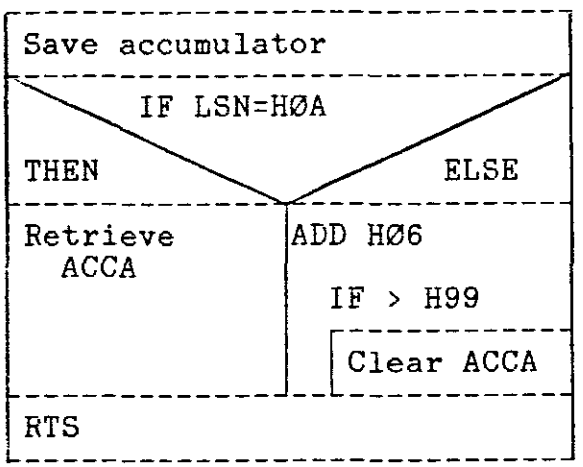
9.4. Demand period reset routine

SPRES

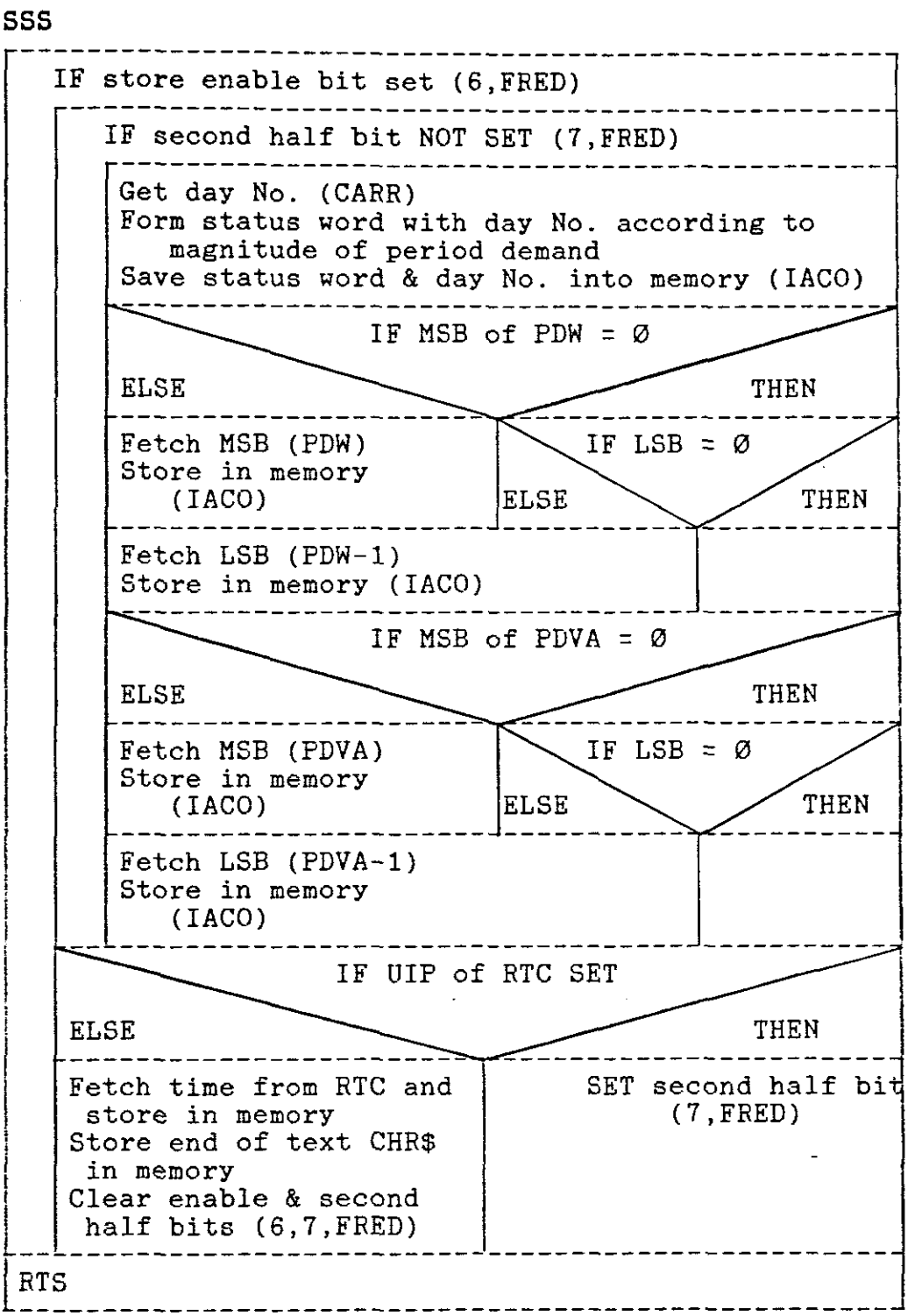


9.5. Decimal adjust accumulator

DAAC

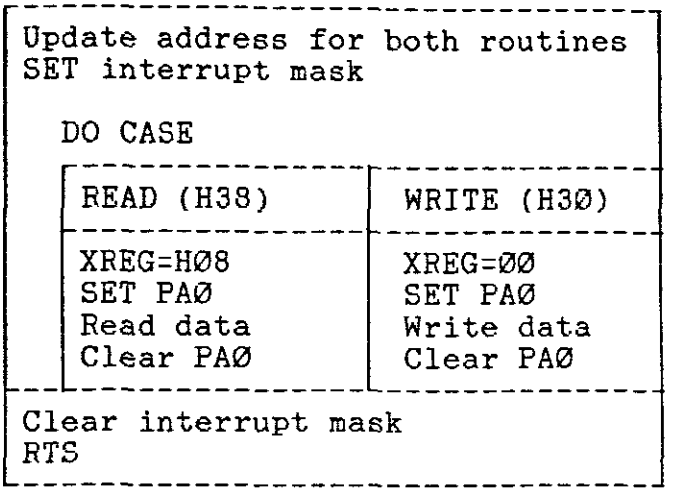


9.6. Store stack of data in memory



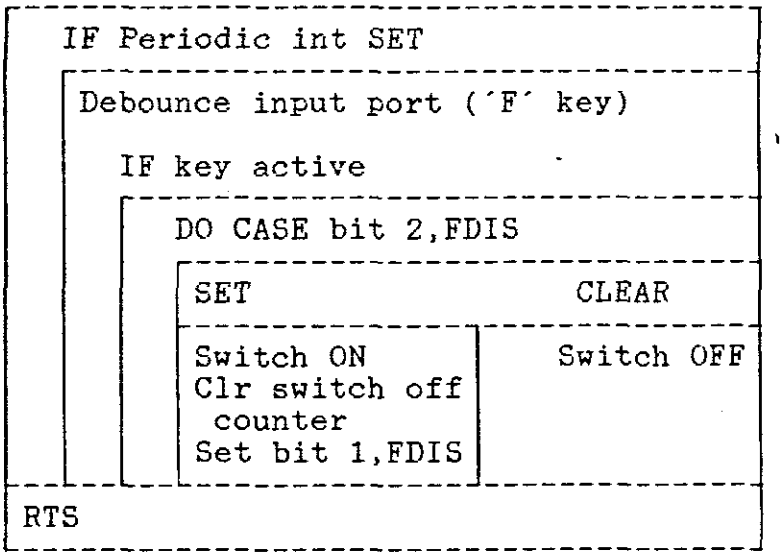
9.7. Data memory Read/Write routine

MRS



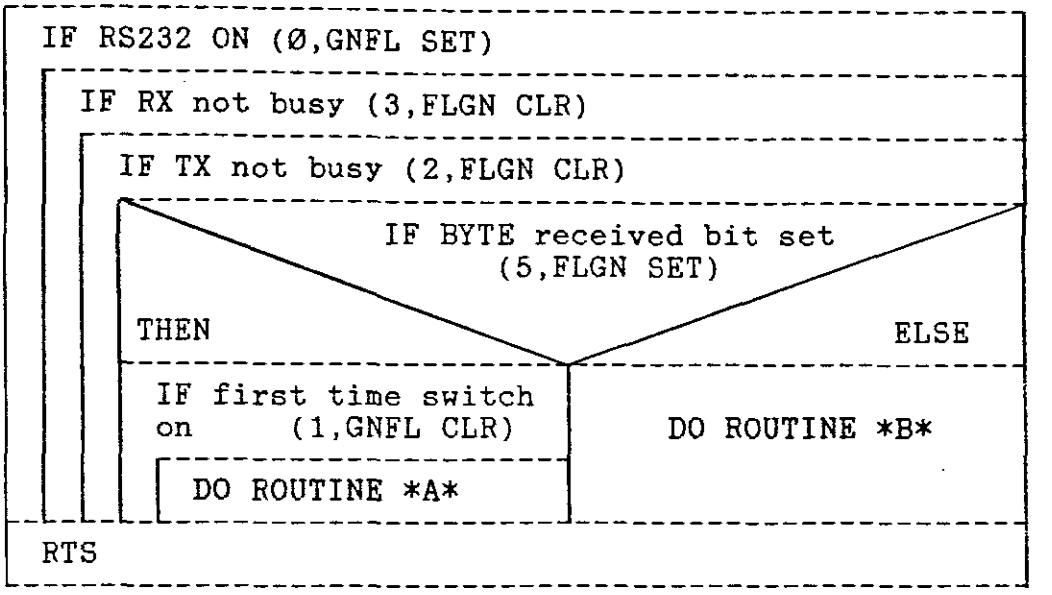
9.8. ON/OFF Switch subroutine

ONOF



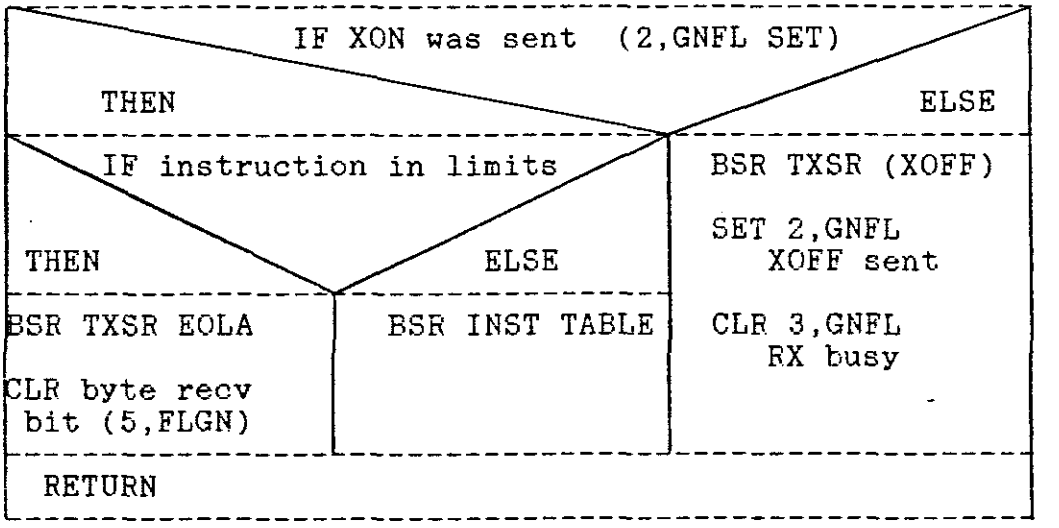
9.9. RS232 Control routine

RTCR



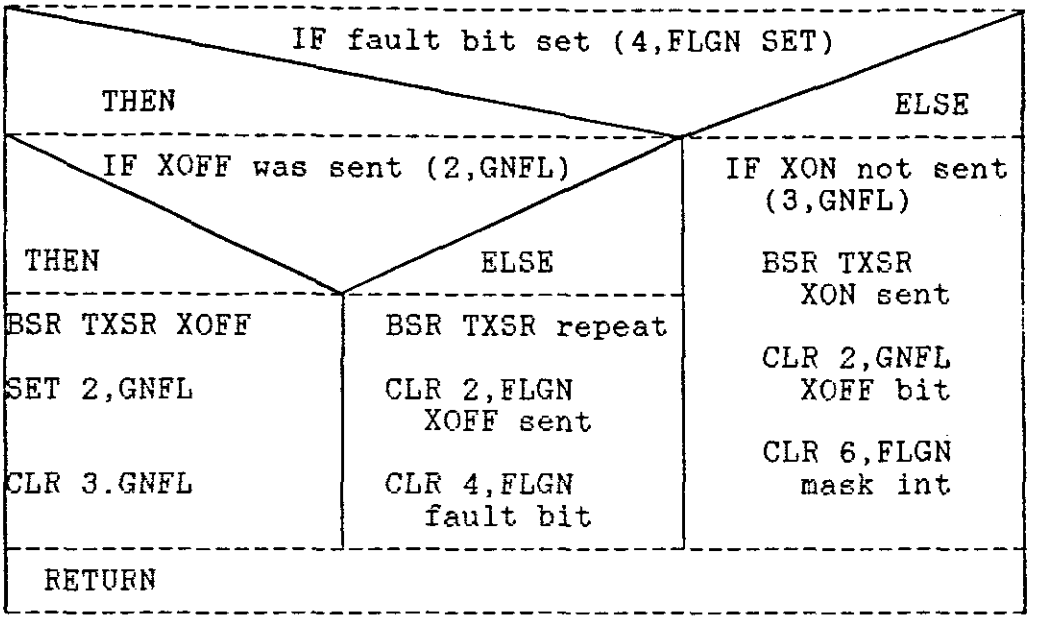
9.9.1. ROUTINE *A*

ROUTINE *A*



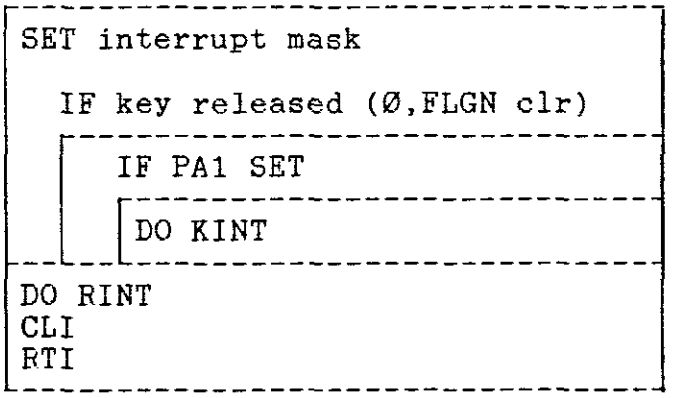
9.9.2. ROUTINE *B*

ROUTINE *B*



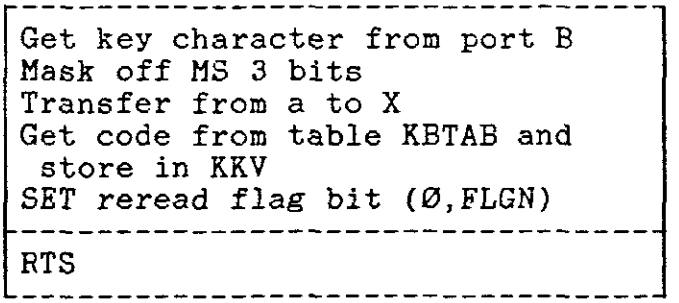
9.1Ø. Main interrupt routine

MINT



9.11. Keyboard interrupt routine

KINT



9.12. Timer interrupt from wait state

TIWS

```
SET BIT 7,FDIS

  IF INT after wait for disp
    Stop timer
    Load H4F into TCR

RTS
```

A P P E N D I X F

L O A D R E C O R D E R
S O F T W A R E L I S T I N G

APPENDIX F

1Ø. LOAD RECORDER SOFTWARE LISTING.

The following pages represents the software listing of the load recorder assembly language program.

```
00001                    OPT    P=64
00002                    TTL    LOAD RECORDER
00003                ***********************************
00004                *     SOFTWARE LISTING FOR THE    *
00005                *     LOAD RECORDER               *
00006                ***********************************
00007
00008
00009                ***************************
00010                *  MACRO INSTRUCTIONS     *
00011                *  FOR MC14680SE2 IC      *
00012                ***************************
00013
00014
00015                *
00016                *     STOP INSTRUCTION
00017                *
00018
00019                STOP   MACR
00020                       FCB    $8E
00021                       ENDM
00022
00023
00024                *
00025                *     WAIT INSTRUCTION
00026                *
00027
00028                WAIT   MACR
00029                       FCB    $8F
00030                       ENDM
00031
00032
00033                *
00034                *     EQUATES
00035                *
00036
00037    000D    A EOLN   EQU    $0D      END OF LINE NO INS CR
00038    0006    A EOLA   EQU    $06      ACK CHR$ IN CR12
00039    0003    A XON    EQU    $03      TX ON INS (ETX)
00040    0002    A XOFF   EQU    $02      TX OFF INS (STX)
00041    0005    A RETX   EQU    $05      REPEAT INS (ENQ)
00042    000E    A RXINL  EQU    $0E      RS232 ON INS
00043    0013    A AREG   EQU    $13      TEMP STORAGE FOR
00044    0014    A XREG   EQU    $14      ACC & XREG
00045    0012    A FLGN   EQU    $12      GENERAL FLAG REG
00046                *                     D0-REREAD KB FLAG
00047                *                     D1-ON/OFF DEBOUNCED
00048                *                     D2-TX BIT
00049                *                     D3-RX BIT
00050                *                     D4-FAULT BIT
00051                *                     D5-BYTE RX BIT
00052                *                     D6-MASK INT BIT RX
00053                *                     D7-MEM FULL BIT
00054    0015    A FRED   EQU    $15      D0-PRESENT DEMAND
00055                *                     D1-MAX DEMAND
00056                *                     D2-TOT DEMAND
00057                *                     D3-KWh TRIG BIT
00058                *                     D4-KVAh TRIG BIT
00059                *                     D5-RESET TRIG BIT
00060                *                     D6-STORE STACK IN MEM
00061                *                     D7-STORE STACK IN MEM
00062    009A    A CW     EQU    $9A      KWh CTR  /D7-TEMP I/P
00063    009B    A CVA    EQU    $9B      KVA CTR  /D6-BEING DEB
00064    009C    A SAH    EQU    $9C      STORAGE ADDRESS
```

```
00065  009D  A SAL   EQU  $9D   PRESENTLY IN
00066  0080  A STMEM EQU  $80   START OF DATA IN 8K MEM
00067  008E  A CR    EQU  $8E   RESET CTR/D5-DEBOUNCED
00068  008F  A CPI   EQU  $8F   PERIODIC INT COUNTER
00069  0021  A PDW   EQU  $21   PRESENT DEM KW
00070  0029  A PDVA  EQU  $29   PRESENT DEM KVA
00071  0023  A MDW   EQU  $23   MAX DEM KW
00072  002B  A MDVA  EQU  $2B   MAX DEM KVA
00073  0026  A TDW   EQU  $26   TOT DEM KW
00074  002E  A TDVA  EQU  $2E   TOT DEM KVA
00075  0016  A BTCT  EQU  $16   BIT CTR FOR RS232
00076  0017  A PRTY  EQU  $17   PARITY CTR RS232
00077  0018  A RXTX  EQU  $18   RX,TX DATA REG
00078  0027  A CARR  EQU  $27   CARRY STOREDGE REG
00079         *                 +DAY NO.
00080         *                 D6-DAY CHNG BIT
00081         *                 D5-ENABLE/DISABLE COUNT
00082  002F  A MEL   EQU  $2F   MULTIPLY ENABLE
00083  0090  A KKV   EQU  $90   KEYBOARD VALUE PRESSED
00084  0010  A FRTC  EQU  $10   FLAG REG FOR RTC
00085         *                 D7-IRQF,D6-PF,D5-AF,
00086         *                 D4-UF--RTC
00087         *                 D2-PF,D1-AF,D0-UF--INT
00088         *                 D3-RESET BIT ROUTINE
00089  0091  A ACTR  EQU  $91   ALARM COUNTER REG
00090  0092  A SPR   EQU  $92   SAMPLE PERIOD REG
00091  0011  A FDIS  EQU  $11   DISP FLAG REG;2-ON/OFF
00092         *                 D1-FIRST TIME,D7-WAIT I
00093         *                 FOR DISP,D0-ON/OFF DEBO
00094         *                 D3-FIRST TIME ENTER MOD
00095         *                 D4-UPDATE FOR MDS 4;5;6
00096         *                 D5-POINT ON DISP
00097         *                 D6-LEADING ZERO BIT
00098  0019  A GNFL  EQU  $19   D0-ENABLE RX/TX
00099         *                 D1-1ST SW ON RS232
00100         *                 D2-XOFF JUST SENT
00101         *                 D3-XON  JUST SENT
00102         *                 D4-MSB FOR 4 BITS
00103         *                 D5-SW OFF RS232 BIT
00104  001A  A RXMD  EQU  $1A   MODE OF RECEIVE
00105  001E  A RSM1  EQU  $1E   RS232 BUSS BUFF
00106  001F  A RSL1  EQU  $1F
00107  001B  A BREG  EQU  $1B
00108  001C  A CREG  EQU  $1C
00109  009E  A RSH   EQU  $9E   RS232 ADD POINTER
00110  009F  A RSL   EQU  $9F
00111  0040  A DREG  EQU  $40
00112  0041  A EREG  EQU  $41
00113  0042  A PLOG  EQU  $42
00114  0043  A CTRS  EQU  $43   COUNTER FOR RS232
00115  0044  A ACTRS EQU  $44   SEC CTR FOR 1MIN OFF
00116  0045  A DOCTR EQU  $45   CTR FOR SWITCH OFF
00117  0046  A MDD   EQU  $46   DATE FOR MAX DEMAND
00118  0047  A MMO   EQU  $47   MONTH FOR MAX DEMAND
00119  0048  A MHH   EQU  $48   HOURS FOR MAX DEMAND
00120  0049  A MMIN  EQU  $49   MIN FOR MAX DEMAND
00121  0093  A MRD   EQU  $93   DISP MODE REG;7-SET
00122         *                 D0-6-MODE
00123  0094  A DAR   EQU  $94   DISP ADDRESS REG
00124  0030  A MEM   EQU  $30   START OF MEM READ/WRIGH
00125  0095  A DEMP  EQU  $95   DEMAND PERIOD
00126  0096  A XVD2  EQU  $96   XREG FOR VD2
00127  0097  A MADH  EQU  $97   ADDRESS LOCI FOR
00128  0098  A MADL  EQU  $98   MEMORY
```

```
00129  0099  A MMRD  EQU  $99    MEMORY FOR MODE
00130  0000  A PORTA EQU  $00
00131  0001  A PORTB EQU  $01
00132  0004  A DDA   EQU  $04    DATA DIRECTION REG
00133  0005  A DDB   EQU  $05
00134  0009  A TCR   EQU  $09    TIMER CONTROL REG
00135  0008  A TDR   EQU  $08    TIMER DATA REG
00136  0180  A DINS  EQU  $180   DISPLAY INSTRUCTION
00137  0188  A DDAT  EQU  $188   DISPLAY DATA
00138  0080  A SEC   EQU  $80    SECONDS-RTC
00139  0081  A SECA  EQU  $81    SECONDS-ALARM
00140  0082  A MIN   EQU  $82    MINUTES
00141  0083  A MINA  EQU  $83    MINUTES ALARM
00142  0084  A HRS   EQU  $84    HOURS
00143  0085  A HRSA  EQU  $85    HOURS ALARM
00144  0086  A DOW   EQU  $86    DAY OF WEEK
00145  0087  A DATE  EQU  $87    DATE OF MONTH
00146  0088  A MON   EQU  $88    MONTH
00147  0089  A YEAR  EQU  $89    YEAR
00148  008A  A ARTC  EQU  $8A    A CONTROL REG-RTC
00149  008B  A BRTC  EQU  $8B    B CONTROL REG-RTC
00150  008C  A CRTC  EQU  $8C    C CONTROL REG-RTC
00151  008D  A DRTC  EQU  $8D    D CONTROL REG-RTC
00152  00A0  A DKW   EQU  $A0    KWh DISP READING
00153  00B0  A DKVA  EQU  $B0    KVA DISP READING
00154  1A80  A MODV1 EQU  $1A80  MODE 1 DISPLAY FORMAT
00155        *                   VARIABLES(07-1 UP;0 PRE
00156        *                   00-3F BLOCK
00157  1B00  A MODV2 EQU  $1B00  MODE 2 DISP FORMAT
00158  1780  A MODV3 EQU  $1780  MODE 3 DISP FORMAT
00159  1700  A MODV4 EQU  $1700  MODE 4;5;6 DISP FORMAT
00160  1A00  A FAV   EQU  $1A00  FIRST AVAILABLE VARIABL
00161  1800  A KBTAB EQU  $1800  KEYBOARD CODES
00162  1820  A U456  EQU  $1820  MD4;5;6 JMP TAB
00163  1840  A JMK   EQU  $1840  KEYBOARD JMP TAB
00164  1C00  A MODF1 EQU  $1C00  DISP FORMAT MOD1
00165  1C20  A MODF2 EQU  $1C20  DISP FORMAT MOD2
00166  1C80  A MODF4 EQU  $1C80  DISP FORMAT MOD4
00167  1CA0  A MODF3 EQU  $1CA0  DISP FORMAT MOD3
00168  1D00  A MODF5 EQU  $1D00  DISP FORMAT MOD5
00169  1D20  A MODF6 EQU  $1D20  DISP FORMAT MOD6
00170  1DA0  A MODF7 EQU  $1DA0  DISP FORMAT MOD7
00171  1D80  A MODFA EQU  $1D80  DISP FORMAT MODA
00172  1E00  A NMOD  EQU  $1E00  DISP FORMAT FOR NAMES
00173  1A40  A JMR   EQU  $1A40  TMP TABLE
00174  1900  A JAT   EQU  $1900  MOVE TO TOP LOC JMP TAB
00175  1940  A JBT   EQU  $1940  MOVE TO PREVIOUS TAB
00176  1980  A JNT   EQU  $1980  NEXT LOC JMP TAB
00177  1880  A SMJT  EQU  $1880  CHANGE JMP TAB
00178  18E0  A ESM   EQU  $18E0  ENABLE SET MODE TAB
00179  1BA0  A RXIN  EQU  $1BA0  JMP INS TAB FOR RS232
00180  1740  A RINVT EQU  $1740  LOOK UP TAB FOR END
00181        *                   ADDRESS FOR RS232 BUFF
```

```
00183
00184                    ****************
00185                    * MAIN PROGRAM  *
00186                    ****************
00187
00188A 0400                       ORG    $400
00189
00190                    *        INITIALIZATION
00191
00192A 0400 9B             MAN    SEI
00193A 0401 9C                    RSP             RESET STACK POINTER
00194A 0402 3F 00          A      CLR    PORTA    ENSURE MAIN MEMORY MAP
00195A 0404 A6 20          A      LDA    #$20     SWITCH KEYBOARD
00196A 0406 B7 01          A      STA    PORTB    OFF
00197A 0408 A6 19          A      LDA    #%00011001 D0;D3;D4-0/PS
00198A 040A B7 04          A      STA    DDA
00199A 040C A6 A0          A      LDA    #%10100000 D7;D5-0/PS
00200A 040E B7 05          A      STA    DDB
00201A 0410 AE 7F          A      LDX    #$7F     CLEAR ON CHIP
00202A 0412 7F               MAN1 CLR    ,X       MEMORY
00203A 0413 5A                    DEX
00204A 0414 A3 10          A      CPX    #$10
00205A 0416 24 FA    0412          BHS    MAN1
00206A 0418 AE BF          A      LDX    #$BF     CLEAR RTC
00207A 041A 7F               MAN2 CLR    ,X       RAM
00208A 041B 5A                    DEX
00209A 041C A3 8E          A      CPX    #$8E
00210A 041E 24 FA    041A          BHS    MAN2
00211A 0420 AE 0F          A      LDX    #$0F     LOAD RAM WITH R/W
00212A 0422 D6 0DA3        A MAN14 LDA    MEMS,X   ROUTINES
00213A 0425 E7 30          A      STA    MEM,X
00214A 0427 5A                    DEX
00215A 0428 A3 FF          A      CPX    #$FF
00216A 042A 26 F6    0422          BNE    MAN14
00217A 042C A6 8F          A      LDA    #$8F     CLR NECESSARY REG
00218A 042E B7 98          A      STA    MADL     IN BK RAM
00219A 0430 3F 97          A      CLR    MADH
00220A 0432 AE 97          A MAN8  LDX    #MADH    ADD OF HIGH ADD
00221A 0434 3F 14          A      CLR    XREG     STORE
00222A 0436 A6 00          A      LDA    #$00     DATA-00
00223A 0438 AD 0C    0446          BSR    MAN15    MRS
00224A 043A 3A 98          A      DEC    MADL
00225A 043C B6 98          A      LDA    MADL
00226A 043E A1 80          A      CMP    #$80
00227A 0440 24 F0    0432          BHS    MAN8
00228A 0442 3C 42          A      INC    PLOG
00229A 0444 20 05    044B          BRA    MAN16
00230A 0446 CC 0DB3        A MAN15 JMP    MRS
00231A 0449 3C 42          A      INC    PLOG
00232                    *
00233                    *        INITIALIZE RTC
00234                    *        PERIODIC INT-125MS
00235                    *        ALARM INT-EVERY MIN
00236                    *        UPDATE IN PROGRESS-ENABLE
00237                    *
00238A 044B A6 79          A MAN16 LDA    #%01111001 10MS;DISABLE CLOCK
00239A 044D B7 BA          A      STA    ARTC
00240A 044F A6 F2          A      LDA    #%11110010 24H MODE
00241A 0451 B7 BB          A      STA    BRTC     ALL INTS ENABLED
00242A 0453 A6 FF          A      LDA    #$FF     ALARM INT-
00243A 0455 B7 B5          A      STA    HRSA     EVERY MIN
00244A 0457 B7 B3          A      STA    MINA
00245A 0459 3F B1          A      CLR    SECA
00246A 045B A6 B7          A      LDA    #$B7
```

```
00247A 045D B7 B9       A       STA   YEAR
00248A 045F A6 01       A       LDA   #$01
00249A 0461 B7 B8       A       STA   MON
00250A 0463 B7 B7       A       STA   DATE
00251A 0465 A6 23       A       LDA   #$23
00252A 0467 B7 B4       A       STA   HRS
00253A 0469 3F B2       A       CLR   MIN
00254A 046B 3F B0       A       CLR   SEC
00255A 046D A6 72       A       LDA   #%01110010 MON SET MODE
00256A 046F B7 BB       A       STA   BRTC
00257A 0471 A6 29       A       LDA   #%00101001 ENABLE CLOCK
00258A 0473 B7 BA       A       STA   ARTC
00259A 0475 A6 01       A       LDA   #$01      ENABLE MODE 1
00260A 0477 B7 93       A       STA   MRD
00261A 0479 B7 99       A       STA   MMRD      MEMORY FOR MODE
00262
00263                     *       INITIALIZE DISPLAY
00264
00265A 047B A6 04       A       LDA   #$04      DELAY FOR
00266A 047D 9A          *       CLI             500MS FOR DISP
00267A 047E               MAN3  STOP  TO        SETTLE DOWN
00268A 047F 05 10 FC 047E       BRCLR 2,FRTC,MAN3
00269A 0482 15 10       A       BCLR  2,FRTC
00270A 0484 4A                  DECA
00271A 0485 26 F7    047E       BNE   MAN3
00272A 0487 C6 0100     A       LDA   DINS      CHECK IF DISP
00273A 048A B7 94       A       STA   DAR       IS BUSY
00274A 048C A6 01       A       LDA   #$01
00275A 048E 0E 94 ED 047E       BRSET 7,DAR,MAN3
00276A 0491 A6 01       A       LDA   #$01      CLEARS ALL DISP
00277A 0493 C7 0100     A       STA   DINS
00278A 0496 A6 4D       A MAN4  LDA   #%01001101 INTERNAL CK;32MICS PR
00279A 0498 B7 09       A       STA   TCR       INT MASKED
00280A 049A A6 34       A       LDA   #52       52*32MICS=1,7MS
00281A 049C B7 08       A       STA   TDR
00282A 049E 1E 11       A       BSET  7,FDIS    FLAG FOR DISP INT FROM
00283A 04A0 A6 0D       A       LDA   #%00001101 WAIT STATE
00284A 04A2 B7 09       A       STA   TCR
00285A 04A4 9A                  CLI
00286A 04A5                     WAIT  WAIT      FOR INT
00287A 04A6 C6 0100     A       LDA   DINS      DISPLAY BUSY?
00288A 04A9 B7 94       A       STA   DAR
00289A 04AB 0E 94 E8 0496       BRSET 7,DAR,MAN4
00290A 04AE 9B                  SEI
00291A 04AF A6 08       A       LDA   #%00001000 DISP OFF,CURSER OFF
00292A 04B1 C7 0100     A       STA   DINS
00293A 04B4 14 11       A       BSET  2,FDIS    DISP OFF FLAG
00294A 04B6 A6 14       A MAN5  LDA   #20       SET TIMER FOR 40MICS DE
00295A 04B8 B7 08       A       STA   TDR
00296A 04BA A6 09       A       LDA   #%00001001
00297A 04BC B7 09       A       STA   TCR
00298A 04BE 9A                  CLI
00299A 04BF                     WAIT
00300A 04C0 C6 0100     A       LDA   DINS
00301A 04C3 B7 94       A       STA   DAR
00302A 04C5 0E 94 EE 04B6       BRSET 7,DAR,MAN5
00303A 04C8 9B                  SEI
00304A 04C9 A6 38       A       LDA   #%00111000 2-LINES
00305A 04CB C7 0100     A       STA   DINS
00306A 04CE A6 14       A MAN6  LDA   #20       40MICS DELAY
00307A 04D0 B7 08       A       STA   TDR
00308A 04D2 A6 09       A       LDA   #%00001001
00309A 04D4 B7 09       A       STA   TCR
00310A 04D6 9A                  CLI
```

```
00311A 04D7                    WAIT
00312A 04D8 C6 0100      A     LDA   DINS
00313A 04DB B7 94        A     STA   DAR
00314A 04DD 0E 94 EE 04CE      BRSET 7,DAR,MAN6
00315A 04E0 1F 11        A     BCLR  7,FDIS
00316A 04E2 9B                 SEI
00317A 04E3 A6 79        A     LDA   #%01111001 STOP RTC
00318A 04E5 B7 8A        A     STA   ARTC
00319A 04E7 B6 8C        A     LDA   CRTC      CLR INTS
00320A 04E9 3F 10        A     CLR   FRTC      CLR ALL RTC FLAGS
00321A 04EB A6 29        A     LDA   #%00101001 START RTC
00322A 04ED B7 8A        A     STA   ARTC
00323A 04EF A6 4B        A     LDA   #$4B      TIMER INT OFF
00324A 04F1 B7 08        A     STA   TDR
00325A 04F3                MAN7  STOP  MAIN    STOP INSTRUCTION
00326A 04F4 AD 17    050D  MAN26 BSR   MAN22   UPDATE COUNTERS FOR
00327                      *                    I/P METERS
00328A 04F6 AD 18    0510        BSR   MAN23   STORE STACK
00329A 04F8 AD 19    0513        BSR   MAN24   RESET
00330A 04FA AD 1A    0516        BSR   MAN10   ON/OFF SUBROUTINE
00331A 04FC 04 11 04 0503 MAN9   BRSET 2,FDIS,MAN13 DO NOT UPDATE DISP
00332A 04FF BE 93        A     LDX   MRD      IF OFF
00333A 0501 AD 19    051C        BSR   MAN12   BR TO JMP TO JMP TAB
00334A 0503 A6 FF        A MAN13 LDA   #$FF     KEY PRESSED?
00335A 0505 B1 90        A     CMP   KKV
00336A 0507 27 20    0529        BEQ   MAN18   ABORT IF NO KEY PRESSED
00337A 0509 AD 0E    0519        BSR   MAN11
00338A 050B 20 18    0525        BRA   MAN17
00339A 050D CC 054F      A MAN22 JMP   CONT
00340A 0510 CC 061F      A MAN23 JMP   SSS      STORE STACK
00341A 0513 CC 05C9      A MAN24 JMP   SPRES    RESET ROUTINE
00342A 0516 CC 0726      A MAN10 JMP   ONOF
00343A 0519 CC 076E      A MAN11 JMP   JKT
00344A 051C CC 075E      A MAN12 JMP   JJT
00345A 051F CC 0766      A MAN21 JMP   DBUS
00346A 0522 CC 0BB2      A MAN25 JMP   RTCR     RS232 ROUTINE
00347A 0525 A6 FF        A MAN17 LDA   #$FF     RESET KKV
00348A 0527 B7 98        A     STA   KKV
00349A 0529 02 00 02 052E MAN18 BRSET 1,PORTA,MAN27
00350A 052C 11 12        A     BCLR  0,FLGN   REREAD KB FLAG
00351A 052E 01 19 06 0537 MAN27 BRCLR 0,GNFL,MAN19 RX TX ENABLE
00352A 0531 AD EF    0522        BSR   MAN25   RS232 ROUTINE
00353A 0533                    WAIT
00354A 0534 CC 04F4      A     JMP   MAN26
00355A 0537 CC 04F3      A MAN19 JMP   MAN7
```

```
00357
00358                    ********************************
00359                    *     INC COUNTER + DEC ADJUST *
00360                    *     X-ADDRESS OF LSB-5       *
00361                    *     A-NO. OF BYTES           *
00362                    ********************************
00363A 053A B7 13      A ICON   STA   AREG     SAVE ACCA
00364A 053C 6C 05      A ICON1  INC   5,X
00365A 053E E6 05      A        LDA   5,X
00366A 0540 AB 00      A        ADD   #$00     CARRY & H.CARRY=0
00367A 0542 AD 08    054C       BSR   ICON3    DAA
00368A 0544 24 05    054B       BCC   ICON2
00369A 0546 5A                  DEX
00370A 0547 3A 13      A        DEC   AREG
00371A 0549 26 F1    053C       BNE   ICON1
00372A 054B 81              ICON2 RTS
00373A 054C CC 0A6F    A ICON3  JMP   DAA
00374
00375                    ************************
00376                    *     COUNT SUBROUTINE *
00377                    ************************
00378A 054F 0B 27 76 05C8 CONT   BRCLR 5,CARR,CON6 CONT DISABLED
00379A 0552 07 15 2A 057F        BRCLR 3,FRED,CON3 TRIGER BIT
00380A 0555 AE 1C      A        LDX   #PDW-5
00381A 0557 A6 02      A        LDA   #$02     2BYTES
00382A 0559 AD DF    053A       BSR   ICON
00383A 055B AE 21      A        LDX   #TDW-5
00384A 055D A6 03      A        LDA   #$03
00385A 055F AD D9    053A       BSR   ICON
00386A 0561 10 15      A        BSET  0,FRED   UPDATE DISP
00387A 0563 14 15      A        BSET  2,FRED   UPDATE DISP
00388A 0565 B6 20      A        LDA   PDW-1
00389A 0567 B1 22      A        CMP   MDW-1
00390A 0569 22 08    0573       BHI   CON1
00391A 056B 26 10    057D       BNE   CON2
00392A 056D B6 21      A        LDA   PDW
00393A 056F B1 23      A        CMP   MDW
00394A 0571 23 0A    057D       BLS   CON2
00395A 0573 B6 21      A CON1   LDA   PDW
00396A 0575 B7 23      A        STA   MDW
00397A 0577 B6 20      A        LDA   PDW-1
00398A 0579 B7 22      A        STA   MDW-1
00399A 057B 12 15      A        BSET  1,FRED   UPDATE DISP
00400A 057D 17 15      A CON2   BCLR  3,FRED
00401A 057F 09 15 46 05C8 CON3   BRCLR 4,FRED,CON6 TRIGER BIT
00402A 0582 AE 24      A        LDX   #PDVA-5
00403A 0584 A6 02      A        LDA   #$02     2BYTES
00404A 0586 AD B2    053A       BSR   ICON
00405A 0588 AE 29      A        LDX   #TDVA-5
00406A 058A A6 03      A        LDA   #$03
00407A 058C AD AC    053A       BSR   ICON
00408A 058E 10 15      A        BSET  0,FRED   UPDATE DISP
00409A 0590 14 15      A        BSET  2,FRED   UPDATE DISP
00410A 0592 B6 28      A        LDA   PDVA-1
00411A 0594 B1 2A      A        CMP   MDVA-1
00412A 0596 22 68    05A0       BHI   CON4
00413A 0598 26 2C    05C6       BNE   CON5
00414A 059A B6 29      A        LDA   PDVA
00415A 059C B1 2B      A        CMP   MDVA
00416A 059E 23 26    05C6       BLS   CON5
00417A 05A0 B6 29      A CON4   LDA   PDVA
00418A 05A2 B7 2B      A        STA   MDVA
00419A 05A4 B6 28      A        LDA   PDVA-1
00420A 05A6 B7 2A      A        STA   MDVA-1
```

```
00421A 05A8 0E BA FD 05A8 CON7  BRSET 7,ARTC,CON7
00422A 05AB B6 B4       A       LDA   HRS
00423A 05AD B7 48       A       STA   MHH
00424A 05AF 0E BA FD 05AF CON8  BRSET 7,ARTC,CON8
00425A 05B2 B6 82       A       LDA   MIN
00426A 05B4 B7 49       A       STA   MMIN
00427A 05B6 0E BA FD 05B6 CON9  BRSET 7,ARTC,CON9
00428A 05B9 B6 87       A       LDA   DATE
00429A 05BB B7 46       A       STA   MDD
00430A 05BD 0E BA FD 05BD CONA  BRSET 7,ARTC,CONA
00431A 05C0 B6 88       A       LDA   MON
00432A 05C2 B7 47       A       STA   MMO
00433A 05C4 12 15       A       BSET  1,FRED  UPDATE DISP
00434A 05C6 19 15       A CON5  BCLR  4,FRED
00435A 05C8 81            CON6  RTS
00436
00437                     **********************
00438                     *     RESET ROUTINE *
00439                     **********************
00440A 05C9 0B 27 25 05F1 SPRES BRCLR 5,CARR,SPRES5 COUNT ENABLE BIT
00441A 05CC 0A 15 07 05D6       BRSET 5,FRED,SPRES3 RESET-TRIG
00442A 05CF 07 10 1F 05F1       BRCLR 3,FRTC,SPRES5 1MIN EXP BIT
00443A 05D2 13 10       A       BCLR  1,FRTC  RESET BIT CTR
00444A 05D4 17 10       A       BCLR  3,FRTC  1MIN EXP BIT
00445A 05D6 1C 15       A SPRES3 BSET 6,FRED  ENABLE STACK ROUTINE
00446A 05D8 AD 18    05F2       BSR   SPRES6  SSS STORE STACK
00447A 05DA 1B 15       A       BCLR  5,FRED  RESET ROUTINE
00448A 05DC 3F 21       A       CLR   PDW     CLR PULSE COUNTERS
00449A 05DE 3F 20       A       CLR   PDW-1
00450A 05E0 3F 29       A       CLR   PDVA
00451A 05E2 3F 28       A       CLR   PDVA-1
00452A 05E4 18 15       A       BSET  0,FRED  UPDATE DISP
00453A 05E6 3F 91       A       CLR   ACTR
00454A 05E8 0E BA FD 05E8 SPRES4 BRSET 7,ARTC,SPRES4
00455A 05EB B6 80       A       LDA   SEC
00456A 05ED B7 44       A       STA   ACTRS
00457A 05EF AD 07    05F8       BSR   IDEMP   INCREMENT DEMAND PER
00458A 05F1 81            SPRES5 RTS
00459A 05F2 CC 061F     A SPRES6 JMP   SSS
00460A 05F5 CC 06A9     A SPRES7 JMP   APPS
00461
00462                     ************************************
00463                     *     SUBROUTINE INCREMENT DEMAND *
00464                     *     NUMBER  A-CURRUPTED         *
00465                     ************************************
00466A 05F8 0D 27 04 05FF IDEMP  BRCLR 6,CARR,IDEMP1 DAYBCHANGE BIT
00467A 05FB 1D 27       A       BCLR  6,CARR  DAYBCHANGE BIT
00468A 05FD 3F 95       A       CLR   DEMP
00469A 05FF 3C 95       A IDEMP1 INC  DEMP
00470A 0601 B6 95       A       LDA   DEMP
00471A 0603 AD 05    060A       BSR   DAAC    DEC ADJUST
00472A 0605 B7 95       A       STA   DEMP
00473A 0607 10 10       A IDEMP2 BSET 0,FRTC  UPDATE DISP MD1
00474A 0609 81                   RTS
00475
00476                     ******************************
00477                     *     DECIMAL ADDJUST ACCA *
00478                     ******************************
00479A 060A B7 13       A DAAC  STA   AREG    SAVE ACCA
00480A 060C A4 0F       A       AND   #$0F    MUSK OFF MSB
00481A 060E A1 0A       A       CMP   #$0A
00482A 0610 26 0A    061C       BNE   DAAC2
00483A 0612 B6 13       A       LDA   AREG
00484A 0614 AB 06       A       ADD   #$06
```

```
00485A 0616 A1 99       A       CMP   #$99
00486A 0618 23 01    061B       BLS   DAAC1
00487A 061A 4F                  CLRA
00488A 061B 81             DAAC1 RTS
00489A 061C B6 13       A DAAC2 LDA   AREG
00490A 061E 81                  RTS
00491
00492                           ******************************
00493                           *     SUBROUTINE STORE STACK *
00494                           *     OF DATA IN MEMORY      *
00495                           ******************************
00496A 061F 0D 15 6E 0690 SSS   BRCLR 6,FRED,SS13 STORE ENABLE BIT
00497A 0622 0E 15 4C 0671       BRSET 7,FRED,SS11 SECOND HALF
00498A 0625 B6 27       A       LDA   CARR    DAY NO.
00499A 0627 A4 0F       A       AND   #$0F
00500A 0629 3D 20       A       TST   PDW-1   MSB=0?
00501A 062B 27 04    0631       BEQ   SS1
00502A 062D AB C0       A       ADD   #$C0    ENABLE BITS
00503A 062F 20 06    0637       BRA   SS2
00504A 0631 3D 21       A SS1   TST   PDW     LSB=0?
00505A 0633 27 02    0637       BEQ   SS2
00506A 0635 AB 40       A       ADD   #$40    ENABLE BIT
00507A 0637 3D 28       A SS2   TST   PDVA-1  MSB=0?
00508A 0639 27 04    063F       BEQ   SS3
00509A 063B AB 30       A       ADD   #$30    ENABLE BITS
00510A 063D 20 06    0645       BRA   SS4
00511A 063F 3D 29       A SS3   TST   PDVA    LSB=0?
00512A 0641 27 02    0645       BEQ   SS4
00513A 0643 AB 10       A       ADD   #$10    ENABLE BIT
00514A 0645 AE 9C       A SS4   LDX   #SAH
00515A 0647 AD 48    0691       BSR   SS14    INC ADD AND STORE
00516A 0649 3D 20       A       TST   PDW-1   MSB=0?
00517A 064B 26 06    0653       BNE   SS5
00518A 064D 3D 21       A       TST   PDW     LSB=0?
00519A 064F 26 06    0657       BNE   SS6
00520A 0651 20 08    065B       BRA   SS7
00521A 0653 B6 28       A SS5   LDA   PDW-1   MSB
00522A 0655 AD 3A    0691       BSR   SS14    INC ADD AND STORE
00523A 0657 B6 21       A SS6   LDA   PDW     LSB
00524A 0659 AD 36    0691       BSR   SS14    INC ADD AND STORE
00525A 065B 3D 28       A SS7   TST   PDVA-1  MSB=0?
00526A 065D 26 06    0665       BNE   SS8
00527A 065F 3D 29       A       TST   PDVA    LSB=0?
00528A 0661 26 06    0669       BNE   SS9
00529A 0663 20 08    066D       BRA   SS10
00530A 0665 B6 28       A SS8   LDA   PDVA-1
00531A 0667 AD 28    0691       BSR   SS14    INC ADD AND STORE
00532A 0669 B6 29       A SS9   LDA   PDVA
00533A 066B AD 24    0691       BSR   SS14    INC ADD AND STORE
00534A 066D B6 95       A SS10  LDA   DEMP    DEMAND PERIOD
00535A 066F AD 20    0691       BSR   SS14    INC ADD AND STORE
00536A 0671 0F 8A 04 0678 SS11  BRCLR 7,ARTC,SS12 UIP BIT
00537A 0674 1E 15       A       BSET  7,FRED  HALF BIT
00538A 0676 20 18    0690       BRA   SS13    RTS
00539A 0678 AE 9C       A SS12  LDX   #SAH
00540A 067A 0E 8A FD 067A SS15  BRSET 7,ARTC,SS15 RTC BUSY
00541A 067D B6 84       A       LDA   HRS     HOURS
00542A 067F AD 10    0691       BSR   SS14    INC ADD AND STORE
00543A 0681 0E 8A FD 0681 SS16  BRSET 7,ARTC,SS16 RTC BUSY
00544A 0684 B6 82       A       LDA   MIN     MINUTES
00545A 0686 AD 09    0691       BSR   SS14    INC ADD AND STORE
00546A 0688 A6 FF       A       LDA   #$FF    END CHR
00547A 068A AD 05    0691       BSR   SS14    FOR STACK
00548A 068C 1F 15       A       BCLR  7,FRED
```

```
0054A 068E 1D 15      A       BCLR  6,FRED
00550A 0690 81          SS13  RTS
00551A 0691 CC 0694     A SS14  JMP   IACO
00552
00553                   ********************************
00554                   *     INC ADD, CHECK OVER FLOW *
00555                   *     AND STORE IN MEMORY      *
00556                   *     X-ADD OF HIGH ADD        *
00557                   ********************************
00558A 0694 AD 18    06A6 IACO  BSR   IACO2   IADD INC ADDRESS
00559A 0696 24 0B    06A3       BCC   IACO1   IF MEM NOT FULL
00560A 0698 1E 12      A        BSET  7,FLGN  MEM FULL BIT
00561A 069A B7 13      A        STA   AREG
00562A 069C 7F                  CLR   ,X
00563A 069D A6 A0      A        LDA   #$A0
00564A 069F E7 01      A        STA   1,X
00565A 06A1 B6 13      A        LDA   AREG
00566A 06A3 AD 47    06EC IACO1 BSR   WHMR
00567A 06A5 81                  RTS
00568A 06A6 CC 06F8     A IACO2 JMP   IADD
00569
00570                   ********************************
00571                   *     SUBROUTINE MAKE ADD POINT *
00572                   *     TO PREVIOUS FF           *
00573                   *     X- ADD OF HIGH ADD       *
00574                   ********************************
00575A 06A9 AD 5F    070A APPS  BSR   DADD    DEC ADDRESS
00576A 06AB 25 06    06B3       BCS   APPS1   RTS
00577A 06AD AD 2D    06DC       BSR   RMMR
00578A 06AF A1 FF      A        CMP   #$FF
00579A 06B1 26 F6    06A9       BNE   APPS
00580A 06B3 81           APPS1  RTS
00581
00582                   ********************************
00583                   *     SUBROUTINE MAKE ADD POINT *
00584                   *     TO NEXT FF               *
00585                   *     X-ADD OF HIGH ADD        *
00586                   ********************************
00587A 06B4 AD 20    06D6 APNS  BSR   APNS4   APNS4 INC ADD
00588A 06B6 24 07    06BF       BCC   APNS2
00589A 06B8 7F                  CLR   ,X
00590A 06B9 A6 B0      A        LDA   #STMEM
00591A 06BB A0 10      A        SUB   #$10
00592A 06BD E7 01      A        STA   1,X
00593A 06BF AD 18    06D9 APNS2 BSR   APNS5   RMMR
00594A 06C1 A1 FF      A        CMP   #$FF
00595A 06C3 26 EF    06B4       BNE   APNS
00596A 06C5 F6                  LDA   ,X      MSB OF ADD
00597A 06C6 B1 9C      A        CMP   SAH     CHECK IF ADD IS=
00598A 06C8 26 08    06D2       BNE   APNS1   PRESENT STORE ADD
00599A 06CA E6 01      A        LDA   1,X     LSB OF ADD
00600A 06CC B1 9D      A        CMP   SAL
00601A 06CE 26 02    06D2       BNE   APNS1
00602A 06D0 20 01    06D3       BRA   APNS3   APPS
00603A 06D2 81           APNS1  RTS
00604A 06D3 CC 06A9     A APNS3 JMP   APPS    DEC UNTIL $FF+RET
00605A 06D6 CC 06F8     A APNS4 JMP   IADD
00606A 06D9 CC 06DC     A APNS5 JMP   RMMR
00607
00608                   ********************************
00609                   *     SUBROUTINE READ FROM MEM *
00610                   *     X-ADD OF HIGH ADD IN MEM *
00611                   *     AREG;XVD2;XREG-CURRUPTED *
00612                   *     RETURNS WITH             *
```

```
00613                   *     A-DATA;X-ADD OF HIGH ADD *
00614                   *********************************
00615A 06DC BF 96     A RMMR  STX   XVD2
00616A 06DE AE 08     A       LDX   #$08    READ MODE
00617A 06E0 BF 14     A       STX   XREG
00618A 06E2 BE 96     A       LDX   XVD2
00619A 06E4 AD 03   06E9      BSR   RMMR1   MRS
00620A 06E6 BE 96     A       LDX   XVD2
00621A 06E8 81                RTS
00622A 06E9 CC 0DB3   A RMMR1 JMP   MRS     MEM READ/WRITE
00623
00624                   *********************************
00625                   *     SUBROUTINE WRITE INTO MEM *
00626                   *     X-ADD OF HIGH ADD IN MEM  *
00627                   *     A-DATA                    *
00628                   *********************************
00629A 06EC BF 96     A WMMR  STX   XVD2    SAVE X
00630A 06EE 3F 14     A       CLR   XREG    WRITE MODE
00631A 06F0 AD 03   06F5      BSR   WMMR1   MRS
00632A 06F2 BE 96     A       LDX   XVD2    RETRIEVE X
00633A 06F4 81                RTS
00634A 06F5 CC 0DB3   A WMMR1 JMP   MRS     MEM READ/WRITE
00635
00636                   *********************************
00637                   *     SUBROUTINE INC TWO MEMLOC *
00638                   *     X-ADD OF HIGH ADD         *
00639                   *********************************
00640A 06F8 98          IADD  CLC             CLEAR CARRY
00641A 06F9 B7 13     A       STA   AREG
00642A 06FB 6C 01     A       INC   1,X
00643A 06FD 26 08   0707      BNE   IADD1
00644A 06FF 7C                INC   ,X
00645A 0700 F6                LDA   ,X
00646A 0701 A1 20     A       CMP   #$20    SETS CARRY
00647A 0703 98                CLC           ENSURE CLR
00648A 0704 26 01   0707      BNE   IADD1
00649A 0706 99                SEC
00650A 0707 B6 13     A IADD1 LDA   AREG
00651A 0709 81                RTS
00652
00653                   *********************************
00654                   *     SUBROUTINE DEC TWO MEM LOC *
00655                   *     X-ADD OF HIGH ADD         *
00656                   *********************************
00657A 070A 98          DADD  CLC
00658A 070B B7 13     A       STA   AREG
00659A 070D 7D                TST   ,X
00660A 070E 26 0A   071A      BNE   DADD1
00661A 0710 E6 01     A       LDA   1,X
00662A 0712 A1 B0     A       CMP   #STMEM  SETS CARRY
00663A 0714 98                CLC           ENSURE CLR
00664A 0715 26 03   071A      BNE   DADD1
00665A 0717 99                SEC
00666A 0718 20 09   0723      BRA   DADD2
00667A 071A 6A 01     A DADD1 DEC   1,X
00668A 071C A6 FF     A       LDA   #$FF
00669A 071E E1 01     A       CMP   1,X
00670A 0720 26 01   0723      BNE   DADD2
00671A 0722 7A                DEC   ,X
00672A 0723 B6 13     A DADD2 LDA   AREG
00673A 0725 81                RTS
00674
00675                   *********************************
00676                   *     SUBROUTINE ON/OFF SWITCH *
```

```
00677                    ********************************
0067BA 0726 05 10 34 075D ONOF   BRCLR  2,FRTC,ON5 PI BIT
0067BA 0729 04 00 0A 0736        BRSET  2,PORTA,ON4 ON/OFF PRESSED
0068BA 072C 02 12 2C 075B        BRSET  1,FLGN,ON3 DEBOUNCED BIT
0068lA 072F 00 11 08 073A        BRSET  0,FDIS,ON1 DEBOUNCE BIT
00682A 0732 10 11      A         BSET   0,FDIS
00683A 0734 20 25    075B        BRA    ON3
00684A 0736 13 12      A ON4     BCLR   1,FLGN   DEBOUNCED BIT
00685A 0738 20 21    075B        BRA    ON3
00686A 073A 11 11      A ON1     BCLR   0,FDIS   DEBOUNCE BIT
00687A 073C 04 00 1C 075B        BRSET  2,PORTA,ON3
00688A 073F 12 12      A         BSET   1,FLGN   DEBOUNCED BIT
00689A 0741 04 11 0B 074F        BRSET  2,FDIS,ON2
00690A 0744 14 11      A         BSET   2,FDIS
00691A 0746 1A 01      A         BSET   5,PORTB  KB-OFF
00692A 0748 A6 08      A         LDA    #%00001000 SWITCH DISP OFF
00693A 074A C7 0100    A         STA    DINS
00694A 074D 20 0C    075B        BRA    ON3
00695A 074F 15 11      A ON2     BCLR   2,FDIS
00696A 0751 1B 01      A         BCLR   5,PORTB  KB-ON
00697A 0753 12 11      A         BSET   1,FDIS   1SISTIMENSWITCH ON
00698A 0755 B6 99      A         LDA    MMRD     MEMORY OF MODE
00699A 0757 B7 93      A         STA    MRD      MODE REG
00700A 0759 3F 45      A         CLR    OOCTR    SWITCH OFF CTR
00701A 075B 15 10      A ON3     BCLR   2,FRTC
00702A 075D 81           ON5     RTS
00703
00704                    ********************************
00705                    *     SUBROUTINE JMP TO JMP TAB  *
00706                    *     FOR SPECIFIC MODE FOR DISP *
00707                    *     X-MODE                     *
00708                    ********************************
00709A 075E B7 13      A JJT     STA    AREG     SAVE AREG
00710A 0760 9F                   TXA
00711A 0761 A4 7F      A         AND    #$7F     MUSK OFF SET BIT
00712A 0763 B7 14      A         STA    XREG
00713A 0765 48                   LSLA            *2
00714A 0766 BB 14      A         ADD    XREG     +XREG=*3
00715A 0768 97                   TAX
00716A 0769 B6 13      A         LDA    AREG
00717A 076B DC 1A40    A         JMP    JMP,X    JMP TO APP JMP INS
00718                    *                      IN JMP TAB
00719
00720                    ********************************
00721                    *     SUBROUTINE JMP TO JMP TAB *
00722                    *     FOR SPECIFIC KEY PRESSED  *
00723                    *     KKV-KEY VAL               *
00724                    ********************************
00725A 076E B7 13      A JKT     STA    AREG     SAVE AREG
00726A 0770 B6 90      A         LDA    KKV      KEY VAL
00727A 0772 A1 09      A         CMP    #$09     NOS OR FUNCTIONS
00728A 0774 22 03    0779        BHI    JKT1
00729A 0776 5F                   CLRX
00730A 0777 20 08    0781        BRA    JKT2
00731A 0779 A0 09      A JKT1    SUB    #$09
00732A 077B B7 14      A         STA    XREG
00733A 077D 48                   LSLA
00734A 077E BB 14      A         ADD    XREG     *3
00735A 0780 97                   TAX
00736A 0781 B6 13      A JKT2    LDA    AREG
00737A 0783 DC 1B40    A         JMP    JMK,X    TO JMP TAB
00738
00739                    ********************************
00740                    *     SUBROUTINE CHECK IF DISP BUSY *
```

```
00741                   *       AND RETURN WHEN NOT BUSY      *
00742                   ************************************
00743
00744A 0786 B7 13       A DBUS  STA   AREG      SAVE ACC
00745A 0788 C6 0100     A DBU1  LDA   DINS      ADD & BUSY FLAG
00746A 078B B7 94       A       STA   DAR
00747A 078D 0E 94 F8 0788       BRSET 7,DAR,DBU1
00748A 0790 B6 13       A       LDA   AREG      RETRIEVE ACC
00749A 0792 81                  RTS
00750
00751                   ********************************
00752                   *       SUBROUTINE GET 2 NUMBERS *
00753                   *       FROM DISP                *
00754                   *       X-ADDRESS OF MSB         *
00755                   ********************************
00756
00757A 0793 AD 10    07A5 GTCD  BSR   GACD      GET A CHR
00758A 0795 A4 0F       A       AND   #$0F      ASCII TO BCD
00759A 0797 48                  LSLA
00760A 0798 48                  LSLA
00761A 0799 48                  LSLA
00762A 079A 48                  LSLA
00763A 079B B7 14       A       STA   XREG      SAVE MSB
00764A 079D 5C                  INX
00765A 079E AD 05    07A5       BSR   GACD      GET A CHR
00766A 07A0 A4 0F       A       AND   #$0F      ASCII TO BCD
00767A 07A2 BA 14       A       ORA   XREG      COMBINE MSB & LSB
00768A 07A4 81                  RTS
00769                   *       A-2 NUMBERS
00770                   *       X-ADD OF LSB IN DISP
00771
00772                   ************************************
00773                   *       SUBROUTINE GET A CHR FROM DISP *
00774                   *       X-ADD OF CHR                   *
00775                   ************************************
00776A 07A5 9F          GACD    TXA
00777A 07A6 AB 80       A       ADD   #$80
00778A 07A8 AD DC    0786       BSR   DBUS
00779A 07AA C7 0100     A       STA   DINS
00780A 07AD AD D7    0786       BSR   DBUS
00781A 07AF C6 0180     A       LDA   DDAT
00782A 07B2 81                  RTS
00783                   *       A-ASCII CHR
00784                   *       X-ADD OF CHR
00785
00786                   ************************************
00787                   *       SUBROUTINE STORE 2 NOS IN DISP *
00788                   *       A- CONTAINS THE 2 NUMBERS      *
00789                   *       X- CONTAINS THE START DISP ADD *
00790                   ************************************
00791
00792A 07B3 B7 14       A STCD  STA   XREG      SAVE 2 NOS
00793A 07B5 A4 F0       A       AND   #$F0      MASK OFF LS4B
00794A 07B7 44                  LSRA
00795A 07B8 44                  LSRA
00796A 07B9 44                  LSRA
00797A 07BA 44                  LSRA
00798A 07BB AB 30       A       ADD   #$30      CONVERT TO ASCII
00799A 07BD AD 13    07D2       BSR   STACH
00800A 07BF B6 14       A       LDA   XREG      GET 2 NOS AGAIN
00801A 07C1 A4 0F       A       AND   #$0F      DO FOR LSNIBBLE
00802A 07C3 AB 30       A       ADD   #$30      TO ASCII
00803A 07C5 5C                  INX
00804A 07C6 0B 11 03 07CC       BRCLR 5,FDIS,STCD1 SKIP POINT
```

```
00805A 07C9 5C                  INX
00806A 07CA 1B 11       A       BCLR  5,FDIS
00807A 07CC AD 04   07D2 STCD1  BSR   STACH
00808A 07CE 81                  RTS
00809                   *       A-ASCII OF SEC CHR
00810                   *       X-ADD OF SEC CHR
00811
00812                   ****************************
00813                   *       JUMP TO SUBROUTINES *
00814                   ****************************
00815
00816A 07CF CC 0786      A JM1   JMP   DBUS
00817
00818                   ****************************************
00819                   *       SUBROUTINE STORE A CHR INTO DISP *
00820                   *       X- OFFSET ADDRESS OF CHR         *
00821                   ****************************************
00822
00823A 07D2 AD FB    07CF STACH  BSR   JM1
00824A 07D4 B7 13       A       STA   AREG
00825A 07D6 9F                  TXA
00826A 07D7 AB 80       A       ADD   #$80      FIX FOR DISP ADDRESS
00827A 07D9 C7 0100     A       STA   DINS      LOCATE CURSER IN DISP
00828A 07DC B6 13       A       LDA   AREG
00829A 07DE AD EF    07CF       BSR   JM1
00830A 07E0 C7 0180     A       STA   DDAT
00831A 07E3 81                  RTS
00832                   *       A-CHR
00833                   *       X-DISP ADD
00834
00835                   ****************************
00836                   *       PREPAIR XREG FOR MODE *
00837                   *       ACC+INDEX-CORRUPTED   *
00838                   ****************************
00839A 07E4 B6 93       A PXM   LDA   MRD       MODE REGISTER
00840A 07E6 A4 7F       A       AND   #$7F      MASK OFF SET BIT
00841A 07E8 B7 13       A       STA   AREG      SAVE AREG
00842A 07EA 48                  LSLA            *3
00843A 07EB BB 13       A       ADD   AREG
00844A 07ED 97                  TAX
00845A 07EE 81                  RTS
00846
00847                   ***********************************
00848                   *       INC MEM ADD, CHECK IF FULL *
00849                   *       AND READ FROM MEMORY       *
00850                   *       X- ADD OF HIGH ADD         *
00851                   ***********************************
00852A 07EF AD 0C    07FD IACR  BSR   IACR2   IADD
00853A 07F1 24 07    07FA       BCC   IACR1   IF MEM NOT FULL
00854A 07F3 1E 12       A       BSET  7,FLGN  MEM FULL BIT
00855A 07F5 7F                  CLR   ,X
00856A 07F6 A6 A0       A       LDA   #$A0
00857A 07F8 E7 01       A       STA   1,X
00858A 07FA AD 04    0800 IACR1 BSR   IACR3   RMMR
00859A 07FC 81                  RTS
00860A 07FD CC 06F8     A IACR2 JMP   IADD     INC ADD
00861A 0800 CC 06DC     A IACR3 JMP   RMMR     READ FROM MEM
00862
00863                   ***********************************
00864                   *       IND TO ASCII,SUM CHECK,    *
00865                   *       AND STORE INTO MEM BLOCK   *
00866                   *       A-MD                       *
00867                   ***********************************
00868A 0803 B7 13       A STAM  STA   AREG      SAVE ACCA
```

```
00869A 0805 B8 1C      A        EOR   CREG     UPDATE SUM CHECK
00870A 0807 B7 1C      A        STA   CREG
00871A 0809 B6 13      A        LDA   AREG
00872A 080B AB 30      A        ADD   #$30     TO ASCII
00873A 080D AE 1E      A        LDX   #RSH1    BUFF BLOCK
00874A 080F AD 01   0812        BSR   STAM1    IACD
00875A 0811 81                  RTS
00876A 0812 CC 0694    A STAM1  JMP   IACD     INC AND STORE
00877
00878                   ********************************
00879                   *     2NOS TO ASCII,SUM CHECK, *
00880                   *     AND STORE INTO MEM BUFF  *
00881                   *     A-2NOS                   *
00882                   ********************************
00883A 0815 B7 1B      A STNM   STA   BREG
00884A 0817 A4 F0      A        AND   #$F0
00885A 0819 44                  LSRA
00886A 081A 44                  LSRA
00887A 081B 44                  LSRA
00888A 081C 44                  LSRA
00889A 081D AD E4   0803        BSR   STAM
00890A 081F B6 1B      A        LDA   BREG
00891A 0821 A4 0F      A        AND   #$0F
00892A 0823 AD DE   0803        BSR   STAM
00893A 0825 81                  RTS
00894
00895                   ********************************
00896                   *     STORE FOUR BYTE DATA NO  *
00897                   ********************************
00898A 0826 B7 1B      A STNM1  STA   BREG
00899A 0828 B6 1F      A        LDA   RSL1
00900A 082A A1 96      A        CMP   #$96     MSD OF NO
00901A 082C 25 1C   084A        BLO   STN12
00902A 082E 27 16   0846        BEQ   STN11
00903A 0830 A1 98      A        CMP   #$98     MSD OF NO
00904A 0832 27 09   083D        BEQ   STN1
00905A 0834 A1 9A      A        CMP   #$9A     MSD OF NO
00906A 0836 27 0E   0846        BEQ   STN11
00907A 0838 A1 9C      A        CMP   #$9C     MSD OF NO
00908A 083A 27 01   083D        BEQ   STN1
00909A 083C 81                  RTS
00910A 083D B6 1B      A STN1   LDA   BREG
00911A 083F 09 19 08 084A       BRCLR 4,GNFL,STN12
00912A 0842 19 19      A        BCLR  4,GNFL
00913A 0844 20 CF   0815        BRA   STNM
00914A 0846 B6 1B      A STN11  LDA   BREG
00915A 0848 20 1C   0866        BRA   STNM3
00916A 084A B6 1B      A STN12  LDA   BREG
00917A 084C 20 00   084E        BRA   STNM2
00918
00919                   ****************************
00920                   *     IF ZERO STORE IN MEM *
00921                   *     A- DATA              *
00922                   ****************************
00923A 084E B7 1B      A STNM2  STA   BREG
00924A 0850 A4 F0      A        AND   #$F0
00925A 0852 27 05   0859        BEQ   STN2
00926A 0854 B6 1B      A        LDA   BREG
00927A 0856 AD BD   0815        BSR   STNM
00928A 0858 81                  RTS
00929A 0859 4F            STN2  CLRA
00930A 085A AE 1E      A        LDX   #RSH1
00931A 085C AD 05   0863        BSR   STN21    IACD
00932A 085E B6 1B      A        LDA   BREG
```

```
00933A 0868 AD A1    0803        BSR   STAM
00934A 086C 81                   RTS
00935A 086D CC 0694    A STN21   JMP   IACO
00936
00937                    ****************************
00938                    *     STORER IN MEM BUF FOR *
00939                    *     FOUR NOS-SET BIT      *
00940                    ****************************
00941A 0866 B7 1B      A STNM3   STA   BREG
00942A 0868 A4 F0      A         AND   #$F0      MUSK OFF LSB
00943A 086A 27 07    0873        BEQ   STN3
00944A 086C 18 19      A         BSET  4,GNFL    BIT FOR MSN OF 4 DIG
00945A 086E B6 1B      A         LDA   BREG
00946A 0870 AD A3    0815        BSR   STNM
00947A 0872 81                   RTS
00948A 0873 B6 1B      A STN3    LDA   BREG
00949A 0875 4D                   TSTA
00950A 0876 26 0D    0885        BNE   STN31
00951A 0878 4F                   CLRA
00952A 0879 AE 1E      A         LDX   #RSH1
00953A 087B AD E6    0863        BSR   STN21     IACO
00954A 087D 4F                   CLRA
00955A 087E AE 1E      A         LDX   #RSH1
00956A 0880 AD E1    0863        BSR   STN21     IACO
00957A 0882 19 19      A         BCLR  4,GNFL
00958A 0884 81                   RTS
00959A 0885 18 19      A STN31   BSET  4,GNFL
00960A 0887 20 D0    0859        BRA   STN2
00961
00962                    ************************
00963                    *     CLR MEM BYTES    *
00964                    *     X-ADD OFFSET     *
00965                    *     A-NO OF BNYTES   *
00966                    ************************
00967A 0889 B7 41      A CTBM    STA   EREG
00968A 088B 4F                   CLRA
00969A 088C AD 05    0893 CTBM1  BSR   CTBM2     IACO
00970A 088E 3A 41      A         DEC   EREG
00971A 0890 26 FA    088C        BNE   CTBM1
00972A 0892 81                   RTS
00973A 0893 CC 0694    A CTBM2   JMP   IACO
00974
00975                    ************************************
00976                    *     SUBROUTINE WHEN NO IS PRESSED *
00977                    ************************************
00978A 0896 0F 93 12 08AB KBN    BRCLR 7,MRD,KBN1 SET BIT
00979A 0899 B6 90      A         LDA   KKV       PRINT NO INTO DISP
00980A 089B AB 30      A         ADD   #$30      TO ASCII A-CHR$
00981A 089D CE 0100    A         LDX   DINS      X-OFFSET ADD
00982A 08A0 AD 0D    08AF        BSR   KBN3      STACH
00983A 08A2 BF 14      A         STX   XREG      SAVE ADDRESS
00984A 08A4 AD 06    08AC        BSR   KBN2      PXM
00985A 08A6 B6 14      A         LDA   XREG      CONTAINS PREVIOUS LOC
00986A 08A8 DC 1980    A         JMP   JNT,X
00987A 08AB 81           KBN1    RTS
00988A 08AC CC 07E4    A KBN2    JMP   PXM
00989A 08AF CC 07D2    A KBN3    JMP   STACH
00990
00991                    ********************************
00992                    *     SUBROUTINE MOVE TO TOP LOC *
00993                    ********************************
00994A 08B2 0F 93 05 08BA KBA    BRCLR 7,MRD,KBA1 IF NOT IN SET MODE
00995A 08B5 AD F5    08AC        BSR   KBN2      PXM XREG FOR MODE
00996A 08B7 DC 1980    A         JMP   JAT,X
```

```
00997A 08BA 81          KBA1  RTS
00998
00999                 ********************************
01000                 *     SUBROUTINE CURSER MOVE TO *
01001                 *     PREVIOUS LOCATION         *
01002                 ********************************
01003A 08BB 0F 93 05 08C3 KBB   BRCLR 7,MRD,KBB1 IF NOT IN SET MODE
01004A 08BE AD EC    08AC       BSR   KBN2    PXM XREG FOR MODE
01005A 08C0 DC 1948    A        JMP   JBT,X
01006A 08C3 81          KBB1  RTS
01007
01008                 ********************************
01009                 *     SUBROUTINE CURSER MOVE TO *
01010                 *     NEXT LOCATION             *
01011                 ********************************
01012A 08C4 0F 93 0A 08D1 KBE   BRCLR 7,MRD,KBE1 IF NOT IN SET MODE
01013A 08C7 AD E3    08AC       BSR   KBN2    PXM XREG FOR MODE
01014A 08C9 C6 0100    A        LDA   DINS
01015A 08CC A4 7F      A        AND   #$7F
01016A 08CE DC 1980    A        JMP   JNT,X
01017A 08D1 81          KBE1  RTS
01018
01019                 ********************************
01020                 *     SUBROUTINE CHANGE COMMAND *
01021                 ********************************
01022A 08D2 0E 93 19 08EE KBC   BRSET 7,MRD,KBC2 IF NOT IN SET MODE
01023A 08D5 BE 93      A        LDX   MRD
01024A 08D7 D6 18E0    A        LDA   ESM,X   CHECK IF ENABLE IS
01025A 08DA 27 11    08ED       BEQ   KBC1    SET FOR MODE
01026A 08DC 1E 93      A        BSET  7,MRD
01027A 08DE D6 1A00    A        LDA   FAV,X
01028A 08E1 AB 80      A        ADD   #$80
01029A 08E3 97                  TAX
01030A 08E4 AD 29    090F       BSR   MCX     MOVE CURSER
01031A 08E6 A6 0D      A        LDA   #%00001101 SWITCH BLINK ON
01032A 08E8 AD 0F    08F9       BSR   KBC3
01033A 08EA C7 0100    A        STA   DINS
01034A 08ED 81          KBC1  RTS
01035A 08EE 1F 93      A KBC2   BCLR  7,MRD
01036A 08F0 B6 93      A        LDA   MRD
01037A 08F2 48                  LSLA          *3
01038A 08F3 BB 93      A        ADD   MRD
01039A 08F5 97                  TAX
01040A 08F6 DC 1880    A        JMP   SWJT,X
01041A 08F9 CC 0786    A KBC3   JMP   DBUS
01042
01043                 ************************
01044                 *     MODE CHANGE KEY  *
01045                 ************************
01046A 08FC 0E 93 0C 090B KBD   BRSET 7,MRD,KBD1 IF IN SET MODE
01047A 08FF B6 93      A        LDA   MRD     SAVE PREV MODE
01048A 0901 B7 99      A        STA   MMRD
01049A 0903 AD 07    090C       BSR   KBD2    UDA
01050A 0905 A6 8A      A        LDA   #$8A    ENABLE MODE SELECT
01051A 0907 B7 93      A        STA   MRD     MODE
01052A 0909 16 11      A        BSET  3,FDIS  FIRST TIME ENTER MODE
01053A 090B 81          KBD1  RTS
01054A 090C CC 100C    A KBD2   JMP   UDA
01055
01056                 ******************************
01057                 *     SUBROUTINE MOVE CURSER *
01058                 *     X-LOC ON DISP          *
01059                 *     D7 OF X=1              *
01060                 ******************************
```

```
01061A 090F AD 04    0915 MCX   BSR   MCX1
01062A 0911 CF 0100     A       STX   DINS
01063A 0914 81                  RTS
01064A 0915 CC 07B6     A MCX1  JMP   DBUS
01065
01066                     ********************************
01067                     *     STORE TWO CHRS AND OMMIT *
01068                     *     LEADING ZEROS            *
01069                     *     A-2 CHRS                 *
01070                     *     X- ADD IN DISP           *
01071                     ********************************
01072A 0918 B7 1B       A STCL  STA   BREG
01073A 091A AD 52    096E       BSR   ST11     GACD
01074A 091C A1 2E       A       CMP   #$2E     .
01075A 091E 27 16    0938       BEQ   ST8
01076A 0920 5C                  INX
01077A 0921 AD 4B    096E       BSR   ST11     GACD
01078A 0923 A1 2E       A       CMP   #$2E     .
01079A 0925 26 19    0940       BNE   ST10
01080A 0927 B6 1B       A       LDA   BREG
01081A 0929 5A                  DEX
01082A 092A 1A 11       A       BSET  5,FDIS   POINT IN DISP
01083A 092C AD 3A    0968       BSR   ST6      STCD
01084A 092E 20 28    0958       BRA   ST2
01085A 0930 0C 11 09 093C ST8   BRSET 6,FDIS,ST9
01086A 0933 1C 11       A       BSET  6,FDIS
01087A 0935 5A                  DEX
01088A 0936 A6 30       A       LDA   #$30     "0"
01089A 0938 AD 2B    0965       BSR   ST5      STACH
01090A 093A AD 2F    096B       BSR   ST7      MTN4
01091A 093C AD 2D    096B ST9   BSR   ST7      MTN4
01092A 093E 20 01    0941       BRA   ST12
01093A 0940 5A             ST10 DEX
01094A 0941 B6 1B       A ST12  LDA   BREG
01095A 0943 0C 11 10 0956       BRSET 6,FDIS,ST1 LEAD ZERO
01096A 0946 A1 00       A       CMP   #$00
01097A 0948 27 12    095C       BEQ   ST3
01098A 094A A1 09       A       CMP   #$09     LSB OR BOTH
01099A 094C 22 08    0956       BHI   ST1
01100A 094E AB 30       A       ADD   #$30     TO ASCII
01101A 0950 AD 19    096B       BSR   ST7      MTN4
01102A 0952 AD 11    0965       BSR   ST5      STACH
01103A 0954 20 02    0958       BRA   ST2
01104A 0956 AD 10    0968 ST1   BSR   ST6      STCD
01105A 0958 1C 11       A ST2   BSET  6,FDIS   LEAD ZERO
01106A 095A 20 08    0964       BRA   ST4
01107A 095C A6 20       A ST3   LDA   #$20     SPACE CHR TO CLR
01108A 095E AD 05    0965       BSR   ST5      STACH
01109A 0960 AD 09    096B       BSR   ST7      MTN4
01110A 0962 AD 01    0965       BSR   ST5
01111A 0964 81             ST4  RTS
01112A 0965 CC 07B2     A ST5   JMP   STACH
01113A 0968 CC 07B3     A ST6   JMP   STCD
01114A 096B CC 10A5     A ST7   JMP   MTN4
01115A 096E CC 07A5     A ST11  JMP   GACD
```

```
01117
01118                    ********************************
01119                    *     SUB UPDATE MULTIPLICAND *
01120                    ********************************
01121A 0971 3A 14      A UMD   DEC   XREG     KWh KVAh MEM
01122A 0973 34 13      A       LSR   AREG     CHECK IF PARTICULAR
01123A 0975 24 08   097F        BCC   UM1      VALUE MUST BE MULLED
01124A 0977 BE 14      A       LDX   XREG     LD INTO MULTIPLICAND
01125A 0979 F6                 LDA   ,X
01126A 097A BE 96      A       LDX   XVD2
01127A 097C F7                 STA   ,X
01128A 097D 3A 96      A       DEC   XVD2
01129A 097F 81           UM1   RTS
01130
01131                    ************************************
01132                    *     READ DATA ROUTINE FROM BK MEM *
01133                    *     MADH MADL - MEM ADDRESS       *
01134                    ************************************
01135A 0980 3C 98      A RMEM  INC   MADL
01136A 0982 AE 08      A       LDX   #$08     READ
01137A 0984 BF 14      A       STX   XREG
01138A 0986 AE 97      A       LDX   #MADH    ADD OF MEM ADD
01139A 0988 AD 01   098B        BSR   RMEM1    MRS
01140A 098A 81                 RTS
01141A 098B CC 0DB3    A RMEM1 JMP   MRS
01142
01143                    ********************************
01144                    *     DO MUL FOR DKW AND DKVA *
01145                    ********************************
01146A 098E AE BD      A WVM   LDX   #$BD     CLR KVA MULTIPLICAND
01147A 0990 7F           WV1   CLR   ,X       MEMORY
01148A 0991 5A                 DEX
01149A 0992 A3 B8      A       CPX   #$B8
01150A 0994 24 FA   0990        BHS   WV1
01151A 0996 A6 BD      A       LDA   #$BD     KVAh MEM BLOCK
01152A 0998 B7 96      A       STA   XVD2
01153A 099A A6 2F      A       LDA   #$2F     KVAh READINGS
01154A 099C B7 14      A       STA   XREG
01155A 099E B6 2F      A       LDA   MEL
01156A 09A0 B7 13      A       STA   AREG
01157A 09A2 34 13      A       LSR   AREG
01158A 09A4 AD CB   0971 WV4   BSR   UMD      UPDATE MULTIPLICAND
01159A 09A6 B6 14      A       LDA   XREG
01160A 09A8 A1 28      A       CMP   #$28
01161A 09AA 26 F8   09A4        BNE   WV4
01162A 09AC B6 2F      A       LDA   MEL      CHECK MODE 5
01163A 09AE A1 30      A       CMP   #$30
01164A 09B0 27 20   09D2        BEQ   WV7
01165A 09B2 AE AD      A       LDX   #$AD     CLR KW
01166A 09B4 7F           WV2   CLR   ,X       MULTIPLICAND
01167A 09B5 5A                 DEX            MEMORY
01168A 09B6 A3 A8      A       CPX   #$A8
01169A 09B8 24 FA   09B4        BHS   WV2
01170A 09BA A6 AD      A       LDA   #$AD     KWh MEM BLOCK
01171A 09BC B7 96      A       STA   XVD2
01172A 09BE A6 27      A       LDA   #$27     KWh READINGS
01173A 09C0 B7 14      A       STA   XREG
01174A 09C2 B6 2F      A       LDA   MEL
01175A 09C4 B7 13      A       STA   AREG
01176A 09C6 34 13      A       LSR   AREG
01177A 09C8 AD A7   0971 WV3   BSR   UMD      UPDATE MULTIPLICAND
01178A 09CA B6 14      A       LDA   XREG
01179A 09CC A1 20      A       CMP   #$20     WHOLE READINGS DONE
01180A 09CE 26 F8   09C8        BNE   WV3
```

```
01181A 09D8 20 56    0A28        BRA   WV6
01182A 09D2 B6 92      A WV7     LDA   SPR       GET CORRECT MUL
01183A 09D4 26 04    09DA        BNE   WV8       VALUE FOR SAMPLE
01184A 09D6 A6 60      A         LDA   #$60      PERIOD REG
01185A 09D8 20 2A    0A04        BRA   WVE
01186A 09DA A1 05      A WV8     CMP   #$05
01187A 09DC 22 04    09E2        BHI   WV9
01188A 09DE A6 12      A         LDA   #$12
01189A 09E0 20 22    0A04        BRA   WVE
01190A 09E2 A1 10      A WV9     CMP   #$10
01191A 09E4 22 04    09EA        BHI   WVA
01192A 09E6 A6 06      A         LDA   #$06
01193A 09E8 20 1A    0A04        BRA   WVE
01194A 09EA A1 15      A WVA     CMP   #$15
01195A 09EC 22 04    09F2        BHI   WVB
01196A 09EE A6 04      A         LDA   #$04
01197A 09F0 20 12    0A04        BRA   WVE
01198A 09F2 A1 20      A WVB     CMP   #$20
01199A 09F4 22 04    09FA        BHI   WVC
01200A 09F6 A6 03      A         LDA   #$03
01201A 09F8 20 0A    0A04        BRA   WVE
01202A 09FA A1 30      A WVC     CMP   #$30
01203A 09FC 22 04    0A02        BHI   WVD
01204A 09FE A6 02      A         LDA   #$02
01205A 0A00 20 02    0A04        BRA   WVE
01206A 0A02 A6 01      A WVD     LDA   #$01
01207A 0A04 AE B0      A WVE     LDX   #DKVA
01208A 0A06 E7 07      A         STA   7,X
01209A 0A08 6F 06      A         CLR   6,X
01210A 0A0A AD 49    0A55        BSR   WV5       MUL
01211A 0A0C AE BD      A         LDX   #$BD
01212A 0A0E 7F           WVF     CLR   ,X
01213A 0A0F 5A                   DEX
01214A 0A10 A3 B8      A         CPX   #$B8
01215A 0A12 24 FA    0A0E        BHS   WVF
01216A 0A14 AE B0      A         LDX   #DKVA
01217A 0A16 E6 05      A         LDA   5,X
01218A 0A18 E7 0D      A         STA   $D,X
01219A 0A1A E6 04      A         LDA   4,X
01220A 0A1C E7 0C      A         STA   $C,X
01221A 0A1E E6 03      A         LDA   3,X
01222A 0A20 E7 0B      A         STA   $B,X
01223A 0A22 A6 89      A         LDA   #$89
01224A 0A24 B7 98      A         STA   MADL
01225A 0A26 20 12    0A3A        BRA   WV10
01226A 0A28 3F 97      A WV6     CLR   MADH      MEM ADD HIGH
01227A 0A2A A6 87      A         LDA   #$87
01228A 0A2C B7 98      A         STA   MADL
01229A 0A2E AD 28    0A58        BSR   WV12      RMEM
01230A 0A30 AE A0      A         LDX   #DKW
01231A 0A32 E7 06      A         STA   6,X
01232A 0A34 AD 22    0A58        BSR   WV12      RMEM
01233A 0A36 AE A0      A         LDX   #DKW
01234A 0A38 E7 07      A         STA   7,X
01235A 0A3A AD 1C    0A58 WV10   BSR   WV12      RMEM
01236A 0A3C AE B0      A         LDX   #DKVA
01237A 0A3E E7 06      A         STA   6,X
01238A 0A40 AD 16    0A58        BSR   WV12      RMEM
01239A 0A42 AE B0      A         LDX   #DKVA
01240A 0A44 E7 07      A         STA   7,X
01241A 0A46 B6 2F      A         LDA   MEL       CHECK MODE 5
01242A 0A48 A1 30      A         CMP   #$30
01243A 0A4A 27 04    0A50        BEQ   WV11
01244A 0A4C AE A0      A         LDX   #DKW
```

```
01245A 0A4E AD 05    0A55        BSR   MV5      MUL
01246A 0A50 AE B0      A MV11    LDX   #DKVA
01247A 0A52 AD 01    0A55        BSR   MV5      MUL
01248A 0A54 81                   RTS
01249A 0A55 CC 0AD7    A MV5     JMP   MUL
01250A 0A58 CC 0980    A MV12    JMP   RMEM
01251
01252                    ***********************************
01253                    *     ADDING OF 1-8 BYTES OF NOS  *
01254                    *     X-ADDRESS OF MEM BANK       *
01255                    ***********************************
01256A 0A5B B7 13       A ADDB    STA   AREG     SAVE ACCA
01257A 0A5D 98                    CLC           CLEAR CARRY
01258A 0A5E E6 05       A ADD1    LDA   5,X      GET SUM
01259A 0A60 E9 0D       A         ADC   $D,X     ADD TO SUM
01260A 0A62 E7 05       A         STA   5,X      STORE IN SUM
01261A 0A64 AD 09    0A6F         BSR   DAA      DEC ADJUST
01262A 0A66 5A                    DEX
01263A 0A67 9F                    TXA
01264A 0A68 A4 05       A         AND   #$05
01265A 0A6A 26 F2    0A5E         BNE   ADD1     DO NEXT BYTE
01266A 0A6C B6 13       A         LDA   AREG     POP ACCA
01267A 0A6E 81                    RTS
01268
01269                    ******************************
01270                    *     SUB ROUTINE DEC ADJUST *
01271                    ******************************
01272A 0A6F 39 27       A DAA     ROL   CARR     STORE CARRY
01273A 0A71 29 07    0A7A         BHCS  DA1      ADD 6 IF HALF C
01274A 0A73 43                    COMA          USE THE COM
01275A 0A74 A4 0F       A         AND   #$0F     TO HOLD CARRY STAT
01276A 0A76 A1 06       A         CMP   #$06     WHEN NO HALF C
01277A 0A78 24 06    0A80         BHS   DA2      DO NOT ADD 6 IF<9
01278A 0A7A E6 05       A DA1     LDA   5,X      ADD 6 TO LSN
01279A 0A7C AB 06       A         ADD   #$06
01280A 0A7E E7 05       A         STA   5,X
01281A 0A80 36 27       A DA2     ROR   CARR     GET STORED CARRY
01282A 0A82 25 09    0A8D         BCS   DA3      ADD 60 IF MAIN C
01283A 0A84 E6 05       A         LDA   5,X
01284A 0A86 43                    COMA          CHECK IF >9
01285A 0A87 A4 F0       A         AND   #$F0
01286A 0A89 A1 60       A         CMP   #$60
01287A 0A8B 24 07    0A94         BHS   DA4      DO NOT ADD IF<9
01288A 0A8D E6 05       A DA3     LDA   5,X      ADD 60 TO SUM
01289A 0A8F AB 60       A         ADD   #$60
01290A 0A91 E7 05       A         STA   5,X
01291A 0A93 99                    SEC           ENSURE CARRY SET
01292A 0A94 81          DA4       RTS
01293
01294                    ************************************
01295                    *     ROTATE LEFT OF 6 BYTES IN MEM  *
01296                    *     X-ADDRESS BANK                 *
01297                    ************************************
01298A 0A95 B7 13       A ROLB    STA   AREG     SAVE ACCA
01299A 0A97 98                    CLC
01300A 0A98 69 0D       A ROL1    ROL   $D,X     ROTATE BYTE
01301A 0A9A 39 27       A         ROL   CARR     SAVE CARRY
01302A 0A9C 5A                    DEX
01303A 0A9D 9F                    TXA
01304A 0A9E AB 06       A         ADD   #$06
01305A 0AA0 36 27       A         ROR   CARR     RETRIEVE CARRY
01306A 0AA2 A4 0F       A         AND   #$0F
01307A 0AA4 26 F2    0A98         BNE   ROL1     DO NEXT BYTE
01308A 0AA6 B6 13       A         LDA   AREG
```

```
01309A 0AA8 81                  RTS
01310
01311                       ************************
01312                       *   SUB ROTATE RIGHT   *
01313                       *   X-MEM BANK         *
01314                       ************************
01315A 0AA9 B7 13     A ROR     STA   AREG     SAVE ACCA
01316A 0AAB A6 04     A         LDA   #$04     DO 4#
01317A 0AAD 66 06     A ROR1    ROR   6,X      MSB
01318A 0AAF 66 07     A         ROR   7,X.     LSB
01319A 0AB1 4A                  DECA
01320A 0AB2 26 F9   0AAD        BNE   ROR1
01321A 0AB4 B6 13     A         LDA   AREG
01322A 0AB6 81                  RTS
01323
01324                       ************************
01325                       *   SUB ROTATE LEFT    *
01326                       *   4#                 *
01327                       *   X-MEM BANK         *
01328                       ************************
01329A 0AB7 B7 96     A ROL     STA   XVD2     SAVE ACCA
01330A 0AB9 A6 04     A         LDA   #$04     DO 4#
01331A 0ABB BE 14     A RO1     LDX   XREG
01332A 0ABD AD D6   0A95        BSR   ROLB     ROTATE 1#
01333A 0ABF 4A                  DECA
01334A 0AC0 26 F9   0ABB        BNE   RO1
01335A 0AC2 B6 96     A         LDA   XVD2     GET ACCA
01336A 0AC4 81                  RTS
01337
01338                       ****************************
01339                       *   MULTIPLY FOR 1 DIGIT   *
01340                       ****************************
01341A 0AC5 B7 96     A DOM     STA   XVD2     SAVE ACCA
01342A 0AC7 E6 07     A         LDA   7,X      GET LSB
01343A 0AC9 A4 0F     A         AND   #$0F     MUSK LSN
01344A 0ACB 27 07   0AD4 DOM1   BEQ   DOM2     ADD TO SUM
01345A 0ACD BE 14     A         LDX   XREG     LSN#
01346A 0ACF AD 8A   0A5B        BSR   ADDB
01347A 0AD1 4A                  DECA
01348A 0AD2 20 F7   0ACB        BRA   DOM1
01349A 0AD4 B6 96     A DOM2    LDA   XVD2     GET ACCA
01350A 0AD6 81                  RTS
01351
01352                       ********************************
01353                       *   SUBROUTINE MULTIPLY        *
01354                       *   X-ADDRESS BLOCK IE A0,B0   *
01355                       *   (X+6TOX+7)#(X+8TOX+D)      *
01356                       *   A-CTR                      *
01357                       ********************************
01358A 0AD7 BF 14     A MUL     STX   XREG     SAVE MEM BANK
01359A 0AD9 9F                  TXA
01360A 0ADA AB 05     A         ADD   #$05     CLR ANSWER MEM
01361A 0ADC 97                  TAX
01362A 0ADD 7F                MUL2  CLR   ,X       LOCATIONS
01363A 0ADE 5A                  DEX
01364A 0ADF B3 14     A         CPX   XREG
01365A 0AE1 24 FA   0ADD        BHS   MUL2
01366A 0AE3 A6 04     A         LDA   #$04     DO FOR 4 DIGITS
01367A 0AE5 BE 14     A MUL1    LDX   XREG
01368A 0AE7 AD DC   0AC5        BSR   DOM      MUL BY ADDITION
01369A 0AE9 BE 14     A         LDX   XREG
01370A 0AEB AD BC   0AA9        BSR   ROR      MULTIPLIER
01371A 0AED AD C8   0AB7        BSR   ROL      MULTIPLICAND
01372A 0AEF 4A                  DECA
```

```
01373A 0AF0 26 F3   0AE5        BNE   MUL1
01374A 0AF2 81                  RTS
01375
01376                    ************************
01377                    *     WRITE TO MEMORY  *
01378                    *     MADH,MADL MEM ADD *
01379                    ************************
01380A 0AF3 3C 98      A WMEM   INC   MADL
01381A 0AF5 3F 14      A        CLR   XREG
01382A 0AF7 AE 97      A        LDX   #MADH
01383A 0AF9 AD 01   0AFC        BSR   WMEM1    MRS
01384A 0AFB 81                  RTS
01385A 0AFC CC 0DB3    A WMEM1  JMP   MRS
```

```
01387
01388                    ****************************
01389                    *      SUBROUTINE RX RS232 *
01390                    ****************************
01391A 0AFF 0C 12 0A 0B0C RXSR   BRSET  6,FLGN,RXSR1 MASK BIT
01392A 0B02 1C 12       A        BSET   6,FLGN
01393A 0B04 16 12       A        BSET   3,FLGN   RX BIT
01394A 0B06 A6 0D       A        LDA    #$0D     TIMER / 32
01395A 0B08 B7 09       A        STA    TCR
01396A 0B0A B7 08       A        STA    TDR
01397A 0B0C 81            RXSR1  RTS
01398
01399                    ************************************
01400                    *      SUBROUTINE TX RS232         *
01401                    *      A-ASCII DATA WITH OUT PARITY *
01402                    ************************************
01403A 0B0D 05 12 0F 0B1F TXSR   BRSET  3,FLGN,TXSR1 RX BIT
01404A 0B10 04 12 0C 0B1F        BRSET  2,FLGN,TXSR1 TX BIT
01405A 0B13 B7 18       A        STA    RXTX
01406A 0B15 14 12       A        BSET   2,FLGN
01407A 0B17 A6 0E       A        LDA    #$0E     TIMER / 64
01408A 0B19 B7 09       A        STA    TCR      TIMER CONTROL REG
01409A 0B1B A6 0D       A        LDA    #$0D     13*64=832MIC SEC
01410A 0B1D B7 08       A        STA    TDR
01411A 0B1F 81            TXSR1  RTS
01412
01413                    ******************************
01414                    *      TIMER INT ROUTINE FOR *
01415                    *      RS232 RX AND TX       *
01416                    ******************************
01417A 0B20 9B            RXR    SEI             SET INT MASK
01418A 0B21 A6 06       A        LDA    #$06     TIMER / 64
01419A 0B23 B7 09       A        STA    TCR      TIMER CONTROL REG
01420A 0B25 07 12 40 0B68        BRCLR  3,FLGN,TXR IF NO RX
01421A 0B28 3D 16       A        TST    BTCT     BIT COUNTER=0?
01422A 0B2A 26 05    0B31        BNE    RX1      CHECK START BIT
01423A 0B2C 0C 01 60 0B8F        BRSET  6,PORTB,TX6 RX I/P SET
01424A 0B2F 20 31    0B62        BRA    RX6
01425A 0B31 B6 16       A RX1    LDA    BTCT
01426A 0B33 A1 09       A        CMP    #$09     DATA COMPLETED
01427A 0B35 24 0D    0B44        BHS    RX4
01428A 0B37 0D 01 03 0B3D        BRCLR  6,PORTB,RX2 DATA=1
01429A 0B3A 98                   CLC             CLEAR CARRY TO ROR
01430A 0B3B 20 03    0B40        BRA    RX3
01431A 0B3D 3C 17       A RX2    INC    PRTY
01432A 0B3F 99                   SEC
01433A 0B40 36 18       A RX3    ROR    RXTX     STORE DATA
01434A 0B42 20 4B    0B8F        BRA    TX6
01435A 0B44 B6 16       A RX4    LDA    BTCT
01436A 0B46 A1 09       A        CMP    #$09     CHECK IF 1ST STOP BIT
01437A 0B48 26 05    0B4F        BNE    RX5
01438A 0B4A 0C 01 15 0B62        BRSET  6,PORTB,RX6 FIRST STOP BIT
01439A 0B4D 20 40    0B8F        BRA    TX6
01440A 0B4F 0C 01 10 0B62 RX5    BRSET  6,PORTB,RX6 SEC STOP BIT
01441A 0B52 34 17       A        LSR    PRTY     PARITY CTR / 2
01442A 0B54 25 0C    0B62        BCS    RX6
01443A 0B56 1A 12       A        BSET   5,FLGN   TOT BYTE RECEIVED
01444A 0B58 B6 18       A        LDA    RXTX
01445A 0B5A AB 10       A        ADD    #$10
01446A 0B5C A4 1F       A        AND    #$1F
01447A 0B5E B7 1A       A        STA    RXMD
01448A 0B60 20 02    0B64        BRA    RX7
01449A 0B62 18 12       A RX6    BSET   4,FLGN   FAULT BIT
01450A 0B64 17 12       A RX7    BCLR   3,FLGN   RX BIT
```

```
01451A 0B66 20 33    0B9B        BRA   TX8
01452A 0B68 05 12 30 0B9B TXR    BRCLR 2,FLGN,TX8 TX BIT
01453A 0B6B 3D 16       A        TST   BTCT     B CTR =0?
01454A 0B6D 26 04    0B73        BNE   TX2
01455A 0B6F 1F 01       A TX1    BCLR  7,PORTB  SET O/P
01456A 0B71 20 1C    0B8F        BRA   TX6
01457A 0B73 B6 16       A TX2    LDA   BTCT
01458A 0B75 A1 08       A        CMP   #$08
01459A 0B77 24 08    0B81        BHS   TX3
01460A 0B79 36 18       A        ROR   RXTX     DATA TO O/P
01461A 0B7B 24 F2    0B6F        BCC   TX1
01462A 0B7D 3C 17       A        INC   PRTY     PARITY COUNTER
01463A 0B7F 20 0C    0B8D        BRA   TX5
01464A 0B81 26 06    0B89 TX3    BNE   TX4
01465A 0B83 34 17       A        LSR   PRTY     PARITY /2
01466A 0B85 24 E8    0B6F        BCC   TX1
01467A 0B87 20 04    0B8D        BRA   TX5
01468A 0B89 A1 09       A TX4    CMP   #$09     FIRST STOP BIT
01469A 0B8B 26 0A    0B97        BNE   TX7
01470A 0B8D 1E 01       A TX5    BSET  7,PORTB
01471A 0B8F 3C 16       A TX6    INC   BTCT
01472A 0B91 A6 0C       A        LDA   #$0C     13*64=832MIC SEC
01473A 0B93 B7 08       A        STA   TDR      TIMER DATA REG
01474A 0B95 20 19    0BB0        BRA   TX9
01475A 0B97 15 12       A TX7    BCLR  2,FLGN   TX BIT
01476A 0B99 1E 01       A        BSET  7,PORTB  TX O/P BIT
01477A 0B9B 3F 16       A TX8    CLR   BTCT
01478A 0B9D 3F 17       A        CLR   PRTY
01479A 0B9F A6 48       A        LDA   #$48     TIMER OFF
01480A 0BA1 B7 09       A        STA   TCR
01481A 0BA3 03 19 0A 0BB0        BRCLR 1,GNFL,TX9
01482A 0BA6 13 19       A        BCLR  1,GNFL   FIRST SW ON RS
01483A 0BA8 1B 12       A        BCLR  5,FLGN   BYTE RX BIT
01484A 0BAA 17 12       A        BCLR  3,FLGN   RX BIT
01485A 0BAC 15 12       A        BCLR  2,FLGN   TX BIT
01486A 0BAE 17 19       A        BCLR  3,GNFL   XON GUST SENT BIT
01487A 0BB0 9A          .  TX9   CLI
01488A 0BB1 80                   RTI
01489
01490                    ******************************
01491                    *     RS232 CONTROL ROUTINE    *
01492                    ******************************
01493A 0BB2 01 19 4E 0C03 RTCR   BRCLR 0,GNFL,RTCR4 RS232 ON
01494A 0BB5 06 12 4B 0C03        BRSET 3,FLGN,RTCR4 RX BIT
01495A 0BB8 04 12 48 0C03        BRSET 2,FLGN,RTCR4 TX BIT
01496A 0BBB 0A 12 1F 0BDD        BRSET 5,FLGN,RTCR2 BYTE RX BIT
01497A 0BBE 09 12 0D 0BCE        BRCLR 4,FLGN,RTCR1 FAULT BIT
01498A 0BC1 05 19 37 0BFB        BRCLR 2,GNFL,RTCR3 XOFF BIT
01499A 0BC4 A6 05       A        LDA   #RETX    REPEAT INS
01500A 0BC6 AD 3F    0C07        BSR   RTCR6    TXSR TX INS
01501A 0BC8 15 19       A        BCLR  2,GNFL   XOFF SENT
01502A 0BCA 19 12       A        BCLR  4,FLGN   FAULT BIT
01503A 0BCC 20 35    0C03        BRA   RTCR4
01504A 0BCE 06 19 32 0C03 RTCR1  BRSET 3,GNFL,RTCR4
01505A 0BD1 A6 03       A        LDA   #XON     TX ON
01506A 0BD3 AD 32    0C07        BSR   RTCR6    TXSR TRANSMIT
01507A 0BD5 16 19       A        BSET  3,GNFL
01508A 0BD7 15 19       A        BCLR  2,GNFL
01509A 0BD9 1D 12       A        BCLR  6,FLGN
01510A 0BDB 20 26    0C03        BRA   RTCR4
01511A 0BDD 02 19 23 0C03 RTCR2  BRSET 1,GNFL,RTCR4 1ST TIME SW ON
01512A 0BE0 05 19 18 0BFB        BRCLR 2,GNFL,RTCR3 XOFF SENT
01513A 0BE3 A6 0E       A        LDA   #RXINL   INS LIMIT
01514A 0BE5 B1 1A       A        CMP   RXMD
```

```
01515A 0BE7 24 0B    0BF1       BHS   RTCR7
01516A 0BE9 A6 06       A       LDA   #EOLA
01517A 0BEB AD 1A    0C07       BSR   RTCR6   TXSR
01518A 0BED 1B 12       A       BCLR  5,FLGN  BYTE RX BIT
01519A 0BEF 20 12    0C03       BRA   RTCR4
01520A 0BF1 B6 1A       A RTCR7 LDA   RXMD
01521A 0BF3 48                  LSLA
01522A 0BF4 B9 1A       A       ADD   RXMD
01523A 0BF6 97                  TAX
01524A 0BF7 AD 0B    0C04       BSR   RTCR5   JMP TO TABLE
01525A 0BF9 20 08    0C03       BRA   RTCR4
01526A 0BFB A6 02       A RTCR3 LDA   #XOFF
01527A 0BFD AD 08    0C07       BSR   RTCR6   TXSR
01528A 0BFF 14 19       A       BSET  2,GNFL  XOFF SENT
01529A 0C01 17 19       A       BCLR  3,GNFL
01530A 0C03 81             RTCR4 RTS
01531A 0C04 DC 1BA0     A RTCR5 JMP   RXIN,X
01532A 0C07 CC 0B0D     A RTCR6 JMP   TXSR
```

```
01534
01535                    ****************************
01536                    *      UPDATE MEM AND FORM *
01537                    *      SUM CHECK           *
01538                    *      RSH-RS232 ADD       *
01539                    *      RSH1-RS232 BUFF ADD *
01540                    ****************************
01541A 0C8A 3F 1C       A BUFU   CLR    CREG
01542A 0C8C 3F 1E       A        CLR    RSH1
01543A 0C8E A6 8F       A        LDA    #$8F
01544A 0C10 B7 1F       A        STA    RSL1
01545A 0C12 AE 9E       A        LDX    #RSH
01546A 0C14 AD 78    0C8E        BSR    BUFU13   IACR
01547A 0C16 B7 40       A        STA    DREG
01548A 0C18 A4 0F       A        AND    #$0F
01549A 0C1A AD 75    0C91        BSR    BUFU14   STAH
01550A 0C1C A6 96       A        LDA    #$96
01551A 0C1E B7 1F       A        STA    RSL1
01552A 0C20 0E 40 0B 0C2E        BRSET  7,DREG,BUFU2
01553A 0C23 0C 40 04 0C2A        BRSET  6,DREG,BUFU1
01554A 0C26 AD 41    0C69        BSR    BUFU10
01555A 0C28 20 06    0C30        BRA    BUFU3
01556A 0C2A AD 48    0C74 BUFU1  BSR    BUFU11
01557A 0C2C 20 02    0C30        BRA    BUFU3
01558A 0C2E AD 51    0C81 BUFU2  BSR    BUFU12
01559A 0C30 0A 40 0B 0C3E BUFU3  BRSET  5,DREG,BUFU5
01560A 0C33 08 40 04 0C3A        BRSET  4,DREG,BUFU4
01561A 0C36 AD 31    0C69        BSR    BUFU10
01562A 0C38 20 06    0C40        BRA    BUFU6
01563A 0C3A AD 38    0C74 BUFU4  BSR    BUFU11
01564A 0C3C 20 02    0C40        BRA    BUFU6
01565A 0C3E AD 41    0C81 BUFU5  BSR    BUFU12
01566A 0C40 A6 94       A BUFU6  LDA    #$94
01567A 0C42 B7 1F       A        STA    RSL1
01568A 0C44 AE 9E       A        LDX    #RSH
01569A 0C46 AD 46    0C8E        BSR    BUFU13   IACR
01570A 0C48 AD 4A    0C94        BSR    BUFU15   STNM1
01571A 0C4A A6 90       A        LDA    #$90
01572A 0C4C B7 1F       A        STA    RSL1
01573A 0C4E AE 9E       A        LDX    #RSH
01574A 0C50 AD 3C    0C8E        BSR    BUFU13
01575A 0C52 AD 40    0C94        BSR    BUFU15
01576A 0C54 AE 9E       A        LDX    #RSH
01577A 0C56 AD 36    0C8E        BSR    BUFU13
01578A 0C58 AD 3A    0C94        BSR    BUFU15
01579A 0C5A A6 9E       A        LDA    #$9E
01580A 0C5C B7 1F       A        STA    RSL1
01581A 0C5E B6 1C       A        LDA    CREG
01582A 0C60 AB 30       A        ADD    #$30
01583A 0C62 AE 1E       A        LDX    #RSH1
01584A 0C64 AD 31    0C97        BSR    BUFU16   IACO
01585A 0C66 3F 1F       A        CLR    RSL1
01586A 0C68 81                   RTS
01587A 0C69 AE 1E       A BUFU10 LDX    #RSH1
01588A 0C6B A6 03       A        LDA    #$03
01589A 0C6D AD 2B    0C9A        BSR    BUFU17   CTBM
01590A 0C6F A6 30       A        LDA    #$30
01591A 0C71 AD 24    0C97        BSR    BUFU16   IACO
01592A 0C73 81                   RTS
01593A 0C74 AE 1E       A BUFU11 LDX    #RSH1
01594A 0C76 A6 02       A        LDA    #$02
01595A 0C78 AD 20    0C9A        BSR    BUFU17   CTBM
01596A 0C7A AE 9E       A        LDX    #RSH
01597A 0C7C AD 10    0C8E        BSR    BUFU13   IACR
```

```
01598A 0C7E AD 14    0C94       BSR   BUFU15  STNM1
01599A 0C80 81                  RTS
01600A 0C81 AE 9E      A BUFU12 LDX   #RSH
01601A 0C83 AD 09    0C8E       BSR   BUFU13  IACR
01602A 0C85 AD 0D    0C94       BSR   BUFU15  STNM1
01603A 0C87 AE 9E      A        LDX   #RSH
01604A 0C89 AD 03    0C8E       BSR   BUFU13  IACR
01605A 0C8B AD 07    0C94       BSR   BUFU15  STNM1
01606A 0C8D 81                  RTS
01607A 0C8E CC 07EF    A BUFU13 JMP   IACR
01608A 0C91 CC 0803    A BUFU14 JMP   STAM
01609A 0C94 CC 0826    A BUFU15 JMP   STNM1
01610A 0C97 CC 0694    A BUFU16 JMP   IACO
01611A 0C9A CC 0889    A BUFU17 JMP   CTBM
01612
01613                     ********************************
01614                     *      INS 00                  *
01615                     *      START INST              *
01616                     *      RESET RS232 ADD POINTERS *
01617                     *      UPDATE BUFFERS          *
01618                     ********************************
01619A 0C9D 3F 9E      A IN00   CLR   RSH     RESET ADD
01620A 0C9F A6 B0      A        LDA   #STNEN  POINTERS
01621A 0CA1 B7 9F      A        STA   RSL
01622A 0CA3 AD 07    0CAC       BSR   IN001   BUFU UPDATE BUFFERS
01623A 0CA5 A6 06      A        LDA   #EOLA   ACK SIG
01624A 0CA7 AD 06    0CAF       BSR   IN002   TXSR TRANSMIT
01625A 0CA9 1B 12      A        BCLR  5,FLGN  BYTE RX BIT
01626A 0CAB 81                  RTS
01627A 0CAC CC 0C0A    A IN001  JMP   BUFU
01628A 0CAF CC 0B0D    A IN002  JMP   TXSR
01629
01630                     ****************************
01631                     *      INS 01-07           *
01632                     *      INFORMATION REQUIRED *
01633                     *      SEND INFO           *
01634                     *      SEND END OF NO      *
01635                     ****************************
01636A 0CB2 BE 1A      A IN01   LDX   RXMD
01637A 0CB4 B6 1F      A        LDA   RSL1
01638A 0CB6 27 07    0CBF       BEQ   IN011
01639A 0CB8 D1 1740    A        CMP   RINVT,X
01640A 0CBB 27 15    0CD2       BEQ   IN013
01641A 0CBD 20 08    0CC7       BRA   IN012
01642A 0CBF 3F 1E      A IN011  CLR   RSH1
01643A 0CC1 5A                  DEX
01644A 0CC2 D6 1740    A        LDA   RINVT,X
01645A 0CC5 B7 1F      A        STA   RSL1
01646A 0CC7 AE 1E      A IN012  LDX   #RSH1
01647A 0CC9 AD 12    0CDD       BSR   IN015   IACR
01648A 0CCB 4D                  TSTA
01649A 0CCC 27 F9    0CC7       BEQ   IN012
01650A 0CCE AD 10    0CE0       BSR   IN016   TXSR
01651A 0CD0 20 0A    0CDC       BRA   IN014
01652A 0CD2 A6 0D      A IN013  LDA   #EOLN
01653A 0CD4 AD 0A    0CE0       BSR   IN016   TXSR
01654A 0CD6 3F 1F      A        CLR   RSL1
01655A 0CD8 15 19      A        BCLR  2,GNFL  XOFF SENT
01656A 0CDA 1B 12      A        BCLR  5,FLGN
01657A 0CDC 81           IN014  RTS
01658A 0CDD CC 07EF    A IN015  JMP   IACR
01659A 0CE0 CC 0B0D    A IN016  JMP   TXSR
01660
01661                     ********************
```

```
01662                    *      INS 08      *
01663                    *     PRESENT INFO *
01664                    **********************
01665A 0CE3 AE 9E      A IN08  LDX   #RSH
01666A 0CE5 AD 09    0CF0      BSR   IN081   APPS
01667A 0CE7 AD 0A    0CF3      BSR   IN082   BUFU
01668A 0CE9 A6 06      A       LDA   #EOLA
01669A 0CEB AD 09    0CF6      BSR   IN083   TXSR
01670A 0CED 1B 12      A       BCLR  5,FLGN  BYTE RX BIT
01671A 0CEF 81                 RTS
01672A 0CF0 CC 06A9    A IN081 JMP   APPS
01673A 0CF3 CC 0C0A    A IN082 JMP   BUFU
01674A 0CF6 CC 0B0D    A IN083 JMP   TXSR
01675
01676                    **********************
01677                    *      INS 09       *
01678                    *     PREVIOUS INFO *
01679                    **********************
01680A 0CF9 AE 9E      A IN09  LDX   #RSH
01681A 0CFB AD F3    0CF0      BSR   IN081   APPS
01682A 0CFD AE 9E      A       LDX   #RSH
01683A 0CFF AD EF    0CF0      BSR   IN081   APPS
01684A 0D01 AD F0    0CF3      BSR   IN082   BUFU
01685A 0D03 A6 06      A       LDA   #EOLA
01686A 0D05 AD EF    0CF6      BSR   IN083   TXSR
01687A 0D07 1B 12      A       BCLR  5,FLGN  BYTE RX BIT
01688A 0D09 81                 RTS
01689
01690                    ******************
01691                    *      INS 0A    *
01692                    *     NEXT INFO  *
01693                    ******************
01694A 0D0A AE 9E      A IN0A  LDX   #RSH
01695A 0D0C AD 09    0D17      BSR   IN0A1   APNS
01696A 0D0E AD E3    0CF3      BSR   IN082   BUFU
01697A 0D10 A6 06      A       LDA   #EOLA
01698A 0D12 AD E2    0CF6      BSR   IN083   TXSR
01699A 0D14 1B 12      A       BCLR  5,FLGN
01700A 0D16 81                 RTS
01701A 0D17 CC 06B4    A IN0A1 JMP   APNS
01702
01703                    ******************
01704                    *      IN0B      *
01705                    *     RS232 OFF  *
01706                    ******************
01707A 0D1A 1A 19      A IN0B  BSET  5,GNFL  OFF BIT
01708A 0D1C A6 06      A       LDA   #EOLA
01709A 0D1E AD D6    0CF6      BSR   IN083   TXSR
01710A 0D20 1B 12      A       BCLR  5,FLGN
01711A 0D22 81                 RTS
01712
01713                    ************************
01714                    *      IN0C            *
01715                    *      START-DATE      *
01716                    *      RATIO & DEVISOR *
01717                    ************************
01718A 0D23 3F 97      A IN0C  CLR   MADH
01719A 0D25 A6 83      A       LDA   #$83
01720A 0D27 B7 98      A       STA   MADL
01721A 0D29 3F 1E      A       CLR   RSH1
01722A 0D2B A6 98      A       LDA   #$98
01723A 0D2D B7 1F      A       STA   RSL1
01724A 0D2F AD 31    0D62 IN0C1 BSR   IN0C5
01725A 0D31 B6 1F      A       LDA   RSL1
```

```
01726A 0D33 A1 96      A       CMP   #$96
01727A 0D35 25 F8   0D2F       BLO   IN0C1
01728A 0D37 3C 98      A       INC   MADL
01729A 0D39 AD 27   0D62 IN0C2 BSR   IN0C5
01730A 0D3B B6 1F      A       LDA   RSL1
01731A 0D3D A1 9E      A       CMP   #$9E
01732A 0D3F 25 F8   0D39       BLO   IN0C2
01733A 0D41 AE 97      A       LDX   #MADH
01734A 0D43 AD 26   0D6B       BSR   IN0C6   IACR
01735A 0D45 AE BF      A       LDX   #$BF
01736A 0D47 BF 1F      A       STX   RSL1
01737A 0D49 AE 1E      A       LDX   #RSH1
01738A 0D4B AD 21   0D6E       BSR   IN0C7   STAM
01739A 0D4D A6 9E      A IN0C3 LDA   #$9E
01740A 0D4F B7 1F      A       STA   RSL1
01741A 0D51 B6 1C      A       LDA   CREG
01742A 0D53 AB 30      A       ADD   #$30    TO ASCII
01743A 0D55 AE 1E      A       LDX   #RSH1
01744A 0D57 AD 1E   0D77       BSR   IN0C10  IACO
01745A 0D59 3F 1F      A       CLR   RSL1
01746A 0D5B A6 06      A       LDA   #EOLA
01747A 0D5D AD 12   0D71       BSR   IN0C8   TXSR
01748A 0D5F 1B 12      A       BCLR  5,FLGN
01749A 0D61 81                 RTS
01750A 0D62 AE 97      A IN0C5 LDX   #MADH
01751A 0D64 AD 05   0D6B       BSR   IN0C6   IACR
01752A 0D66 AE 1E      A       LDX   #RSH1
01753A 0D68 AD 0A   0D74       BSR   IN0C9   STNM1
01754A 0D6A 81                 RTS
01755A 0D6B CC 07EF    A IN0C6 JMP   IACR
01756A 0D6E CC 0803    A IN0C7 JMP   STAM
01757A 0D71 CC 0B0D    A IN0C8 JMP   TXSR
01758A 0D74 CC 0826    A IN0C9 JMP   STNM1
01759A 0D77 CC 0694    A IN0C10 JMP  IACO
01760
01761                    ********************
01762                    *      IN50D       *
01763                    *    START-TIME    *
01764                    *      SPR         *
01765                    ********************
01766A 0D7A 3F 97      A IN0D  CLR   MADH
01767A 0D7C A6 80      A       LDA   #$80
01768A 0D7E B7 98      A       STA   MADL
01769A 0D80 3F 1E      A       CLR   RSH1
01770A 0D82 A6 90      A       LDA   #$90
01771A 0D84 B7 1F      A       STA   RSL1
01772A 0D86 AD DA   0D62 IN0D1 BSR   IN0C5
01773A 0D88 B6 1F      A       LDA   RSL1
01774A 0D8A A1 94      A       CMP   #$94
01775A 0D8C 25 F8   0D86       BLO   IN0D1
01776A 0D8E B6 92      A       LDA   SPR
01777A 0D90 AE 1E      A       LDX   #RSH1
01778A 0D92 AD E0   0D74       BSR   IN0C9   STNM1
01779A 0D94 CC 0D4D    A       JMP   IN0C3   SUM CHECK+EOLA
01780
01781                    **********************
01782                    *    INSTRUCTION 0E  *
01783                    **********************
01784A 0D97 1E 01      A IN0E  BSET  7,PORTB TX IN IDLE STATE
01785A 0D99 A6 06      A       LDA   #EOLA
01786A 0D9B AD 03   0DA0       BSR   IN0E1   TXSR
01787A 0D9D 1B 12      A       BCLR  5,FLGN  BYTE RX BIT
01788A 0D9F 81                 RTS
01789A 0DA0 CC 0B0D    A IN0E1 JMP   TXSR
```

```
01791                   ****************************
01792                   *      8K MEMORY ROUTINES  *
01793                   ****************************
01794                   *80-ID LENGTH/FILE
01795                   *81-HH  /
01796                   *82-MM  /
01797                   *83-SS  /START OF RECORDING
01798                   *84-YY  /IN 8K MEM
01799                   *85-MM  /
01800                   *86-DD  /
01801                   *87-DEMP/
01802                   *88,89-KWh RATIO
01803                   *8A,8B-KVAh RATIO
01804
01805
01806                   *   A-DATA WHEN ENTERING
01807A 0DA3 18 00      A MEMS   BSET   0,PORTA  CHANGE MEM MAP
01808A 0DA5 C7 1AAA    A        STA    $1AAA    STA INTO ADDRESS
01809A 0DA8 11 00      A        BCLR   0,PORTA  CHNG MEM
01810A 0DAA 81                  RTS
01811
01812A 0DAB 10 00      A MEMR   BSET   0,PORTA
01813A 0DAD C6 1AAA    A        LDA    $1AAA    GET DATA
01814A 0DB0 11 00      A        BCLR   0,PORTA
01815A 0DB2 81                  RTS
01816
01817                   *      A-DATA WHEN RETURNING
01818
01819                   ***********************************
01820                   *      SUBROUTINE STORE/READ DATA *
01821                   *      IN 8K MEMORY               *
01822                   *      A-DATA                     *
01823                   *      X-LOC OF HIGH ADD OF MEM   *
01824                   *      XREG-00-STORE              *
01825                   *           00-READ               *
01826                   ***********************************
01827A 0DB3 B7 13      A MRS    STA    AREG     SAVE AREG
01828A 0DB5 F6                  LDA    ,X       GET HIGH ADD
01829A 0DB6 B7 33      A        STA    MEM+$3   STORE HIGH
01830A 0DB8 B7 3B      A        STA    MEM+$B   ADD IN BOTH
01831A 0DBA 5C                  INX
01832A 0DBB F6                  LDA    ,X       GET LOW ADD
01833A 0DBC B7 34      A        STA    MEM+$4   STORE LOW ADD
01834A 0DBE B7 3C      A        STA    MEM+$C   IN BOTH
01835A 0DC0 BE 14      A        LDX    XREG     READ/STORE
01836A 0DC2 B6 13      A        LDA    AREG     DATA
01837A 0DC4 9B                  SEI
01838A 0DC5 AD 02    0DC9       BSR    MRS1     DO ACTION
01839A 0DC7 9A                  CLI
01840A 0DC8 81                  RTS
01841A 0DC9 EC 30      A MRS1   JMP    MEM,X    READ OR STORE
01842
```

```
01844                    ************************************
01845                    *     SUBROUTINE CHANGE TIME+DATE    *
01846                    ************************************
01847A 0DCB A6 79      A CHT   LDA   #%01111001 DISABLE RTC
01848A 0DCD B7 8A      A       STA   ARTC
01849A 0DCF A6 F2      A       LDA   #%11110010 SET MODE
01850A 0DD1 B7 8B      A       STA   BRTC
01851A 0DD3 1F 93      A       BCLR  7,MRD    CLR SET MODE
01852A 0DD5 BE 93      A       LDX   MRD
01853A 0DD7 DE 1A00    A       LDX   FAV,X
01854A 0DDA AD 34    0E10      BSR   CHT1     GET 2 CHRS FROM DISP
01855A 0DDC B7 89      A       STA   YEAR     YEAR
01856A 0DDE AD 33    0E13      BSR   CHT2     MAKE X POINT TO NEXT VA
01857A 0DE0 AD 2E    0E10      BSR   CHT1
01858A 0DE2 B7 88      A       STA   MON      MONTH
01859A 0DE4 AD 2D    0E13      BSR   CHT2     MAKE X POINT NEXT
01860A 0DE6 AD 28    0E10      BSR   CHT1
01861A 0DE8 B7 87      A       STA   DATE     DATE
01862A 0DEA AD 27    0E13      BSR   CHT2
01863A 0DEC AD 22    0E10      BSR   CHT1
01864A 0DEE B7 84      A       STA   HRS      HOURS
01865A 0DF0 AD 21    0E13      BSR   CHT2
01866A 0DF2 AD 1C    0E10      BSR   CHT1
01867A 0DF4 B7 82      A       STA   MIN      MINUTES
01868A 0DF6 AD 1B    0E13      BSR   CHT2
01869A 0DF8 AD 16    0E10      BSR   CHT1
01870A 0DFA B7 80      A       STA   SEC      SECONDS
01871A 0DFC A6 72      A       LDA   #%01110010 NON SET MODE
01872A 0DFE B7 8B      A       STA   BRTC
01873A 0E00 A6 29      A       LDA   #%00101001
01874A 0E02 B7 8A      A       STA   ARTC
01875A 0E04 BF 14      A       STX   XREG
01876A 0E06 AE 0C      A       LDX   #%00001100 BLINK OFF
01877A 0E08 AD 0C    0E16      BSR   CHT3     DBUS
01878A 0E0A CF 0100    A       STX   DINS
01879A 0E0D BE 14      A       LDX   XREG
01880A 0E0F 81                 RTS
01881A 0E10 CC 0793    A CHT1  JMP   GTCD     GET 2 NUMBERS FROM DISP
01882A 0E13 CC 0E19    A CHT2  JMP   MTNV     MAKE X POINT TO NEXT VA
01883A 0E16 CC 0786    A CHT3  JMP   DBUS
01884
01885                    ****************************************
01886                    *     SUBROUTINE MOVE TO NEXT VAR(TIME)  *
01887                    *     X-PREVS ADDRESS                    *
01888                    ****************************************
01889A 0E19 B7 13      A MTNV  STA   AREG     SAVE AREG
01890A 0E1B 9F                 TXA
01891A 0E1C AB 20      A       ADD   #$20
01892A 0E1E 97                 TAX
01893A 0E1F D6 1A80    A       LDA   MODV1,X
01894A 0E22 A4 7F      A       AND   #$7F
01895A 0E24 97                 TAX
01896A 0E25 B6 13      A       LDA   AREG
01897A 0E27 81                 RTS
01898                    *     X-NEXT ADDRESS
01899
01900                    ***************************
01901                    *     SUBROUTINE MODE 1   *
01902                    ***************************
01903A 0E28 0E 93 32 0E5D MOD1  BRSET 7,MRD,MD19 SET MODE
01904A 0E2B 06 11 03 0E31       BRSET 3,FDIS,MD18 1ST TIME ENTER MOD
01905A 0E2E 03 11 1E 0E4F       BRCLR 1,FDIS,MD11 1ST SWITCH ON
01906A 0E31 0E 8A 2B 0E5F MD18  BRSET 7,ARTC,MD12 UPDATE IN PROGRESS
01907A 0E34 AD 2A    0E60       BSR   MD13     SUBT
```

```
01908A 0E36 AD 31    0E69        BSR   MD16    MTNV
01909A 0E38 B6 92       A        LDA   SPR
01910A 0E3A AD 30    0E6C        BSR   MD17    STCD
01911A 0E3C AE 1E       A        LDX   #$1E    DO FOR DEMP NO
01912A 0E3E B6 95       A        LDA   DEMP
01913A 0E40 AD 2A    0E6C        BSR   MD17    STCD
01914A 0E42 13 11       A        BCLR  1,FDIS  1ST ENTERE MOD
01915A 0E44 17 11       A        BCLR  3,FDIS  1ST SWITCH ON
01916A 0E46 AD 1B    0E63        BSR   MD14    DBUS
01917A 0E48 A6 0C       A        LDA   #%00001100 SWITCH DISP ON
01918A 0E4A C7 0100     A        STA   DINS
01919A 0E4D 20 0E    0E5D        BRA   MD19
01920A 0E4F 01 10 0D 0E5F MD11   BRCLR 0,FRTC,MD12 PI BIT
01921A 0E52 0E 8A 0A 0E5F        BRSET 7,ARTC,MD12 UPDATE IN PROGRESS
01922A 0E55 AD 0F    0E66        BSR   MD15    SVDT
01923A 0E57 AE 1E       A        LDX   #$1E    LOC IN DISP
01924A 0E59 B6 95       A        LDA   DEMP    DEMAND PERIOD
01925A 0E5B AD 0F    0E6C        BSR   MD17    STCD
01926A 0E5D 11 10       A MD19   BCLR  0,FRTC  UF BIT
01927A 0E5F 81             MD12  RTS
01928A 0E60 CC 0E6F     A MD13   JMP   SUDT
01929A 0E63 CC 0786     A MD14   JMP   DBUS
01930A 0E66 CC 0E9D     A MD15   JMP   SVDT
01931A 0E69 CC 0E19     A MD16   JMP   MTNV
01932A 0E6C CC 07B3     A MD17   JMP   STCD
01933
01934                    ************************************
01935                    *     SUBROUTINE UPDATE DAY AND TIME *
01936                    ************************************
01937A 0E6F AD 72    0EE3 SUDT   BSR   SVDT1   DISP BUSY
01938A 0E71 A6 01       A        LDA   #%00000001 CLEAR DISP
01939A 0E73 C7 0100     A        STA   DINS
01940A 0E76 AE 00       A        LDX   #$00
01941A 0E78 D6 1C00     A SUD2   LDA   MODF1,X  LOAD CHR FROM TABLE
01942A 0E7B A1 20       A        CMP   #$20    AND SKIP IF SPACE
01943A 0E7D 27 02    0E81        BEQ   SUD1
01944A 0E7F AD 65    0EE6        BSR   SVDT3   STORE CHR IN DISP
01945A 0E81 5C             SUD1  INX
01946A 0E82 A3 1F       A        CPX   #$1F    1ST & 3RD LINE DONE?
01947A 0E84 23 F2    0E78        BLS   SUD2
01948A 0E86 AD 5B    0EE3        BSR   SVDT1
01949A 0E88 A6 C0       A        LDA   #$C0
01950A 0E8A C7 0100     A        STA   DINS
01951A 0E8D AE 40       A        LDX   #$40    DO LINES 40 TO 5F
01952A 0E8F D6 1C00     A SUD3   LDA   MODF1,X
01953A 0E92 A1 20       A        CMP   #$20
01954A 0E94 27 02    0E98        BEQ   SUD4
01955A 0E96 AD 4E    0EE6        BSR   SVDT3   STACH
01956A 0E98 5C             SUD4  INX
01957A 0E99 A3 5F       A        CPX   #$5F
01958A 0E9B 23 F2    0E8F        BLS   SUD3
01959A 0E9D B6 93       A SVDT   LDA   HRD     GET ADDRESS OF FIRST
01960A 0E9F A4 7F       A        AND   #$7F    VARIABLE OFDDISP
01961A 0EA1 97                   TAX
01962A 0EA2 DE 1A00     A        LDX   FAV,X
01963A 0EA5 0E 8A FD 0EA5 SVDT5  BRSET 7,ARTC,SVDT5 RTC BUSY
01964A 0EA8 B6 89       A        LDA   YEAR    FETCH YEAR FROM RTC
01965A 0EAA AD 3D    0EE9        BSR   SVDT4   STORE YEAR IN DISP
01966A 0EAC 5C                   INX           MAKE X POINT TO
01967A 0EAD 5C                   INX           NEXT LOC
01968A 0EAE 0E 8A FD 0EAE SVDT6  BRSET 7,ARTC,SVDT6 RTC BUSY
01969A 0EB1 B6 88       A        LDA   MON     MONTH
01970A 0EB3 AD 34    0EE9        BSR   SVDT4   STCD
01971A 0EB5 5C                   INX
```

```
01972A 0EB6 5C                  INX
01973A 0EB7 0E BA FD 0EB7 SVDT7 BRSET  7,ARTC,SVDT7 RTC BUSY
01974A 0EBA B6 87       A       LDA    DATE     DO FOR DATE
01975A 0EBC AD 2B    0EE9       BSR    SVDT4    STCD
01976A 0EBE B6 93       A       LDA    MRD      GOTO VAR ON
01977A 0EC0 A4 7F       A       AND    #$7F     NEXT LINE
01978A 0EC2 97                  TAX
01979A 0EC3 D6 1A00     A       LDA    FAV,X
01980A 0EC6 AB 40       A       ADD    #$40
01981A 0EC8 97                  TAX
01982A 0EC9 0E BA FD 0EC9 SVDT8 BRSET  7,ARTC,SVDT8 RTC BUSY
01983A 0ECC B6 84       A       LDA    HRS      DO FOR HOURS
01984A 0ECE AD 19    0EE9       BSR    SVDT4    STCD
01985A 0ED0 5C                  INX
01986A 0ED1 5C                  INX
01987A 0ED2 0E BA FD 0ED2 SVDT9 BRSET  7,ARTC,SVDT9 RTC BUSY
01988A 0ED5 B6 82       A       LDA    MIN      MINUTES FROM RTC
01989A 0ED7 AD 10    0EE9       BSR    SVDT4    STCD
01990A 0ED9 5C                  INX
01991A 0EDA 5C                  INX
01992A 0EDB 0E BA FD 0EDB SVDTA BRSET  7,ARTC,SVDTA RTC BUSY
01993A 0EDE B6 80       A       LDA    SEC
01994A 0EE0 AD 07    0EE9       BSR    SVDT4    STCD
01995A 0EE2 81                  RTS
01996A 0EE3 CC 8786     A SVDT1 JMP    DBUS
01997A 0EE6 CC 0702     A SVDT3 JMP    STACH
01998A 0EE9 CC 8783     A SVDT4 JMP    STCD
01999
02000                     ****************************
02001                     *     MODE1 MOVE TO TOP LOC *
02002                     ****************************
02003A 0EEC AD 09    0EF7 MT1   BSR    MTL      MAKE X POINT TO ADD
02004A 0EEE DE 1A80     A       LDX    MODV1,X
02005A 0EF1 AD 01    0EF4       BSR    MT11     MCX MOVE CURSER
02006A 0EF3 81                  RTS
02007A 0EF4 CC 090F     A MT11  JMP    MCX
02008
02009                     ****************************************
02010                     *     SUBROUTINE FOR MOVE TO TOP LOCATION *
02011                     ****************************************
02012A 0EF7 C6 0100     A MTL   LDA    DINS
02013A 0EFA A4 7F       A       AND    #$7F
02014A 0EFC 4C                  INCA
02015A 0EFD B7 94       A       STA    DAR
02016A 0EFF 0C 94 04 0F06       BRSET  6,DAR,MTL1
02017A 0F02 1C 94       A       BSET   6,DAR
02018A 0F04 20 02    0F08       BRA    MTL2
02019A 0F06 1D 94       A MTL1  BCLR   6,DAR
02020A 0F08 BE 94       A MTL2  LDX    DAR
02021A 0F0A 81                  RTS
02022                     *     X-CONTAINS OFFSET ADD IN TAB
02023
02024                     ********************************
02025                     *     MODE1 MOVE TO PREVIOUS LOC *
02026                     ********************************
02027A 0F0B C6 0100     A MP1   LDA    DINS
02028A 0F0E A4 7F       A       AND    #$7F
02029A 0F10 97                  TAX
02030A 0F11 DE 1A80     A       LDX    MODV1,X
02031A 0F14 AD 01    0F17       BSR    MP11     MCX
02032A 0F16 81                  RTS
02033A 0F17 CC 090F     A MP11  JMP    MCX
02034
02035                     ****************************
```

```
02036                    *      MODE1 MOVE TO NEXT LOC *
02037                    *      A-PREVIOUS LOC         *
02038                    ********************************
02039A 0F1A AB 20      A MN1    ADD    #$20     ADD OFFSET OF TABLE
02040A 0F1C 97                  TAX
02041A 0F1D DE 1A80    A        LDX    MODV1,X
02042A 0F20 AD 01    0F23       BSR    MN11     MCX
02043A 0F22 81                  RTS
02044A 0F23 CC 890F    A MN11   JMP    MCX
02045
02046                    ********************
02047                    *      CHANGE MODE1 *
02048                    ********************
02049A 0F26 AD 07    0F2F CM1    BSR    CM11     CHANGE TIME & DATE
02050A 0F28 AD 08    0F32        BSR    CM12     MTNV
02051A 0F2A AD 09    0F35        BSR    CM13     GTCD
02052A 0F2C B7 92      A         STA    SPR      SAMPLE PERIOD REG
02053A 0F2E 81                   RTS
02054A 0F2F CC 0DCB    A CM11   JMP    CHT
02055A 0F32 CC 0E19    A CM12   JMP    MTNV
02056A 0F35 CC 0793    A CM13   JMP    GTCD
```

```
02058
02059                    ************************
02060                    *      SUBROUTINE MODE2 *
02061                    ************************
02062A 0F38 06 11 03 0F3E MOD2   BRSET  3,FDIS,MD20 1STSTIMEMENTEREMOD
02063A 0F3B 03 11 12 0F50        BRCLR  1,FDIS,MD22 1ST TIME SWITCH ON
02064A 0F3E AD 23    0F63 MD20   BSR    UD2      UPDATE DISP WITH M2
02065A 0F40 02 11 04 0F47        BRSET  1,FDIS,MD21 1ST TIMERSWITCH ON
02066A 0F43 17 11       A        BCLR   3,FDIS   1ST TIME ENTER MOD
02067A 0F45 20 09    0F50        BRA    MD22
02068A 0F47 13 11       A MD21   BCLR   1,FDIS
02069A 0F49 AD 06    0F51        BSR    MD23     DBUS
02070A 0F4B A6 0C       A        LDA    #%00001100 SWITCH DISP ON
02071A 0F4D C7 0100     A        STA    DINS
02072A 0F50 81                MD22   RTS
02073A 0F51 CC 0786     A MD23   JMP    DBUS
02074
02075                    ******************************
02076                    *      MODE2 MAKE X POINT TO *
02077                    *      NEXT VAR              *
02078                    ******************************
02079A 0F54 B7 13       A MTN2   STA    AREG     SAVE AREG
02080A 0F56 9F                   TXA
02081A 0F57 AB 20       A        ADD    #$20
02082A 0F59 97                   TAX
02083A 0F5A D6 1B00     A        LDA    MODV2,X
02084A 0F5D A4 7F       A        AND    #$7F
02085A 0F5F 97                   TAX
02086A 0F60 B6 13       A        LDA    AREG
02087A 0F62 81                   RTS
02088
02089                    ******************************
02090                    *     M2 SUB UPDATE DISPLAY *
02091                    ******************************
02092A 0F63 AD 61    0FC6 UD2    BSR    VD23     DBUS DISP BUSY
02093A 0F65 A6 01       A        LDA    #%00000001 CLR DISP
02094A 0F67 C7 0100     A        STA    DINS
02095A 0F6A AE 00       A        LDX    #$00
02096A 0F6C D6 1C20     A UD22   LDA    MODF2,X
02097A 0F6F A1 20       A        CMP    #$20
02098A 0F71 27 02    0F75        BEQ    UD21
02099A 0F73 AD 54    0FC9        BSR    VD24     STACH
02100A 0F75 5C                UD21   INX
02101A 0F76 A3 1F       A        CPX    #$1F     1&3 LINE DONE
02102A 0F78 23 F2    0F6C        BLS    UD22
02103A 0F7A AE 40       A        LDX    #$40
02104A 0F7C D6 1C20     A UD23   LDA    MODF2,X
02105A 0F7F A1 20       A        CMP    #$20
02106A 0F81 27 02    0F85        BEQ    UD24
02107A 0F83 AD 44    0FC9        BSR    VD24     STACH
02108A 0F85 5C                UD24   INX
02109A 0F86 A3 5F       A        CPX    #$5F     2&4 LINE DONE
02110A 0F88 23 F2    0F7C        BLS    UD23
02111A 0F8A AE 02       A VD2    LDX    #$02     MODE2 INITIAL DISP
02112A 0F8C DE 1A00     A        LDX    FAV,X    VARIABLE
02113A 0F8F BF 96       A        STX    XVD2
02114A 0F91 3F 97       A        CLR    MADH
02115A 0F93 A6 80       A        LDA    #$80
02116A 0F95 B7 98       A        STA    MADL
02117A 0F97 AD 17    0FB0        BSR    VD21     GET HOURS
02118A 0F99 BE 96       A        LDX    XVD2     DISP ADD
02119A 0F9B AD 2F    0FCC        BSR    VD25     STCD
02120A 0F9D BF 96       A        STX    XVD2
02121A 0F9F AD 1A    0FBB        BSR    VD22     MIN
```

```
02122A 0FA1 AD 18   0FBB       BSR   VD22    SEC
02123A 0FA3 AD 16   0FBB       BSR   VD22    YEARS
02124A 0FA5 AD 14   0FBB       BSR   VD22    MONTH
02125A 0FA7 AD 12   0FBB       BSR   VD22    DATE
02126A 0FA9 AD 05   0FB0       BSR   VD21    GET DEMP
02127A 0FAB AE 4E      A       LDX   #$4E     POSITION OF DEMP
02128A 0FAD AD 1D   0FCC       BSR   VD25    STCD
02129A 0FAF 81                 RTS
02130A 0FB0 3C 98      A VD21  INC   MADL    ADDRESS
02131A 0FB2 AE 08      A       LDX   #$08    READ
02132A 0FB4 BF 14      A       STX   XREG
02133A 0FB6 AE 97      A       LDX   #MADH
02134A 0FB8 AD 15   0FCF       BSR   VD26    MRS
02135A 0FBA 81                 RTS
02136A 0FBB AD F3   0FB0 VD22  BSR   VD21
02137A 0FBD BE 96      A       LDX   XVD2
02138A 0FBF AD 93   0F54       BSR   MTN2
02139A 0FC1 AD 09   0FCC       BSR   VD25    STCD
02140A 0FC3 BF 96      A       STX   XVD2
02141A 0FC5 81                 RTS
02142A 0FC6 CC 0786    A VD23  JMP   DBUS
02143A 0FC9 CC 07D2    A VD24  JMP   STACH
02144A 0FCC CC 07B3    A VD25  JMP   STCD
02145A 0FCF CC 0DB3    A VD26  JMP   MRS
```

```
02147
02148                    ******************************
02149                    *     MODEA MOVE TO TOP LOC  *
02150                    ******************************
02151A 0FD2 AE 0A      A MTA   LDX   #$0A     MODA
02152A 0FD4 DE 1A00    A       LDX   FAV,X    FIRST VAR
02153A 0FD7 AD 0E    0FE7      BSR   MTA2     GACD
02154A 0FD9 4C                 INCA
02155A 0FDA A1 38      A       CMP   #$38
02156A 0FDC 25 02    0FE0      BLO   MTA1
02157A 0FDE A6 31      A       LDA   #$31
02158A 0FE0 AD 0B    0FED MTA1 BSR   MTA4     STACH
02159A 0FE2 A4 0F      A       AND   #$0F     TO BCD
02160A 0FE4 AD 04    0FEA      BSR   MTA3     MNA+7
02161A 0FE6 81                 RTS
02162A 0FE7 CC 0745    A MTA2  JMP   GACD
02163A 0FEA CC 0FF8    A MTA3  JMP   MNA+7    FIX NAME &CURS-1ST INST
02164A 0FED CC 0702    A MTA4  JMP   STACH
02165
02166                    **********************************
02167                    *     MODEA MOVE TO PREVIOUS LOC *
02168                    **********************************
02169A 0FF0 81           MPA   RTS
02170
02171                    ******************************
02172                    *     MODEA MOVE TO NEXT LOC *
02173                    ******************************
02174A 0FF1 B6 90      A MNA   LDA   KKV      PRINT NAMES IN
02175A 0FF3 A1 0E      A       CMP   #$0E     IF NEXT KEY PRESSED
02176A 0FF5 26 01    0FF8      BNE   MNA3     RETURN
02177A 0FF7 81                 RTS
02178A 0FF8 AD 0E    1008 MNA3 BSR   MNA2     NAME DISP
02179A 0FFA AE 0A      A       LDX   #$0A
02180A 0FFC D6 1A00    A       LDA   FAV,X
02181A 0FFF A3 80      A       ADD   #$80
02182A 1001 97                 TAX
02183A 1002 AD 01    1005      BSR   MNA1     MCX
02184A 1004 81                 RTS
02185A 1005 CC 098F    A MNA1  JMP   MCX
02186A 1008 CC 1004    A MNA2  JMP   NAME
02187
02188                    ******************************
02189                    *     MODEA MAKE X POINT TO  *
02190                    *     NEXT VAR               *
02191                    ******************************
02192A 100B 81           MTNA  RTS
02193
02194                    ******************************
02195                    *     MA SUB UPDATE DISPLAY  *
02196                    ******************************
02197A 100C AD 4C    105A UDA  BSR   VDA3     DBUS DISP BUSY
02198A 100E A6 01      A       LDA   #%00000001 CLR DISP
02199A 1010 C7 0100    A       STA   DINS
02200A 1013 AE 00      A       LDX   #$00
02201A 1015 D6 1D80    A UDA2  LDA   MODFA,X
02202A 1018 A1 20      A       CMP   #$20
02203A 101A 27 02    101E      BEQ   UDA1
02204A 101C AD 36    1054      BSR   VDA1     STACH
02205A 101E 5C           UDA1  INX
02206A 101F A3 1F      A       CPX   #$1F     1&3 LINE DONE
02207A 1021 23 F2    1015      BLS   UDA2
02208A 1023 AE 40      A       LDX   #$40
02209A 1025 D6 1D60    A UDA3  LDA   MODFA,X
02210A 1028 A1 20      A       CMP   #$20
```

```
02211A 102A 27 02   102E        BEQ   UDA4
02212A 102C AD 26   1054        BSR   VDA1     STACH
02213A 102E 5C            UDA4  INX
02214A 102F A3 5F       A       CPX   #$5F     2&4 LINE DONE
02215A 1031 23 F2   1025        BLS   UDA3
02216A 1033 AE 0A       A VDA   LDX   #$0A
02217A 1035 DE 1A00     A       LDX   FAV,X
02218A 1038 B6 93       A       LDA   MRD
02219A 103A AB 30       A       ADD   #$30     BCD TO ASCII
02220A 103C AD 16   1054        BSR   VDA1     STACH
02221A 103E B6 93       A       LDA   MRD      PRINT NAME IN DISP
02222A 1040 AD 42   1084        BSR   NAME
02223A 1042 AE 0A       A       LDX   #$0A
02224A 1044 D6 1A00     A       LDA   FAV,X
02225A 1047 AB 80       A       ADD   #$80
02226A 1049 97                  TAX
02227A 104A AD 0B   1057        BSR   VDA2     MCX
02228A 104C A6 DD       A       LDA   #%00001101 BLINK ON
02229A 104E AD 0A   105A        BSR   VDA3     DBUS
02230A 1050 C7 0100     A       STA   DINS
02231A 1053 81                  RTS
02232A 1054 CC 0702     A VDA1  JMP   STACH
02233A 1057 CC 090F     A VDA2  JMP   MCX
02234A 105A CC 0786     A VDA3  JMP   DBUS
02235
02236                     ***********************
02237                     *      MODEA ROUTINE *
02238                     ***********************
02239A 105D 81            MODA  RTS             JUST RETURN
02240
02241                     ***********************
02242                     *      MODEA CHANGE *
02243                     ***********************
02244A 105E AE 0A       A CMA   LDX   #$0A
02245A 1060 DE 1A00     A       LDX   FAV,X
02246A 1063 AD 19   107E        BSR   CMA1     GACD
02247A 1065 A4 0F       A       AND   #$0F     ASCII TO BCD
02248A 1067 27 04   106D        BEQ   CMA4
02249A 1069 A1 08       A       CMP   #$08
02250A 106B 25 05   1072        BLO   CMA3
02251A 106D 1E 93       A CMA4  BSET  7,MRD
02252A 106F CC 0FF8     A       JMP   MNA+7
02253A 1072 B7 93       A CMA3  STA   MRD
02254A 1074 B7 99       A       STA   MMRD     UPDATE MEM MODE
02255A 1076 A6 0C       A       LDA   #%00001100 BLINK OFF
02256A 1078 AD 07   1081        BSR   CMA2     DBUS
02257A 107A C7 0100     A       STA   DINS
02258A 107D 81                  RTS
02259A 107E CC 07A5     A CMA1  JMP   GACD
02260A 1081 CC 0786     A CMA2  JMP   DBUS
02261
02262                     ******************************
02263                     *      PRINT NAMES INTO DISP *
02264                     ******************************
02265A 1084 43            NAME  LSLA            MOVE MODE NO
02266A 1085 48                  LSLA            TO MSB
02267A 1086 48                  LSLA
02268A 1087 48                  LSLA
02269A 1088 B7 14       A       STA   XREG
02270A 108A A6 50       A       LDA   #$50
02271A 108C B7 96       A       STA   XVD2     DISP ADD
02272A 108E BE 14       A NAM1  LDX   XREG
02273A 1090 D6 1E00     A       LDA   NMOD,X
02274A 1093 BE 96       A       LDX   XVD2
```

```
02275A 1095 AD 0B   10A2       BSR   NAM3    STACH
02276A 1097 3C 14      A NAM2  INC   XREG
02277A 1099 3C 96      A       INC   XVD2
02278A 109B A6 5F      A       LDA   #$5F
02279A 109D B1 96      A       CMP   XVD2
02280A 109F 24 ED   108E       BHS   NAM1
02281A 10A1 81                 RTS
02282A 10A2 CC 0702    A NAM3  JMP   STACH
```

```
02284
02285                    ******************************
02286                    *     MODE4 MAKE X POINT TO  *
02287                    *     NEXT VAR               *
02288                    ******************************
02289A 10A5 B7 13      A MTN4  STA   AREG     SAVE AREG
02290A 10A7 9F                 TXA
02291A 10A8 AB 20      A       ADD   #$20
02292A 10AA 97                 TAX
02293A 10AB D6 17FF    A       LDA   MODV4,X
02294A 10AE A4 7F      A       AND   #$7F
02295A 10B0 97                 TAX
02296A 10B1 B6 13      A       LDA   AREG
02297A 10B3 81                 RTS
02298
02299                    ******************************
02300                    *     M4 SUB UPDATE DISPLAY  *
02301                    ******************************
02302A 10B4 AD 31    10E7 UD4   BSR   UD46     DBUS
02303A 10B6 A6 01      A        LDA   #%00000001 CLR DISP
02304A 10B8 C7 0100    A        STA   DINS
02305A 10BB AD 2D    10EA       BSR   UD47     ADDD
02306A 10BD AE 00      A        LDX   #$00
02307A 10BF D6 1C80    A UD42   LDA   MODF4,X
02308A 10C2 A1 20      A        CMP   #$20
02309A 10C4 27 02    10C8       BEQ   UD41
02310A 10C6 AD 1C    10E4       BSR   UD45     STACH
02311A 10C8 5C           UD41    INX
02312A 10C9 AD 22    10ED       BSR   UD48     BDDD
02313A 10CB A3 1F      A        CPX   #$1F     1&3 LINE DONE
02314A 10CD 23 F8    10BF       BLS   UD42
02315A 10CF AE 40      A        LDX   #$40
02316A 10D1 D6 1C80    A UD43   LDA   MODF4,X
02317A 10D4 A1 20      A        CMP   #$20
02318A 10D6 27 02    10DA       BEQ   UD44
02319A 10D8 AD 0A    10E4       BSR   UD45     STACH
02320A 10DA 5C           UD44    INX
02321A 10DB AD 10    10ED       BSR   UD48     BDDD
02322A 10DD A3 5F      A        CPX   #$5F     2&4 LINE DONE
02323A 10DF 23 F0    10D1       BLS   UD43
02324A 10E1 CC 10F0    A        JMP   VD4
02325A 10E4 CC 07D2    A UD45   JMP   STACH
02326A 10E7 CC 0786    A UD46   JMP   DBUS
02327A 10EA CC 122B    A UD47   JMP   ADDD
02328A 10ED CC 123B    A UD48   JMP   BDDD
02329
02330                    ********************************
02331                    *     SUB UPDATE VARIABLES FOR *
02332                    *     MODES 4;5;6              *
02333                    ********************************
02334A 10F0 B6 93      A VD4   LDA   MRD      CHECK MODE 5
02335A 10F2 A4 7F      A       AND   #$7F
02336A 10F4 A1 05      A       CMP   #$05
02337A 10F6 26 1E    1116      BNE   VD48
02338A 10F8 AE 10      A       LDX   #$10
02339A 10FA B6 46      A       LDA   MDD
02340A 10FC AD 79    1177      BSR   VD49     STCD
02341A 10FE AE 13      A       LDX   #$13
02342A 1100 B6 47      A       LDA   MMO
02343A 1102 AD 73    1177      BSR   VD49     STCD
02344A 1104 AE 17      A       LDX   #$17
02345A 1106 B6 48      A       LDA   MHH
02346A 1108 AD 6D    1177      BSR   VD49
02347A 110A AE 1A      A       LDX   #$1A
```

```
02348A 110C B6 49      A       LDA   MMIN
02349A 110E AD 67   1177       BSR   VD49
02350A 1110 AE 50      A       LDX   #$50
02351A 1112 BF 14      A       STX   XREG
02352A 1114 20 2E   1144       BRA   VD4A
02353A 1116 1B 11      A VD48  BCLR  6,FDIS  LEADING ZERO BIT
02354A 111B AE 04      A       LDX   #$04
02355A 111A DE 1A00    A       LDX   FAV,X
02356A 111D BF 14      A       STX   XREG
02357A 111F A6 01      A       LDA   #$01    CTR IN MEM
02358A 1121 B7 96      A       STA   XVD2
02359A 1123 BE 96      A VD41  LDX   XVD2    CTR IN MEM
02360A 1125 E6 A0      A       LDA   DKW,X   FETCH DATA
02361A 1127 BE 14      A       LDX   XREG
02362A 1129 AD 49   1174       BSR   VD47    STORE CHRS LEAD ZERO
02363A 112B 3C 96      A       INC   XVD2    CTR IN MEM
02364A 112D B6 96      A       LDA   XVD2    CTR IN MEM
02365A 112F A1 06      A       CMP   #$06    WHOLE NO DONE
02366A 1131 26 07   113A       BNE   VD43
02367A 1133 0C 11 04 113A      BRSET 6,FDIS,VD43
02368A 1136 A6 30      A       LDA   #$30    ZERO
02369A 1138 AD 34   116E       BSR   VD45    STACH
02370A 113A AD 35   1171 VD43  BSR   VD46    MTN4
02371A 113C BF 14      A       STX   XREG
02372A 113E B6 96      A       LDA   XVD2    CTR IN MEM
02373A 1140 A1 06      A       CMP   #$06
02374A 1142 26 DF   1123       BNE   VD41
02375A 1144 1D 11      A VD4A  BCLR  6,FDIS  LEAD ZERO
02376A 1146 A6 01      A       LDA   #$01    CTR IN MEM
02377A 1148 B7 96      A       STA   XVD2
02378A 114A BE 96      A VD42  LDX   XVD2    CTR IN MEM
02379A 114C E6 B0      A       LDA   DKVA,X
02380A 114E BE 14      A       LDX   XREG
02381A 1150 AD 22   1174       BSR   VD47    STCL
02382A 1152 3C 96      A       INC   XVD2    CTR IN MEM
02383A 1154 B6 96      A       LDA   XVD2    CTR IN MEM
02384A 1156 A1 06      A       CMP   #$06
02385A 1158 26 07   1161       BNE   VD44
02386A 115A 0C 11 04 1161      BRSET 6,FDIS,VD44
02387A 115D A6 30      A       LDA   #$30    ZERO
02388A 115F AD 0D   116E       BSR   VD45    STACH
02389A 1161 AD 0E   1171 VD44  BSR   VD46    MTN4
02390A 1163 BF 14      A       STX   XREG
02391A 1165 B6 96      A       LDA   XVD2
02392A 1167 A1 06      A       CMP   #$06
02393A 1169 26 DF   114A       BNE   VD42
02394A 116B 19 11      A       BCLR  4,FDIS  UPD BIT
02395A 116D 81                 RTS
02396A 116E CC 07D2    A VD45  JMP   STACH
02397A 1171 CC 10A5    A VD46  JMP   MTN4
02398A 1174 CC 0918    A VD47  JMP   STCL
02399A 1177 CC 07B3    A VD49  JMP   STCD
02400
02401                          **********************
02402                          *     SUB MODE 4;5;6 *
02403                          **********************
02404A 117A 06 11 0C 1189 MOD4  BRSET 3,FDIS,MD41 1ST ENT
02405A 117D 02 11 09 1189      BRSET 1,FDIS,MD41 1ST SW ON
02406A 1180 AD 23   11A5       BSR   ORGD
02407A 1182 09 11 19 119E      BRCLR 4,FDIS,MD42 UPD BIT
02408A 1185 AD 1B   11A2       BSR   MD44    VD4
02409A 1187 20 15   119E       BRA   MD42
02410A 1189 10 15      A MD41  BSET  0,FRED   PD BIT
02411A 118B 12 15      A       BSET  1,FRED   MD BIT
```

```
02412A 118D 14 15      A        BSET   2,FRED   TD BIT
02413A 118F AD 14    11A5       BSR    ORGD
02414A 1191 AD 47    11DA       BSR    UD456
02415A 1193 17 11      A        BCLR   3,FDIS
02416A 1195 13 11      A        BCLR   1,FDIS
02417A 1197 AD 06    119F       BSR    MD43     DBUS
02418A 1199 A6 0C      A        LDA    #%00001100 DISP ON
02419A 119B C7 0100    A        STA    DINS
02420A 119E 81           MD42   RTS
02421A 119F CC 0786    A MD43   JMP    DBUS
02422A 11A2 CC 10F0    A MD44   JMP    VD4
02423
02424                   ********************************
02425                   *      SUBROUTINE ORGINIZE DATA *
02426                   *      FOR MODES 4;5;6          *
02427                   ********************************
02428A 11A5 B6 93      A ORGD   LDA    MRD      MODE
02429A 11A7 A1 04      A        CMP    #$04     MOD4?
02430A 11A9 26 0B    11B6       BNE    ORGD1
02431A 11AB 01 15 28 11D6       BRCLR  0,FRED,ORGD4 PD BIT
02432A 11AE A6 C0      A        LDA    #%11000000 PD VAL TO MUL
02433A 11B0 B7 2F      A        STA    MEL
02434A 11B2 11 15      A        BCLR   0,FRED   PD BIT
02435A 11B4 20 1C    11D2       BRA    ORGD3
02436A 11B6 A1 05      A ORGD1  CMP    #$05     MOD5?
02437A 11B8 26 0B    11C5       BNE    ORGD2
02438A 11BA 03 15 19 11D6       BRCLR  1,FRED,ORGD4 MD BIT
02439A 11BD A6 30      A        LDA    #%00110000 MD VAL TO MUL
02440A 11BF B7 2F      A        STA    MEL
02441A 11C1 13 15      A        BCLR   1,FRED   MD BIT
02442A 11C3 20 0D    11D2       BRA    ORGD3
02443A 11C5 A1 06      A ORGD2  CMP    #$06     MOD6?
02444A 11C7 26 0D    11D6       BNE    ORGD4
02445A 11C9 05 15 0A 11D6       BRCLR  2,FRED,ORGD4 TD BIT
02446A 11CC A6 0E      A        LDA    #%00001110 TD VAL TO MUL
02447A 11CE B7 2F      A        STA    MEL
02448A 11D0 15 15      A        BCLR   2,FRED   TD BIT
02449A 11D2 AD 03    11D7 ORGD3 BSR    ORGD5    WVM
02450A 11D4 18 11      A        BSET   4,FDIS   UPD BIT
02451A 11D6 81           ORGD4  RTS
02452A 11D7 CC 098E    A ORGD5  JMP    WVM
02453
02454                   ****************************
02455                   *      SUB UPDATE MD4;5;6 *
02456                   *      DATA & VAR         *
02457                   ****************************
02458A 11DA B6 93      A UD456  LDA    MRD
02459A 11DC A0 04      A        SUB    #$04
02460A 11DE B7 13      A        STA    AREG
02461A 11E0 48                  LSLA           *3
02462A 11E1 BB 13      A        ADD    AREG
02463A 11E3 97                  TAX
02464A 11E4 DC 1820    A        JMP    U456,X
02465
02466
02467                   ******************************
02468                   *      M5 SUB UPDATE DISPLAY *
02469                   ******************************
02470A 11E7 AD 3F    1228 UD5   BSR    UD56     DBUS
02471A 11E9 A6 01      A        LDA    #%00000001 CLR DISP
02472A 11EB C7 0100    A        STA    DINS
02473A 11EE AD 3B    122B       BSR    ADDD
02474A 11F0 AE 00      A        LDX    #$00
02475A 11F2 D6 1D00    A UD52   LDA    MODF5,X
```

```
02476A 11F5 A1 20      A        CMP   #$20
02477A 11F7 27 02   11FB        BEQ   UD51
02478A 11F9 AD 2A   1225        BSR   UD55     STACH
02479A 11FB 5C            UD51  INX
02480A 11FC AD 3D   123B        BSR   BDDD
02481A 11FE A3 0F      A        CPX   #$0F     1&3 LINE DONE
02482A 1200 23 F0   11F2        BLS   UD52
02483A 1202 D6 1D00    A UD57   LDA   MODF5,X
02484A 1205 A1 20      A        CMP   #$20
02485A 1207 27 02   120B        BEQ   UD58
02486A 1209 AD 1A   1225        BSR   UD55
02487A 120B 5C            UD58  INX
02488A 120C A3 1F      A        CPX   #$1F
02489A 120E 23 F2   1202        BLS   UD57
02490A 1210 AE 40      A        LDX   #$40
02491A 1212 D6 1D00    A UD53   LDA   MODF5,X
02492A 1215 A1 20      A        CMP   #$20
02493A 1217 27 02   121B        BEQ   UD54
02494A 1219 AD 0A   1225        BSR   UD55     STACH
02495A 121B 5C            UD54  INX
02496A 121C AD 1D   123B        BSR   BDDD
02497A 121E A3 5F      A        CPX   #$5F     2&4 LINE DONE
02498A 1220 23 F0   1212        BLS   UD53
02499A 1222 CC 18F0    A        JMP   VD4
02500A 1225 CC 07D2    A UD55   JMP   STACH
02501A 1228 CC 0786    A UD56   JMP   DBUS
02502
02503                    **********************
02504                    *     ROUTINE ADDD *
02505                    **********************
02506A 122B 3F 97      A ADDD   CLR   MADH
02507A 122D A6 8C      A        LDA   #$8C
02508A 122F B7 93      A        STA   MADL
02509A 1231 AE 97      A        LDX   #MADH
02510A 1233 AD 03   1238        BSR   ADDD1    RMMR
02511A 1235 B7 1B      A        STA   BREG
02512A 1237 81                  RTS
02513A 1238 CC 06DC    A ADDD1  JMP   RMMR
02514
02515                    **********************
02516                    *     ROUTINE BDDD *
02517                    **********************
02518A 123B 9F            BDDD  TXA
02519A 123C AB 10      A        ADD   #$10
02520A 123E A4 1F      A        AND   #$1F
02521A 1240 BB 1B      A        ADD   BREG
02522A 1242 A1 0A      A        CMP   #$0A
02523A 1244 26 05   124B        BNE   BDD1
02524A 1246 A6 2E      A        LDA   #$2E     "."
02525A 1248 AD 02   124C        BSR   BDD2     STACH
02526A 124A 5C                  INX
02527A 124B 81            BDD1  RTS
02528A 124C CC 07D2    A BDD2   JMP   STACH
```

```
02530
02531                    ****************************
02532                    *     M6 SUB UPDATE DISPLAY *
02533                    ****************************
02534A 124F AD 31   1282 UD6   BSR   UD66    DRUS
02535A 1251 A6 01     A       LDA   #%00000001 CLR DISP
02536A 1253 C7 0100   A       STA   DINS
02537A 1256 AD D3   122B      BSR   ADDD
02538A 1258 AE 00     A       LDX   #$00
02539A 125A D6 1D20   A UD62  LDA   MODF6,X
02540A 125D A1 20     A       CMP   #$20
02541A 125F 27 02   1263      BEQ   UD61
02542A 1261 AD 1C   127F      BSR   UD65    STACH
02543A 1263 5C          UD61  INX
02544A 1264 AD D5   123B      BSR   BDDD
02545A 1266 A3 1F     A       CPX   #$1F    1&3 LINE DONE
02546A 1268 23 F0   125A      BLS   UD62
02547A 126A AE 40     A       LDX   #$40
02548A 126C D6 1D20   A UD63  LDA   MODF6,X
02549A 126F A1 20     A       CMP   #$20
02550A 1271 27 02   1275      BEQ   UD64
02551A 1273 AD 0A   127F      BSR   UD65    STACH
02552A 1275 5C          UD64  INX
02553A 1276 AD C3   123B      BSR   BDDD
02554A 1278 A3 5F     A       CPX   #$5F    2&4 LINE DONE
02555A 127A 23 F0   126C      BLS   UD63
02556A 127C CC 10F0   A       JMP   VD4
02557A 127F CC 0702   A UD65  JMP   STACH
02558A 1282 CC 0786   A UD66  JMP   DRUS
```

```
02560
02561                    ******************************
02562                    *     MODE3 MOVE TO TOP LOC  *
02563                    ******************************
02564A 1285 AD 09   1290 MT3   BSR   MT32    PREPAIR X FOR TAB
02565A 1287 DE 1780    A       LDX   MODV3,X  GET ADDRESS FROM TAB
02566A 128A AD 01   128D       BSR   MT31    MCX  MOVE CURSER
02567A 128C 81                 RTS
02568A 128D CC 090F    A MT31  JMP   MCX
02569A 1290 CC 0EF7    A MT32  JMP   MTL
02570
02571                    ***********************************
02572                    *     MODE3 MOVE TO PREVIOUS LOC  *
02573                    ***********************************
02574A 1293 C6 0180    A MP3   LDA   DINS    GET PRESENT ADDRESS
02575A 1296 A4 7F      A       AND   #$7F
02576A 1298 97                 TAX
02577A 1299 DE 1780    A       LDX   MODV3,X  FROM TAB
02578A 129C AD 01   129F       BSR   MP31    MCX
02579A 129E 81                 RTS
02580A 129F CC 090F    A MP31  JMP   MCX
02581
02582                    ********************************
02583                    *     MODE3 MOVE TO NEXT LOC   *
02584                    ********************************
02585A 12A2 AB 20      A MN3   ADD   #$20    ADD OFFSET OF TAB
02586A 12A4 97                 TAX
02587A 12A5 DE 1780    A       LDX   MODV3,X
02588A 12A8 AD 01   12AB       BSR   MN31    MCX
02589A 12AA 81                 RTS
02590A 12AB CC 090F    A MN31  JMP   MCX
02591
02592                    ******************************
02593                    *     MODE3 MAKE X POINT TO  *
02594                    *     NEXT VAR               *
02595                    ******************************
02596A 12AE B7 13      A MTN3  STA   AREG    SAVE AREG
02597A 12B0 9F                 TXA
02598A 12B1 AB 20      A       ADD   #$20
02599A 12B3 97                 TAX
02600A 12B4 D6 1780    A       LDA   MODV3,X
02601A 12B7 A4 7F      A       AND   #$7F
02602A 12B9 97                 TAX
02603A 12BA B6 13      A       LDA   AREG
02604A 12BC 81                 RTS
02605
02606                    ******************************
02607                    *     SUBROUTINE CHNG MOD3   *
02608                    ******************************
02609A 12BD 3F 97      A CM3   CLR   MADH    HIGH ADD IN MEM
02610A 12BF A6 87      A       LDA   #$87
02611A 12C1 B7 98      A       STA   MADL
02612A 12C3 AE 03      A       LDX   #$03    MOD3
02613A 12C5 DE 1A00    A       LDX   FAV,X
02614A 12C8 BF 96      A       STX   XVD2    DISP  ADDRESS
02615A 12CA AD 27   12F3       BSR   CM32    GTCD
02616A 12CC BF 96      A       STX   XVD2
02617A 12CE AD 29   12F9       BSR   CM34    WRITE TO MEM
02618A 12D0 AD 16   12E8       BSR   CM31
02619A 12D2 AD 14   12E8       BSR   CM31
02620A 12D4 AD 12   12E8       BSR   CM31
02621A 12D6 BE 96      A       LDX   XVD2
02622A 12D8 AD D4   12AE       BSR   MTN3
02623A 12DA AD 20   12FC       BSR   CM35    GACD
```

```
02624A 12DC A4 0F      A        AND   #$0F     ASCII TO BCD
02625A 12DE AD 19   12F9        BSR   CM34     WMEM
02626A 12E0 AE 0C      A        LDX   #Z00001100 BLINK OFF
02627A 12E2 AD 12   12F6        BSR   CM33     DBUS
02628A 12E4 CF 0100    A        STX   DINS
02629A 12E7 81                  RTS
02630A 12E8 BE 96      A CM31   LDX   XVD2
02631A 12EA AD C2   12AE        BSR   MTN3
02632A 12EC AD 05   12F3        BSR   CM32     GTCD
02633A 12EE BF 96      A        STX   XVD2
02634A 12F0 AD 07   12F9        BSR   CM34     WMEM
02635A 12F2 81                  RTS
02636A 12F3 CC 0793    A CM32   JMP   GTCD
02637A 12F6 CC 0786    A CM33   JMP   DBUS
02638A 12F9 CC 0AF3    A CM34   JMP   WMEM
02639A 12FC CC 07A5    A CM35   JMP   GACD
02640
02641                           ****************************
02642                           *      SUBROUTINE MODE 3 *
02643                           ****************************
02644A 12FF 0E 93 15 1317 MOD3   BRSET 7,MRD,MD32 SET MODE
02645A 1302 06 11 05 130A        BRSET 3,FDIS,MD31 1ST TIME ENT MOD
02646A 1305 02 11 02 130A        BRSET 1,FDIS,MD31 1ST TIME SW ON
02647A 1308 20 0D    1317        BRA   MD32
02648A 130A AD 0F    131B MD31   BSR   UD3
02649A 130C 13 11      A         BCLR  1,FDIS
02650A 130E 17 11      A         BCLR  3,FDIS
02651A 1310 AD 06    1318        BSR   MD33     DBUS
02652A 1312 A6 0C      A         LDA   #Z00001100
02653A 1314 C7 0100    A         STA   DINS
02654A 1317 81           MD32   RTS
02655A 1318 CC 0786    A MD33   JMP   DBUS
02656
02657                           ******************************
02658                           *      M3 SUB UPDATE DISPLAY *
02659                           ******************************
02660A 131B AD 5C    1379 UD3    BSR   VD34     DBUS DISP BUSY
02661A 131D A6 01      A         LDA   #Z00000001 CLR DISP
02662A 131F C7 0100    A         STA   DINS
02663A 1322 AE 00      A         LDX   #$00
02664A 1324 D6 1CA0    A UD32    LDA   MODF3,X
02665A 1327 A1 20      A         CMP   #$20
02666A 1329 27 02    132D        BEQ   UD31
02667A 132B AD 4F    137C        BSR   VD35     STACH
02668A 132D 5C           UD31   INX
02669A 132E A3 1F      A         CPX   #$1F     1&3 LINE DONE
02670A 1330 23 F2    1324        BLS   UD32
02671A 1332 AE 40      A         LDX   #$40
02672A 1334 D6 1CA0    A UD33    LDA   MODF3,X
02673A 1337 A1 20      A         CMP   #$20
02674A 1339 27 02    133D        BEQ   UD34
02675A 133B AD 3F    137C        BSR   VD35     STACH
02676A 133D 5C           UD34   INX
02677A 133E A3 5F      A         CPX   #$5F     2&4 LINE DONE
02678A 1340 23 F2    1334        BLS   UD33
02679A 1342 AE 03      A VD3     LDX   #$03     MOD3
02680A 1344 DE 1A00    A         LDX   FAV,X
02681A 1347 BF 96      A         STX   XVD2
02682A 1349 3F 97      A         CLR   MADH
02683A 134B A6 87      A         LDA   #$87
02684A 134D B7 98      A         STA   MADL
02685A 134F AD 22    1373        BSR   VD32     RMEM
02686A 1351 BE 96      A         LDX   XVD2
02687A 1353 AD 21    1376        BSR   VD33     STCD
```

```
02688A 1355 BF 96      A        STX   XVD2
02689A 1357 AD 0F   1368        BSR   VD31
02690A 1359 AD 0D   1368        BSR   VD31
02691A 135B AD 0B   1368        BSR   VD31
02692A 135D AD 14   1373        BSR   VD32     RMEM
02693A 135F AB 30      A        ADD   #$30     TO ASCII
02694A 1361 BE 96      A        LDX   XVD2
02695A 1363 AD 1A   137F        BSR   VD36     MTN3
02696A 1365 AD 15   137C        BSR   VD35     STACH
02697A 1367 81                  RTS
02698A 1368 AD 09   1373 VD31   BSR   VD32     RMEM
02699A 136A BE 96      A        LDX   XVD2
02700A 136C AD 11   137F        BSR   VD36     MTN3
02701A 136E AD 06   1376        BSR   VD33     STCD
02702A 1370 BF 96      A        STX   XVD2
02703A 1372 81                  RTS
02704A 1373 CC 0980    A VD32   JMP   RMEM
02705A 1376 CC 0783    A VD33   JMP   STCD
02706A 1379 CC 0786    A VD34   JMP   DBUS
02707A 137C CC 0782    A VD35   JMP   STACH
02708A 137F CC 12AE    A VD36   JMP   MTN3
```

```
02710
02711                   ***************************
02712                   *     ROUTINES FOR MODE7 *
02713                   *     PULSE LOGGING      *
02714                   ***************************
02715A 1382 81          MT7   RTS
02716A 1383 81          MP7   RTS
02717A 1384 81          MTN7  RTS
02718
02719                   ************************
02720                   *     MOVE TO NEXT VAR *
02721                   ************************
02722A 1385 AE 07      A MN7   LDX   #$07     MD7
02723A 1387 D6 1A00    A       LDA   FAV,X    MAKE X POINT
02724A 138A AB 80      A       ADD   #$80     TO FIRST VAR
02725A 138C 97                 TAX
02726A 138D AD 1C    13AB      BSR   MN74     MCX
02727A 138F B6 90      A       LDA   KKV      NO PRESSED
02728A 1391 26 04    1397      BNE   MN71
02729A 1393 3C 90      A       INC   KKV
02730A 1395 20 08    139F      BRA   MN72
02731A 1397 A1 03      A MN71  CMP   #$03
02732A 1399 23 0F    13AA      BLS   MN73
02733A 139B A6 03      A       LDA   #$03
02734A 139D B7 90      A       STA   KKV
02735A 139F B6 90      A MN72  LDA   KKV
02736A 13A1 AB 30      A       ADD   #$30
02737A 13A3 CE 0100    A       LDX   DINS
02738A 13A6 AD 06    13AE      BSR   MN75     STACH
02739A 13A8 20 DB    1385      BRA   MN7
02740A 13AA 81          MN73  RTS
02741A 13AB CC 090F    A MN74  JMP   MCX
02742A 13AE CC 07D2    A MN75  JMP   STACH
02743
02744                   ************************
02745                   *     MODE 7 ROUTINE *
02746                   ************************
02747A 13B1 0E 93 15 13C9 MOD7  BRSET 7,MSD,MD72 SET MODE
02748A 13B4 06 11 05 13BC      BRSET 3,FDIS,MD71 1ST ENT MD
02749A 13B7 02 11 02 13BC      BRSET 1,FDIS,MD71 1ST SW ON
02750A 13BA 20 0D    13C9      BRA   MD72
02751A 13BC AD 0F    13CD MD71 BSR   UD7
02752A 13BE 13 11      A       BCLR  1,FDIS
02753A 13C0 17 11      A       BCLR  3,FDIS
02754A 13C2 AD 06    13CA      BSR   MD73     DBUS
02755A 13C4 A6 0C      A       LDA   #%00001100
02756A 13C6 C7 0100    A       STA   DINS
02757A 13C9 81          MD72  RTS
02758A 13CA CC 0786    A MD73  JMP   DBUS
02759
02760                   ******************************
02761                   *     UPDATE DISP FOR MODE 7 *
02762                   ******************************
02763A 13CD AD 41    1410 UD7   BSR   VD73     DBUS
02764A 13CF A6 01      A       LDA   #%00000001 CLR DISP
02765A 13D1 C7 0100    A       STA   DINS
02766A 13D4 AE 00      A       LDX   #$00
02767A 13D6 D6 1DA0    A UD72  LDA   MODF7,X
02768A 13D9 A1 20      A       CMP   #$20     SPACE
02769A 13DB 27 02    13DF      BEQ   UD71
02770A 13DD AD 2B    140A      BSR   VD71     STACH
02771A 13DF 5C          UD71  INX
02772A 13E0 A3 1F      A       CPX   #$1F     1&3 LINE DONE
02773A 13E2 23 F2    13D6      BLS   UD72
```

```
02774A 13E4 AE 40      A       LDX   #$40
02775A 13E6 D6 1DA8    A UD73  LDA   MODF7,X
02776A 13E9 A1 20      A       CMP   #$20
02777A 13EB 27 02   13EF       BEQ   UD74
02778A 13ED AD 1B   140A       BSR   VD71     STACH
02779A 13EF 5C            UD74 INX
02780A 13F0 A3 5F      A       CPX   #$5F     264 LINE DONE
02781A 13F2 23 F2   13E6       BLS   UD73
02782A 13F4 AE 07      A VD7   LDX   #$07
02783A 13F6 DE 1A00    A       LDX   FAV,X
02784A 13F9 B6 42      A       LDA   PLOG
02785A 13FB AB 30      A       ADD   #$30
02786A 13FD AD 0B   140A       BSR   VD71     STACH
02787A 13FF AE 07      A       LDX   #$07
02788A 1401 D6 1A00    A       LDA   FAV,X
02789A 1404 AB 80      A       ADD   #$80
02790A 1406 97                 TAX
02791A 1407 AD 04   140D       BSR   VD72     MCX
02792A 1409 81                 RTS
02793A 140A CC 0782    A VD71  JMP   STACH
02794A 140D CC 090F    A VD72  JMP   MCX
02795A 1410 CC 0786    A VD73  JMP   DBUS
02796
02797                          ********************************
02798                          *     CHANGE ROUTINE FOR MODE 7 *
02799                          ********************************
02800A 1413 AE 07      A CM7   LDX   #$07
02801A 1415 DE 1A00    A       LDX   FAV,X
02802A 1418 AD 1B   1435       BSR   CM75     GACD
02803A 141A A4 0F      A       AND   #$0F     TO BCD
02804A 141C B1 42      A       CMP   PLOG
02805A 141E 27 08   1428       BEQ   CM7H
02806A 1420 A1 01      A       CMP   #$01     IF CONTINUE
02807A 1422 26 07   142B       BNE   CM71
02808A 1424 1A 27      A       BSET  5,CARR   ENABLE SWITCH ON
02809A 1426 B7 42      A       STA   PLOG
02810A 1428 CC 14B6    A CM7H  JMP   CM74
02811A 142B A1 02      A CM71  CMP   #$02
02812A 142D 26 09   1438       BNE   CM72
02813A 142F 1B 27      A       BCLR  5,CARR   STOP COUNT
02814A 1431 B7 42      A       STA   PLOG
02815A 1433 20 F3   1428       BRA   CM7H
02816A 1435 CC 07A5    A CM75  JMP   GACD
02817A 1438 3F 95      A CM72  CLR   DEMP
02818A 143A 3C 95      A       INC   DEMP
02819A 143C B7 42      A       STA   PLOG     PULSE LOG
02820A 143E 3F 9C      A       CLR   SAH
02821A 1440 A6 B0      A       LDA   #STMEM
02822A 1442 4A                 DECA
02823A 1443 B7 9D      A       STA   SAL
02824A 1445 AE 9C      A       LDX   #SAH
02825A 1447 A6 FF      A       LDA   #$FF
02826A 1449 AD 73   14BE       BSR   CM76     IACD
02827A 144B 1A 27      A       BSET  5,CARR
02828A 144D 1D 27      A       BCLR  6,CARR   DAY CHANG BIT
02829A 144F 0E 8A FD 144F CM73 BRSET 7,ARTC,CM73 RTC BUSY
02830A 1452 A6 01      A       LDA   #$01
02831A 1454 B7 86      A       STA   DOW
02832A 1456 B6 27      A       LDA   CARR
02833A 1458 A4 F0      A       AND   #$F0
02834A 145A AB 01      A       ADD   #$01
02835A 145C B7 27      A       STA   CARR
02836A 145E 3F 97      A       CLR   MADH
02837A 1460 A6 80      A       LDA   #$80
```

```
02838A 1462 B7 98       A       STA   MADL
02839A 1464 AE 97       A       LDX   #MADH
02840A 1466 0E BA FD 1466 CM7B  BRSET 7,ARTC,CM7B RTC BUSY
02841A 1469 B6 84       A       LDA   HRS
02842A 146B B7 48       A       STA   MHH      HRS FOR MAX DEMAND
02843A 146D AD 4F    14BE       BSR   CM76     IACD
02844A 146F 0E BA FD 146F CM7C  BRSET 7,ARTC,CM7C RTC BUSY
02845A 1472 B6 82       A       LDA   MIN
02846A 1474 B7 49       A       STA   MMIN     MIN FOR MAX DEMAND
02847A 1476 AD 46    14BE       BSR   CM76     IACD
02848A 1478 0E BA FD 1478 CM7D  BRSET 7,ARTC,CM7D RTC BUSY
02849A 147B B6 80       A       LDA   SEC
02850A 147D AD 3F    14BE       BSR   CM76     IACD
02851A 147F 0E BA FD 147F CM7E  BRSET 7,ARTC,CM7E RTC BUSY
02852A 1482 B6 89       A       LDA   YEAR
02853A 1484 AD 38    14BE       BSR   CM76     IACD
02854A 1486 0E BA FD 1486 CM7F  BRSET 7,ARTC,CM7F RTC BUSY
02855A 1489 B6 88       A       LDA   MON
02856A 148B B7 47       A       STA   MMD      MONTH FOR MAX DEMAND
02857A 148D AD 2F    14BE       BSR   CM76     IACD
02858A 148F 0E BA FD 148F CM7G  BRSET 7,ARTC,CM7G RTC BUSY
02859A 1492 B6 87       A       LDA   DATE
02860A 1494 B7 46       A       STA   MDD      DATE FOR MAX DEMAND
02861A 1496 AD 26    14BE       BSR   CM76     IACD
02862A 1498 B6 95       A       LDA   DEMP
02863A 149A AD 22    14BE       BSR   CM76     IACD
02864A 149C AE 26       A       LDX   #TDW     CLR PULSE
02865A 149E 7F            CM79  CLR   ,X
02866A 149F 5A                  DEX
02867A 14A0 A3 20       A       CPX   #PDW-1
02868A 14A2 24 FA    149E       BHS   CM79
02869A 14A4 AE 2E       A       LDX   #TDVA
02870A 14A6 7F            CM7A  CLR   ,X
02871A 14A7 5A                  DEX
02872A 14A8 A3 28       A       CPX   #PDVA-1
02873A 14AA 24 FA    14A6       BHS   CM7A
02874A 14AC 3F 9A       A       CLR   CW
02875A 14AE 3F 9B       A       CLR   CVA
02876A 14B0 3F 8E       A       CLR   CR
02877A 14B2 17 15       A       BCLR  3,FRED
02878A 14B4 19 15       A       BCLR  4,FRED
02879A 14B6 AE 0C       A CM74  LDX   #%000001100
02880A 14B8 AD 07    14C1       BSR   CM77     DBUS
02881A 14BA CF 0100     A       STX   DINS
02882A 14BD 81                  RTS
02883A 14BE CC 0694     A CM76  JMP   IACD
02884A 14C1 CC 0786     A CM77  JMP   DBUS
02885
```

```
02887                    ************************
02888                    *     MAIN INT ROUTINE *
02889                    ************************
02890
02891A 14C4 9B           MINT   SEI
02892A 14C5 00 12 05 14CD       BRSET  0,FLGN,MIN1 KEYISTILL PRESSED
02893A 14C8 03 00 02 14CD       BRCLR  1,PORTA,MIN1 KEYBOARD STR PULSE
02894A 14CB AD 27    14F4       BSR    KINT
02895A 14CD AD 37    1506 MIN1  BSR    RINT
02896A 14CF 01 19 0A 14DC       BRCLR  0,GNFL,MIN2 RS232-OFF
02897A 14D2 0D 01 1A 14EF       BRCLR  6,PORTB,MIN3 NO INT FROM RS232
02898A 14D5 0A 12 17 14EF       BRSET  5,FLGN,MIN3 BYTE RX BIT
02899A 14D8 AD 17    14F1       BSR    MIN4     RXSR
02900A 14DA 20 13    14EF       BRA    MIN3
02901A 14DC 0C 01 10 14EF MIN2  BRSET  6,PORTB,MIN3 RS232 NOT CONECTED
02902A 14DF 10 19       A       BSET   0,GNFL   SWITCH ON
02903A 14E1 1A 12       A       BSET   5,FLGN
02904A 14E3 A6 0F       A       LDA    #$0F     20MS DELAY
02905A 14E5 B7 09       A       STA    TCR
02906A 14E7 A6 A0       A       LDA    #$A0
02907A 14E9 B7 03       A       STA    TDR
02908A 14EB 12 19       A       BSET   1,GNFL
02909A 14ED 1E 01       A       BSET   7,PORTB  O/P IDLE STATE
02910A 14EF 9A             MIN3 CLI
02911A 14F0 80                  RTI
02912A 14F1 CC 0AFF     A MIN4  JMP    RXSR
02913
02914                    ********************************
02915                    *     KEYBOARD INT SUBROUTINE *
02916                    ********************************
02917
02918A 14F4 B6 01       A KINT  LDA    PORTB    GET KEY CHR
02919A 14F6 A4 1F       A       AND    #$1F     MASK OFF MS3B
02920A 14F8 97                  TAX
02921A 14F9 D6 1800     A       LDA    KBTAB,X  GET CODE FROM TABLE
02922A 14FC B7 90       A       STA    KKV
02923A 14FE 3F 45       A       CLR    DUCTR
02924A 1500 10 12       A       BSET   0,FLGN   SET REREAD FLAG-KB
02925A 1502 81                  RTS
02926
02927                    ************************
02928                    *     RTC INT SUBROUTINE *
02929                    ************************
02930
02931A 1503 CC 159C     A RIN11 JMP    RIN3     JMP INST FOR LONG BR
02932A 1506 B6 0C       A RINT  LDA    CRTC     INT STATUS
02933A 1508 BA 10       A       ORA    FRTC
02934A 150A B7 10       A       STA    FRTC
02935A 150C 0F 10 F4 1503       BRCLR  7,FRTC,RIN11 TOT INT GENERATED?
02936A 150F 0D 10 40 1552       BRCLR  6,FRTC,RIN1 PERIODIC INT
02937A 1512 3D 0F       A       TST    CPI      IF CTR=0 FOR 1ST TIME
02938A 1514 27 33    1549       BEQ    INP5     TIME MAKE=10(80MS)
02939A 1516 3A 0F       A       DEC    CPI      PERIODIC INT CTR
02940A 1518 3D 0F       A       TST    CPI
02941A 151A 26 31    154D       BNE    INP4
02942A 151C 14 10       A       BSET   2,FRTC
02943A 151E 03 10 0D 152E       BRCLR  1,FRTC,RINB SAMP PERIOD ENDED
02944A 1521 B6 44       A       LDA    ACTRS    SEC STORED 1MIN
02945A 1523 0E 91 FD 1523 RINA  BRSET  7,ACTR,RINA
02946A 1526 B1 80       A       CMP    SEC
02947A 1528 25 04    152E       BLO    RINB
02948A 152A 16 10       A       BSET   3,FRTC
02949A 152C 13 10       A       BCLR   1,FRTC
02950A 152E 0C 01 06 1537 RINB  BRSET  6,PORTB,RIN7 CHECK IF RS232 IS
```

```
02951A 1531 A6 1E       A       LDA   #$1E     DISCONECTED FOR MORE
02952A 1533 B7 43       A       STA   CTRS     THAN 3SEC AND
02953A 1535 20 12     1549      BRA   INP5     SWITCH RS232 OFF
02954A 1537 01 19 0F 1549 RIN7  BRCLR 0,GNFL,INP5
02955A 153A 06 12 0C 1549       BRSET 3,FLGN,INP5
02956A 153D 3A 43       A       DEC   CTRS     RS232 COUNTER
02957A 153F 26 08     1549      BNE   INP5
02958A 1541 11 19       A       BCLR  0,GNFL   OFF RS232
02959A 1543 1F 01       A       BCLR  7,PORTB
02960A 1545 A6 48       A       LDA   #$48     TIMER INT OFF
02961A 1547 B7 08       A       STA   TDR
02962A 1549 A6 0A       A INP5  LDA   #$0A     100MS
02963A 154B B7 0F       A       STA   CPI
02964A 154D 0B 27 02 1552 INP4  BRCLR 5,CARR,RIN1 IF COUNT DISABLED
02965A 1550 AD 54     15A6      BSR   INPT     CHECK I/PS POINTS
02966A 1552 09 10 02 1557 RIN1  BRCLR 4,FRTC,RIN2 UPDATE INT
02967A 1555 10 10       A       BSET  0,FRTC
02968A 1557 0B 10 42 159C RIN2  BRCLR 5,FRTC,RIN3 ALARM INT
02969A 155A 04 11 13 1570       BRSET 2,FDIS,RIN10
02970A 155D 3C 45       A       INC   DOCTR
02971A 155F B6 45       A       LDA   DOCTR
02972A 1561 A1 0A       A       CMP   #$0A     10MIN
02973A 1563 25 0B     1570      BLO   RIN10
02974A 1565 3F 45       A       CLR   DOCTR
02975A 1567 14 11       A       BSET  2,FDIS
02976A 1569 1A 01       A       BSET  5,PORTB
02977A 156B A6 08       A       LDA   #%00001000
02978A 156D C7 0100     A       STA   DINS     DISP OFF
02979A 1570 B6 27       A RIN10 LDA   CARR
02980A 1572 A4 0F       A       AND   #$0F
02981A 1574 0E 8A FD 1574 RIN8  BRSET 7,ARTC,RIN8 RTC BUSY
02982A 1577 B1 86       A       CMP   DOW
02983A 1579 27 0D     1588      BEQ   RIN5
02984A 157B B6 27       A       LDA   CARR
02985A 157D A4 F0       A       AND   #$F0
02986A 157F 0E 8A FD 157F RIN9  BRSET 7,ARTC,RIN9 RTC BUSY
02987A 1582 BA 86       A       ORA   DOW
02988A 1584 B7 27       A       STA   CARR
02989A 1586 1C 27       A       BSET  6,CARR   DAY CHNG BIT
02990A 1588 0B 27 11 159C RIN5  BRCLR 5,CARR,RIN3
02991A 158B 3C 91       A       INC   ACTR
02992A 158D B6 91       A       LDA   ACTR
02993A 158F AD 12     15A3      BSR   RIN6
02994A 1591 B7 91       A       STA   ACTR
02995A 1593 B1 92       A       CMP   SPR
02996A 1595 23 05     159C      BLS   RIN3
02997A 1597 06 10 02 159C       BRSET 3,FRTC,RIN3
02998A 159A 12 10       A       BSET  1,FRTC   SAMPLED PER ENDED
02999A 159C B6 10       A RIN3  LDA   FRTC
03000A 159E A4 0F       A       AND   #$0F     MAKE MS4B-0
03001A 15A0 B7 10       A       STA   FRTC
03002A 15A2 81                  RTS
03003A 15A3 CC 660A     A RIN6  JMP   DAAC
03004
03005                           ************************************
03006                           *     SUBROUTINE FOR READING I/PS  *
03007                           ************************************
03008A 15A6 1B 80       A INPT  BSET  4,PORTA  LEDS OF OPTOS ON
03009A 15A8 0E 00 02 15AD       BRSET 7,PORTA,INP1 KW I/P
03010A 15AB 1E 9A       A       BSET  7,CW     KW CTR TEMP I/P
03011A 15AD 0C 00 02 15B2 INP1  BRSET 6,PORTA,INP2 KVA I/P
03012A 15B0 1E 9B       A       BSET  7,CVA    KVA CTR TEMP I/P
03013A 15B2 0A 00 02 15B7 INP2  BRSET 5,PORTA,INP3 RESET I/P
03014A 15B5 1E 8E       A       BSET  7,CR     RESET CTR TEMP I/P
```

```
03015A 15B7 19 88       A INP3   BCLR   4,PORTA  LEDS OFF
03016A 15B9 0C 9A 23 15DF IN1    BRSET  6,CW,IN10 BEING DEBOUNCED BIT
03017A 15BC 0E 9A 19 15D8        BRSET  7,CW,IN11 TEMP I/P BIT
03018A 15BF 0B 9A 36 15F8        BRCLR  5,CW,IN12 ;DEBOUNCED BIT
03019A 15C2 B6 9A       A        LDA    CW       ;
03020A 15C4 A4 0F       A        AND    #$0F     ;
03021A 15C6 A1 03       A        CMP    #$03     ;
03022A 15C8 24 04    15CE        BHS    IN14     ;ALLOW 30MS OFF
03023A 15CA 3C 9A       A        INC    CW       ;BEFORE NEXT
03024A 15CC 20 2A    15F8        BRA    IN12     ;
03025A 15CE B6 9A       A IN14   LDA    CW       ;
03026A 15D0 A4 F0       A        AND    #$F0     ;
03027A 15D2 B7 9A       A        STA    CW       ;
03028A 15D4 1B 9A       A        BCLR   5,CW     DEBOUNCED BIT
03029A 15D6 20 20    15F8        BRA    IN12
03030A 15D8 0A 9A 15 15F0 IN11   BRSET  5,CW,IN13 DEBOUNCED BIT
03031A 15DB 1C 9A       A        BSET   6,CW     BEING DEBOUNCED
03032A 15DD 20 19    15F8        BRA    IN12
03033A 15DF 3C 9A       A IN10   INC    CW       KW COUNTER
03034A 15E1 B6 9A       A        LDA    CW
03035A 15E3 A4 0F       A        AND    #$0F     MASK OFF FLAGS
03036A 15E5 A1 07       A        CMP    #$07     70MS DEBOUNCE PER
03037A 15E7 25 0F    15F8        BLO    IN12
03038A 15E9 0F 9A 04 15F0        BRCLR  7,CW,IN13 TEMP BIT
03039A 15EC 1A 9A       A        BSET   5,CW     DEBOUNCED BIT
03040A 15EE 16 15       A        BSET   3,FRED   TRIGGER BIT
03041A 15F0 1D 9A       A IN13   BCLR   6,CW
03042A 15F2 B6 9A       A        LDA    CW       CLR COUNTER
03043A 15F4 A4 F0       A        AND    #$F0
03044A 15F6 B7 9A       A        STA    CW
03045A 15F8 1F 9A       A IN12   BCLR   7,CW     TEMP BIT
03046A 15FA 0C 9B 23 1620 IN2    BRSET  6,CVA,IN20 BEING DEBOUNCED BIT
03047A 15FD 0E 9B 19 1619        BRSET  7,CVA,IN21 TEMP I/P BIT
03048A 1600 0B 9B 36 1639        BRCLR  5,CVA,IN22 ;
03049A 1603 B6 9B       A        LDA    CVA      ;
03050A 1605 A4 0F       A        AND    #$0F     ;ALLOW 30MS OFF
03051A 1607 A1 03       A        CMP    #$03     ;
03052A 1609 24 04    160F        BHS    IN24     ;BEFORE NEXT
03053A 160B 3C 9B       A        INC    CVA      ;PULSE
03054A 160D 20 2A    1639        BRA    IN22     ;
03055A 160F B6 9B       A IN24   LDA    CVA      ;
03056A 1611 A4 F0       A        AND    #$F0     ;
03057A 1613 B7 9B       A        STA    CVA      ;
03058A 1615 1B 9B       A        BCLR   5,CVA    DEBOUNCED BIT
03059A 1617 20 20    1639        BRA    IN22
03060A 1619 0A 9B 15 1631 IN21   BRSET  5,CVA,IN23 DEBOUNCED BIT
03061A 161C 1C 9B       A        BSET   6,CVA    BEING DEBOUNCED
03062A 161E 20 19    1639        BRA    IN22
03063A 1620 3C 9B       A IN20   INC    CVA      KVA COUNTER
03064A 1622 B6 9B       A        LDA    CVA
03065A 1624 A4 0F       A        AND    #$0F     MASK OFF FLAGS
03066A 1626 A1 07       A        CMP    #$07     70MS DEBOUNCE PER
03067A 1628 25 0F    1639        BLO    IN22
03068A 162A 0F 9B 04 1631        BRCLR  7,CVA,IN23 TEMP BIT
03069A 162D 1A 9B       A        BSET   5,CVA    DEBOUNCED BIT
03070A 162F 18 15       A        BSET   4,FRED   TRIGGER BIT
03071A 1631 1D 9B       A IN23   BCLR   6,CVA
03072A 1633 B6 9B       A        LDA    CVA      CLR COUNTER
03073A 1635 A4 F0       A        AND    #$F0
03074A 1637 B7 9B       A        STA    CVA
03075A 1639 1F 9B       A IN22   BCLR   7,CVA    TEMP BIT
03076A 163B 0C 8E 23 1661 IN3    BRSET  6,CR,IN30 BEING DEBOUNCED BIT
03077A 163E 0E 8E 19 165A        BRSET  7,CR,IN31 TEMP I/P BIT
03078A 1641 0B 8E 36 167A        BRCLR  5,CR,IN32 ;
```

```
03079A 1644 B6 8E      A       LDA   CR      ;
03080A 1646 A4 0F      A       AND   #$0F    ;ALLOW 30MS DELAY
03081A 1648 A1 03      A       CMP   #$03    ;
03082A 164A 24 04   1650       BHS   IN34    ;BEFORE NEXT
03083A 164C 3C 8E      A       INC   CR      ;PULSE
03084A 164E 20 2A   167A       BRA   IN32    ;
03085A 1650 B6 8E      A IN34  LDA   CR      ;
03086A 1652 A4 F0      A       AND   #$F0    ;
03087A 1654 B7 8E      A       STA   CR      ;
03088A 1656 1B 8E      A       BCLR  5,CR    DEBOUNCED BIT
03089A 1658 20 20   167A       BRA   IN32
03090A 165A 0A 8E 15 1672 IN31  BRSET 5,CR,IN33 DEBOUNCED BIT
03091A 165D 1C 8E      A       BSET  6,CR    BEING DEBOUNCED
03092A 165F 20 19   167A       BRA   IN32
03093A 1661 3C 8E      A IN30  INC   CR      RESET COUNTER
03094A 1663 B6 8E      A       LDA   CR
03095A 1665 A4 0F      A       AND   #$0F    MASK OFF FLAGS
03096A 1667 A1 07      A       CMP   #$07    70MS DEBOUNCE PER
03097A 1669 25 0F   167A       BLO   IN32
03098A 166B 0F 8E 04 1672      BRCLR 7,CR,IN33 TEMP BIT
03099A 166E 1A 8E      A       BSET  5,CR    DEBOUNCED BIT
03100A 1670 1A 15      A       BSET  5,FRED  TRIGGER BIT
03101A 1672 1D 8E      A IN33  BCLR  6,CR
03102A 1674 B6 8E      A       LDA   CR      CLR COUNTER
03103A 1676 A4 F0      A       AND   #$F0
03104A 1678 B7 8E      A       STA   CR
03105A 167A 1F 8E      A IN32  BCLR  7,CR    TEMP BIT
03106A 167C 81                 RTS
03107
03108                    ********************************
03109                    *     TIMER INT FROM WAIT STATE *
03110                    ********************************
03111
03112A 167D 0F 11 05 1685 TIWS  BRCLR 7,FDIS,TIW1
03113A 1680 A6 48      A       LDA   %Z0104000 STOP TIMER
03114A 1682 B7 09      A       STA   TCR
03115A 1684 80                 RTI
03116A 1685 CC 0820    A TIW1  JMP   RXR
03117
03118
```

```
03120                   ************************************
03121                   *     MEMORY ADDRESS FOR PARTICULAR *
03122                   *     INSTRUCTION TO HAULT ON       *
03123                   ************************************
03124A 1740                     ORG   RINVT
03125A 1740  8F90    A          FDB   $8F90
03126A 1742  9294    A          FDB   $9294
03127A 1744  969A    A          FDB   $969A
03128A 1746  9E9F    A          FDB   $9E9F
03129
03130                   ********************************
03131                   *     JMP TABLE FOR RS232 INS  *
03132                   ********************************
03133A 1BA0                     ORG   RXIN
03134A 1BA0 CC 0C9D  A          JMP   IN00
03135A 1BA3 CC 0CB2  A          JMP   IN01
03136A 1BA6 CC 0CB2  A          JMP   IN01
03137A 1BA9 CC 0CB2  A          JMP   IN01
03138A 1BAC CC 0CB2  A          JMP   IN01
03139A 1BAF CC 0CB2  A          JMP   IN01
03140A 1BB2 CC 0CB2  A          JMP   IN01
03141A 1BB5 CC 0CB2  A          JMP   IN01
03142A 1BB8 CC 0CE3  A          JMP   IN08
03143A 1BBB CC 0CF9  A          JMP   IN09
03144A 1BBE CC 0D0A  A          JMP   IN0A
03145A 1BC1 CC 0D1A  A          JMP   IN0B
03146A 1BC4 CC 0D23  A          JMP   IN0C
03147A 1BC7 CC 0D7A  A          JMP   IN0D
03148A 1BCA CC 0D97  A          JMP   IN0E
03149
03150                   ********************************
03151                   *     PREVIOUS VAR ADD TAB FOR *
03152                   *     MOD3 IN DISPLAY          *
03153                   ********************************
03154A 1790                     ORG   MODV3+$10
03155A 1790  CE90    A          FDB   $CE90
03156A 1792  9192    A          FDB   $9192
03157A 1794  93      A          FCB   $93
03158
03159A 17CE                     ORG   MODV3+$4E
03160A 17CE  D3      A          FCB   $D3
03161
03162A 17D0                     ORG   MODV3+$50
03163A 17D0  93D0    A          FDB   $93D0
03164A 17D2  D1D2    A          FDB   $D1D2
03165A 17D4  D3      A          FCB   $D3
03166
03167                   ****************************
03168                   *     NEXT VAR ADD TAB FOR *
03169                   *     MOD3 IN DISPLAY      *
03170                   ****************************
03171A 17B0                     ORG   MODV3+$30
03172A 17B0  9192    A          FDB   $9192
03173A 17B2  93D0    A          FDB   $93D0
03174
03175A 17EE                     ORG   MODV3+$6E
03176A 17EE  90      A          FCB   $90
03177
03178A 17F0                     ORG   MODV3+$70
03179A 17F0  D1D2    A          FDB   $D1D2
03180A 17F2  D3CE    A          FDB   $D3CE
03181
03182                   **************************
03183                   *     DATA+VAR UPDATE FOR *
```

```
03184                   *       MD4;5;6             *
03185                   ****************************
03186A 1820                     ORG     U456
03187A 1820 CC 1084     A       JMP     UD4
03188A 1823 CC 11E7     A       JMP     UD5
03189A 1826 CC 124F     A       JMP     UD6
03190
03191                   ********************************
03192                   *       NEXT VAR TAB FOR MD4;5;6 *
03193                   ********************************
03194A 1730                     ORG     MODV4+$30
03195A 1730   9192      A       FDB     $9192
03196A 1732   9394      A       FDB     $9394
03197A 1734   9596      A       FDB     $9596
03198A 1736   9798      A       FDB     $9798
03199A 1738   99D0      A       FDB     $99D0
03200A 173A   D0D0      A       FDB     $D0D0
03201
03202A 1770                     ORG     MODV4+$70
03203A 1770   D1D2      A       FDB     $D1D2
03204A 1772   D3D4      A       FDB     $D3D4
03205A 1774   D5D6      A       FDB     $D5D6
03206A 1776   D7D8      A       FDB     $D7D8
03207A 1778   D990      A       FDB     $D990
03208A 177A   9090      A       FDB     $9090
03209
03210
03211                   ************************************
03212                   *       DISPLAY FORMATS FOR DIFFERENT *
03213                   *       MODES FOR LOAD RECORDER       *
03214                   ************************************
03215
03216                   **********************************
03217                   *       DISPLAY FORMATS FOR NAMES *
03218                   **********************************
03219A 1E00                     ORG     NMOD
03220A 1E00   20        A       FCC     %   (not used)   %
03221A 1E10   20        A       FCC     % (date & time)  %
03222A 1E20   20        A       FCC     %(start of test) %
03223A 1E30   20        A       FCC     %(pulse factors) %
03224A 1E40   20        A       FCC     %(present demand)%
03225A 1E50   20        A       FCC     % (max. demand)  %
03226A 1E60   20        A       FCC     % (tot. demand)  %
03227A 1E70   20        A       FCC     %  (pulse leg.)  %
03228A 1E80   20        A       FCC     %   (not used)   %
03229A 1E90   20        A       FCC     %   (not used)   %
03230
03231                   ******************************
03232                   *       DISP FORMMAT FOR MODE2 *
03233                   ******************************
03234A 1C00                     ORG     MODF1
03235A 1C00   44        A       FCC     &DATE     / /  &
03236A 1C10   50        A       FCC     &PERIOD No.    &
03237
03238A 1C40                     ORG     MODF1+$40
03239A 1C40   54        A       FCC     &TIME     :  :  &
03240A 1C50   50        A       FCC     &PERIOD DUR.   &
03241
03242A 1C20                     ORG     MODF2
03243A 1C20   54        A       FCC     &Test started on &
03244A 1C30   54        A       FCC     &TIME     :  :  &
03245
03246A 1C60                     ORG     MODF2+$40
03247A 1C60   44        A       FCC     &DEMAND PERIOD   &
```

```
03248A 1C78  44      A      FCC   &DATE     / /  &
03249
03250A 1C80                 ORG   MODF4
03251A 1C80  58      A      FCC   &Present period  &
03252A 1C90  20      A      FCC   &           KWh &
03253
03254A 1CC0                 ORG   MODF4+$40
03255A 1CC0  64      A      FCC   &demand:        &
03256A 1CD0  20      A      FCC   &           KVAh&
03257
03258A 1CA0                 ORG   MODF3
03259A 1CA0  44      A      FCC   &Demand pulse   &
03260A 1CB0  20      A      FCC   &    KWh/pulse &
03261
03262A 1CE0                 ORG   MODF3+$40
03263A 1CE0  66      A      FCC   &factors:  (/1E )&
03264A 1CF0  20      A      FCC   &    KVAh/pulse &
03265
03266A 1D00                 ORG   MODF5
03267A 1D00  4D      A      FCC   &Maximum demand  &
03268A 1D10  20      A      FCC   & /     h      &
03269
03270A 1D40                 ORG   MODF5+$40
03271A 1D40  72      A      FCC   &recorded:      &
03272A 1D50  20      A      FCC   &           KVA &
03273
03274A 1D20                 ORG   MODF6
03275A 1D20  54      A      FCC   &Tot. accumulated&
03276A 1D30  20      A      FCC   &           KWh &
03277
03278A 1D60                 ORG   MODF6+$40
03279A 1D60  64      A      FCC   &demand:        &
03280A 1D70  20      A      FCC   &           KVAh&
03281
03282A 1D80                 ORG   MODFA
03283A 1D80  53      A      FCC   &Select display  &
03284A 1D90  20      A      FCC   &               &
03285
03286A 1DC0                 ORG   MODFA+$40
03287A 1DC0  6D      A      FCC   &mode.          &
03288A 1DD0  20      A      FCC   &               &
03289
03290A 1DA0                 ORG   MODF7
03291A 1DA0  70      A      FCC   &pulse logging  &
03292A 1DB0  32      A      FCC   &2 -STOP        &
03293
03294A 1DE0                 ORG   MODF7+$40
03295A 1DE0  31      A      FCC   &1 -CONTINUE    &
03296A 1DF0  33      A      FCC   &3 -START       &
03297
03298                  ************************
03299                  *     PREVIOUS LOC TABLE *
03300                  ************************
03301A 1B18                 ORG   MODV2+$18
03302                  *     NOT NEEDED NO SET MODE
03303
03304                  ********************
03305                  *     NEXT VAR LOC *
03306                  ********************
03307A 1B38                 ORG   MODV2+$38
03308A 1B38  999B    A      FDB   $999B
03309A 1B3A  9B9C    A      FDB   $9B9C
03310A 1B3C  9E9E    A      FDB   $9E9E
03311A 1B3E  9FD8    A      FDB   $9FD8
```

```
03312
03313A 187B                    ORG   MODV2+$7B
03314A 187B   D9DB     A       FDB   $D9DB
03315A 187A   DBDC     A       FDB   $DBDC
03316A 187C   DEDE     A       FDB   $DEDE
03317A 187E   DF98     A       FDB   $DF98
03318
03319
03320
03321                  ******************
03322                  *     JMP TABLE    *
03323                  ******************
03324A 1A40                    ORG   JMR
03325A 1A40 CC 0406    A       JMP   MAN
03326A 1A43 CC 0E28    A       JMP   MOD1     MODE 1
03327A 1A46 CC 0F38    A       JMP   MOD2     MODE 2
03328A 1A49 CC 12FF    A       JMP   MOD3     MODE 3
03329A 1A4C CC 117A    A       JMP   MOD4     MODE 4
03330A 1A4F CC 117A    A       JMP   MOD4     MODE 5
03331A 1A52 CC 117A    A       JMP   MOD4     MODE 6
03332A 1A55 CC 1381    A       JMP   MOD7     MODE 7
03333A 1A5E                    ORG   JMR+30
03334A 1A5E CC 165D    A       JMP   MODA
03335
03336                  ************************
03337                  *     KEYBOARD JMP TAB *
03338                  ************************
03339A 1840                    ORG   JMK
03340A 1840 CC 0896    A       JMP   KBN
03341A 1843 CC 08B2    A       JMP   KBA
03342A 1846 CC 08BB    A       JMP   KBB
03343A 1849 CC 08D2    A       JMP   KBC
03344A 184C CC 08FC    A       JMP   KBD
03345A 184F CC 08C4    A       JMP   KBE
03346A 1852 CC 1855    A       JMP   KBF
03347
03348A 1855 81         KBF     RTS
03349
03350                  ********************************
03351                  *     MOVE TO TOP LOC JMP TAB  *
03352                  ********************************
03353A 1900                    ORG   JAT
03354A 1900 CC 0406    A       JMP   MAN
03355A 1903 CC 0EEC    A       JMP   MT1
03356A 1906 CC 0EEC    A       JMP   MT1
03357A 1909 CC 1285    A       JMP   MT3
03358A 1915                    ORG   JAT+21
03359A 1915 CC 1382    A       JMP   MT7
03360A 191E                    ORG   JAT+30
03361A 191E CC 0FD2    A       JMP   MTA
03362
03363                  ***********************************
03364                  *     MOVE TO PREVIOUS LOC JMP TAB  *
03365                  ***********************************
03366A 1940                    ORG   JBT
03367A 1940 CC 0400    A       JMP   MAN
03368A 1943 CC 0F0B    A       JMP   MP1
03369A 1946 CC 0F0B    A       JMP   MP1
03370A 1949 CC 1293    A       JMP   MP3
03371A 1955                    ORG   JBT+21
03372A 1955 CC 1383    A       JMP   MP7
03373A 195E                    ORG   JBT+30
03374A 195E CC 0FF0    A       JMP   MPA
03375
```

```
03376                   ********************************
03377                   *     MOVE TO NEXT LOC JMP TAB *
03378                   ********************************
03379A 1980                     ORG   JNT
03380A 1980 CC 0400   A         JMP   MAN
03381A 1983 CC 0F1A   A         JMP   MN1
03382A 1986 CC 0F1A   A         JMP   MN1
03383A 1989 CC 12A2   A         JMP   MN3
03384A 1995                     ORG   JNT+21
03385A 1995 CC 1385   A         JMP   MN7
03386A 199E                     ORG   JNT+30
03387A 199E CC 0FF1   A         JMP   MNA
03388
03389                   ***********************
03390                   *     CHANGE JMP TAB *
03391                   ***********************
03392A 1880                     ORG   SMJT
03393A 1880 CC 0400   A         JMP   MAN
03394A 1883 CC 0F26   A         JMP   CM1
03395A 1886 CC 0F26   A         JMP   CM1
03396A 1889 CC 12BD   A         JMP   CM3
03397A 1895                     ORG   SMJT+21
03398A 1895 CC 1413   A         JMP   CM7
03399A 189E                     ORG   SMJT+30
03400A 189E CC 185E   A         JMP   CMA
03401
03402                   ****************************
03403                   *     ENABLE SET MODE TAB *
03404                   ****************************
03405A 18E0                     ORG   ESM
03406A 18E0    00       A       FCB   $00
03407A 18E1    EE       A       FCB   $EE      MODE1 ENABLE
03408A 18E2    00       A       FCB   $00      MODE2 DISABLE
03409A 18E3    EE       A       FCB   $EE      MODE3 ENABLE
03410A 18E4    00       A       FCB   $00      MODE4 DISABLE
03411A 18E5    00       A       FCB   $00      MODE5 DISABLE
03412A 18E6    00       A       FCB   $00      MODE6 DISABLE
03413A 18E7    EE       A       FCB   $EE      MODE7 ENABLE
03414
03415                   ***********************
03416                   *     KEYBOARD TABLE *
03417                   ***********************
03418A 1800                     ORG   KBTAB
03419A 1800    0E       A       FCB   $0E
03420A 1801    0A       A       FCB   $0A
03421A 1802    00       A       FCB   $00
03422A 1803    0B       A       FCB   $0B
03423A 1804    0C       A       FCB   $0C
03424A 1805    09       A       FCB   $09
03425A 1806    08       A       FCB   $08
03426A 1807    07       A       FCB   $07
03427A 1808    0D       A       FCB   $0D
03428A 1809    06       A       FCB   $06
03429A 180A    05       A       FCB   $05
03430A 180B    04       A       FCB   $04
03431A 180C    0F       A       FCB   $0F
03432A 180D    03       A       FCB   $03
03433A 180E    02       A       FCB   $02
03434A 180F    01       A       FCB   $01
03435A 1810    10       A       FCB   $10
03436A 1811    11       A       FCB   $11
03437A 1812    12       A       FCB   $12
03438A 1813    13       A       FCB   $13
03439
```

```
03440                   ******************************
03441                   *     FIRST VARIABLE TABLE *
03442                   ******************************
03443A 1A00                     ORG   FAV
03444A 1A00   00      A         FCB   $00
03445A 1A01   08      A         FCB   $08
03446A 1A02   18      A         FCB   $18
03447A 1A03   10      A         FCB   $10
03448A 1A04   10      A         FCB   $10
03449A 1A05   10      A         FCB   $10
03450A 1A06   10      A         FCB   $10
03451A 1A07   0F      A         FCB   $0F
03452A 1A0A                     ORG   FAV+$A
03453A 1A0A   4F      A         FCB   $4F
03454
03455                   ***********************
03456                   *     DISP VARIABLES *
03457                   ***********************
03458A 1A08                     ORG   MODV1+$8
03459A 1A08   DE08    A         FDB   $DE08
03460A 1A0A   8989    A         FDB   $8989
03461A 1A0C   8B8C    A         FDB   $8B8C
03462A 1A0E   8C8E    A         FDB   $8C8E
03463A 1A90   8FC8    A         FDB   $8FC8
03464
03465A 1A28                     ORG   MODV1+$28
03466A 1A28   8988    A         FDB   $8988
03467A 1A2A   8B8C    A         FDB   $8B8C
03468A 1A2C   8E8E    A         FDB   $8E8E
03469A 1A2E   8FC8    A         FDB   $8FC8
03470A 1A30   C8      A         FCB   $C8
03471
03472A 1A68                     ORG   MODV1+$68
03473A 1A68   C9C8    A         FDB   $C9C8
03474A 1A6A   CBCC    A         FDB   $CBCC
03475A 1A6C   CECE    A         FDB   $CECE
03476A 1A6E   CFDD    A         FDB   $CFDD
03477A 1A70   88      A         FCB   $88
03478A 1A7D                     ORG   MODV1+$7D
03479A 1A7D   DE88    A         FDB   $DE88
03480
03481A 1AC8                     ORG   MODV1+$48
03482A 1AC8   8FC8    A         FDB   $8FC8
03483A 1ACA   C9C9    A         FDB   $C9C9
03484A 1ACC   CBCC    A         FDB   $CBCC
03485A 1ACE   CCCE    A         FDB   $CCCE
03486A 1AD0   CF88    A         FDB   $CF88
03487A 1AD2                     ORG   MODV1+$5D
03488A 1ADD   CFDD    A         FDB   $CFDD
03489
03490                   ****************
03491                   *     VECTORES *
03492                   ****************
03493A 1FF6                     ORG   $1FF6
03494A 1FF6   167D    A         FDB   TIWS
03495A 1FF8   0B2B    A         FDB   RXR
03496A 1FFA   14C4    A         FDB   MINT
03497A 1FFC   0400    A         FDB   MAN
03498A 1FFE   0400    A         FDB   MAN
03499
03500                           END
TOTAL ERRORS 00000--00000
```

```
0091 ACTR   00089*00453 02945 02991 02992 02994
0044 ACTRS  00115*00456 02944
0A5E ADD1   01258*01265
0A5B ADDB   01256*01346
122B ADDD   02327 02473 02506*02537
1238 ADDD1  02510 02513*
06B4 APNS   00587*00595 01701
06D2 APNS1  00598 00601 00603*
06BF APNS2  00588 00593*
06D3 APNS3  00602 00604*
06D6 APNS4  00587 00605*
06D9 APNS5  00593 00606*
06A9 APPS   00468 00575*00579 00604 01672
06B3 APPS1  00576 00580*
0013 AREG   00043*00363 00370 00479 00483 00489 00561 00565 00641
            00650 00658 00672 00709 00716 00725 00736 00744 00748
            00824 00828 00841 00843 00868 00871 01122 01156 01157
            01175 01176 01256 01266 01298 01308 01315 01321 01827
            01836 01889 01896 02079 02066 02289 02296 02460 02462
            02596 02603
008A ARTC   00148*00239 00258 00318 00322 00421 00424 00427 00430
            00454 00536 00540 00543 01848 01874 01906 01921 01963
            01968 01973 01982 01987 01992 02829 02840 02844 02848
            02851 02854 02858 02981 02986
124B BDD1   02523 02527*
124C BDD2   02525 02528*
123B BDDD   02328 02480 02496 02518*02544 02553
001B BREG   00107*00883 00890 00898 00910 00914 00916 00923 00926
            00932 00941 00945 00948 01072 01080 01094 02511 02521
008B BRTC   00149*00241 00256 01850 01872
0016 BTCT   00075*01421 01425 01435 01453 01457 01471 01477
0C0A BUFU   01541*01627 01673
0C2A BUFU1  01553 01556*
0C69 BUFU10 01554 01561 01587*
0C74 BUFU11 01556 01563 01593*
0C81 BUFU12 01553 01565 01602*
0C8E BUFU13 01546 01589 01574 01577 01597 01601 01604 01607*
0C91 BUFU14 01549 01608*
0C94 BUFU15 01570 01575 01578 01598 01602 01605 01609*
0C97 BUFU16 01584 01591 01610*
0C9A BUFU17 01589 01595 01611*
0C2E BUFU2  01552 01558*
0C38 BUFU3  01555 01557 01559*
0C3A BUFU4  01560 01563*
0C3E BUFU5  01559 01565*
0C40 BUFU6  01562 01564 01566*
0027 CARR   00078*00378 00440 00486 00467 00498 01272 01281 01301
            01305 02898 02813 02827 02828 02832 02835 02964 02979
            02984 02988 02989 02990
0DCB CHT    01847*02054
0E10 CHT1   01854 01857 01860 01863 01866 01869 01881*
0E13 CHT2   01856 01859 01862 01865 01868 01882*
0E16 CHT3   01877 01883*
0F26 CM1    02049*03394 03395
0F2F CM11   02849 02054*
0F32 CM12   02850 02055*
0F35 CM13   02851 02056*
12BD CM3    02609*03396
12E8 CM31   02613 02619 02620 02630*
12F3 CM32   02615 02632 02636*
12F6 CM33   02627 02637*
12F9 CM34   02617 02625 02634 02638*
12FC CM35   02623 02639*
1413 CM7    02800*03398
```

```
142B CM71   02807 02811*
1438 CM72   02812 02817*
144F CM73   02829*02829
14B6 CM74   02810 02879*
1435 CM75   02802 02816*
14BE CM76   02826 02843 02847 02850 02853 02857 02861 02863 02883*
14C1 CM77   02880 02884*
149E CM79   02865*02868
14A6 CM7A   02870*02873
1466 CM7B   02840*02840
146F CM7C   02844*02844
1478 CM7D   02848*02848
147F CM7E   02851*02851
1486 CM7F   02854*02854
148F CM7G   02858*02858
1428 CM7H   02805 02810*02815
105E CMA    02244*03400
107E CMA1   02246 02259*
1081 CMA2   02256 02260*
1072 CMA3   02250 02253*
106D CMA4   02248 02251*
0573 CON1   00390 00395*
057D CON2   00391 00394 00400*
057F CON3   00379 00401*
05A0 CON4   00412 00417*
05C6 CON5   00413 00416 00434*
05C8 CON6   00378 00401 00435*
05A8 CON7   00421*00421
05AF CON8   00424*00424
05B6 CON9   00427*00427
05BD CONA   00430*00430
054F CONT   00339 00376*
008F CPI    00068*02937 02939 02940 02963
008E CR     00067*02876 03014 03076 03077 03078 03079 03083 03085
            03087 03088 03090 03091 03093 03094 03098 03099 03101
            03102 03104 03105
001C CREG   00108*00869 00870 01541 01581 01741
008C CRTC   00150*00319 02932
0089 CTBM   00967*01611
008C CTBM1  00969*00971
0093 CTBM2  00969 00973*
0043 CTRS   00114*02952 02956
009B CVA    00063*02875 03012 03046 03047 03048 03049 03053 03055
            03057 03058 03060 03061 03063 03064 03068 03069 03071
            03072 03074 03075
009A CW     00062*02874 03010 03016 03017 03018 03019 03023 03025
            03027 03028 03030 03031 03033 03034 03038 03039 03041
            03042 03044 03045
0A7A DA1    01273 01278*
0A80 DA2    01277 01281*
0A8D DA3    01282 01288*
0A94 DA4    01287 01292*
0A6F DAA    00373 01261 01272*
060A DAAC   00471 00479*03003
061B DAAC1  00486 00489*
061C DAAC2  00482 00489*
070A DADD   00575 00657*
071A DADD1  00660 00664 00667*
0723 DADD2  00666 00670 00672*
0094 DAR    00123*00273 00275 00288 00289 00301 00302 00313 00314
            00746 00747 02015 02016 02017 02019 02020
0087 DATE   00145*00250 00428 01861 01974 02859
0788 DBU1   00745*00747
0786 DBUS   00345 00744*00778 00780 00816 01641 01664 01883 01929
```

```
054C ICON3  00367 00373*
05F8 IDEMP  00457 00466*
05FF IDEMP1 00466 00469*
0687 IDEMP2 00473*
0C9D IN00   01619*03134
0CAC IN001  01622 01627*
0CAF IN002  01624 01628*
0CB2 IN01   01636*03135 03136 03137 03138 03139 03140 03141
0CBF IN011  01638 01642*
0CC7 IN012  01641 01646*01649
0CD2 IN013  01640 01652*
0CDC IN014  01651 01657*
0CDD IN015  01647 01656*
0CE0 IN016  01650 01653 01659*
0CE3 IN08   01665*03142
0CF0 IN081  01666 01672*01681 01683
0CF3 IN082  01667 01673*01684 01696
0CF6 IN083  01669 01674*01686 01698 01709
0CF9 IN09   01680*03143
0D0A IN0A   01694*03144
0D17 IN0A1  01695 01701*
0D1A IN0B   01707*03145
0D23 IN0C   01718*03146
0D2F IN0C1  01724*01727
0D77 IN0C10 01744 01759*
0D39 IN0C2  01729*01732
0D4D IN0C3  01739*01779
0D62 IN0C5  01724 01729 01750*01772
0D6B IN0C6  01734 01751 01755*
0D6E IN0C7  01738 01756*
0D71 IN0C8  01747 01757*
0D74 IN0C9  01753 01758*01778
0D7A IN0D   01766*03147
0D86 IN0D1  01772*01775
0D97 IN0E   01784*03148
0DA0 IN0E1  01786 01789*
15B9 IN1    03016*
15DF IN10   03016 03033*
15D8 IN11   03017 03030*
15FB IN12   03018 03024 03029 03032 03037 03045*
15F0 IN13   03030 03038 03041*
15CE IN14   03022 03025*
15FA IN2    03046*
1620 IN20   03046 03063*
1619 IN21   03047 03060*
1639 IN22   03048 03054 03059 03062 03067 03075*
1631 IN23   03060 03068 03071*
160F IN24   03052 03055*
163B IN3    03076*
1661 IN30   03076 03093*
165A IN31   03077 03090*
167A IN32   03078 03084 03089 03092 03097 03105*
1672 IN33   03090 03098 03101*
1650 IN34   03082 03085*
15AD INP1   03009 03011*
15B2 INP2   03011 03013*
15B7 INP3   03013 03015*
154D INP4   02941 02964*
1549 INP5   02938 02953 02954 02955 02957 02962*
15A6 INPT   02965 03088*
1900 JAT    00174*00996 03353 03358 03360
1948 JBT    00175*01005 03366 03371 03373
075E JJT    00344 00709*
076E JKT    00343 00725*
```

```
0779 JKT1   00728 00731*
0781 JKT2   00730 00736*
07CF JM1    00816*00823 00829
1840 JMK    00163*00737 03339
1A40 JMR    00173*00717 03324 03333
1980 JNT    00176*00986 01016 03379 03384 03386
08B2 KBA    00994*03341
08BA KBA1   00994 00997*
08BB KBB    01003*03342
08C3 KBB1   01003 01006*
08D2 KBC    01022*03343
08ED KBC1   01025 01034*
08EE KBC2   01022 01035*
08F9 KBC3   01032 01041*
08FC KBD    01046*03344
090B KBD1   01046 01053*
096C KBD2   01049 01054*
08C4 KBE    01012*03345
08D1 KBE1   01012 01017*
1855 KBF    03346 03348*
0896 KBN    00978*03340
08AB KBN1   00978 00987*
08AC KBN2   00984 00988*00995 01004 01013
08AF KBN3   00982 00989*
1800 KBTAB  00161*02921 03418
14F4 KINT   02894 02916*
0090 KKV    00083*00335 00348 00726 00979 02174 02727 02729 02734
            02735 02922
0097 MADH   00127*00219 00220 01138 01226 01362 01718 01733 01750
            01766 02114 02133 02506 02509 02609 02682 02836 02839
0098 MADL   00128*00218 00224 00225 01135 01224 01228 01380 01720
            01728 01768 02116 02130 02508 02611 02684 02838
0400 MAN    00192*03325 03354 03367 03380 03393 03497 03498
0412 MAN1   00202*00205
0516 MAN10  00330 00342*
0519 MAN11  00337 00343*
051C MAN12  00333 00344*
0503 MAN13  00331 00334*
0422 MAN14  00212*00216
0446 MAN15  00223 00233*
044B MAN16  00229 00238*
0525 MAN17  00338 00347*
0529 MAN18  00336 00349*
0537 MAN19  00351 00355*
041A MAN2   00207*00210
051F MAN21  00345*
050D MAN22  00326 00339*
0510 MAN23  00328 00340*
0513 MAN24  00329 00341*
0522 MAN25  00346*00352
04F4 MAN26  00326*00354
052E MAN27  00349 00351*
047E MAN3   00267*00268 00271 00275
0496 MAN4   00278*00289
04B6 MAN5   00294*00302
04CE MAN6   00306*00314
04F3 MAN7   00325*00355
0432 MAN8   00220*00227
04FC MAN9   00331*
098F MCX    01030 01061*02007 02033 02044 02185 02233 02568 02580
            02590 02741 02794
0915 MCX1   01061 01064*
0E4F MD11   01905 01920*
0E5F MD12   01906 01920 01921 01927*
```

```
0E60 MD13   01907 01928*
0E63 MD14   01916 01929*
0E66 MD15   01922 01930*
0E69 MD16   01908 01931*
0E6C MD17   01910 01913 01925 01932*
0E31 MD18   01904 01906*
0E5D MD19   01903 01919 01926*
0F3E MD20   02062 02064*
0F47 MD21   02065 02068*
0F50 MD22   02063 02067 02072*
0F51 MD23   02069 02073*
130A MD31   02645 02646 02648*
1317 MD32   02644 02647 02654*
1318 MD33   02651 02655*
1189 MD41   02404 02405 02410*
119E MD42   02407 02409 02420*
119F MD43   02417 02421*
11A2 MD44   02408 02422*
13BC MD71   02748 02749 02751*
13C9 MD72   02747 02750 02757*
13CA MD73   02754 02758*
0046 MDD    00117*00429 02339 02860
002B MDVA   00072*00411 00415 00418 00420
0023 MDW    00071*00389 00393 00396 00398
002F MEL    00082*01155 01162 01174 01241 02433 02440 02447
0038 MEM    00124*00213 01829 01830 01833 01834 01841
0DAB MEMR   01812*
0DA3 MEMS   00212 01807*
0046 MHH    00119*00423 02345 02842
0082 MIN    00140*00253 00425 00544 01867 01988 02845
14CD MIN1   02892 02893 02895*
14DC MIN2   02896 02901*
14EF MIN3   02897 02898 02900 02901 02910*
14F1 MIN4   02899 02912*
0083 MINA   00141*00244
14C4 MINT   02891*03496
0049 MMIN   00120*00426 02348 02846
0047 MMO    00118*00432 02342 02856
0099 MMRD   00129*00261 00698 01048 02254
0F1A MN1    02839*03381 03382
0F23 MN11   02842 02844*
12A2 MN3    02585*03383
12AB MN31   02588 02590*
1385 MN7    02722*02739 03385
1397 MN71   02728 02731*
139F MN72   02730 02735*
13AA MN73   02732 02740*
13AB MN74   02726 02741*
13AE MN75   02738 02742*
0FF1 MNA    02163 02174*02252 03387
1005 MNA1   02183 02185*
1008 MNA2   02178 02186*
0FF8 MNA3   02176 02179*
0E28 MOD1   01903*03326
0F38 MOD2   02662*03327
12FF MOD3   02644*03328
117A MOD4   02404*03329 03330 03331
13B1 MOD7   02747*03332
105D MODA   02239*03334
1C00 MODF1  00164*01941 01952 03234 03238
1C20 MODF2  00165*02696 02184 03242 03246
1CA0 MODF3  00167*02664 02672 03258 03262
1CE0 MODF4  00166*02307 02316 03250 03254
1D00 MODF5  00168*02475 02483 02491 03266 03270
```

```
1D20 MODF6  00169*02539 02548 03274 03278
1DA0 MODF7  00170*02767 02775 03290 03294
1D80 MODFA  00171*02201 02209 03282 03286
1A80 MODV1  00154*01893 02004 02030 02041 03458 03465 03472 03478
            03481 03487
1B00 MODV2  00157*02083 03301 03307 03313
1780 MODV3  00158*02565 02577 02587 02600 03154 03159 03162 03171
            03175 03178
1780 MODV4  00159*02293 03194 03202
0088 MON    00146*00249 00431 01858 01969 02855
0F0B MP1    02027*03368 03369
0F17 MP11   02031 02033*
1293 MP3    02574*03370
129F MP31   02578 02580*
1383 MP7    02716*03372
0FF0 MPA    02169*03374
0093 MRD    00121*00260 00332 00699 00839 00978 00994 01003 01012
            01022 01023 01026 01035 01036 01038 01046 01047 01051
            01851 01852 01903 01959 01976 02218 02221 02251 02253
            02334 02428 02458 02644 02747
0DB3 MRS    00230 00622 00634 01141 01385 01627*02145
0DC9 MRS1   01838 01841*
0EEC MT1    02003*03355 03356
0EF4 MT11   02005 02007*
1285 MT3    02564*03357
128D MT31   02566 02568*
1290 MT32   02564 02569*
1382 MT7    02715*03359
0FD2 MTA    02151*03361
0FE0 MTA1   02156 02158*
0FE7 MTA2   02153 02162*
0FEA MTA3   02160 02163*
0FED MTA4   02158 02164*
0EF7 MTL    02003 02012*02569
0F06 MTL1   02016 02019*
0F08 MTL2   02018 02020*
0F54 MTN2   02079*02138
12AE MTN3   02596*02622 02631 02708
10A5 MTN4   01114 02289*02397
1384 MTN7   02717*
100B MTNA   02192*
0E19 MTNV   01882 01889*01931 02055
0AD7 MUL    01249 01358*
0AE5 MUL1   01367*01373
0ADD MUL2   01362*01365
108E NAM1   02272*02280
1097 NAM2   02276*
10A2 NAM3   02275 02282*
1084 NAME   02186 02222 02265*
1E80 NMOD   00172*02273 03319
073A ON1    00681 00686*
074F ON2    00689 00695*
075B ON3    00680 00683 00685 00687 00694 00701*
0736 ON4    00679 00684*
075D ON5    00678 00702*
0726 ONOF   00342 00678*
0045 OOCTR  00116*00700 02923 02970 02971 02974
11A5 ORGD   02406 02413 02428*
11B6 ORGD1  02430 02436*
11C5 ORGD2  02437 02443*
11D2 ORGD3  02435 02442 02449*
11D6 ORGD4  02431 02438 02444 02445 02451*
11D7 ORGD5  02449 02452*
0029 PDVA   00070*00402 00410 00414 00417 00419 00458 00451 00507
```

```
                00511 00525 00527 00530 00532 02872
0021 PDW        00069*00380 00388 00392 00395 00397 00448 00449 00500
                00504 00516 00518 00521 00523 02867
0042 PLOG       00113*00228 00231 02784 02804 02809 02814 02819
0000 PORTA      00130*00194 00349 00679 00687 01607 01809 01812 01814
                02893 03008 03009 03011 03013 03015
0001 PORTB      00131*00196 00691 00696 01423 01428 01438 01440 01455
                01470 01476 01784 02897 02901 02909 02918 02958 02959
                02976
0017 PRTY       00076*01431 01441 01462 01465 01470
07E4 PXM        00839*00988
0005 RETX       00041*01499
1552 RIN1       02936 02964 02966*
1570 RIN10      02969 02973 02979*
1503 RIN11      02931*02935
1557 RIN2       02966 02968*
159C RIN3       02931 02968 02990 02996 02997 02999*
1588 RIN5       02983 02990*
15A3 RIN6       02993 03003*
1537 RIN7       02950 02954*
1574 RIN8       02981*02981
157F RIN9       02986*02986
1523 RINA       02945*02945
152E RINB       02943 02947 02950*
1506 RINT       02895 02932*
1740 RINVT      00180*01639 01644 03124
0980 RMEM       01135*01250 02764
098B RMEM1      01139 01141*
06DC RMMR       00577 00606 00615*00861 02513
06E9 RMMR1      00619 00622*
0ABB RO1        01331*01334
0AB7 ROL        01329*01371
0A98 ROL1       01300*01307
0A95 ROLB       01298*01332
0AA9 ROR        01315*01370
0AAD ROR1       01317*01320
009E RSH        00109*01545 01568 01573 01576 01596 01600 01603 01619
                01665 01680 01682 01694
001E RSH1       00105*00873 00930 00952 00955 01542 01583 01587 01593
                01642 01646 01721 01737 01743 01752 01769 01777
009F RSL        00110*01621
001F RSL1       00106*00899 01544 01551 01567 01572 01580 01585 01637
                01645 01654 01723 01725 01730 01736 01740 01745 01771
                01773
0BB2 RTCR       00346 01493*
0BCE RTCR1      01497 01504*
0BDD RTCR2      01496 01511*
0BFB RTCR3      01498 01512 01526*
0C03 RTCR4      01493 01494 01495 01503 01504 01510 01511 01519 01525
                01530*
0C04 RTCR5      01524 01531*
0C07 RTCR6      01508 01506 01517 01527 01532*
0BF1 RTCR7      01515 01526*
0B31 RX1        01422 01425*
0B3D RX2        01428 01431*
0B40 RX3        01430 01433*
0B44 RX4        01427 01435*
0B4F RX5        01437 01440*
0B62 RX6        01424 01438 01440 01442 01449*
0B64 RX7        01448 01450*
1BA0 RXIN       00179*01531 03133
000E RXINL      00042*01513
001A RXMD       00104*01447 01514 01520 01522 01636
0B20 RXR        01417*03116 03495
```

```
0AFF RXSR   01391*02912
0B0C RXSR1  01391 01397*
0018 RXTX   00077*01405 01433 01444 01460
009C SAH    00064*00514 00539 00597 02820 02824
009D SAL    00065*00600 02823
0080 SEC    00138*00254 00455 01870 01993 02849 02946
0081 SECA   00139*00245
1880 SMJT   00177*01048 03392 03397 03399
0092 SPR    00090*01182 01776 01909 02052 02995
05C9 SPRES  00341 00440*
05D6 SPRES3 00441 00445*
05E8 SPRES4 00454*00454
05F1 SPRES5 00440 00442 00458*
05F2 SPRES6 00446 00459*
05F5 SPRES7 00460*
0631 SS1    00501 00504*
066D SS10   00529 00534*
0671 SS11   00497 00536*
0678 SS12   00536 00539*
0690 SS13   00496 00538 00550*
0691 SS14   00515 00522 00524 00531 00533 00535 00542 00545 00547
            00551*
067A SS15   00540*00540
0681 SS16   00543*00543
0637 SS2    00503 00505 00507*
063F SS3    00508 00511*
0645 SS4    00510 00512 00514*
0653 SS5    00517 00521*
0657 SS6    00519 00523*
065B SS7    00520 00525*
0665 SS8    00526 00530*
0669 SS9    00528 00532*
061F SSS    00340 00459 00496*
0956 ST1    01095 01099 01104*
0940 ST10   01079 01093*
096E ST11   01073 01077 01115*
0941 ST12   01092 01094*
0958 ST2    01084 01103 01105*
095C ST3    01097 01107*
0964 ST4    01106 01111*
0965 ST5    01089 01102 01108 01110 01112*
0968 ST6    01083 01104 01113*
0968 ST7    01090 01091 01101 01109 01114*
0930 ST8    01075 01085*
093C ST9    01085 01091*
07D2 STACH  00799 00807 00823*00989 01112 01997 02143 02164 02232
            02282 02325 02396 02500 02528 02557 02707 02742 02793
0803 STAM   00868*00889 00892 00933 01608 01756
0812 STAM1  00874 00876*
07B3 STCD   00792*01113 01932 01998 02144 02399 02705
07CC STCD1  00804 00807*
0918 STCL   01072*02398
00B0 STMEM  00066*00590 00662 01620 02821
083D STN1   00904 00908 00910*
0846 STN11  00902 00906 00914*
084A STN12  00901 00911 00916*
0859 STN2   00925 00929*00960
0863 STN21  00931 00935*00953 00956
0873 STN3   00943 00948*
0885 STN31  00950 00959*
0815 STNM   00883*00913 00927 00946
0826 STNM1  00898*01609 01758
084E STNM2  00917 00923*
0866 STNM3  00915 00941*
```

```
0E81 SUD1   01943 01945*
0E78 SUD2   01941*01947
0E8F SUD3   01952*01958
0E98 SUD4   01954 01956*
0E6F SUDT   01928 01937*
0E9D SVDT   01930 01959*
0EE3 SVDT1  01937 01948 01996*
0EE6 SVDT3  01944 01955 01997*
0EE9 SVDT4  01965 01978 01975 01984 01989 01994 01998*
0EA5 SVDT5  01963*01963
0EAE SVDT6  01968*01968
0EB7 SVDT7  01973*01973
0EC9 SVDT8  01982*01982
0ED2 SVDT9  01987*01987
0EDB SVDTA  01992*01992
0089 TCR    00134*00279 00284 00297 00309 01395 01408 01419 01480
            02905 03114
0088 TDR    00135*00281 00295 00307 00324 01396 01410 01473 02907
            02961
002E TDVA   00074*00405 02869
0026 TDW    00073*00383 02864
1685 TIW1   03112 03116*
167D TIWS   03112*03494
0B6F TX1    01455*01461 01466
0B73 TX2    01454 01457*
0B91 TX3    01459 01464*
0B89 TX4    01464 01468*
0B8D TX5    01463 01467 01470*
0B8F TX6    01423 01434 01439 01456 01471*
0B97 TX7    01469 01475*
0B9B TX8    01451 01452 01477*
0BB0 TX9    01474 01481 01487*
0B68 TXR    01420 01452*
0B0D TXSR   01403*01532 01628 01659 01674 01757 01789
0B1F TXSR1  01403 01404 01411*
182B U456   00162*02464 03186
0F63 UD2    02064 02092*
0F75 UD21   02098 02100*
0F6C UD22   02096*02102
0F7C UD23   02104*02116
0F85 UD24   02106 02108*
131B UD3    02648 02660*
132D UD31   02666 02668*
1324 UD32   02664*02670
1334 UD33   02672*02678
133D UD34   02674 02676*
10B4 UD4    02302*03187
10C8 UD41   02309 02311*
10BF UD42   02307*02314
10D1 UD43   02316*02323
10DA UD44   02318 02320*
10E4 UD45   02310 02319 02325*
11DA UD456  02414 02458*
10E7 UD46   02302 02326*
10EA UD47   02305 02327*
10ED UD48   02312 02321 02328*
11E7 UD5    02470*03188
11FB UD51   02477 02479*
11F2 UD52   02475*02482
1212 UD53   02491*02498
121B UD54   02493 02495*
1225 UD55   02478 02486 02494 02500*
1228 UD56   02470 02501*
1202 UD57   02483*02489
```

```
120B UD58   02485 02487*
124F UD6    02534*03189
1263 UD61   02541 02543*
125A UD62   02539*02546
126C UD63   02548*02555
1275 UD64   02550 02552*
127F UD65   02542 02551 02557*
1282 UD66   02534 02558*
13CD UD7    02751 02763*
13DF UD71   02769 02771*
13D6 UD72   02767*02773
13E6 UD73   02775*02781
13EF UD74   02777 02779*
100C UDA    01054 02197*
101E UDA1   02203 02205*
1015 UDA2   02201*02207
1025 UDA3   02209*02215
102E UDA4   02211 02213*
097F UM1    01123 01129*
0971 UMD    01121*01158 01177
0F8A VD2    02111*
0FB0 VD21   02117 02126 02130*02136
0FBB VD22   02121 02122 02123 02124 02125 02136*
0FC6 VD23   02092 02142*
0FC9 VD24   02099 02107 02143*
0FCC VD25   02119 02128 02139 02144*
0FCF VD26   02134 02145*
1342 VD3    02679*
1368 VD31   02689 02690 02691 02698*
1373 VD32   02685 02692 02698 02704*
1376 VD33   02687 02701 02705*
1379 VD34   02660 02706*
137C VD35   02667 02675 02696 02707*
137F VD36   02695 02700 02708*
10F0 VD4    02324 02334*02422 02499 02556
1123 VD41   02359*02374
114A VD42   02378*02393
113A VD43   02366 02367 02370*
1161 VD44   02385 02386 02389*
116E VD45   02369 02388 02396*
1171 VD46   02370 02387 02397*
1174 VD47   02362 02381 02398*
1116 VD48   02337 02353*
1177 VD49   02340 02343 02346 02349 02399*
1144 VD4A   02352 02375*
13F4 VD7    02782*
140A VD71   02770 02778 02786 02793*
1409 VD72   02791 02794*
1418 VD73   02763 02795*
1033 VDA    02216*
1054 VDA1   02204 02212 02220 02232*
1057 VDA2   02227 02233*
105A VDA3   02197 02229 02234*
0AF3 WMEM   01380*02638
0AFC WMEM1  01383 01385*
06EC WMMR   00586 00629*
06F5 WMMR1  00631 00634*
0990 WV1    01147*01150
0A3A WV10   01225 01235*
0A50 WV11   01243 01246*
0A58 WV12   01229 01232 01235 01238 01250*
09B4 WV2    01166*01169
09C8 WV3    01177*01180
09A4 WV4    01156*01161
```

```
0A55 WV5    01210 01245 01247 01249*
0A28 WV6    01181 01226*
09D2 WV7    01164 01182*
09DA WV8    01183 01186*
09E2 WV9    01187 01190*
09EA WVA    01191 01194*
09F2 WVB    01195 01198*
09FA WVC    01199 01202*
0A02 WVD    01203 01206*
0A04 WVE    01185 01189 01193 01197 01201 01205 01207*
0A0E WVF    01212*01215
098E WVH    01146*02452
0002 XOFF   00040*01526
0003 XON    00039*01505
0014 XREG   00044*00221 00617 00630 00712 00714 00732 00734 00763
            00767 00792 00800 00983 00985 01121 01124 01137 01154
            01159 01173 01178 01331 01345 01358 01364 01367 01369
            01381 01835 01875 01879 02132 02269 02272 02276 02351
            02356 02361 02371 02380 02398
0096 XVD2   00126*00615 00618 00620 00629 00632 01126 01128 01152
            01171 01329 01335 01341 01349 02113 02118 02120 02137
            02140 02271 02274 02277 02279 02358 02359 02363 02364
            02372 02377 02378 02382 02383 02391 02614 02616 02621
            02630 02633 02681 02686 02688 02694 02699 02702
0089 YEAR   00147*00247 01855 01964 02852
```

A P P E N D I X G

H P 8 5
S O F T W A R E D E S C R I P T I O N

APPENDIX G

11. HP85 SOFTWARE DESCRIPTION.

11.1. Down loading program.

11.1.1.

Line numbers: 1Ø - 88Ø

Routine: Main routine.

Description: This routine is the main downloading routine. The functions of this routine are as follows:

i) Initializes the screen and inputs an option for analysis.

ii) Initializes the RS232.

iii) Creates the necessary data files.

iv) Downloads the data and stores it into the file.

v) Computes the total accumulated and maximum demand.

11.1.2.

Line numbers: 89Ø - 128Ø

Routine: Enters stack of data.

Description: This subroutine downloads a stack of data from the load recorder. The stack of data consists of seven numbers. The variable contained in I$ is used as an instruction to the load recorder. The I$ is initialized before executing this subroutine. This instruction is sent

before the seven values are down loaded. The instructions are as follows:

i)	Start	ØØ
ii)	Present information	Ø8
iii)	Previous information	Ø9
iv)	Next information	ØA
v)	RS232 off	ØB
vi)	Start date	ØC
vii)	Start time	ØD
viii)	Acknowledge	ØE

11.1.3. Line numbers: 129Ø - 153Ø

Routine: Timer routine.

Description: This timer subroutine is executed when timer 1 in the HP85 times out. Timer 1 is used as an error check to moniter RS232 communications. Timer 1 would time out if no response occurs from the load recorder.

11.1.4. Line numbers: 154Ø - 169Ø

Routine: Interrupt.

Description: This subroutine is the RS232 interrupt routine. It is executed when the RS232 part has received a string of data followed by a carriage return.

11.1.5. Line numbers: 1700 - 1820

Routine: Sum check.

Description: This subroutine computes the sum check for the data that has been received.

11.1.6. Line numbers: 1830 - 1880

Routine: Reset RS232.

Description: Resets the RS232 port.

11.1.7. Line numbers: 1890 - 1960

Routine: Maximum and total.

Description: This subroutine computes the maximum and total accumulated values.

11.1.8. Line numbers: 1970 - 2080

Routine: Initialize RS232.

Description: This subroutine initializes the RS232 port in the HP85 for the load recorder.

11.1.9. Line numbers: 2090 - 2230

Routine: Creates files.

Description: Inputs the file names and creates the necessary files on tape for analysis.

11.1.10. Line numbers: 2240

Description: This chain instruction loads and executes the DRAW program from the tape.

11.2. Draw programs

11.2.1.

Line numbers: 1Ø - 43Ø

Routine: Main draw routine.

Description: This routine is the main draw routine which is executed from the download program. The functions of this routine are as follows:

i) Opens the required data files.

ii) Initializes variables and Bplots.

iii) Inputs the data from the files and plots the graph for each day.

iv) Transfers control to the calculation routine.

11.2.2.

Line numbers: 44Ø - 56Ø

Routine: Bplot subroutine.

Description: This subroutine is called from the main DRAW routine. It creates the Bplots from the data in the program for the graph.

11.2.3.

Line numbers: 57Ø - 63Ø

Routine: Inputs stack from file.

Description: Inputs a stack of data from the data file. The stack consists of numbers.

11.2.4. Line numbers: 640 - 730

Routine: Opens files.

Description: This subroutine has the following functions:

i) Inputs data file names and headings of results.

ii) Opens the required data files.

11.2.5. Line numbers: 740 - 780

Routine: Error.

Description: This subroutine is executed on an occurrence of an error during program execution. The function of this routine is to display the error on the screen.

11.2.6. Line numbers: 790 - 1380

Routine: Draw first screen of graph.

Description: This subroutine creates and draws the first graph screen. The first screen consists of headings and labelled axes. The X axis represents the time and the Y axis represents the KW, KVA and power factor. The scale of the X axis is calculated from the maximum demand of the investigation.

11.2.7. Line numbers: 1390 - 1570

Routine: Draws axes for new day.

Description: This subroutine clears the screen and

draws the axes for a new day.

11.2.8. Line numbers: 1580 - 1830

Routine: Plots graph.

Description: This subroutine plots the graph for a demand period obtained from the data files.

11.2.9. Line numbers 1840 - 2020

Routine: Prints heading.

Description: This subroutine prints the first heading of the results. The information that is printed in this routine is as follows:

i) Heading of results.

ii) Start of tests.

iii) Demand period.

iv) Maximum demand for KW and KVA.

v) Total consumption for KWh and KVAh.

11.2.10. Line numbers: 2040 - 2170

Routine: Computes totals.

Description: This subroutine computes the total accumulated KWh and KVAh for one week. These values are used in the tariff calculations.

11.2.11. Line numbers: 2180 - 2990

Routine: Calculations.

Description: The results are computed for one week to calculate the monthly tariff expense of the consumer. This subroutine calculates the following:

i) Combined maximum demand for the week.

ii) Consumption for the week.

iii) Average power factor for the week.

iv) Load factor.

v) Cost of General rate for the month.

vi) Cost of large user rate for the month.

vii) Effect of improvement on power factor.

A P P E N D I X H

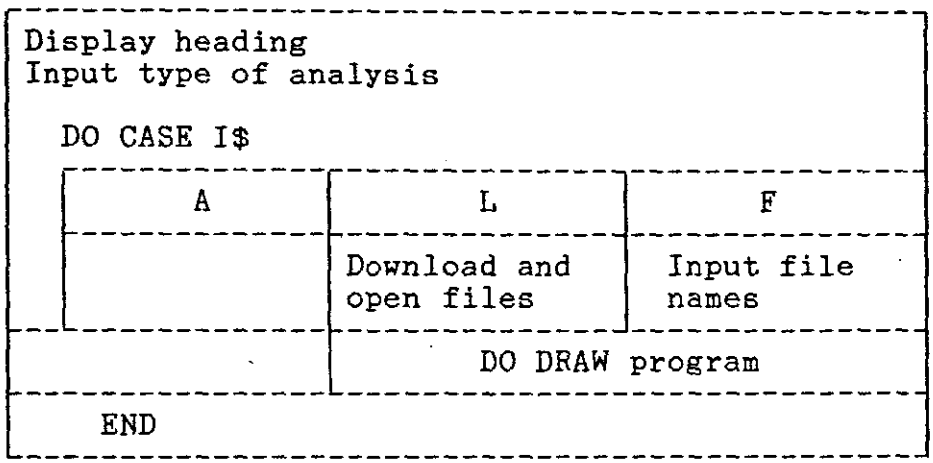
H P 8 5 F L O W C H A R T S

APPENDIX H

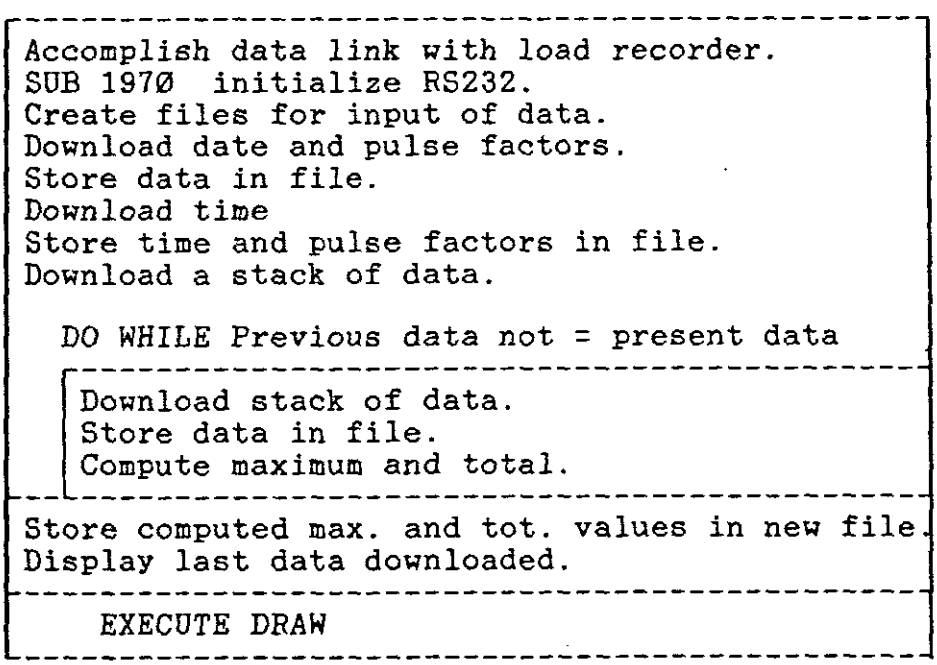
12. HP85 FLOWCHARTS.

12.1. Downloading flowcharts.

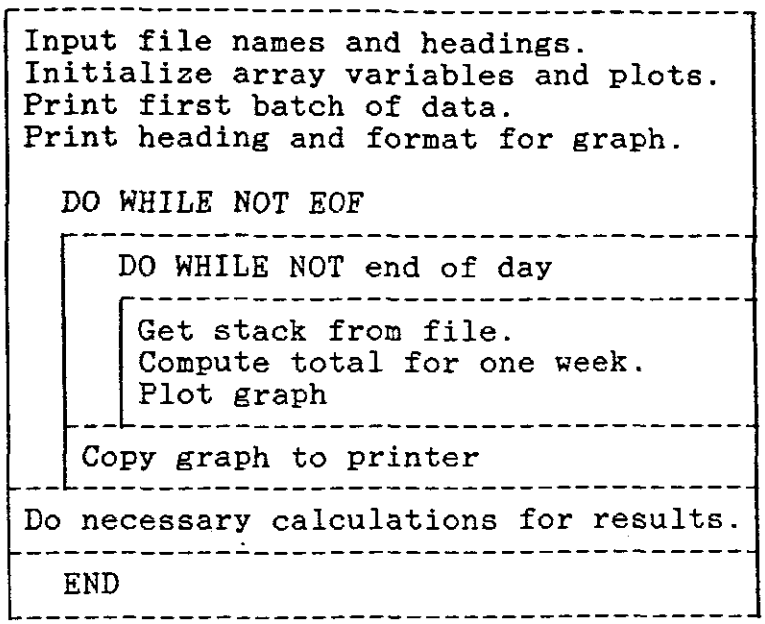
12.1.1. Main HP85 program.



12.1.2. Download and open files.



12.2. **Draw program.**



A P P E N D I X I

H P 8 5
S O F T W A R E L I S T I N G

APPENDIX I

13. HP85 SOFTWARE LISTING.

13.1. Download program (LRCD.PRG).

The following basic program listing is the main HP85 program used to download the data from the load recorder.

```
 10 CLEAR
 20 COM REAL N$[6],I$[1]
 30 DISP "    ******************
    ******"
 40 DISP "    * ANALYSIS PROGRAM
     FOR *"
 50 DISP "    *     LOAD RECORDE
    R    *"
 60 DISP "    ******************
    ******"
 70 DISP
 80 DIM D(7)
 90 W1=0 @ W2=0 @ V1=0 @ V2=0
100 DIM E(7)
110 FOR I=1 TO 7
120 E(I)=0
130 NEXT I
140 DISP "IS ANALYSIS REQUIRED F
    OR"
150 DISP "DATA STORED ON A FILE
    OR"
160 DISP "IN THE LOAD RECORDER"
170 DISP "(F;L OR A-ABORT)"
180 INPUT I$
190 IF I$="A" THEN 2250
200 IF I$="L" THEN 230
210 IF I$="F" THEN 880
220 GOTO 170
230 REM DOWN LOAD AND OPEN FILES
240 DISP "ATTACH LOAD RECORDER T
    O"
250 DISP "HP WITH RS232 CORD"
260 DISP "(PRESS 'CONT' WHEN REA
    DY)"
270 PAUSE
280 WAIT 200
290 GOSUB 1970 @ REM INITIALIZE
    RS232
300 ENABLE INTR 10;112
310 GOSUB 1830 @ REM RESET INTS
320 ON INTR 10 GOSUB 1540
330 ON TIMER# 1,2000 GOSUB 1290
340 C=1
350 CONTROL 10,9 ; 133@ REM TXON
    ,RESET RX Q
360 I$=CHR$(62)
370 OUTPUT 10 USING "#,K" ; I$
380 J=0
390 IF J=0 THEN 390
400 IF J=1 THEN 350
410 IF J<>2 THEN 380
420 OFF TIMER# 1
430 GOSUB 2090 @ REM CREATE FILE
    S
440 I$=CHR$(60)
450 GOSUB 890 @ REM GET STACK OF
     DATA
460 W=D(5)/10^D(1)
470 V=D(6)/10^D(1)
480 PRINT# 1 ; D(2),D(3),D(4)
```

```
490 I$=CHR$(61)
500 GOSUB 890 @ REM GET STACK
510 PRINT# 1 ; D(2),D(3),D(4)
520 PRINT# 1 ; W,V
530 I$="0"
540 GOSUB 890 @ REM GET STACK
550 GOSUB 1700 @ REM SUM CHECK
560 IF D(7)=S THEN 610
570 IF I$="0" THEN 540
580 IF I$=CHR$(58) THEN 590 ELSE
     600
590 I$="8"
600 GOTO 540
610 FOR L=1 TO 6
620 IF D(L)=E(L) THEN 740
630 FOR L=1 TO 6
640 E(L)=D(L)
650 NEXT L
660 BEEP 50,30
670 BEEP 200,30
680 DISP D(1);D(2);D(3);D(4)
690 IF D(1)=11 THEN 750
700 GOSUB 2190 @ REM STORE DATA
    IN FILE
710 GOSUB 1890 @ REM MAX + TOT
720 I$=CHR$(58)
730 GOTO 540
740 NEXT L
750 ASSIGN# 1 TO *
760 CREATE N$&"'",1,256
770 ASSIGN# 1 TO N$&"'"
780 PRINT# 1 ; W1,V1,W2,V2
790 ASSIGN# 1 TO *
800 DISP "LAST DATA DOWN LOADED
    WAS FOR;"
810 DISP "DAY No. ";D(1)
820 DISP "TIME ";D(2);"h";D(3)
830 DISP "DEMAND PERIOD No. ";D(
    4)
840 DISP
850 DISP "DETACH LOAD RECORDER"
860 DISP "(PRESS 'CONT' WHEN REA
    DY)"
870 GOTO 880
880 GOTO 2240
890 REM ENTER STACK OF DATA
900 STATUS 10,9 ; S9@ REM CHECK
    TXON/OFF
910 IF S9>=128 THEN 960
920 WAIT 500
930 STATUS 10,9 ; S9
940 IF S9>=128 THEN 960
950 CONTROL 10,9 ; 133
960 ON TIMER# 1,2000 GOSUB 1290
970 GOSUB 1830 @ REM RESET INTS
980 ENABLE INTR 10;112
990 OUTPUT 10 USING "#,K" ; I$
1000 J=0
1010 IF J=0 THEN 1010
1020 IF J=1 THEN 970
```

```
1030 IF J>2 THEN 1000
1040 OFF TIMER# 1
1050 FOR I=1 TO 7
1060 STATUS 10,9 ; S9@ REM CHECK
      TXON/OFF
1070 IF S9>=128 THEN 1120
1080 WAIT 500
1090 STATUS 10,9 ; S9
1100 IF S9>=128 THEN 1120
1110 CONTROL 10,9 ; 133
1120 ON TIMER# 1,2000 GOSUB 1290
1130 GOSUB 1830 @ REM RESET INTS
1140 ENABLE INTR 10;112
1150 OUTPUT 10 USING "#,K" ; I
1160 J=0
1170 IF J=0 THEN 1170
1180 IF J=1 THEN 1130
1190 IF J=2 THEN 1130
1200 IF J>3 THEN 1130
1210 IF I<7 THEN 1250
1220 D(I)=NUM(D$)
1230 D(I)=BINAND(D(I),15)
1240 GOTO 1260
1250 D(I)=VAL(D$)
1260 OFF TIMER# 1
1270 NEXT I
1280 RETURN
1290 REM TIMER#1 SUBROUTINE
1300 C=C+1
1310 BEEP 6,300
1320 IF C>5 THEN 1370
1330 STATUS 10,9 ; S9@ REM CHECK
      TXON/OFF
1340 IF S9>128 THEN 1520
1350 WAIT 500
1360 GOTO 1520
1370 OFF TIMER# 1
1380 WAIT 500
1390 DISP "NO REPLY FROM LOAD RE
     CORDER!"
1400 BEEP 5,2000
1410 DISP
1420 DISP "DETACH LOAD RECORDER"
1430 DISP "PRESS 'CONT' WHEN DON
     E"
1440 PAUSE
1450 DISP
1460 DISP "RECONNECT LOAD RECORD
     ER"
1470 DISP "AND PRESS 'CONT'"
1480 DISP
1490 PAUSE
1500 ON TIMER# 1,2000 GOSUB 1290
1510 C=1
1520 J=1
1530 RETURN
1540 REM INT ROUTINE
1550 STATUS 10,1 ; S1@ REM GET I
     NT CAUSE
1560 IF S1>16 THEN 1660
```

```
1570 S1=BINAND(S1,16)
1580 IF S1<16 THEN 1660
1590 ENTER 10 USING "%,%K" ; D$
1600 STATUS 10,11 ; S1@ REM GET
     END OF TEXT CAUSE
1610 S1=BINAND(S1,6)
1620 IF S1=2 THEN 1680
1630 IF S1<>4 THEN 1660
1640 J=3
1650 GOTO 1690
1660 J=1
1670 GOTO 1690
1680 J=2
1690 RETURN
1700 REM SUM CHECK
1710 S=0
1720 FOR T=1 TO 6
1730 D=IP(D(T)/1000)
1740 S=BINEOR(S,D)
1750 D=IP(FP(D(T)/1000)*10)
1760 S=BINEOR(S,D)
1770 D=IP(FP(D(T)/100)*10)
1780 S=BINEOR(S,D)
1790 D=FP(D(T)/10)*10
1800 S=BINEOR(S,D)
1810 NEXT T
1820 RETURN
1830 REM RESET INT STATUS ON RS2
     32
1840 STATUS 10,1 ; S1
1850 IF S1=0 THEN 1880
1860 CONTROL 10,9 ; 133@ REM RES
     ET RX Q
1870 GOTO 1840
1880 RETURN
1890 REM MAX & TOT
1900 IF W1>D(5) THEN 1920
1910 W1=D(5)
1920 IF V1>D(6) THEN 1940
1930 V1=D(6)
1940 W2=W2+D(5)
1950 V2=V2+D(6)
1960 RETURN
1970 REM INITIALIZE RS232 ROUTIN
     E
1980 RESET 10 @ REM RESET RS232
     INTERFACE
1990 CONTROL 10,16 ; 0@ REM NO E
     OL FROM HP
2000 CONTROL 10,3 ; 8@ REM 1200
     BAUD
2010 CONTROL 10,4 ; 30@ REM 7ASC
     II;2STOPS; EVEN PARITY
2020 CONTROL 10,12 ; 6@ REM EOLA
     -ACK
2030 CONTROL 10,13 ; 13@ REM EOL
     N-CR
2040 CONTROL 10,14 ; 2@ REM XOFF
     -STX
2050 CONTROL 10,15 ; 3@ REM XON-
     ETX
```

```
2060 CONTROL 10,11 ; 198@ REM TR
     ANSMIT ON/OFF CONTROL-ON;TE
     RMINATE ON EOLA+EOLN
2070 CONTROL 10,1 ; 112@ REM GEN
     ERATE INT ON FRAMING & PARI
     TY ERROR AND RECEIVED DATA
     AVAILABLE
2080 RETURN
2090 REM ROUTINE TO CREATE FILES
2100 DISP "APPROXIMATE DURATION
     OF"
2110 DISP "TEST (DAY)"
2120 INPUT N
2130 DISP "NEW NAME REQUIRED OF
     FILE TO"
2140 DISP "STORE DATA? (6 LETTER
     S MAX)"
2150 INPUT N$
2160 CREATE N$,N,4352
2170 ASSIGN# 1 TO N$
2180 RETURN
2190 REM STORE DATA IN FILE
2200 FOR I=1 TO 6
2210 PRINT# 1 ; D(I)
2220 NEXT I
2230 RETURN
2240 CHAIN "DRAW"
2250 END
```

13.2. Draw program (DRAW.PRG).

The following basic program listing is the HP85 program used to plot the graph, compute and calculate the results.

```
 10 COM REAL N$[6],I$[1]
 20 REM MAIN PROGRAM FOR DRAWING
 30 CLEAR
 40 GOSUB 640 @ REM ENTER NAMES
 50 GOTO 70
 60 GOSUB 710
 70 DIM W$[5]
 80 DIM V$[5]
 90 DIM P$[5]
100 J=0
110 GCLEAR
120 GOSUB 440 @ REM BPLOTS
130 ON ERROR GOSUB 740
140 IF J=1 THEN 2030
150 READ# 2 ; W1,V1,W2,V2
160 IF J=1 THEN 2030
170 READ# 1 ; S1,S2,S3
180 IF J=1 THEN 2030
190 READ# 1 ; S4,S5,S6
200 IF J=1 THEN 2030
210 READ# 1 ; W,V
220 IF J=1 THEN 2030
230 GOSUB 1840
240 GOSUB 790 @ REM PRINT 1ST SC
    REEN
250 Y=Y*5
260 D(7)=0
270 W9=0
280 W8=0
290 V9=0
300 V8=0
310 X8=0
320 GOSUB 570 @ REM GET STACK
330 IF J=1 THEN 400
340 GOSUB 2040
350 IF D(7)=D(1) THEN 370
360 GOSUB 1390 @ REM NEW DAY
370 D(7)=D(1)
380 GOSUB 1580 @ REM PLOT GRAPH
390 GOTO 320
400 COPY
410 GOTO 2030
420 ASSIGN# 1 TO *
430 ASSIGN# 2 TO *
440 REM SUBROUTINE TO CREATE BPL
    OTS
450 FOR I=1 TO 5
460 READ W3,V3,P3
470 W$[I,I]=CHR$(W3)
480 V$[I,I]=CHR$(V3)
490 P$[I,I]=CHR$(P3)
500 NEXT I
510 DATA 112,136,32
520 DATA 136,80,112
530 DATA 136,32,112
540 DATA 136,80,112
550 DATA 112,136,32
560 RETURN
570 REM GET STACK OF DATA FROM F
    ILE
```

```
580 J=0
590 FOR I=1 TO 6
600 READ# 1 ; D(I)
610 IF J=1 THEN 630
620 NEXT I
630 RETURN
640 REM ENTER NAMES & OPEN FILES
650 IF I$="L" THEN 680
660 DISP "ENTER NAME OF DATA FIL
    E"
670 INPUT N$
680 DISP
690 DISP "HEADING OF RESULTS"
700 INPUT M$
710 ASSIGN# 1 TO N$
720 ASSIGN# 2 TO N$&"'"
730 RETURN
740 REM ERROR ROUTINE
750 IF ERRN=72 THEN 770
760 DISP "ERROR";ERRN
770 J=1
780 RETURN
790 SCALE 0,240,-10,100
800 MOVE 5,95
810 LABEL "GRAPH SHOWING"
820 MOVE 5,88
830 LABEL "ELECTRICITY"
840 MOVE 5,81
850 LABEL "CONSUMPTION FOR"
860 MOVE 5,74
870 LABEL "THIS PERIOD"
880 MOVE 5,75
890 MOVE 5,70
900 MOVE 5,65
910 MOVE 5,60
920 MOVE 60,60
930 MOVE 5,50
940 MOVE 5,45
950 MOVE 5,35
960 MOVE 5,30
970 MOVE 5,25
980 MOVE 50,30
990 MOVE 50,25
1000 MOVE 5,15
1010 MOVE 5,10
1020 MOVE 5,5
1030 MOVE 50,10
1040 MOVE 50,5
1050 YAXIS 140,10
1060 LDIR 50
1070 MOVE 120,30
1080 LABEL "COS"&CHR$(15)
1090 MOVE 115,60
1100 BPLOT P$,1
1110 MOVE 155,30
1120 LABEL "KW"
1130 MOVE 150,60
1140 BPLOT W$,1
1150 MOVE 170,30
1160 LABEL "KVA"
```

```
1170 MOVE 165,60
1180 BPLOT V$,1
1190 LDIR 0
1200 MOVE 125,-2
1210 LABEL "0"
1220 MOVE 125,47
1230 LABEL ".5"
1240 MOVE 125,96
1250 LABEL "1"
1260 Y=(IP(V1/10)+1)*V*2
1270 FOR I=1 TO 5
1280 IF I<5 THEN 1310
1290 MOVE 185,20*I-5
1300 GOTO 1320
1310 MOVE 185,20*I-3
1320 LABEL VAL$(I*Y*60/S6)
1330 NEXT I
1340 MOVE 140,0
1350 DRAW 240,0
1360 MOVE 145,-8
1370 LABEL "TIME (hrs)"
1380 RETURN
1390 REM CLEAR AND FORM AXIS FOR
      NEW DAY
1400 COPY
1410 GCLEAR
1420 SCALE 0,240,-10,100
1430 XAXIS 0,10
1440 YAXIS 0,10,0,100
1450 MOVE 0,-8
1460 LABEL "0"
1470 FOR I=1 TO 11
1480 IF I<5 THEN 1510
1490 MOVE I*20-5,-8
1500 GOTO 1520
1510 MOVE I*20-2,-8
1520 LABEL VAL$(I*2)
1530 NEXT I
1540 MOVE 5,90
1550 LABEL "DAY "&VAL$(D(1))
1560 T1=0
1570 RETURN
1580 REM PLOT GRAPHS
1590 W5=D(5)*W/Y*100
1600 V5=D(6)*V/Y*100
1610 P5=D(5)*W/(D(6)*V)*100
1620 T2=D(2)*10+D(3)/60*10
1630 IF D(1)>1 THEN 1660
1640 IF D(4)>1 THEN 1660
1650 GOTO 1720
1660 MOVE T1,W4
1670 DRAW T2,W5
1680 MOVE T1,V4
1690 DRAW T2,V5
1700 MOVE T1,P4
1710 DRAW T2,P5
1720 MOVE T2,W5
1730 GOTO 1790
1740 BPLOT W$,1
1750 MOVE T2,V5
```

```
1760 BPLOT V$,1
1770 MOVE T2,P5
1780 BPLOT P$,1
1790 W4=W5
1800 V4=V5
1810 P4=P5
1820 T1=T2
1830 RETURN
1840 PRINT "ELECTRICITY TARIFF I
     NVESTIGATION"
1850 PRINT "RESULTS FOR ";M$
1860 PRINT
1870 PRINT "Start of test period
     :"
1880 PRINT S1;"/";S2;"/";S3;TAB(
     20);S4;"h";S5
1890 PRINT
1900 PRINT "Demand Period:";TAB(
     20);S6;"min"
1910 PRINT
1920 PRINT "Max. demand:"
1930 IMAGE 16X,6D.4D,4A,/
1940 PRINT USING 1930 ; W1*W*60/
     S6;"KW"
1950 PRINT USING 1930 ; V1*V*60/
     S6;"KVA"
1960 PRINT
1970 PRINT "Tot. consumption:"
1980 IMAGE 14X,8D.4D,4A,/
1990 PRINT USING 1980 ; W2*W;"KW
     h"
2000 PRINT USING 1980 ; V2*V;"KV
     Ah"
2010 PRINT USING "6/"
2020 RETURN
2030 GOTO 2180
2040 REM OBTAIN TOTAL FOR WEEK
2050 IF W8=1 THEN 2170
2060 IF V8>0 THEN 2120
2070 IF D(1)<2 THEN 2170
2080 V8=1
2090 W9=W9+D(5)
2100 V9=V9+D(6)
2110 GOTO 2170
2120 IF D(1)=2 THEN 2150
2130 X8=1
2140 GOTO 2090
2150 IF X8<>1 THEN 2090
2160 W8=1
2170 RETURN
2180 REM CALCULATIONS
2190 PRINT
2200 PRINT "**********************
     ********"
2210 PRINT USING "6/"
2220 PRINT "Combined maximum dem
     and:"
2230 PRINT USING 1930 ; V1*V*60/
     S6;"KVA"
2240 PRINT
```

```
2250 PRINT "Consumption for the
     week:"
2260 PRINT USING 1980 ; W9*W;"KW
     .h"
2270 PRINT
2280 PRINT "Power factor:"
2290 PRINT USING "16X,Z.DD" ; W9
     *W/(V9*V)
2300 PRINT
2310 PRINT "Load factor:"
2320 PRINT USING "16X,DD.D,A" ;
     V9*V*100/(V1*V*(60/S6)*7*24
     );"%"
2330 PRINT
2340 PRINT "MONTHLY EXPENSE FOR:
     "
2350 PRINT
2360 PRINT "**********************
     *********"
2370 PRINT
2380 PRINT "GENERAL RATE:"
2390 PRINT
2400 PRINT "Service charge:"
2410 PRINT USING 2470 ; "R";2.2
2420 PRINT
2430 PRINT "First 1500 units:"
2440 PRINT USING 2470 ; "R";219.
     9
2450 PRINT
2460 PRINT "Exeeding units:"
2470 IMAGE 20X,A,6D.DD
2480 PRINT USING 2470 ; "R";(W9*
     4*W-1500)*.112
2490 PRINT
2500 PRINT "TOTAL:"
2510 PRINT USING 2470 ; "R";2.2+
     219.9+(W9*4*W-1500)*.112
2520 PRINT "**********************
     *********"
2530 PRINT USING "3/"
2540 PRINT "L.V. LARGE USER RATE
     :"
2550 PRINT
2560 PRINT "Service charge:"
2570 PRINT USING 2470 ; "R";30.8
2580 PRINT
2590 PRINT "Demand charge:"
2600 D(1)=13.6*V1*V*60/S6
2610 PRINT USING 2470 ; "R";D(1)
2620 PRINT
2630 PRINT "Energy charge:"
2640 D(2)=W9*4*W*.0486
2650 PRINT USING 2470 ; "R";D(2)
2660 PRINT
2670 PRINT "TOTAL"
2680 PRINT USING 2470 ; "R";30.8
     +D(1)+D(2)
2690 PRINT "**********************
     *********"
2700 PRINT
```

```
2710 PRINT "POWER FACTOR IMPROVE
     MENT:"
2720 PRINT
2730 PRINT "Power factor at maxi
     mum demand:"
2740 D(3)=W1*W*60/S6/(V1*V*60/S6
     )
2750 PRINT USING "16X,Z.DD" ; D(
     3)
2760 IF IP(D(3)*10)<9 THEN 2790
2770 PRINT "Improvement of power
      factor is  not justified."
2780 GOTO 2980
2790 PRINT "With a power factor
     of 0.9"
2800 PRINT
2810 PRINT "The L.V. LARGE USER
     RATE would be:"
2820 PRINT
2830 PRINT "Service charge:"
2840 PRINT
2850 PRINT USING 2470 ; "R";30.8
2860 PRINT
2870 PRINT "Demand charge:"
2880 D(1)=13.6*(W1*W*60/S6)/.9
2890 PRINT USING 2470 ; "R";D(1)
2900 PRINT
2910 PRINT "Energy charge:"
2920 D(2)=W9*4*W*.0486
2930 PRINT USING 2470 ; "R";D(2)
2940 PRINT
2950 PRINT "Total:"
2960 PRINT USING 2470 ; "R";30.8
     +D(1)+D(2)
2970 PRINT
2980 PRINT "**********************
     **********"
2990 END
```

APPENDIX J

EMULATOR
HARDWARE DESCRIPTION

APPENDIX J

14. EMULATOR HARDWARE DESCRIPTION.

14.1. EXORCISER SECTION.

14.1.1. Programmable timer module.

The 684Ø PTM chip is used to generate timing signals between the two micros. It contains three 16 bit timers which can be programmed to various modes, viz, astable mode, one shot mode and pulse count mode. The PTM is decoded to H21ØØ location of the exorciser's USER memory map. The clock frequency of the timer is derived from the E (enable) pulse of the exorciser. Timer 1 is programmed in the astable mode, generating a frequency of 1ØØkHz. Timer 2 is programmed in the one shot retriggerable mode with a pulse duration of 5Ø counts (5Ø x 1 micro sec). Timer 3 is programmed to generate a 5 micro second pulse in the one shot mode, which is used to generate an interrupt to the 68Ø5 micro. This interrupt pulse enables the 68Ø5 micro from the stop mode after the memory block has been transfered to the 68Ø5. This interrupt is initiated under the exorciser program control.

The block diagram of the PTM is shown in figure 14.1. The one shot timer (T2) is triggered from either T1 or the load instruction (LI) signal of the 68Ø5 micro. The LI signal is triggered each time an instruction is fetched by the 68Ø5 micro during program execution. During the 68Ø5 target system program

execution, timer 2 is triggered by timer 1 (astable output). This prevents timer 2 from timing out during users STOP and WAIT instruction modes.

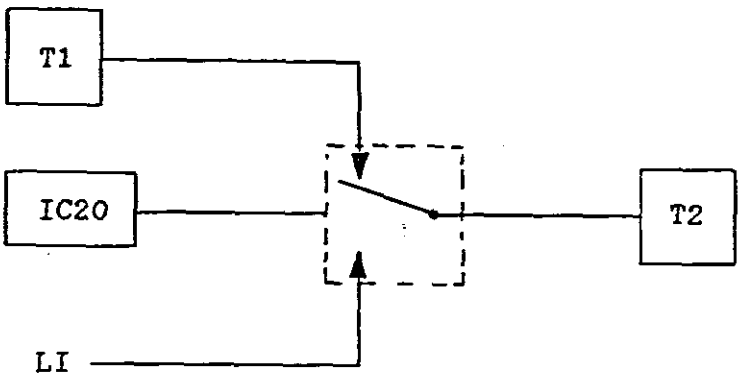


FIGURE 14.1 EMULATOR PTM BLOCK DIAGRAM.

The software interrupt (SWI) routine of the 68Ø5 micro changes the trigger source to the LI pin before executing the stand by STOP instruction. The LI trigger pulses will terminate once in the stop mode causing timer 2 to time out. The PTM then generates an interrupt to the exorciser.

14.1.2. Data bus buffer.

The inverted data signals on the mother board of the exorciser are buffered through U1 (74620) which is an octal bi-derictional inverting buffer. The address decode signal AND R/W is used to control the DIR control pins of U1 (pins 1 and 19). This logic is used to enable the emulator board from address decode and to control the direction with the R/W signal. This ensures no buss contention in the exorciser.

14.1.3. Emulator eprom.

The 68Ø9 subroutines and instructions that are used for controlling the operation of the emulator board are programmed in a 2k eprom contained on the emulator board. It is addressed to the executive memory map of the exorciser from HAØØØ to HA7FF. The valid executive address (VXA) signal is used in the address decoding for the eprom. This eprom simplifies the initialization procedure of the emulator board.

14.1.4. Output latches.

Quad D type latches (7475) are used for output control signals on the emulator board. These output latches are write only memory locations. The latches are as follows:

14.1.4.1. IC19.

This output latch is known as the memory select latch. The data stored on this latch enables the various blocks of the 8k memory with the aid of an address decode prom. This latch consists of 4 bits i.e. DØ to D3 which selects various address decoding pages (A4 to A7) in the EMUROM. The memory select latch is address decoded to H2Ø1Ø in the USER memory map of the exorciser. This hardware feature enables the 8k memory map to be controlled and selected according to the user application.

14.1.4.2. IC2Ø.

This quad latch is divided into 2 dual latches

which are selected at two different memory locations.

i) Latch 1.

This latch is enabled from address decoder IC1Ø which is selected at memory location H2Ø18 of the USER memory map. Input pins 2 and 3 are connected to the data signals DØ and D1. This latch drives two signals, namely, the ram LED and the RESET. The ram LED (pin 16) is used to indicate 68Ø5 program execution. The RESET (pin 15) is used to reset the 68Ø5 micro on the reset line for initialization. Both these signals are controlled by the exorciser.

ii) Latch 2.

This latch is addressed by IC12 from the 68Ø5 part of the emulator. This latch is known as the one shot latch. The input lines (pin 6 and 7) are driven by ADØ and AD1 of the 68Ø5 main data buss. This enables the latch to be serviced from both the 68Ø5 and 68Ø9 micros. It is address decoded to H1FF5. The output of this latch controls the source of the one shot trigger (T2 of PTM). The 1ØØkHz astable frequency (T1) is selected by the exorciser when the memory is transferred to the 68Ø5. The trigger source is changed to the LI pin of the 68Ø5 when the memory needs to be transferred back to the 68Ø9. The memory can

only be transferred to the 68Ø9 after the 68Ø5 micro has halted (i.e. entered the stop mode) for change over. The LI pin would stop generating pulses as soon as the 68Ø5 micro enters the stop mode causing T2 to time out. A 1ØØkHz trigger source is used during emulation to overcome the memory transfer when the user STOP or the WAIT instructions are used in the target system program.

14.1.4.3. IC21.

This latch is also divided into two dual latches. Both are selected by the exorciser and are driven by DØ and D1 of the exorciser data buss.

i) Latch 1.

This is addressed at the memory location H2ØØØ of the USER memory map. This latch is known as the memory buffer control latch. The output signal at pin 15 is used as an indicator of memory transfer. Pin 16 is used to enable two complementary buffers to allow the 8k ram to be transfered between the 68Ø5 and the 68Ø9. When pin 16 is a logic 1 the 68Ø5 buffer is enabled, connecting the 8k ram to the 68Ø5. When pin 16 is a logic Ø the 68Ø9 buffer is enabled, connecting the 8k ram to the 68Ø9.

ii) Latch 2.

This is addressed at memory location H2ØØ8 of the USER memory map. This latch is known as

the load instruction (LI) pin latch, which is used to change the state of the LI pin in the target system. This feature is mainly used as a select line if different pages of memory maps are used in the target system.

14.1.5. The 68Ø9 exorciser buffer.

This buffer (buffer 1) is the main buffer interfacing the 68Ø5 with the 68Ø9 buss signals. It consists of three octal buffers and the control logic.

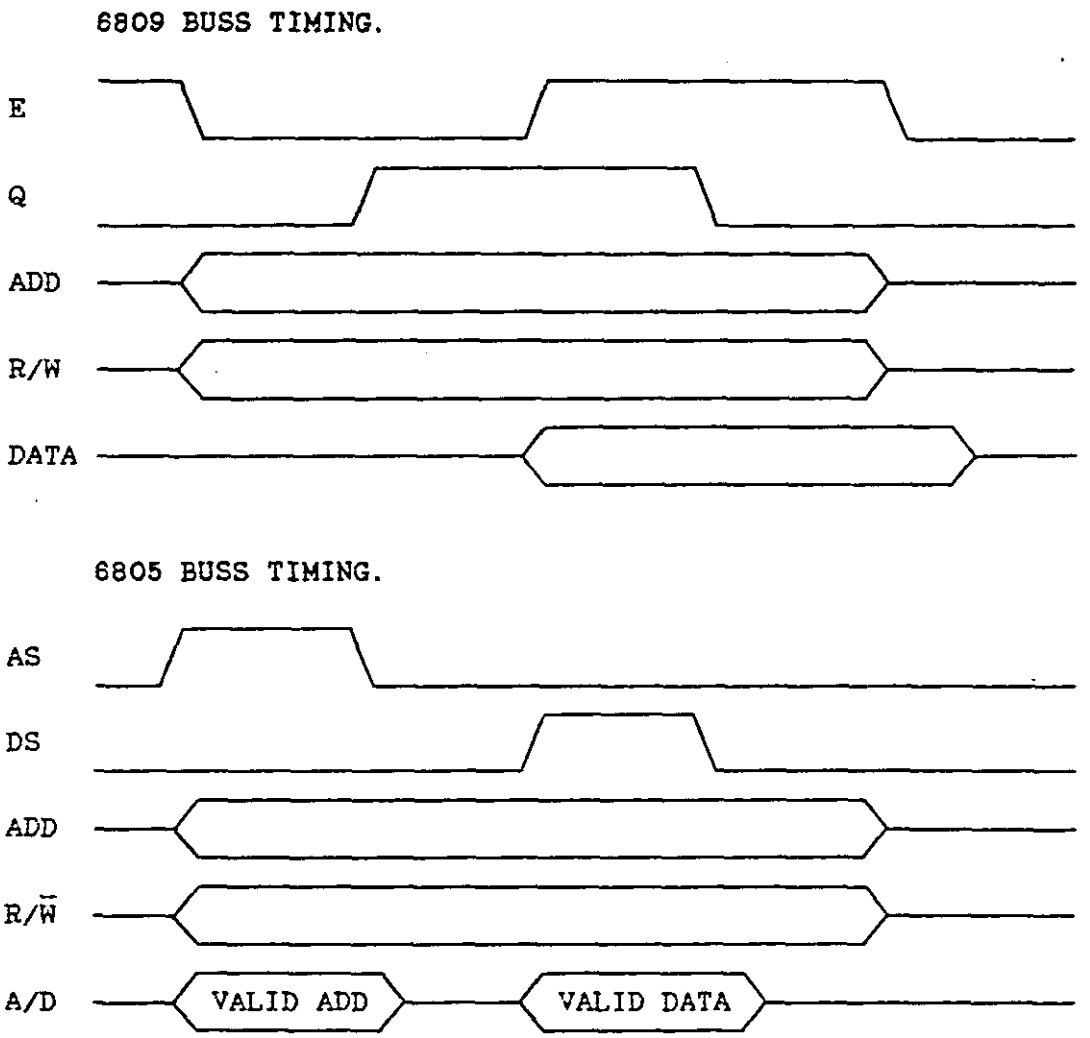


FIGURE 14.2 EMULATOR BUSS TIMING DIAGRAM.

The function of the buffer is as follows:

14.1.5.1. Control signal logic.

The buffer block contains the gate logic which converts the 68Ø9 control buss to the 68Ø5 control buss. From the buss timing diagram figure 14.2 on page 14-6 the following truth table was drawn up.

E	Q	AS	DS
Ø	Ø	1	Ø
Ø	1	Ø	Ø
1	Ø	Ø	Ø
1	1	Ø	1

TABLE 14.1 EMULATOR CONTROL LOGIC TRUTH TABLE.

From the above table the boolean expression for AS, DS and R/W is as follows:

$$AS = \overline{E} \cdot \overline{Q} = \overline{\overline{E} \cdot \overline{Q}}$$

$$DS = E \cdot Q = \overline{\overline{E} \cdot \overline{Q}}$$

$$R/\overline{W} = R/\overline{W}$$

These control signals and the high address buss (A8 to A12) are buffered through U17.

14.1.5.2. Address data multiplexing.

The buffers U15 (address AØ to A7) and U16 (data DØ to D7) is controlled to achieve an enabled multiplexed address/data buss for the 68Ø5 micro. The U16 is an octal bi-directional buffer with a

direction (DR) and enable (OE) control lines. The R/W signal of the 68Ø9 is used to control direction (DR) of the buffer. As seen on the bus timing in figure 14.2 the address (AØ to A7) is multiplexed to the 68Ø5 buss when the E signal is a logic Ø and the data (DØ to D7) is multiplexed to the buss when the E signal is a logic 1. The E signal and the buffer enable signal obtained from the address decode and buffer enable latch is used to achieve the multiplexing. The output enable (OE) control signals of U15 and U16 is as follows:

$$U15\ \overline{OE} = \overline{\overline{E} \cdot BE}$$

$$U16\ \overline{OE} = \overline{E \cdot \overline{BE}}$$

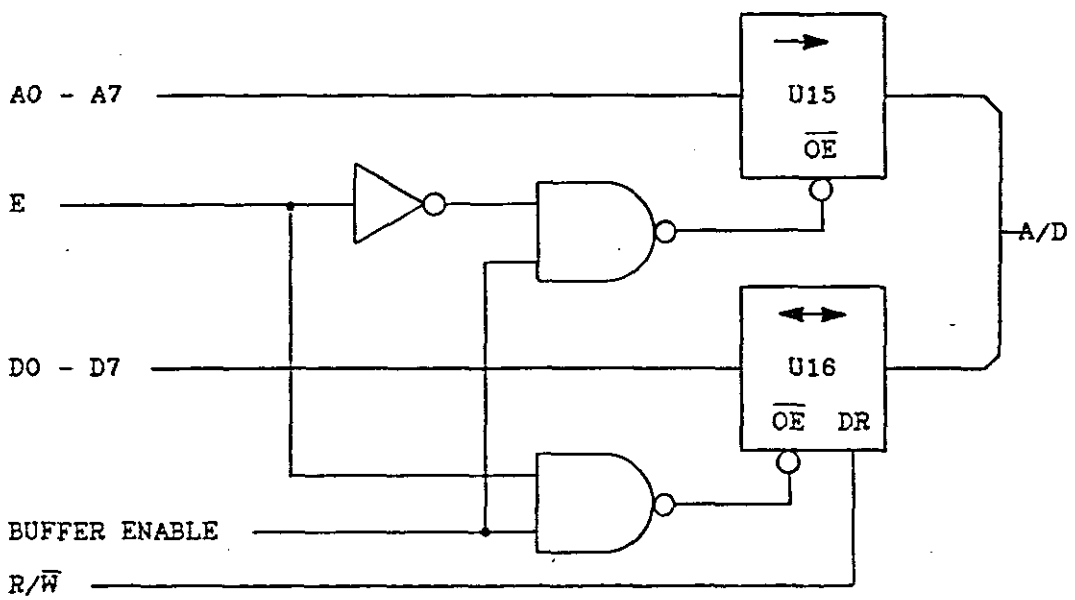


FIG 14.3 EMULATOR MULTIPLEXING BLOCK DIAGRAM.

14.1.5.3. Buffer enable.

The select signal, pin 16 of IC21 is connected to the entire buffer block to enable the buffer when the 68Ø5 map is transferred to the exorciser. A further select signal is obtained from the address decoder to address the 68Ø5 map from HØ to H1FFF in the USER memory map of the exorciser.

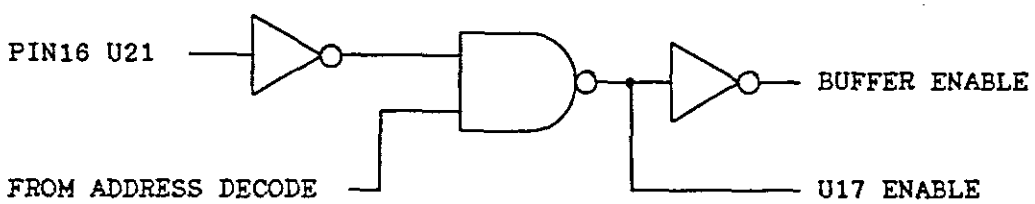


FIGURE 14.4 EMULATOR BUFFER ENABLE LOGIC.

To select the entire buffer from the exorciser pin 16 of U21 must be a logic Ø and the memory must be addressed to activate the address decoder.

14.2. 68Ø5 SECTION.

14.2.1. The 68Ø5 main buffer.

This buffer is complementary enabled with the 68Ø9 buffer to allow the 68Ø5 memory map to be shared between the two microprocessors. It consists of two main octal buffers, viz:

14.2.1.1. Address and control buffer (U25).

It is an octal unidirectional buffer (74LS244)

which is enabled when pin 16 of U21 is a logic 1.

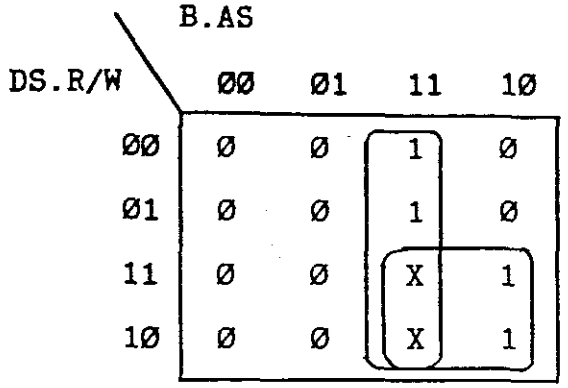
14.2.1.2. Address data buffer (U23).

Due to the multiplexed address/data buss a more complex control logic was required to control and select U23. Pin 16 of U21 (represented as B) is used to select the buffer. This signal together with AS, DS and R/W is used to calculate the logic required for the buffer signals DR and OE. The truth table below and Karnough maps were used to calculate these buffer control signals.

INPUT				OUTPUT	
B	AS	DS	R/W	OE	DR
Ø	Ø	Ø	Ø	Ø	X
Ø	Ø	Ø	1	Ø	X
Ø	Ø	1	Ø	Ø	X
Ø	Ø	1	1	Ø	X
Ø	1	Ø	Ø	Ø	X
Ø	1	Ø	1	Ø	X
Ø	1	1	Ø	Ø	X
Ø	1	1	1	Ø	X
1	Ø	Ø	Ø	Ø	X
1	Ø	Ø	1	Ø	X
1	Ø	1	Ø	1	Ø
1	Ø	1	1	1	1
1	1	Ø	Ø	1	Ø
1	1	Ø	1	1	Ø
1	1	1	Ø	X	X
1	1	1	1	X	X

TABLE 14.2 EMULATOR OE AND DR TRUTH TABLE.

i) The Karnough map for OE.



The Boolean expression for the output enable OE is as follows:

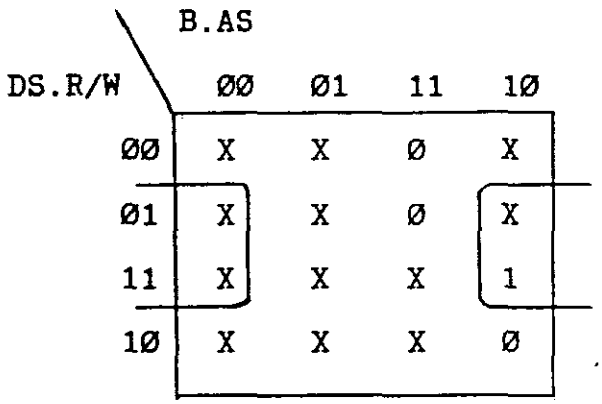
$$OE = B \cdot AS + B \cdot DS$$

$$OE = B \cdot (AS + DS)$$

$$\overline{OE} = \overline{B} + \overline{AS} \cdot \overline{DS}$$

$$\overline{OE} = B \cdot \overline{\overline{AS} \cdot \overline{DS}}$$

ii) The Karnough map for DR.



The Boolean expression for the direction control is as follows:

$$DR = \overline{AS} \cdot R/\overline{W}$$

$$DR = \overline{\overline{AS} \cdot R/\overline{W}}$$

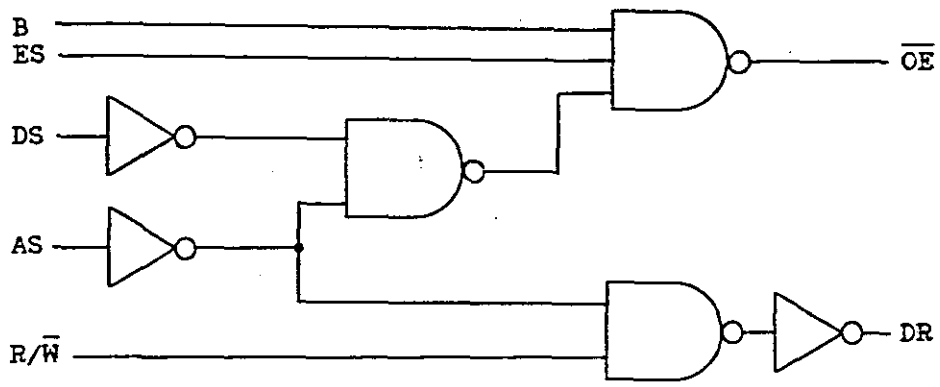


FIGURE 14.5 EMULATOR OE AND DR CIRCUIT LOGIC.

The additional select signal ES in the above figure is used to disable the buffer when the hardware ABORT is activated.

14.2.2. Software interrupt Abort logic.

The 68Ø5 user program is stoped at the required break points by inserting the Software Interrupt (SWI) instructions in the program. The 68Ø5 machine code instruction for SWI is H83. These break point instructions are inserted in the memory by the exorciser at the required addresses. The 68Ø5 program will only stop when it executes this SWI instruction. These break point (SWI) instructions are sometimes not executed due to errors in the user program. To

overcome this problem a Software Interrupt (SWI) Abort logic was designed to manually insert a H83 instruction on the data buss for the next load instruction. The user could stop the program at any time by pressing the ABORT switch on the emulator board. Basically this logic disables the data buss buffer (U22) and enables two tristate buffers which inserts a H83 on the data buss when the switch is activated during a load instruction. The timing diagram below illustrates the operation of this logic. The input signals to this logic is the data strobe (DS); the load instruction (LI) and the switch.

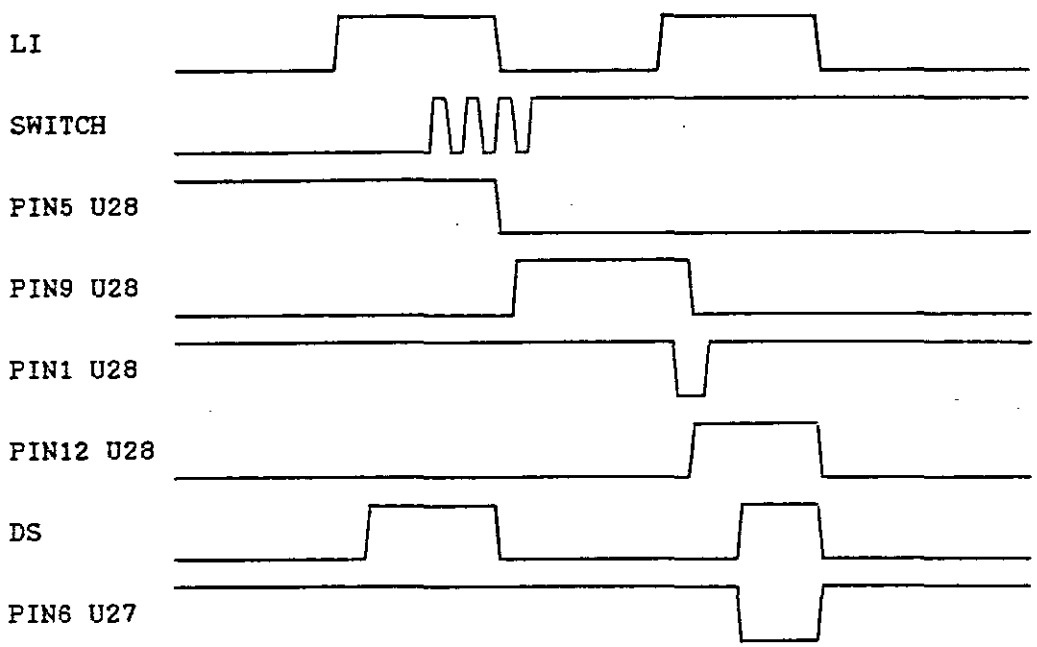


FIGURE 14.6 EMULATOR ABORT LOGIC TIMING DIAGRAM.

14.2.3. Ram Block.

This block consists of an 8k ram chip (6264) which is accessable by both micros depending on the active buffer. The chip select is controlled by the address decoding logic and the control line OE and WE is derived from the DS and R/W signals.

14.3. POD.

The pod consists of a buffer and a 40 way IC connector which is used for debugging. The pod replaces the 6805 micro on the target system. Connected to the 40 way connector are the following:

i) address/data buss

ii) address buss

iii) control buss

iv) port lines

This pod is used to emulate the 6805 micro of the target system. It enables the target system to be accessed by the exorciser.

The data buss is buffered through an octal bi-directional buffer (74LS245) on the POD to maintain data signal levels during a read operation.
The control signals for this buffer were calculated as follows:

$$OE = B \cdot (AS + DS)$$

but B does not exists for this buffer.

$$\therefore \quad \overline{OE} = \overline{AS + DS}$$

$$DR = \overline{AS} \cdot R/\overline{W}$$

$$\therefore \quad DR = \overline{AS + \overline{R}/W}$$

As indicated in the circuit diagram (Appendix L) the three NOR gates are used to implement the two control signals.

14.4. ADDRESS DECODING.

The memory map of the emulator board is indicated in figure 15.1 page 15-1. It consists of the Exorciser USER memory map with a common 68Ø5 map. The common 68Ø5 map is addressed from HØØØØ to H1FFF.

14.4.1. Exorciser.

The address decoding logic used for the exorciser memory map consists of a 256 by 4 bit ROM (MOTROM) and a three to eight line decoder (74138). The contents of the MOTROM is shown in figure 15.2 on page 15-2.

14.4.2. 68Ø5 map.

The 68Ø5 memory map consists of an 8k ram which is address decoded by two ROMS. The EMUROM is used, in combination with U19, to address the RAM in 2k blocks under different software control, as required. The STKROM is used to decode the bottom 156 bytes of the RAM for the 68Ø5 micro. This is to enable and disable the RAM for the internal I/O and timer registers of the 68Ø5 micro. The contents of the EMUROM and the STKROM is shown in figure 15.3 and figure 15.4 on pages 15-3 and 15-4 respectively.

The intergrated circuit (U12) is used to address decode latch 2 of IC2Ø from the 68Ø5 section of the

emulator board at memory address H1FF5. This enables access from the 68Ø5 micro as well as the exorciser.

A P P E N D I X K

E M U L A T O R
M E M O R Y M A P A N D
A D D R E S S D E C O D E R O M S

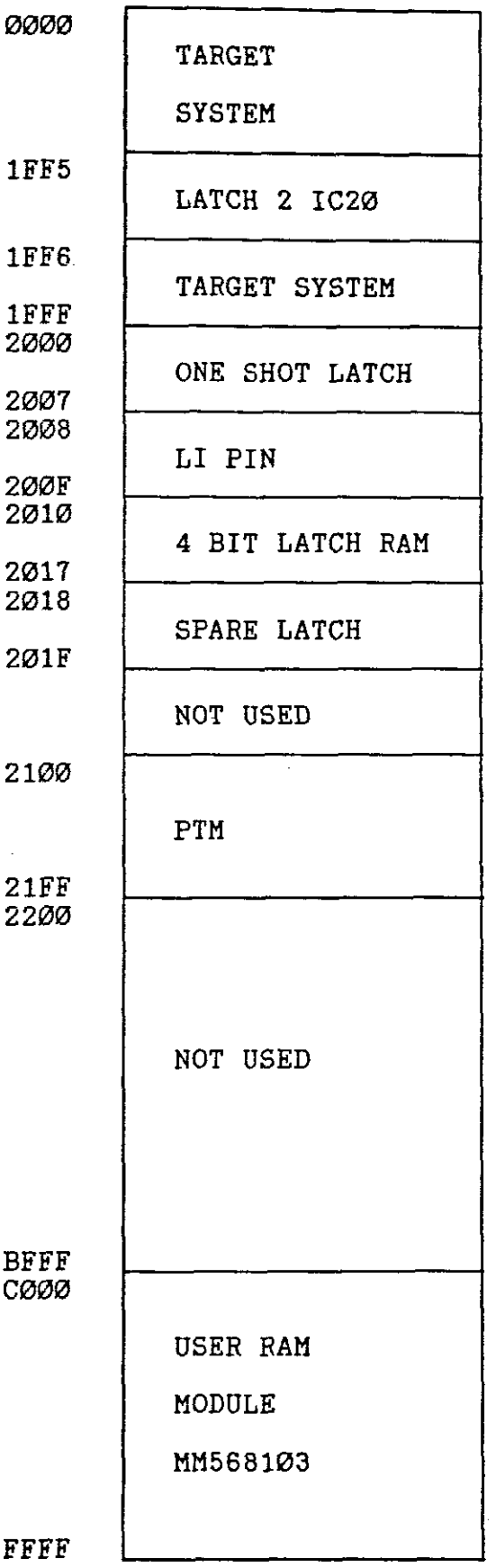


FIGURE 15.1 EMULATOR MEMORY MAP

ØØØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØ1Ø	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØ2Ø	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØ3Ø	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØ4Ø	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØ5Ø	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØ6Ø	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØ7Ø	Ø6	Ø6	Ø6	Ø6	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØ8Ø	ØA	ØA	ØA	ØA	ØA	ØA	ØA	ØA	ØA	ØA	ØA	ØA	ØA	ØA	ØA	ØA
ØØ9Ø	Ø3	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØAØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØBØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØCØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØDØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØEØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØFØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ

FIGURE 15.2 MOTOROLA ROM (MOTROM) CONTENTS.

```
ØØØØ  Ø3 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø9
ØØ1Ø  Ø3 ØØ ØØ ØØ Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø9
ØØ2Ø  Ø3 Ø1 Ø1 Ø1 ØØ ØØ ØØ ØØ Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø9
ØØ3Ø  Ø3 ØØ ØØ ØØ ØØ ØØ ØØ ØØ Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø9
ØØ4Ø  Ø3 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 ØØ ØØ ØØ ØØ Ø1 Ø1 Ø1 Ø9
ØØ5Ø  Ø3 ØØ ØØ ØØ Ø1 Ø1 Ø1 Ø1 ØØ ØØ ØØ ØØ Ø1 Ø1 Ø1 Ø9
ØØ6Ø  Ø3 Ø1 Ø1 Ø1 ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ Ø1 Ø1 Ø1 Ø9
ØØ7Ø  Ø3 ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ Ø1 Ø1 Ø1 Ø9
ØØ8Ø  Ø3 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 ØØ ØØ ØØ Ø8
ØØ9Ø  Ø3 ØØ ØØ ØØ Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 ØØ ØØ ØØ Ø8
ØØAØ  Ø3 Ø1 Ø1 Ø1 ØØ ØØ ØØ ØØ Ø1 Ø1 Ø1 Ø1 ØØ ØØ ØØ Ø8
ØØBØ  Ø3 ØØ ØØ ØØ ØØ ØØ ØØ ØØ Ø1 Ø1 Ø1 Ø1 ØØ ØØ ØØ Ø8
ØØCØ  Ø3 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 Ø1 ØØ ØØ ØØ ØØ ØØ ØØ ØØ Ø8
ØØDØ  Ø3 ØØ ØØ ØØ Ø1 Ø1 Ø1 Ø1 ØØ ØØ ØØ ØØ ØØ ØØ ØØ Ø8
ØØEØ  Ø3 Ø1 Ø1 Ø1 ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ Ø8
ØØFØ  Ø3 ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ Ø8
```

FIGURE 15.3 EMULATOR ROM (EMUROM) CONTENTS.

ØØØØ	ØØ	ØØ	Ø1	Ø1	ØØ	ØØ	Ø1	Ø1	ØØ	ØØ	Ø1	Ø1	ØØ	ØØ	ØØ	ØØ
ØØ1Ø	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØ2Ø	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØ3Ø	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØ4Ø	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØ5Ø	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØ6Ø	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØ7Ø	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ
ØØ8Ø	Ø1	Ø1	Ø3	Ø3	Ø1	Ø1	Ø3	Ø3	Ø1	Ø1	Ø3	Ø3	Ø3	Ø3	Ø3	Ø3
ØØ9Ø	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1
ØØAØ	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1
ØØBØ	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1
ØØCØ	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1
ØØDØ	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1
ØØEØ	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1
ØØFØ	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1	Ø1

FIGURE 15.4 STACK ROM (STKROM) CONTENTS.

A P P E N D I X L

E M U L A T O R
C I R C U I T D I A G R A M

CIRCUIT DIAGRAM OF MC146805E2 EMULATOR

CITY OF CAPE TOWN ELECTRICITY DEPARTMENT	DRAWN			SCALE	REV.
D/S MANAGER	COMPILED			FILE	
C.E.E. F.L.U. BARNES	CHECKED			DR2793	

APPENDIX M

EMULATOR
SOFTWARE DESCRIPTION

APPENDIX M

17. EMULATOR SOFTWARE DESCRIPTION.

This section contains a detail description of the emulator software subroutines.

17.1. 68Ø5 EMULATOR SOFTWARE.

These 68Ø5 operating system routines are loaded into the 68Ø5 target system memory map from H1EBØ to H1FFF.

17.1.1. SWIR H1F8Ø

Routine: Software interrupt routine

Routines called: DUPM

Description: This routine is executed when a SWI breakpoint is encountered in the target system program. SWIR is a software interrupt vector. The functions of this routine are:

i) It branches to copy the memory routine, to store a known program counter in the stack.

ii) It changes the one shot trigger to the LI pin.

iii) It transfers control over to the exorciser by executing a stop instruction.

A RTI instruction is used to return to the target system program once, the

control is transfered back to the 68Ø5 by generating an interrrupt.

17.1.2. DUPM H1F8E

Routine: Copy memory routine.

Description: The internal memory addresses (HØ to HF) are mirror copied to H1FØØ to H1FØF in the 8k ram. This enables the exorciser to excess the internal I/O through the mirror image. Executing this subroutine also causes a known program counter to be stored in the stack. The stack pointer can be calculated from this known program counter.

17.1.3. RINT H1F9A

Routine: Recovery interrupt routine.

Description: The vector address H1F9A (in memory locations H1FF1 and H1FF2) is changed with the target system interrupt vector (in vector address H1FFA/B) by the exorciser to enable this routine to be executed by the 68Ø5 micro after an interrupt. The functions of this routine are:

i) It restores the user interrupt vectors.

ii) It copies the modified interrupt states from the stack to the memory

image H1FØØ.

iii) It copies the entire image stack block H1FØF - H1F7F to HØF - H7F.

iv) It modifies the timer interrupt mask.

The RTI instruction causes a return to the SWIR routine which then causes a return to the target system program.

17.1.4. REST H1FE2

Routine: Restart routine

Routines called: SWIR from a SWI instruction.

Description: It is a cold start routine that is executed after a hardware reset which is generated by the exorciser in the BEGN (68Ø9) routine. The function of this routine is to clear the ram memory of the internal 68Ø5 and generate a SWI interrupt, so that the SWIR routine is executed to transfer control back to the exorciser. This routine is part of the initialization of the emulator board.

17.1.5. CINT H1EBØ

Routine: Change I/O status interrupt routine.

Routines called: DUPM

Description: This routine is executed when the internal I/O of the 68Ø5 needs to be changed. The vector address H1EBØ (in

memory location H1FEF and H1FFØ) is changed with the target system interrupt vector (in vector address H1FFA/B) by the exorciser to enable this routine to be executed after an interrupt. The functions of this routine are:

i) It restores the user interrupt vectors.

ii) It makes the program counter point to the STOP instruction of the SWIR routine.

iii) It changes the required I/O memory address with the index in HØC.

iv) It copies the memory block (DUPM).

v) It changes the one shot trigger to LI pin.

The stop instruction is executed in the SWIR routine after executing the RTI to transfer control to the exorciser.

17.1.6. VECTORS H1FEF

Description: CINT and RINT vectors are temporary stored in memory locations H1FEF and H1FF1. These vectors are interchanged with the main interrupt vectors to execute the required interrupt routine.

17.2. THE 68Ø9 EXORCISER EMULATOR SOFTWARE.

These subroutines are contained in the 2716 eprom on the emulator board addressed in the executive memory map from HEØØØ to HE7FF.

17.2.1. BEGN HEØØØ

Routine: Begins instruction.

Routines called: SSTA; SSTB; SLDA; SLDB.

Description: This is the main emulator initialization routine. It is accessed by executing the BEGN instruction on the exorciser when in the USER memory map. The functions of this routine are:

i) It enables the 8k block (buffer) to the exorciser and holds the reset of the 68Ø5 active.

ii) It changes the one shot trigger to the astable timer output.

iii) It loads the interrupt vector and updates the jump table for all additional instructions.

iv) It clears the break point addresses.

v) It loads the boot program and vectors (68Ø5 emulator code) into the 8k ram.

vi) It initializes the timer chip.

vii) It transfers the memory to the 68Ø5 micro.

viii) It removes the reset to allow the REST routine to be executed in the 68Ø5.

x) It waits for the interrupts and flashes the run LED.

17.2.2. STAA HE11D

Routine: Stores ACCA in USER map.

Description: Stores a byte contained in the accumulator A into the memory location (address contained in the X index register) of the USER memory map.

17.2.3. SLDA HE125

Routine: Loads ACCA from USER map.

Description: Loads the memory byte (address of which is contained in the X index register) from the USER memory map into the accumulator A.

17.2.4. STAB HE12D

Routine: Stores ACCB in USER memory.

Description: Stores a byte contained in the accumulator A into the memory location (address contained in the X index register) of the USER memory map.

17.2.5. SLDB HE135

Routine: Loads ACCB from USER memory.

Description: Loads the memory byte (address of which is contained in the X index register) from the USER memory map into the accumulator A.

17.2.6. INTR HE18Ø

Routine: Interrupt routine.

Routines called: SSTA; SLDA; SSTB; SLDB; XBUG.

Description: The 68Ø5 micro stops to transfer control to the exorciser causing the one shot to generate an interrupt to the exorciser. This would cause this routine to be executed. The functions of this routine are as follows:

i) It disables the timers and transfer the memory to the exorciser.

ii) It copies the stack area and computes the stack pointer from the known PC (H1F82).

iii) It copies the relavent I/O memory.

iv) It restores the program information into the break point (SWI) locations.

v) It updates the appropriate 68Ø9 registers with the 68Ø5 registers.

vi) It transfers the control to the monitor by executing the XBUG of

the exorciser.

This would finally envoke the EXBUG mode of the exorciser.

17.2.7. CONP H2EBA

Routine: Continues program instruction.

Routines called: STAA; SLDA; STAB; SLDB; BEG3.

Description: This routine is an added EXBUG instruction which is executed by CONP on the command line. The functions of this routine are as follows:

i) It restores the modified registers into 68Ø5 stack.

ii) It copies the internal mirror block to H1F.

iii) It swops the RINT vectors in the 68Ø5 memory map for recovery.

iv) It stores the SWI instructions into the program code for the breakpoints.

v) It enables the one shot timer and transfers control to the 68Ø5.

The BEG3 lable is finally executed which places the 68Ø9 micro in a wait for interrupt mode.

17.2.8. CHNG HE36E

Routine: Changes specific memory instruction.

Routines called: STAA; SLDA; STAB; SLDB; CON2

Description: This routine is an added EXBUG instruction which is executed by CHNG on the command line of the exorciser. The functions of this routine are as follows:

i) It stores the address of the memory change byte.

ii) It swops the CINT interrupt vector in the 68Ø5 memory map.

iii) It transfers the control to the 68Ø5 by executing CON2 in the CONP routine.

This transfer would cause the CINT routine in the 68Ø5 to be executed to change the required memory byte.

17.2.9. BPST HE3BD

Routine: Break point set instruction.

Routines called: XBUG

Description: This routine is an added XBUG command routine which is executed to set a break point in the break point stack. The break point address is stored in the Y register before executing this routine.

17.2.1Ø. BPCL HE3DC

Routine: Break point clear instruction.

Routine called: XBUG

Description: This routine is an added XBUG command

routine which is executed to clear a break point in the break point stack. The break point to be cleared is stored in the Y register before executing this routine.

17.2.11. BPAC HE3FB

Routine: Break point all clear instruction.

Routine called: XBUG

Description: This routine is an added XBUG command routine which is executed to clear all break points in the break point stack.

17.2.12. BPDS HE411

Routine: Break point display instruction.

Routine called: XBUG

Description: This routine is an added XBUG command routine which is executed to display the break points.

17.2.13. CTBEG HE14Ø

Routine: Instruction table.

Description: This table is the look up instruction table used by the 68Ø9 exorciser to reference the added user XBUG commands. The begining address (CTBEG) and the ending address (CTEND) of this table is stored into the XBUG ram by the initialization instruction (BEGN).

A P P E N D I X N

E M U L A T O R F L O W C H A R T S

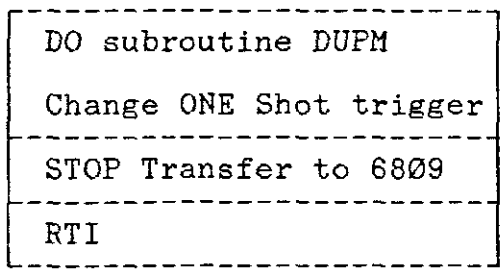
APPENDIX N

18. EMULATOR FLOWCHARTS

18.1. 6805 Emulator flowcharts.

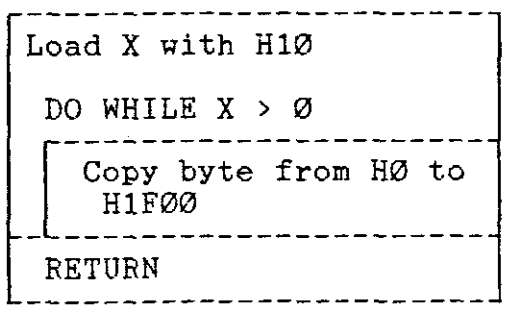
18.1.1. Software interrupt routine.

SWIR



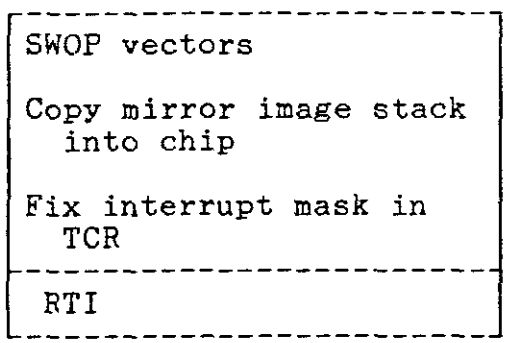
18.1.2. Duplicate memory routine.

DUPM



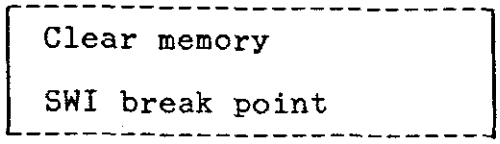
18.1.3. Return interrupt routine.

RINT



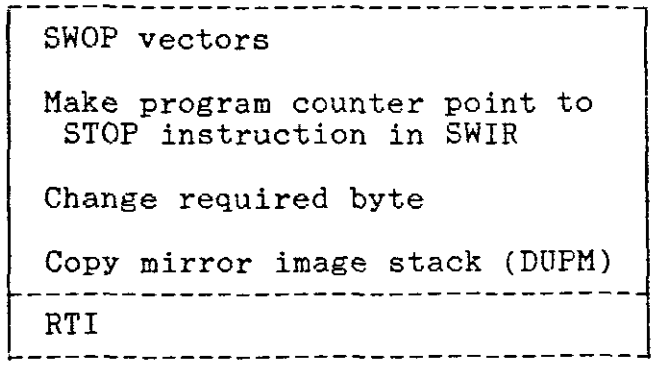
18.1.4. Reset routine.

REST



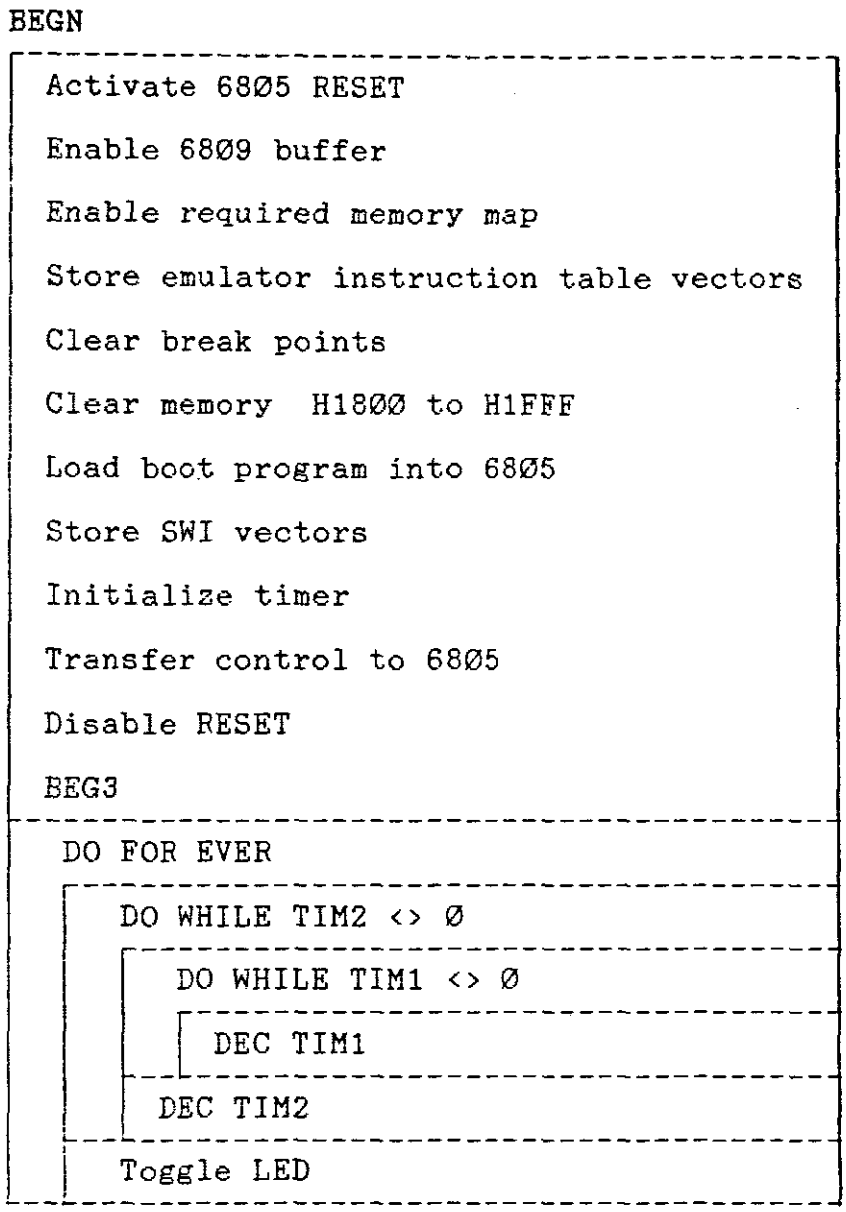
18.1.5. Change interrupt routine.

CINT



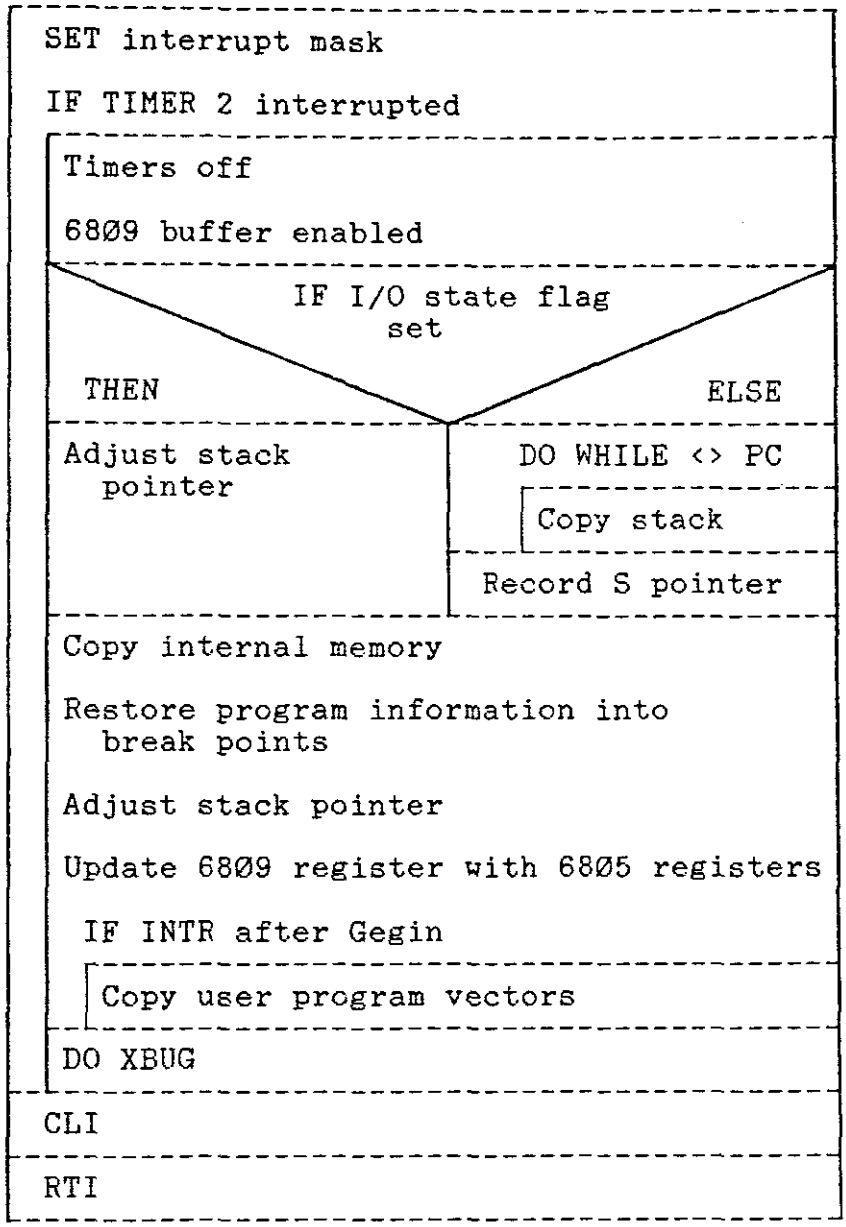
18.2. 68Ø9 Emulator flowcharts.

18.2.1. Begin instruction.



18.2.2. Interrupt routine.

INTR



18.2.3. Continue program instruction.

CONP

```
Restore modified vectors from EXEC
 map

Copy stack area

Swop Return interrupt vectors in 68Ø5

Store SWI instructions into break
 points

CON2
```

```
Astable to ONE shot

Enable 68Ø5 buffer

Execute BEG3
```

18.2.4. Change memory instruction.

CHNG

```
Store address of memory change

SWOP interrupt vectors with CINT
 routine in 68Ø5 map

Execute CON2
```

18.2.5. Break point set instruction.

BPST

```
Record Y register break
 point address
```

18.2.6. Break point clear instruction.

BPCL

```
Clear Y register break
 point address
```

18.2.7. Break point all clear.

BPAC

```
Clear all break points in
 meory
```

18.2.8. Break point display.

BPDS

```
Display all 8 break points
```

A P P E N D I X O

E M U L A T O R
S O F T W A R E L I S T I N G

APPENDIX O

19. EMULATOR SOFTWARE LISTING.

19.1. The 68Ø5 emulator software listing.

The following pages represents the software listing of the 68Ø5 emulator board assembly language program.

```
00001                           OPT    P=64
00002                           TTL    6805 EMULATOR SOFTWARE
00003                     ************************
00004                     *  6805 PROGRAM FOR    *
00005                     *  EMULATOR CARD       *
00006                     ************************
00007
00008                     ****************
00009                     *      EQUATES *
00010                     ****************
00011
00012          1FF5     A LIE    EQU    $1FF5    ONE SHOT TRIG
00013          1F0A     A CHIO   EQU    $1F0A    I/O CH FLAG
00014          1FF3     A SPH    EQU    $1FF3    STACK POINTER
00015          1FF4     A SPL    EQU    $1FF4    ADDRESS
00016          1FEE     A INTC   EQU    $1FEE    INT COUNTER
00017          1FFA     A INTH   EQU    $1FFA    INT VECTOR
00018          1FFB     A INTL   EQU    $1FFB
00019          1FF1     A RINH   EQU    $1FF1    RECOVERY INT
00020          1FF2     A RINL   EQU    $1FF2    VECTOR
00021          1FEF     A CINH   EQU    $1FEF    CHG I/O STAT
00022          1FF0     A CINL   EQU    $1FF0    VECTOR
00023          1F0C     A CHG    EQU    $1F0C    I/O CHG ADDRESS
00024          008E     A STP    EQU    $8E
00025          1F09     A TIMC1  EQU    $1F09    TIMER CONTROL REG
00026          000D     A TIMC   EQU    $0D      TIMER CONTROL REG
00027          0009     A TCR    EQU    $09      TIMER CONTROL REG
00028
00029
00030                     ***********************************
00031                     *      MACRO INSTRUCTION FOR STOP *
00032                     ***********************************
00033
00034                     STOP   MACR
00035                            FCB    STP
00036                            ENDM
00037
```

```
00039
00040                    ***********************************
00041                    *      SOFTWARE INTERRUPT ROUTINE *
00042                    ***********************************
00043
00044A 1F80                       ORG    $1F80
00045
00046A 1F80 AD 0C    1F8E SWIR    BSR    DUPM     COPY MEM & KNOWN PC
00047A 1F82 A6 81       A        LDA    #$81     ONE SHOT TRIG OVER
00048A 1F84 C7 1FF5     A        STA    LIE      TO LI PIN
00049A 1F87 A6 00       A        LDA    #$00
00050A 1F89 C7 1F0A     A        STA    CHIO     CLR CHG I/O FLAG
00051A 1F8C                      STOP
00052A 1F8D 80                   RTI             CONTINUE
00053
00054
00055                    ******************************
00056                    *      COPY MEMORY SUBROUTINE *
00057                    ******************************
00058
00059A 1F8E AE 10       A DUPM   LDX    #$10     COPY MEMORY
00060A 1F90 D6 FFFF     A DUP1   LDA    -1,X     FROM 0-F
00061A 1F93 D7 1EFF     A        STA    $1EFF,X  TO
00062A 1F96 5A                   DEX             1F00-1F0F
00063A 1F97 26 F7    1F90       BNE    DUP1
00064A 1F99 81                   RTS
00065
00066
00067                    ****************************
00068                    *      RECOVERY INT ROUTINE *
00069                    ****************************
00070
00071A 1F9A C6 1FFA     A RINT   LDA    INTH     SWOP
00072A 1F9D CE 1FF1     A        LDX    RINH     VECTORS
00073A 1FA0 C7 1FF1     A        STA    RINH
00074A 1FA3 CF 1FFA     A        STX    INTH
00075A 1FA6 C6 1FFB     A        LDA    INTL
00076A 1FA9 CE 1FF2     A        LDX    RINL
00077A 1FAC C7 1FF2     A        STA    RINL
00078A 1FAF CF 1FFB     A        STX    INTL
00079A 1FB2 C6 1FF4     A        LDA    SPL
00080A 1FB5 A0 05       A        SUB    #$5
00081A 1FB7 97                   TAX
00082A 1FB8 A6 05       A        LDA    #$5
00083A 1FBA C7 1FEE     A        STA    INTC
00084A 1FBD F6             RIN1  LDA    ,X       RESTORE INT STAT
00085A 1FBE D7 1F00     A        STA    $1F00,X  FROM STACK
00086A 1FC1 C6 1FEE     A        LDA    INTC
00087A 1FC4 4A                   DECA
00088A 1FC5 C7 1FEE     A        STA    INTC
00089A 1FC8 27 03    1FCD       BEQ    RIN2
00090A 1FCA 5A                   DEX
00091A 1FCB 20 F0    1FBD       BRA    RIN1
00092A 1FCD AE 70       A RIN2   LDX    #$70     TFR STACK MEM
00093A 1FCF D6 1F0F     A RIN3   LDA    $1F0F,X  BACK INTO CHIP
00094A 1FD2 E7 0F       A        STA    $0F,X
00095A 1FD4 5A                   DEX
00096A 1FD5 26 F8    1FCF       BNE    RIN3
00097A 1FD7 C6 1F09     A        LDA    TIMC1
00098A 1FDA E7 0D       A        STA    TIMC
00099A 1FDC 0C 0D 02 1FE1        BRSET  6,TIMC,RIN4 FIX INT MASK
00100A 1FDF 1D 09       A        BCLR   6,TCR    IN TCR
00101A 1FE1 80             RIN4  RTI
00102
```

```
00103
00104                    ************************
00105                    *      RESTART ROUTINE *
00106                    ************************
00107
00108A 1FE2 AE 70       A REST   LDX    #$70     CLEAR MEMORY
00109A 1FE4 6F 0F       A RES1   CLR    $F,X
00110A 1FE6 5A                   DEX
00111A 1FE7 26 FB    1FE4        BNE    RES1
00112A 1FE9 83                   SWI
00113
00114
00115                    ****************************
00116                    *      INT CHANGE I/O STAT *
00117                    ****************************
00118
00119A 1EB0                      ORG    $1EB0
00120
00121A 1EB0 C6 1FFA     A CINT   LDA    INTH     SWOP VECTORS
00122A 1EB3 CE 1FEF     A        LDX    CINH
00123A 1EB6 C7 1FEF     A        STA    CINH
00124A 1EB9 CF 1FFA     A        STX    INTH
00125A 1EBC C6 1FFB     A        LDA    INTL
00126A 1EBF CE 1FF0     A        LDX    CINL
00127A 1EC2 C7 1FF0     A        STA    CINL
00128A 1EC5 CF 1FFB     A        STX    INTL
00129A 1EC8 C6 1FF4     A        LDA    SPL      MAKE PC POINT
00130A 1ECB A0 05       A        SUB    #$05     TO STOP
00131A 1ECD 97                   TAX             INSTRUCTION
00132A 1ECE 7A                   DEC    ,X
00133A 1ECF CE 1F0C     A        LDX    CHG
00134A 1ED2 A6 0B       A        LDA    #$0B
00135A 1ED4 B7 0C       A        STA    $0C
00136A 1ED6 D6 1F00     A        LDA    $1F10,X  CHANGE I/O
00137A 1ED9 F7                   STA    ,X
00138A 1EDA C6 1F09     A        LDA    TIMC1    FIX TCR
00139A 1EDD B7 0D       A        STA    TIMC
00140A 1EDF 0C 0D 02 1EE4        BRSET  6,TIMC,CIN3
00141A 1EE2 1D 09       A        BCLR   6,TCR
00142A 1EE4 AD 6B    1EF1 CIN3   BSR    CIN2     COPY MEMORY
00143A 1EE6 A6 80       A        LDA    #$80
00144A 1EE8 C7 1F0A     A        STA    CHIO
00145A 1EEB A6 01       A        LDA    #$01
00146A 1EED C7 1FF5     A        STA    LIE
00147A 1EF0 80                   RTI
00148A 1EF1 CC 1F6E     A CIN2   JMP    DUPM
00149
00150
00151                    ************************
00152                    *      INTERRUPT VECTORS *
00153                    ************************
00154
00155A 1FEF                      ORG    $1FEF
00156A 1FEF    1EB0     A        FDB    CINT
00157A 1FF1    1F9A     A        FDB    RINT
00158A 1FFC                      ORG    $1FFC
00159A 1FFC    1F80     A        FDB    SWIR
00160A 1FFE    1FE2     A        FDB    REST
00161
00162
00163                            END
TOTAL ERRORS 00000--00000
```

```
1F0C CHG    00023*00133
1F0A CHID   00013*00050 00144
1EF1 CIN2   00142 00148*
1EE4 CIN3   00140 00142*
1FEF CINH   00021*00122 00123
1FF0 CINL   00022*00126 00127
1EB0 CINT   00121*00156
1F9B DUP1   00060*00063
1FBE DUPN   00046 00059*00148
1FEE INTC   00016*00083 00086 00088
1FFA INTH   00017*00071 00074 00121 00124
1FFB INTL   00018*00075 00078 00125 00128
1FF5 LIE    00012*00048 00146
1FE4 RES1   00109*00111
1FE2 REST   00108*00160
1FBD RIN1   00084*00091
1FCD RIN2   00089 00092*
1FCF RIN3   00093*00096
1FE1 RIN4   00099 00101*
1FF1 RINH   00019*00072 00073
1FF2 RINL   00020*00076 00077
1F9A RINT   00071*00157
1FF3 SPH    00014*
1FF4 SPL    00015*00079 00129
008E STP    00024*00051
1F8B SWIR   00046*00159
0009 TCR    00027*00100 00141
000D TIMC   00026*00098 00099 00139 00140
1F09 TIMC1  00025*00097 00138
```

19.2. The 68Ø9 exorciser software listing.

The following pages represents the software listing of the 68Ø9 exorciser assembly language program.

```
00001                    OPT    P=64
00002                    TTL    6809 EMULATOR SOFTWARE
00003                  **************************
00004                  * 6809 PRG FOR EMULATOR  *
00005                  * CARD                   *
00006                  **************************
00007
00008          FFF0  A MEMM  EQU   $FFF0   USER MEM MAP CHOISE
00009          FF16  A PCOR  EQU   $FF16   APPROPRIATE REGS
00010          FF18  A USP   EQU   $FF18   IN EXEC MAP
00011          FF1A  A INRY  EQU   $FF1A
00012          FF1C  A INRX  EQU   $FF1C
00013          FF1E  A DPR   EQU   $FF1E
00014          FF1F  A AAR   EQU   $FF1F
00015          FF20  A BBR   EQU   $FF20
00016          FF21  A CCR   EQU   $FF21
00017          FF22  A SSP   EQU   $FF22
00018          4000  A PCON  EQU   $4000   PC FLAG
00019          2000  A MCHG  EQU   $2000   MEM BUFF
00020          1F0A  A CHIO  EQU   $1F0A   CHANGE I/O
00021          1FF3  A SPH   EQU   $1FF3   STACK P H
00022          1FF4  A SPL   EQU   $1FF4   STACK P L
00023          F5C2  A XBUG  EQU   $F5C2   RET TO XBUG LOC
00024          1FFA  A INTH  EQU   $1FFA   05 INT VECTOR
00025          1FFB  A INTL  EQU   $1FFB
00026          1FF1  A RINH  EQU   $1FF1   RET INT VECTOR
00027          1FF2  A RINL  EQU   $1FF2
00028          1FEF  A CINH  EQU   $1FEF   CHG INT VECTOR
00029          1FF0  A CINL  EQU   $1FF0
00030          1F0C  A CHG   EQU   $1F0C   HOLDS CHG ADD
00031          000C  A CHG1  EQU   $000C   HOLDS CHG ADD
00032          2021  A CR2   EQU   $2021   CONTROL REG 2
00033          2020  A CR13  EQU   $2020   CONTROL REG 1/3
00034          2022  A LAT1  EQU   $2022
00035          2023  A LAL1  EQU   $2023
00036          2024  A LAT2  EQU   $2024
00037          2025  A LAL2  EQU   $2025
00038          2026  A LAT3  EQU   $2026
00039          2027  A LAL3  EQU   $2027
00040          1FF5  A LIE   EQU   $1FF5   LI/E LOCATION
00041          2010  A MEM   EQU   $2010
00042          2018  A LED   EQU   $2018   FLASH LED
00043          4003  A LEDM  EQU   $4003
00044          4001  A TIM1  EQU   $4001
00045          4002  A TIM2  EQU   $4002
00046          FFF8  A INRH  EQU   $FFF8   09TINT VECTOR
00047          FFF9  A INRL  EQU   $FFF9
00048          0000  A PORTA EQU   $0000
00049          0001  A PORTB EQU   $0001
00050          0004  A DDA   EQU   $0004
00051          0005  A DDB   EQU   $0005
00052          0008  A TDR   EQU   $0008
00053          0009  A TCR   EQU   $0009
00054          E4AF  A BOOT  EQU   $E4AF
00055          E180  A INTRR EQU   $E180
00056          400F  A BP0   EQU   $400F   BREAK POINT 0
00057          4010  A BP1   EQU   $4010   BREAK POINT 1-8
00058          4020  A PGRI  EQU   $4020   PRG CONT OF BP
00059          F01E  A XOUT4H EQU  $F01E   4 CHRS TO SCREEN
00060          F021  A XPCRLF EQU  $F021   C/R L/F TO SCREEN
00061          4030  A VECT  EQU   $4030
00062          FF0E  A TABEG EQU   $FF0E   JMP TABLE
00063          FF10  A TAEND EQU   $FF10   VECTORS
00064
```

```
00066                    **************************
00067                    * BEGIN INSTRUCTION FOR *
00068                    *   BOOTING             *
00069                    *   INITIALIZE FTM      *
00070                    *   WAIT FOR INTS       *
00071                    **************************
00072
0007A E000                          ORG    $E000
0007A E000 86   00      A BEGN      LDAA   #$00     RESET 05
0007A E002 8E   2018    A           LDX    #LED
0007A E005 17   0115 E11D           LBSR   SSTA
0007A E008 86   02      A           LDAA   #$02     BUF-NOT
0007A E00A 8E   2000    A           LDX    #NCHG    LED-OFF
0007A E00D 17   010D E11D           LBSR   SSTA
0007A E010 8E   FFF0    A           LDX    #MEMM    ENABLE
0007A E013 17   010F E125           LBSR   SLDA     REQUIRED
0007A E016 8A   08      A           ORA    #$08     MEMORY
0007A E018 81   10      A           CMPA   #$10     MAP  OR
0007A E01A 2D   01   E01D           BLT    BEG6     DISABLE
0007A E01C 4F                       CLRA            COMPLETELY IF
0007A E01D 8E   2010    A BEG6      LDX    #MEM     EXT MEM
0007A E020 17   00FA E11D           LBSR   SSTA     REQUIRED
0007A E023 86   08      A           LDAA   #$08     ASTABLE TO
0007A E025 8E   1FF5    A           LDX    #LIE     ONE SHOT
0007A E028 17   00F2 E11D           LBSR   SSTA
0007A E02B 8E   E180    A           LDX    #INTRR   LOAD INT
0007A E02E BF   FFF8    A           STX    INRH     VECTOR
0007A E031 8E   E140    A           LDX    #CTBEG   STORE JMP
0007A E034 BF   FF0E    A           STX    TABEG    TABLE
0007A E037 8E   E169    A           LDX    #CTEND   VECTORS
0007A E03A BF   FF10    A           STX    TAEND
0007A E03D 108E 000F    A           LDY    #$F      SAVE VECTORS
0007A E041 30   A9 1FF0 A BEG5      LEAX   $1FF0,Y  OF USER
0007A E045 17   00DD E125           LBSR   SLDA     PROGRAM
0007A E048 A7   A9 4030 A           STA    VECT,Y
0007A E04C 31   3F                  DEY
0007A E04E 26   F1   E041           BNE    BEG5
0007A E050 8E   0010    A           LDX    #$10     CLR BREAK
0007A E053 6F   89 400F A BEG4      CLR    BP0,X    POINT
0007A E057 30   1F                  DEX             ADDRESSES
0007A E059 26   F8   E053           BNE    BEG4
0007A E05B 108E 0130    A           LDY    #$0130
0007A E05F 86   00      A           LDAA   #$00
0007A E061 30   A9 1ECF A BEG1      LEAX   $1ECF,Y  CLR MEMORY
0007A E065 17   00B5 E11D           LBSR   SSTA
0007A E068 31   3F                  DEY             FROM 1800
0007A E06A 26   F5   E061           BNE    BEG1     TO 1FFF
0007A E06C 108E 0150    A           LDY    #$0150   LOAD BOOT
0007A E070 A6   A9 E4AF A BEG2      LDAA   BOOT,Y   PROGRAM INTO
0007A E074 30   A9 1EAF A           LEAX   $1EAF,Y  6805 MEM
0007A E078 17   00A2 E11D           LBSR   SSTA
0007A E07B 31   3F                  DEY
0007A E07D 26   F1   E070           BNE    BEG2
0007A E07F 8E   1FFC    A           LDX    #$1FFC   GET SWI
0007A E082 17   00A0 E125           LBSR   SLDA     VECTOR
0007A E085 8E   1FFD    A           LDX    #$1FFD   AND STORE
0007A E088 17   00AA E135           LBSR   SLDB     IN USER
0007A E08B 8E   4030    A           LDX    #VECT    PROGRAM
0007A E08E ED   0C      A           STD    $C,X     VECTORS
00125
00126                       *       INITILIZE TIMER
00127
0007A E090 86   F6      A           LDAA   #$F6     TIMER 2,ONE
0007A E092 8E   2021    A           LDX    #CR2     SHOT
```

```
00138A E095 17  0085 E11D       LBSR  SSTA
00131A E098 86  B6     A        LDAA  #$B6     TIMER 3 ONE
00132A E09A 8E  2028   A        LDX   #CR13    SHOT
00133A E09D 8D  7E   E11D       BSR   SSTA
00134A E09F 86  F7     A        LDAA  #$F7
00135A E0A1 8E  2021   A        LDX   #CR2
00136A E0A4 8D  77   E11D       BSR   SSTA
00137A E0A6 86  97     A        LDAA  #$97     TIMER 1
00138A E0A8 8E  2028   A        LDX   #CR13    ASTABLE
00139A E0AB 8D  7B   E11D       BSR   SSTA
00140A E0AD 86  01     A        LDAA  #$01     100KZ
00141A E0AF 8E  2022   A        LDX   #LAT1
00142A E0B2 8D  69   E11D       BSR   SSTA
00143A E0B4 86  04     A        LDAA  #$04
00144A E0B6 8E  2023   A        LDX   #LAL1
00145A E0B9 8D  62   E11D       BSR   SSTA
00146A E0BB 86  00     A        LDAA  #$00     50 MIC SEC
00147A E0BD 8E  2024   A        LDX   #LAT2
00148A E0C0 8D  5B   E11D       BSR   SSTA
00149A E0C2 86  32     A        LDAA  #$32
00150A E0C4 8E  2025   A        LDX   #LAL2
00151A E0C7 8D  54   E11D       BSR   SSTA
00152A E0C9 86  08     A        LDAA  #$08     5IMIC SEC
00153A E0CB 8E  2026   A        LDX   #LAT3
00154A E0CE 8D  4D   E11D       BSR   SSTA
00155A E0D0 86  05     A        LDAA  #$05
00156A E0D2 8E  2027   A        LDX   #LAL3
00157A E0D5 8D  46   E11D       BSR   SSTA
00158A E0D7 86  01     A        LDAA  #$01     BUF-6805
00159A E0D9 8E  2008   A        LDX   #MCHG    LED ON
00160A E0DC 8D  3F   E11D       BSR   SSTA
00161A E0DE 86  96     A        LDAA  #$96
00162A E0E0 8E  2028   A        LDX   #CR13    TIMERS OPERATE
00163A E0E3 8D  38   E11D       BSR   SSTA
00164A E0E5 86  02     A        LDAA  #$02     DISABLE RESET
00165A E0E7 8E  2018   A        LDX   #LED     TO 6805
00166A E0EA 1C  EF       BEG3   CLI
00167A E0EC 7D  4001   A MAN1   TST   TIM1     WAIT FOR INTS
00168A E0EF 26  26   E117       BNE   MAN4
00169A E0F1 7D  4002   A        TST   TIM2
00170A E0F4 26  1E   E114       BNE   MAN3
00171A E0F6 7D  4003   A        TST   LEDM
00172A E0F9 27  0D   E108       BEQ   MAN2
00173A E0FB 7F  4003   A        CLR   LEDM
00174A E0FE 86  02     A        LDAA  #$02
00175A E100 8E  2018   A        LDX   #LED
00176A E103 8D  18   E11D       BSR   SSTA
00177A E105 7E  E114   A        JMP   MAN3
00178A E108 86  FF     A MAN2   LDAA  #$FF
00179A E10A B7  4003   A        STAA  LEDM
00180A E10D 86  03     A        LDAA  #$03
00181A E10F 8E  2018   A        LDX   #LED
00182A E112 8D  09   E11D       BSR   SSTA
00183A E114 7A  4002   A MAN3   DEC   TIM2
00184A E117 7A  4001   A MAN4   DEC   TIM1
00185A E11A 7E  E0EC   A        JMP   MAN1
00186
00187
00188                    **********************
00189                    *     SUBROUTINE STAA *
00190                    **********************
00191
00192A E11D B5  FCFC   A SSTA   BITA  $FCFC
00193A E120 20  00   E122       BRA   SST1
```

```
00194A E122 A7   84     A SST1  STAA   ,X
00195A E124 39                  RTS
00196
00197                   **********************
00198                   *     SUBROUTINE LDAA *
00199                   **********************
00200
00201A E125 B5   FCFC   A SLDA  BITA   $FCFC
00202A E128 20   00   E12A      BRA    LLD1
00203A E12A A6   84     A LLD1  LDAA   ,X
00204A E12C 39                  RTS
00205
00206                   **********************
00207                   *     SUBROUTINE STAB *
00208                   **********************
00209
00210A E12D B5   FCFC   A SSTB  BITA   $FCFC
00211A E130 20   00   E132      BRA    SST2
00212A E132 E7   84     A SST2  STAB   ,X
00213A E134 39                  RTS
00214
00215                   **********************
00216                   *     SUBROUTINE LDAB *
00217                   **********************
00218
00219A E135 B5   FCFC   A SLDB  BITA   $FCFC
00220A E138 20   00   E13A      BRA    LLD2
00221A E13A E6   84     A LLD2  LDAB   ,X
00222A E13C 39                  RTS
00223
00224
00225                   ************************
00226                   *     INTERRUPT ROUTINE *
00227                   ************************
00228
00229
00230A E180                      ORG    INTRR
00231
00232A E180 1A   10      INTR    SEI
00233A E182 8E   2021   A        LDX    #CR2     GET STAT
00234A E185 8D   9E   E125       BSR    SLDA
00235A E187 84   02     A        ANDA   #$02     TIMER2 INTRUPTED?
00236A E189 1027 012A E2B7       LBEQ   INT5
00237A E18D C6   97     A        LDAB   #$97
00238A E18F 8E   2020   A        LDX    #CR13    TIMERS OFF
00239A E192 8D   99   E12D       BSR    SSTB
00240A E194 86   02     A        LDAA   #$02     BUF-NOT
00241A E196 8E   2000   A        LDX    #MCHG    LED OFF
00242A E199 8D   82   E11D       BSR    SSTA
00243A E19B 8E   1F0A   A        LDX    #CHID    I/O STAT FLAG
00244A E19E 8D   85   E125       BSR    SLDA
00245A E1A0 84   80     A        ANDA   #$80     DO I/O IF D7
00246A E1A2 26   34   E1D8       BNE    INT3     SET
00247A E1A4 7F   4000   A        CLR    PCON
00248A E1A7 108E 0080   A        LDY    #$80
00249A E1AB 30   A4     A INT1   LEAX   ,Y       COPY STACK INFO
00250A E1AD 8D   86   E135       BSR    SLDB
00251A E1AF 30   1F              DEX
00252A E1B1 17   FF71 E125       LBSR   SLDA
00253A E1B4 1083 1F82   A        CMPD   #$1F82   KNOWN PC
00254A E1B8 26   18   E1D2       BNE    INT2
00255A E1BA 1F   20     A        TFR    Y,D      STACK P MEM
00256A E1BC 8E   1FF3   A        LDX    #SPH
00257A E1BF 17   FF5B E11D       LBSR   SSTA
```

```
00258A E1C2 8E  1FF4    A       LDX   #SPL
00259A E1C5 17  FF65 E12D       LBSR  SSTB
00260A E1C8 CC  0000    A       LDD   #$00     CLR D
00261A E1CB 30  A4      A       LEAX  ,Y
00262A E1CD 17  FF4D E11D       LBSR  SSTA
00263A E1D0 20  1C   E1EE       BRA   INT4
00264A E1D2 31  3F          INT2 LEAY -1,Y
00265A E1D4 26  D5   E1AB       BNE   INT1
00266A E1D6 20  16   E1EE       BRA   INT4
00267A E1D8 8E  1F0A    A INT3  LDX   #CHIO
00268A E1DB 86  00      A       LDAA  #$00
00269A E1DD 17  FF3D E11D       LBSR  SSTA
00270A E1E0 8E  1FF4    A       LDX   #SPL     ADJUST STACK
00271A E1E3 17  FF3F E125       LBSR  SLDA
00272A E1E6 80  05      A       SUBA  #$05     POINTER
00273A E1E8 8E  1FF4    A       LDX   #SPL
00274A E1EB 17  FF2F E11D       LBSR  SSTA
00275A E1EE 8E  1F00    A INT4  LDX   #$1F00
00276A E1F1 17  FF31 E125       LBSR  SLDA     COPY RELEVANT
00277A E1F4 8E  0000    A       LDX   #PORTA   MEM FROM
00278A E1F7 17  FF23 E11D       LBSR  SSTA
00279A E1FA 8E  1F01    A       LDX   #$1F01   $00 TO $0F
00280A E1FD 17  FF25 E125       LBSR  SLDA
00281A E200 8E  0001    A       LDX   #PORTB
00282A E203 17  FF17 E11D       LBSR  SSTA
00283A E206 8E  1F04    A       LDX   #$1F04
00284A E209 17  FF19 E125       LBSR  SLDA
00285A E20C 8E  0004    A       LDX   #DDA
00286A E20F 17  FF0B E11D       LBSR  SSTA
00287A E212 8E  1F05    A       LDX   #$1F05
00288A E215 17  FF0D E125       LBSR  SLDA
00289A E218 8E  0005    A       LDX   #DDB
00290A E21B 17  FEFF E11D       LBSR  SSTA
00291A E21E 8E  1F08    A       LDX   #$1F08
00292A E221 17  FF01 E125       LBSR  SLDA
00293A E224 8E  0008    A       LDX   #TDR
00294A E227 17  FEF3 E11D       LBSR  SSTA
00295A E22A 8E  1F09    A       LDX   #$1F09
00296A E22D 17  FEF5 E125       LBSR  SLDA
00297A E230 8E  0009    A       LDX   #TCR
00298A E233 17  FEE7 E11D       LBSR  SSTA
00299A E236 108E 4018   A       LDY   #BP1     RESTORE
00300A E23A AE  A4      A INT7  LDX   ,Y       PROGRAM
00301A E23C 27  06   E244       BEQ   INT8     INFORMATION
00302A E23E A6  A8 10   A       LDAA  $10,Y    INTO USER
00303A E241 17  FED9 E11D       LBSR  SSTA     PROGRAM
00304A E244 31  22      A INT8  LEAY  $2,Y
00305A E246 108C 4020   A       CMPY  #PGRI
00306A E24A 26  EE   E23A       BNE   INT7
00307A E24C 8E  1FF4    A       LDX   #SPL     ADJUST
00308A E24F 17  FEE3 E135       LBSR  SLDB     STACK POINTER
00309A E252 CB  05      A       ADDB  #$05
00310A E254 17  FED6 E120       LBSR  SSTB
00311A E257 8E  1FF4    A INT6  LDX   #SPL     GET STACK
00312A E25A 17  FED8 E135       LBSR  SLDB     POINTER
00313A E25D 8E  1FF3    A       LDX   #SPH
00314A E260 17  FEC2 E125       LBSR  SLDA
00315A E263 1F  03      A       TFR   D,U
00316A E265 FF  FF18    A       STU   USP      \GET APPROPRIATE
00317A E268 1F  01      A       TFR   D,X      \REGS FROM EXEC MAP
00318A E26A 17  FEC8 E135       LBSR  SLDB     \AND STORE IN
00319A E26D 30  1F              LEAX -1,X      \6805 STACK
00320A E26F 17  FEB3 E125       LBSR  SLDA     \AREA
00321A E272 5A                  DECB           \PC 1 LESS
```

```
00322A E273 FD   FF16    A        STD   PCOR     /
00323A E276 30   1F               TFR   D,X      /
00324A E278 17   FEBA E135        LBSR  SLDB     /
00325A E27B 4F                    CLRA           /
00326A E27C FD   FF1C    A        STD   INRX     /
00327A E27F 30   1F               TFR   D,X      /
00328A E281 17   FEA1 E125        LBSR  SLDA     /
00329A E284 B7   FF1F    A        STAA  AAR      /
00330A E287 30   1F               TFR   D,X      /
00331A E289 17   FE99 E125        LBSR  SLDA     /
00332A E28C B7   FF20    A        STAA  BBR      /
00333A E28F FC   FF16    A        LDD   PCOR     CHECK IF
00334A E292 1083 1F00    A        CMPD  #$1F00   INTRR IS
00335A E296 2D   1C   E2B4        BLT   INT9     EXECUTED AFTER
00336A E298 108E 0006    A        LDY   #$6      BEGN AND
00337A E29C EC   A9 4030 A INT10  LDD   VECT,Y   COPY
00338A E2A0 30   A9 1FF0 A        LEAX  $1FF0,Y  USER PROGRAM
00339A E2A4 17   FE76 E11D        LBSR  SSTA     VECTORS
00340A E2A7 30   01      A        LEAX  1,X
00341A E2A9 17   FE81 E12D        LBSR  SSTB
00342A E2AC 31   22      A        LEAY  2,Y
00343A E2AE 108C 0010    A        CMPY  #$10
00344A E2B2 26   E8   E29C        BNE   INT10
00345A E2B4 7E   F5C2    A INT9   JMP   XBUG
00346A E2B7 1C   EF       INT5    CLI
00347A E2B9 3B                    RTI
00348
00349                    ********************************
00350                    * CONTINUE PROGRAM INSTRUCTION *
00351                    ********************************
00352
00353
00354A E2BA 8E   1FF3    A CONP   LDX   #SPH    \RESTORE
00355A E2BD 17   FE65 E125        LBSR  SLDA     \MODIFIED
00356A E2C0 8E   1FF4    A        LDX   #SPL    \STACK FROM
00357A E2C3 17   FE6F E135        LBSR  SLDB     \EXEC MAP
00358A E2C6 1F   01      A        TFR   D,X      \
00359A E2C8 FC   FF16    A        LDD   PCOR     \
00360A E2CB 17   FE5F E12D        LBSR  SSTB     \
00361A E2CE 30   1F               TFR   D,X      \
00362A E2D0 17   FE4A E11D        LBSR  SSTA     \
00363A E2D3 FC   FF1C    A        LDD   INRX     /
00364A E2D6 30   1F               TFR   D,X      /
00365A E2D8 17   FE52 E12D        LBSR  SSTB     /
00366A E2DB B6   FF1F    A        LDAA  AAR      /
00367A E2DE 30   1F               TFR   D,X      /
00368A E2E0 17   FE3A E11D        LBSR  SSTA     /
00369A E2E3 B6   FF20    A        LDAA  BBR      /
00370A E2E6 30   1F               TFR   D,X      /
00371A E2E8 17   FE32 E11D        LBSR  SSTA     /
00372A E2EB 108E 0070    A        LDY   #$70
00373A E2EF 30   2F      A CON1   LEAX  $0F,Y    COPY STACK AREA
00374A E2F1 17   FE31 E125        LBSR  SLDA     FROM PAGE 0 TO
00375A E2F4 30   A9 1F0F A        LEAX  $1F0F,Y  PAGE 1F
00376A E2F8 17   FE22 E11D        LBSR  SSTA
00377A E2FB 31   3F               LEAY  -1,Y
00378A E2FD 26   F0   E2EF        BNE   CON1
00379A E2FF 8E   1FFA    A        LDX   #INTH    INT VECTOR
00380A E302 17   FE20 E125        LBSR  SLDA     RET INT VECTOR
00381A E305 8E   1FFB    A        LDX   #INTL    SWAP VECTORS
00382A E308 17   FE2A E135        LBSR  SLDB
00383A E30B 1F   03      A        TFR   D,U
00384A E30D 8E   1FF1    A        LDX   #RINH
00385A E310 17   FE12 E125        LBSR  SLDA
```

```
00386A E313 BE   1FF2    A        LDX   #RINL
00387A E316 17   FE1C E135        LBSR  SLDB
00388A E319 BE   1FFB    A        LDX   #INTL
00389A E31C 17   FE0E E12D        LBSR  SSTB
00390A E31F BE   1FFA    A        LDX   #INTH
00391A E322 17   FDF8 E11D        LBSR  SSTA
00392A E325 1F   30      A        TFR   U,D
00393A E327 BE   1FF1    A        LDX   #RINH
00394A E32A 17   FDF0 E11D        LBSR  SSTA
00395A E32D BE   1FF2    A        LDX   #RINL
00396A E330 17   FDFA E12D        LBSR  SSTB
00397A E333 108E 4010    A        LDY   #BP1     STORE SWI
00398A E337 AE   A4      A CON4   LDX   ,Y       INSTRUCTION
00399A E339 27   10   E34B        BEQ   CON3     INTO USER
00400A E33B BC   FF16    A        CMPX  PCOR     PROGRAM
00401A E33E 27   0B   E34B        BEQ   CON3
00402A E340 17   FDE2 E125        LBSR  SLDA
00403A E343 A7   A8 10   A        STAA  $10,Y
00404A E346 86   83      A        LDAA  #$83
00405A E348 17   FDD2 E11D        LBSR  SSTA
00406A E34B 31   22      A CON3   LEAY  2,Y
00407A E34D 108C 4020    A        CMPY  #PGRI
00408A E351 26   E4   E337        BNE   CON4
00409A E353 86   00      A CON2   LDAA  #$00     ASTABLE TO
00410A E355 BE   1FF5    A        LDX   #LIE     ONE SHOT
00411A E358 17   FDC2 E11D        LBSR  SSTA     TRIGGER
00412A E35B 86   01      A        LDAA  #$01     BUFF-6805
00413A E35D BE   2000    A        LDX   #MCHG    LED ON
00414A E360 17   FDBA E11D        LBSR  SSTA
00415A E363 86   96      A        LDAA  #$96     TIIMERS OPERATE
00416A E365 BE   2020    A        LDX   #CR13
00417A E368 17   FDB2 E11D        LBSR  SSTA
00418A E36B 7E   E0EA    A        JMP   BEG3
00419
00420                     ***********************************
00421                     * CHANGE SPECIFIC MEM INSTRUCTION *
00422                     ***********************************
00423
00424A E36E BE   000C    A CHNG   LDX   #CHG1    ADDRESS OF MEM CHG
00425A E371 17   FDC1 E135        LBSR  SLDB
00426A E374 BE   1F0C    A        LDX   #CHG
00427A E377 17   FDB3 E12D        LBSR  SSTB
00428A E37A 4F                    CLRA
00429A E37B 1F   01      A        TFR   D,X
00430A E37D 17   FDA5 E125        LBSR  SLDA
00431A E380 30   89 1F00 A        LEAX  $1F00,X
00432A E384 17   FD96 E11D        LBSR  SSTA
00433A E387 BE   1FFA    A        LDX   #INTH    SWAP VECTORS
00434A E38A 17   FD98 E125        LBSR  SLDA
00435A E38D BE   1FFB    A        LDX   #INTL
00436A E390 17   FDA2 E135        LBSR  SLDB
00437A E393 1F   03      A        TFR   D,U
00438A E395 BE   1FEF    A        LDX   #CINH
00439A E398 17   FD8A E125        LBSR  SLDA
00440A E39B BE   1FF0    A        LDX   #CINL
00441A E39E 17   FD94 E135        LBSR  SLDB
00442A E3A1 BE   1FFB    A        LDX   #INTL
00443A E3A4 17   FD86 E12D        LBSR  SSTB
00444A E3A7 BE   1FFA    A        LDX   #INTH
00445A E3AA 17   FD78 E11D        LBSR  SSTA
00446A E3AD 1F   30      A        TFR   U,D
00447A E3AF BE   1FEF    A        LDX   #CINH
00448A E3B2 17   FD68 E11D        LBSR  SSTA
00449A E3B5 BE   1FF0    A        LDX   #CINL
```

```
00450A E3B8 17   FD72 E12D       LBSR  SSTB
00451A E3BB 20   96   E353       BRA   CON2
00452
00453
00454                    ************************************
00455                    *     BREAK POINT SET INSTRUCTION *
00456                    ************************************
00457
00458A E3BD 8E   4010    A BPST   LDX   #BP1     Y REG CONTAINS
00459A E3C0 8C   4020    A BPS1   CMPX  #PGRI    THE BREAK
00460A E3C3 27   14   E3D9       BEQ   BPS2     POINT
00461A E3C5 10AE 81      A       LDY   ,X++
00462A E3C8 26   F6   E3C0       BNE   BPS1
00463A E3CA 10BE FF1A    A       LDY   INRY
00464A E3CE 10AF 63      A       STY   ,--X
00465A E3D1 108E 0000    A       LDY   #$00
00466A E3D5 10BF FF1A    A       STY   INRY
00467A E3D9 7E   F5C2    A BPS2   JMP   XBUG
00468
00469
00470                    ************************************
00471                    *     BREAK POINT CLEAR INSTRUCTION *
00472                    ************************************
00473
00474A E3DC 8E   4010    A BPCL   LDX   #BP1     Y REG CONTAINS
00475A E3DF 10BE FF1A    A       LDY   INRY     BREAK POINT
00476A E3E3 8C   4020    A BPC1   CMPX  #PGRI
00477A E3E6 27   10   E3F8       BEQ   BPC2
00478A E3E8 10AC 81      A       CPY   ,X++
00479A E3EB 26   F6   E3E3       BNE   BPC1
00480A E3ED 108E 0000    A       LDY   #$00
00481A E3F1 10AF 63      A       STY   ,--X
00482A E3F4 10BF FF1A    A       STY   INRY
00483A E3F8 7E   F5C2    A BPC2   JMP   XBUG
00484
00485
00486                    ***********************************
00487                    *     BREAK POINT ALL CLEAR INST *
00488                    ***********************************
00489
00490A E3FB 8E   4010    A BPAC   LDX   #BP1
00491A E3FE 108E 0000    A       LDY   #$00
00492A E402 10AF 81      A BPA1   STY   ,X++
00493A E405 8C   4020    A       CMPX  #PGRI
00494A E408 26   F8   E402       BNE   BPA1
00495A E40A 10BF FF1A    A       STY   INRY
00496A E40E 7E   F5C2    A       JMP   XBUG
00497
00498
00499                    *********************************
00500                    *     BREAK POINT DISPLAY INST *
00501                    *********************************
00502
00503A E411 108E 4010    A BPDS   LDY   #BP1
00504A E415 BD   FD21    A       JSR   XPCRLF
00505A E418 30   A1      A BPD1   LEAX  ,Y++
00506A E41A BD   F01E    A       JSR   XOUT4H
00507A E41D 108C 4020    A       CMPY  #PGRI
00508A E421 26   F5   E418       BNE   BPD1
00509A E423 7E   F5C2    A       JMP   XBUG
00510
00511
00512
00513                    ************************
```

```
00514                    *     INSTRUCTION TABLE *
00515                    ************************
00516
00517A E140                        ORG   $E140
00518A E140     42      A CTBEG    FCC   /BEGN/
00519A E144     E000    A          FDB   BEGN
00520A E146     43      A          FCC   /CONP/
00521A E14A     E2BA    A          FDB   CONP
00522A E14C     43      A          FCC   /CHNG/
00523A E150     E36E    A          FDB   CHNG
00524A E152     42      A          FCC   /BPST/
00525A E156     E3BD    A          FDB   BPST
00526A E158     42      A          FCC   /BPCL/
00527A E15C     E3DC    A          FDB   BPCL
00528A E15E     42      A          FCC   /BPAC/
00529A E162     E3FB    A          FDB   BPAC
00530A E164     42      A          FCC   /BPDS/
00531A E168     E411    A          FDB   BPDS
00532           E169    A CTEND    EQU   $E169
00533
00534
00535                              END
TOTAL ERRORS 00000--00000
TOTAL WARNINGS 00000--00000
```

```
FF1F AAR    00014*00329 00366
FF20 BBR    00015*00332 00369
E061 BEG1   00109*00112
E070 BEG2   00114*00118
E0EA BEG3   00166*00418
E053 BEG4   00104*00106
E041 BEG5   00098*00102
E01D BEG6   00084 00086*
E000 BEGN   00074*00519
E4AF BOOT   00054*00114
400F BP0    00056*00104
4010 BP1    00057*00299 00397 00458 00474 00490 00503
E402 BPA1   00492*00494
E3FB BPAC   00490*00529
E3E3 BPC1   00476*00479
E3F8 BPC2   00477 00483*
E3DC BPCL   00474*00527
E418 BPD1   00505*00508
E411 BPDS   00503*00531
E3C0 BPS1   00459*00462
E3D9 BPS2   00460 00467*
E3BD BPST   00458*00525
FF21 CCR    00016*
1F0C CHG    00030*00426
000C CHG1   00031*00424
1F0A CHI0   00020*00243 00267
E36E CHNG   00424*00523
1FEF CINH   00028*00438 00447
1FF0 CINL   00029*00440 00449
E2EF CON1   00373*00378
E353 CON2   00409*00451
E34B CON3   00399 00401 00406*
E337 CON4   00398*00408
E2BA CONP   00354*00521
2020 CR13   00033*00132 00138 00162 00238 00416
2021 CR2    00032*00129 00135 00233
E140 CTBEG  00093 00518*
```

```
E169 CTEND  00095 00532*
0004 DDA    00050*00285
0005 DDB    00051*00289
FF1E DPR    00013*
FFF8 INRH   00046*00092
FFF9 INRL   00047*
FF1C INRX   00012*00326 00363
FF1A INRY   00011*00463 00466 00475 00482 00495
E1AB INT1   00249*00265
E29C INT10  00337*00344
E1D2 INT2   00254 00264*
E1D8 INT3   00246 00267*
E1EE INT4   00263 00266 00275*
E2B7 INT5   00236 00346*
E257 INT6   00311*
E23A INT7   00300*00306
E244 INT8   00301 00304*
E2B4 INT9   00335 00345*
1FFA INTH   00024*00379 00390 00433 00444
1FFB INTL   00025*00381 00388 00435 00442
E180 INTR   00232*
E180 INTRR  00055*00091 00230
2023 LAL1   00035*00144
2025 LAL2   00037*00150
2027 LAL3   00039*00156
2022 LAT1   00034*00141
2024 LAT2   00036*00147
2026 LAT3   00039*00153
2018 LED    00042*00075 00165 00175 00181
4003 LEDH   00043*00171 00173 00179
1FF5 LIE    00040*00089 00410
E12A LLD1   00202 00203*
E13A LLD2   00220 00221*
E0EC MAN1   00167*00185
E108 MAN2   00172 00178*
E114 MAN3   00170 00177 00183*
E117 MAN4   00168 00184*
2008 MCHG   00019*00078 00159 00241 00413
2010 MEM    00041*00086
FFF8 MEMM   00008*00080
4000 PCON   00018*00247
FF16 PCOR   00009*00322 00333 00359 00400
4020 PGRI   00056*00385 00407 00459 00476 00493 00507
0000 PORTA  00048*00277
0001 PORTB  00049*00281
1FF1 RINH   00026*00384 00393
1FF2 RINL   00027*00386 00395
E125 SLDA   00081 00099 00120 00201*00234 00244 00252 00271 00276
            00280 00284 00288 00292 00296 00314 00320 00328 00331
            00355 00374 00380 00385 00402 00430 00434 00439
E135 SLDB   00122 00219*00250 00308 00312 00318 00324 00357 00382
            00387 00425 00436 00441
1FF3 SPH    00021*00256 00313 00354
1FF4 SPL    00022*00258 00270 00273 00307 00311 00356
FF22 SSP    00017*
E122 SST1   00193 00194*
E132 SST2   00211 00212*
E11D SSTA   00176 00179 00087 00090 00110 00116 00130 00133 00136
            00139 00142 00145 00148 00151 00154 00157 00160 00163
            00176 00182 00192*00242 00257 00262 00269 00274 00278
            00282 00286 00290 00294 00298 00303 00339 00362 00368
            00371 00376 00391 00394 00405 00411 00414 00417 00432
            00445 00448
E12D SSTB   00210*00239 00259 00310 00341 00360 00365 00389 00396
```

```
                00427 00443 00450
  FF0E TABEG    00062*00094
  FF10 TAEND    00063*00096
  0009 TCR      00153*00297
  0008 TDR      00152*00293
  4001 TIM1     00044*00167 00184
  4002 TIM2     00045*00169 00183
  FF18 USP      00010*00316
  4030 VECT     00061*00108 00123 00337
  F5C2 XBUG     00023*00345 00467 00483 00496 00509
  F01E XOUT4H   00059*00506
  F021 XPCRLF   00060*00504
```

B I B L I O G R A P H Y

BIBLIOGRAPHY.

BOOKS.

Brey, B. B. 1984. Microprocessor/Hardware Interfacing and Applications. Columbus, Charles E. Merill Publishing Company.

Strangio, C. E. 1980. Digital Electronics: Fundamental Concepts and Applications. New Jersey, Prentice-Hall Publishers.

JOURNALS.

Dum, S. & Philpott, D. 1978. In-Circuit Emulation Aids Microprocessor System Design. Electronic Engineering, November No.615, pages 87-90.

Ferguson, J. D. 1984. In-Circuit Emulation. Electronics & Wireless World, June, pages 53-56.

Koehler, B. 1980. 8051 Single-Chip Microcomputer Architechiral Specification and Functional Description. Intel Corporation, Preliminary.

Ulmer, D. J. 1979. Architecture Based on Multiple MCPs, Buses Boosts In-Circuit Emulation. Electronic Design, April 26, No.9, pages 52-55.

Wharton, J. 1980. An Introduction to Intel MCS-51 Single-Chip Microcomputer Family. Intel Corporation. AP69.

Wharton, J. 1980. Using the Intel MCS-51 Boolean Processing Capabilities. Intel Corporation. AP70.

MANUALS.

Hewlett-Packard. 1979. Owner's Manual and Programming Guide. USA. Hewlett-Packard Company.

Hewlett-Packard. 1982. HP82939A Serial Interface Owner's Manual Series 80. USA. Hewlett-Packard Company.

Hewlett-Packard. 1983. I/O Rom Owner's Manual Series 80. USA. Hewlett-Packard Company.

Motorola. 1979. M6809 EXORterm Development System User's Guide. Motorola Inc.

DATA BOOKS.

General Instruments. 1980 Catalog of Optoelectronic Products. California. General instruments. pages 177-180.

Intel. 1986. **Memory Components Handbook.** Santa Clara. Intel Corporation. page 4-21 - 4-34.

Motorola Semiconductors. 1982. **Microprocessor Data Manual.** Switzerland. Motorola Inc. pages 4-406; 4-985; 4-1046.

Motorola Semiconductors. 1983. **MC1468705G2 8 bit Eprom Microcomputer.** Motorola Inc. AD1976.

National Semiconductor. 1984. Logic Data book. Volume 1 and 2. California. National Semiconductor Corp.

A N N E X U R E 1

L O A D R E C O R D E R
D I S P L A Y D O C U M E N T A T I O N

OPTREX

OPTREX's Liquid Crystal Dot Matrix Display Module DMC Series can easily be connected to a micro computer by using LSI s which contain sophisticated control circuits, character generators, etc. DMC Series is most suitable for use with micro computer peripheral, word processor, POS terminal and telecommunication systems etc.

APPLICATION

- Personal Computer
- Electronic Typewriter
- Word Processor
- Facsimile
- Plain Paper Copier
- Telecommunication Systems
- Instrument Devices
- POS Terminal
- Other Peripherals

Item / Model No.	Display Format	Display Fonts	Module Size WxHxT(mm)	View Area WxH(mm)	Character Size WxH(mm)
DMC 16106A	16 characters 1 line	5x10+Cursor	80×36×10	64.5×13	3.2×8.2 (3.2×5.95)
DMC 16106C	16 characters 1 line	5x7+Cursor	80×36×10	64.5×13	3.2×5.95
DMC 16117	16 characters 1 line	5x7+Cursor	80×36×10	64.5×13	3.2×5.95
DMC 16106B	16 characters 1 line	5x7+Cursor	130×44×10	99×13	4.9×9.5
DMC 16207	16 characters 2 lines	5x7+Cursor	84×44×11	61×16	2.95×5.55
DMC 20215	20 characters 2 lines	5x7+Cursor	116×44×11	83×18.6	3.2×5.55
DMC 32216	32 characters 2 lines	5x7+Cursor	175×44×11	130×18.6	3.2×5.55
DMC 40209	40 characters 2 lines	5x7+Cursor	220×53×10	155×19	2.9×6.05
DMC 40218	40 characters 2 lines	5x7+Cursor	182×33.5×11	152.5×16.5	3.2×5.55

FEATURES

(1) 5 x 7 dots cursor display (DMC16106A available for use with 5 x 10 dots and cursor display.)
(2) 8-bit or 4-bit MPU interface is available.
(3) 160 JIS type characters such as Alphabet, Numeral and Kana and 32 special characters and symbols can be displayed by internal character generator (ROM).
(4) Random symbols can be displayed by internal character generator (RAM).
(5) Many instructional functions by means of program such as "clear display", "home cursor", "on/off cursor", "blink character", "shift display", "shift cursor", "read/write display data", etc. are available.
(6) Compact and light weight design enable easy assembly on device.
(7) Single "+5V" power supply.
(8) Low power consumption
(9) DMC Series is most suitable for use with micro computer peripheral, word processor, POS terminal and telecommunication systems etc.

Dot Size WxH(mm)	Preferred		Operating Method	Power Consumption TYP (mW)	Weight (g)	Operating Temperature	Storage Temperature
	V_{DD}-V_{SS}	V_{EE}-V_{SS}					
0.6×0.7	+5	—	1/11D−1/4B	10	30	0 ~ 50°C	−20°C ~ +70°C
0.6×0.7	+5	—	1/8D−1/4B	10	30	〃	〃
0.6×0.7	+5	—	1/11D−1/5B	10	30	〃	〃
0.9×1.05	+5	—	1/8D−1/4B	10	55	〃	〃
0.55×0.65	+5	—	1/16D−1/5B	12	45	〃	〃
0.6×0.65	+5	—	1/16D−1/5B	12	45	〃	〃
0.6×0.65	+5	—	1/16D−1/5B	15	75	〃	〃
0.52×0.6	+5	—	1/16D−1/5B	15	90	〃	〃
0.6×0.65	+5	—	1/16D−1/5B	15	75	〃	〃

ELECTRICAL CHARACTERISTICS

Item	Symbol	Test Condition	Standard Value			Unit
			min.	typ.	max.	
Input "High" Voltage	V_{IH}	—	2.2	—	Vcc	V
Input "Low" Voltage	V_{IL}	—	-0.3	—	0.6	V
Output "High" Voltage	V_{OH}	$-I_{OH}$=0.205mA	2.4	—	—	V
Output "Low" Volatage	V_{OL}	I_{OL}=1.2mA	—	—	0.4	V
Power Supply Current	Icc	Vcc=5.0V	—	0.5	5	mA

※Vcc=5.0V ± 5 %, Ta=25°C

ABSOLUTE MAXIMUM RATINGS

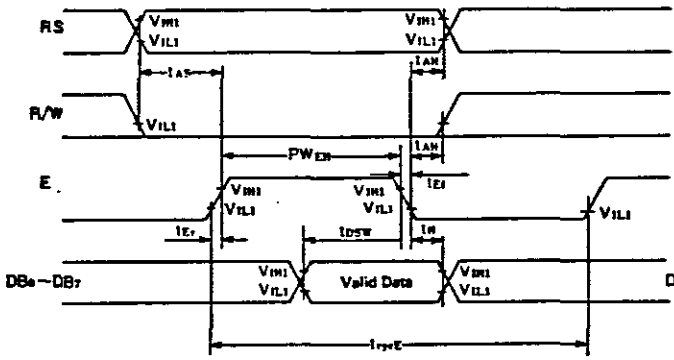
Item	Symbol	Test Condition	Standard Value			Unit
			min.	typ.	max.	
Power Supply Voltage for Logic	Vcc −Vss	—	0	—	7	V
Power Supply Voltage for LCD Drive	Vcc −Vee	—	0	—	13.5	V
Imput Voltage	V_I	—	Vss	—	Vcc	V
Operating Temperature	Ta		0	—	+50	°C
Storage Temperature	Tstg		−20	—	+70	°C

TIMING CHART

Item	Symbol	Measuring Condition	Standard Value			Unit
			min.	typ.	max.	
Enable Cycle Time	T_{CYCE}	See Figs. 1 and 2	1000	—	—	nS
Enable Pulse Width, High Level	PW_{EH}	See Figs. 1 and 2	450	—	—	nS
Enable Rise and Decay Time	t_{Er},t_{Ef}	See Figs. 1 and 2	—	—	25	nS
Address Setup Time, RS, R/W–E	t_{AS}	See Figs. 1 and 2	140	—	—	nS
Data Delay Time	t_{DDR}	See Figs. 2	—	—	320	nS
Data Setup Time	t_{DSW}	See Figs. 1	195	—	—	nS
Data Hold Time	t_H	See Figs. 1	10	—	—	nS
Data Hold Time	t_{DHR}	See Figs. 2	20	—	—	nS
Address Hold Time	t_{AH}	See Figs. 1 and 2	10	—	—	nS

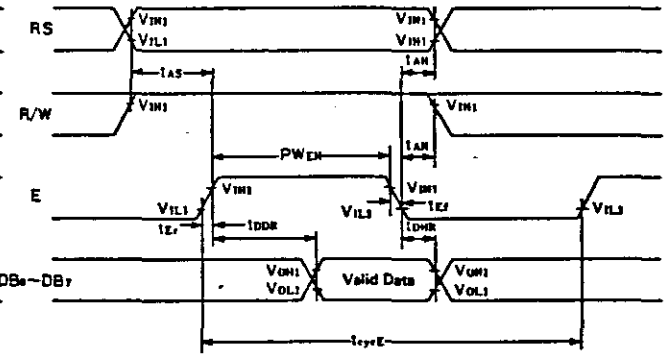
※Vcc=5.0V ± 5 %, Ta=25°C

● FIG. 1 WRITE OPERATION



(Write Data from MPU to MODULE)

● FIG. 2 READ OPERATION



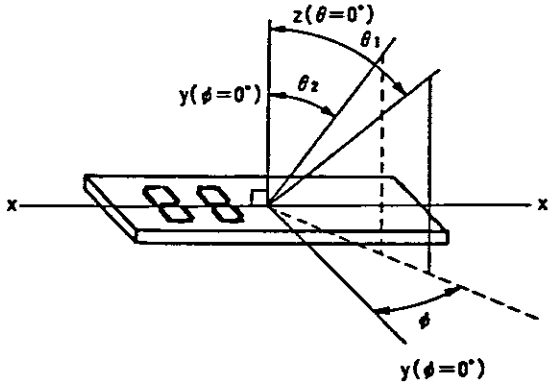
(Reading Data from MODULE to MPU)

OPTICAL CHARACTERISTICS

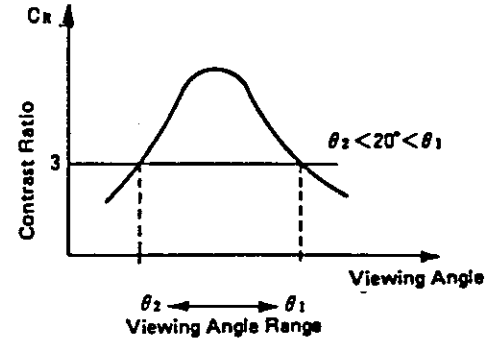
Item	Symbol	Test Condition	Standard Value			Unit
			min.	typ.	max.	
Liquid Crystal Drive Voltage (Recommended value) 1/8 duty	Vcc −Vee (V_D)	Ta = 0°C	4.0	—	—	V
		Ta = 25°C	—	4.1	—	V
		Ta = 50°C	—	—	4.2	V
Visual Angle Range	$\theta_1 - \theta_2$	$C_R = 3$	30	—	—	deg.
Contrast Ratio	C_R	$\theta_2 = 20°$, $\phi = 0°$	10	—	—	
Rise Time	τr	$V_D = 4.1V$, $\theta = 20°$	—	100	200	mS
Decay Time	τd	$V_D = 4.1V$, $\theta = 20°$	—	100	200	mS
Liquid Crystal Drive Voltage (Recommended Value) 1/11 duty	Vcc −Vee (V_D)	Ta = 0°C	4.4	—	—	V
		Ta = 25°C	—	4.5	—	V
		Ta = 50°C	—	—	4.6	V
Visual Angle Range	$\theta_1 - \theta_2$	$C_R = 3$	25	—	—	deg.
Contrast Ratio	C_R	$\theta_2 = 20°$, $\phi = 0°$	8	—	—	
Rise Time	τr	$V_D = 4.5V$, $\theta = 20°$	—	120	240	mS
Decay Time	τd	$V_D = 4.5V$, $\theta = 20°$	—	120	240	mS
Liquid Crystal Drive Voltage (Recommended Value) 1/16 duty	Vcc −Vee (V_D)	Ta = 0°C	4.7	4.8	4.9	V
		Ta = 25°C	4.4	4.5	4.6	V
		Ta = 50°C	4.1	4.2	4.3	V
Visual Angle Range	$\theta_1 - \theta_2$	$C_R = 3$	20	—	—	deg.
Contrast Ratio	C_R	$\theta_2 = 20°$, $\phi = 0°$	6	—	—	
Rise Time	τr	$V_D = 4.5V$, $\theta = 20°$	—	120	240	mS
Decay Time	τd	$V_D = 4.5V$, $\theta = 20°$	—	120	240	mS

※Vcc = 5.0V ± 5%, Ta = 25℃

Ⓐ Definition of Viewing Angle θ and ϕ

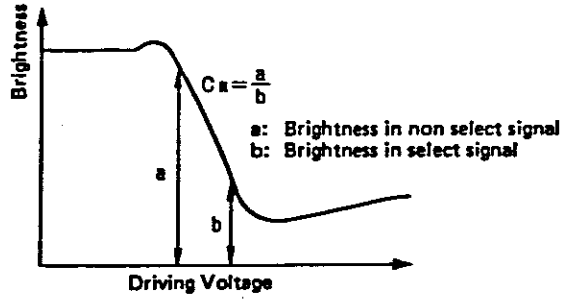


Ⓑ Definition of Viewing Angles θ_2 and θ_1

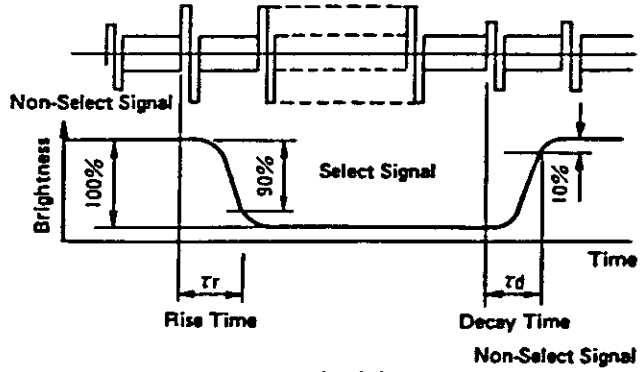


Ⓒ Definition of Contrast Ratio CR

Viewing Direction



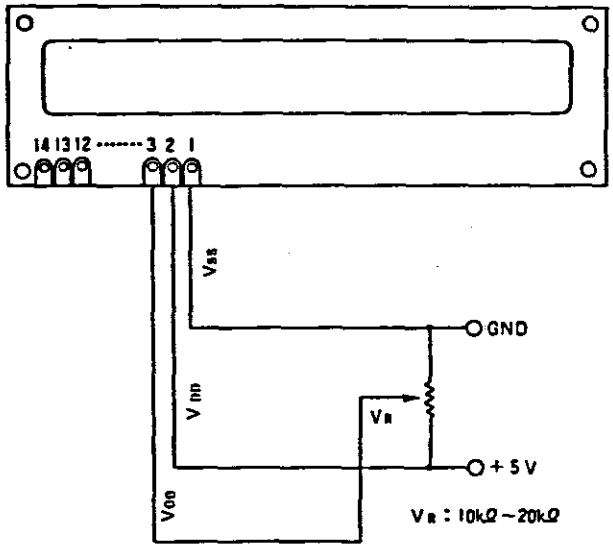
Ⓓ Definition of Optical Response Time



Those time that the brightness of lighting segment reaches 90% from 0% is τr and that reaches 10% from 100% is τd.

EXAMPLE OF POWER SUPPLY

DMC Module

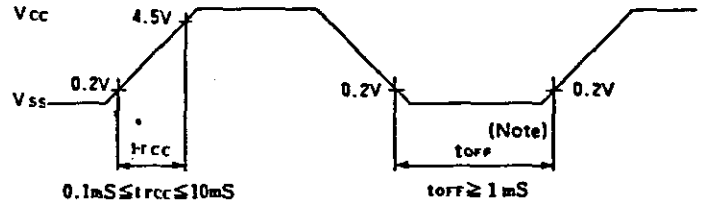


■ NOTE: When the voltage of Vee is different from the recommended voltage, the viewing angle may be changed.

POWER SUPPLY RESET

The internal reset circuit will not be correctly operated, when the following power supply condition is not satisfied. In this case, please perform initial setting according to the instruction.

Item	Symbol	Measuring Condition	Standard Value min.	typ.	max.	Unit
Power Supply Rise Time	trcc	——	0.1	—	10	mS
Power Supply OFF Time	tOFF	——	1	—	—	mS



Note: The item t OFF defines the time when the power supply is off, when the power supply shuts down mementarily or repeats on-off state.

PIN ASSIGNMENT

Pin No.	Symbol	Level	Function
1	Vss	——	Power Supply Voltage 0V (GND)
2	VDD	——	Power Supply Voltage +5V
3	Vee	——	Power Supply for Liquid Crystal Drive
4	RS	H/L	Register Select H: Data Input L: Instruction Input
5	R/W	H/L	H: Data Read (Module → MPU) L: Data Write (Module ← MPU)
6	E	H, H→L	Enable Signal
7	DB ϕ	H/L	Data Bus Line
8	DB 1	H/L	〃
9	DB 2	H/L	〃
10	DB 3	H/L	〃
11	DB 4	H/L	〃
12	DB 5	H/L	〃
13	DB 6	H/L	〃
14	DB 7	H/L	〃

■ NOTE: In the data bus line, data transfer is performed two times by the 4-bit or one time by the 8-bit in order to interface with 4-bit or 8-bit MPU.

* In case interface data length is 4-bit

The data is transferred by using only four buses of DB4 — DB7 and the buses of DBϕ — DB3 are not used. The data transfer to MPU is completed by transferring the data of 4 bits twice.

Transfer of upper four bits and low four bits is performed in sequence.

* in case interface data lentgth is 8 bit,

Data transfer is performed by using eight buses of DB0 — DB7.

INSTRUCTIONS

Instruction	Code										Description	Execute Time (max.)
	RS	R/W	DB 7	DB 6	DB 5	DB 4	DB 3	DB 2	DB 1	DB 0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears all display and returns the cursor to the home position (Address 0).	1.64mS
Cursor At Home	0	0	0	0	0	0	0	0	1	*	Returns the cursor to the home position (Address 0). Also returns the display being shifted to the original position. DD RAM contents remain unchanged.	1.64mS
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets the cursor move direction and specifies or not to shift the display. These operations are performed during data write and read.	40μS
Display On/Off Control	0	0	0	0	0	0	1	D	C	B	Sets ON/OFF of all display (D) cursor ON/OFF (C), and blink of cursor position character (B).	40μS
Cursor/Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Moves the cursor and shifts the display without changing DD RAM contents.	40μS
Function Set	0	0	0	0	1	DL	N	F	*	*	Sets interface data length (DL) number of display lines (L) and character font (F).	40μS
CG RAM Address Set	0	0	0	1	ACG						Sets the CG RAM address. CG RAM data is sent and received after this setting.	40μS
DD RAM Address Set	0	0	1	ADD							Sets the DD RAM address. DD RAM data is sent and received after this setting	40μS
Busy Flag/ Address Read	0	1	BF	AC							Reads Busy flag (BF) indicating internal operation is being performed and reads address counter contents.	40μS
CG RAM/DD RAM Data Write	1	0	WRITE DATA								Writes data into DD RAM or CG RAM.	40μS
CG RAM/DD RAM Data Read	1	1	READ DATA								Reads data from DD RAM or CG RAM.	40μS

■ NOTE:

Code	Description	Execute Time (max.)
I/D = 1: Increment I/D = 0: Decrement S = 1: With display shift S/C = 1: Display shift S/C = 0: Cursor movement R/L = 1: Shift to the right R/L = 0: Shift to the left DL = 1: 8-bit DL = 0: 4-bit N = 1: 2 lines N = 0: 1 line F = 1: 5 x 10 dots F = 0: 5 x 7 dots BF = 1: Internal operation is being performed. BF = 0: Instruction acceptable.	DD RAM: Display Data RAM CG RAM: Character Generator RAM ACG: CG RAM Address ADD: DD RAM Address Corresponds to cursor address. AC: Address Counter, used for both DD RAM and CG RAM * : Invalid DON'T CARE	fcp or fosc = 250 kHz However, when frequency changes, execution time also changes. Ex. When fcp or fosc = 270 kHz, $40\mu S \times \frac{250}{270} = 37\mu S$

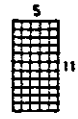
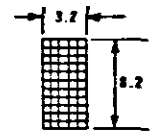
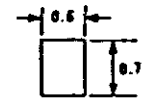
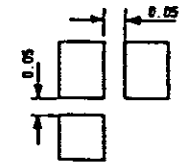
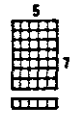
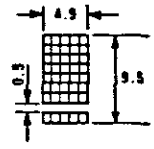
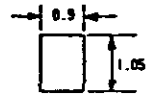
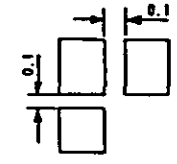
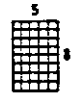
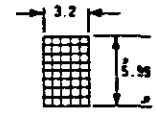
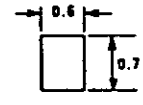
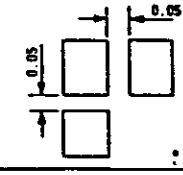
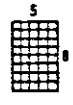
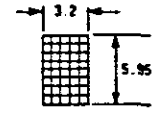
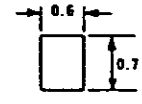
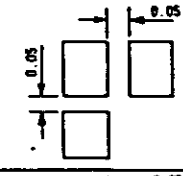
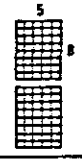
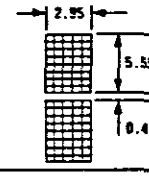
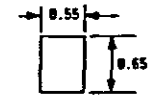
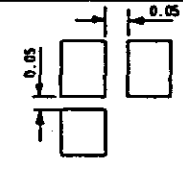
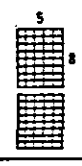
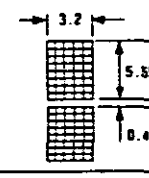
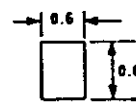
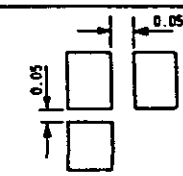
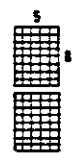
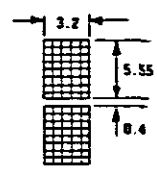
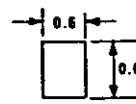
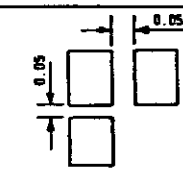
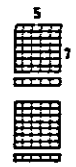
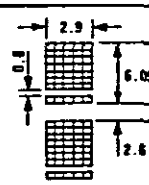
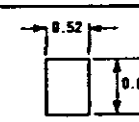
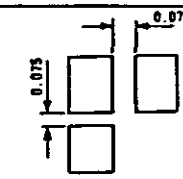
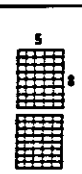
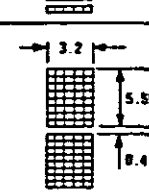
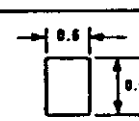
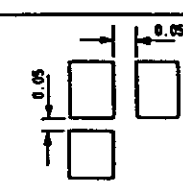
■ NOTE: For details in program, refer to the User's Manual which is separately provided.

CHARACTER FONT TABLE (Correspondence between Characters Codes and Character Pattern)

Lower 4 bit \ Upper 4 bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
××××0000	CG RAM (1)												
××××0001	(2)												
××××0010	(3)												
××××0011	(4)												
××××0100	(5)												
××××0101	(6)												
××××0110	(7)												
××××0111	(8)												
××××1000	(1)												
××××1001	(2)												
××××1010	(3)												
××××1011	(4)												
××××1100	(5)												
××××1101	(6)												
××××1110	(7)												
××××1111	(8)												

* CG RAM: Character pattern area which can freely be rewritten by program.

DISPLAY DOT PATTERN

Model Number	Character Configuration	Character Size mm	Dot Size mm	Dot Pitch mm
DMC16106A (16 characters 1 line)				
DMC16106B (16 characters 1 line)				
DMC16106C (16 characters 1 line)				
DMC16117 (16 characters 1 line)				
DMC16207 (16 characters 2 lines)				
DMC20215 (20 characters 2 lines)				
DMC32216 (32 characters 2 lines)				
DMC40209 (40 characters 2 lines)				
DMC40218 (40 characters 2 lines)				

A N N E X U R E 2

6 8 Ø 9 E X O R T E R M D E V E L O P M E N T
S Y S T E M U S E R S G U I D E

5.9.3.1 Adding EXbug Commands. The user has the ability to add as many four-character commands as desired. The only limiting factor is memory size. In order to implement this feature, the user must have a table of his commands and the actual commands stored in the Executive memory map (if the Dual Map mode is in use), and must have told EXbug where his command table resides. The user command table format must be as follows:

Example:

```
CTBEG EQU*       Command table beginning
      FCC/CMD1/  Four character command
      FDB CMD1E  Entry address of command
      FCC/CMD2/  Four character command
      FDB CMD2E  Entry address of command
       .  .       .
       .  .       .
       .  .       .
       .  .       .
      FCC/CMDN/  Four character command
      FDB CMDNE  Entry address of command
CTBENDEQU*       Command table end
```

Once the user command table is stored in memory, EXbug must be informed of its location by having the beginning address of the table (the value of CTBEG in the above example) put at locations $FF0E and $FF0F, while the ending address of the table (the value of CTBEND in the above example) is put at locations $FF10 and $FF11. In both of these cases, the addresses are loaded into memory in the order of most significant byte first, followed by the least significant byte. If the command table and commands are loaded from a tape, the tape may contain an object code that will properly initialize these locations. This object code may be generated by the ORG and FDB statements in the source. For the above example, the source code required to generate the proper object code to initialize these locations would be:

```
Example:  ORG $FF0E
          FDB CTBEG, CTBEND
```

NOTE: An ORG statement or END statement would be required after the two source lines shown above, so that the object code would not be produced at location $FF12 or beyond.

If the command table and commands are loaded from an MDOS file, a short program can be included in the file that would initialize these locations and then give control to EXbug when it is executed. Programs cannot be loaded from the disk at these locations.

Pressing the ABORT pushbutton will not modify locations $FF0E through $FF11. However, pressing the RESTART pushbutton will cause these locations to be restored to the EXbug values. These locations will also be restored to the EXbug values when power is initially applied. Thus, following a restart, the user must restore the beginning and ending addresses of his table (if required) in memory locations $FF0E through $FF11. If the user does not wish to add commands, no operation is needed.

On entry to the user command, the stack pointer will be pointing at two locations below the top of the EXbug stack area; the A accumulator will contain $20; and the contents of the B accumulator and X, Y, U, CC, and DP registers will be unspecified. None of these values need be restored before returning to EXbug. However, IRQ and FIRQ are normally made while EXbug is running. User commands that are intended to return to EXbug without affecting the current states of EXbug variables, should return by jumping to location $F5C2. User programs that are intended to return to EXbug and initialize EXbug variables, should return by jumping to location $F564.

```
00618                          *
00619                          * EXBUG POWER-UP INITIALIZATION
00620                          *
00621           F2AA    A START  EQU   *
00622A F2AA 10CE FFE5  A        LDS   #XSTACK  INIT SP
00623                          * INITIALIZE RAM
00624A F2AE CE   FF02  A        LDU   #ATOP+2
00625A F2B1 8E   83FF  A        LDX   #TOPTGT
00626A F2B4 AF   5E    A        STX   -2,U     ATOP,U
00627A F2B6 8E   FB61  A        LDX   #CRCMD
00628A F2B9 AF   4C    A        STX   $C,U     CMDBEG-ATOP,U
00629A F2BB 30   06    A        LEAX  6,X
00630A F2BD AF   4E    A        STX   $E,U     CMDEND-ATOP,U
00631A F2BF 4F                  CLRA
00632A F2C0 5F                  CLRB
00633A F2C1 FD   FFE6  A        STD   Q
00634A F2C4 ED   C4    A        STD   0,U
00635                          * INITIALIZE HALT ON ADDRESS PIA
00636A F2C6 CE   FCF4  A        LDU   #ACIASC  PIA BASE ADDRESS
00637A F2C9 CC   FF3C  A        LDD   #$FF3C   A DATA TO OUTPUT
00638A F2CC ED   44    A        STD   4,U      A DIR+CNTL
00639A F2CE C6   04    A        LDB   #$4      B DATA TO OUTPUT
00640A F2D0 ED   46    A        STD   6,U      B DIR+CNTL
00641                          * INITIALIZE MAP CONTROL PIA
00642A F2D2 CC   7F7F  A        LDD   #$7F7F   A+B DATA OUTPUT
00643A F2D5 ED   48    A        STD   8,U      A+B DIR
00644A F2D7 86   2C    A        LDA   #$2C     A CNTL
00645A F2D9 ED   4A    A        STD   10,U     A+B CNTL
00646                          * COME UP IN EXEC MAP
00647A F2DB 17   0824 FB02      LBSR  EXEC
00648                          * INITIALIZE ACIA
00649A F2DE C6   03    A        LDB   #$3      RESET ACIA
00650A F2E0 E7   C4    A        STB   0,U
00651A F2E2 C6   31    A        LDB   #$31     ASSUME 2 STOP BITS
00652A F2E4 E7   C4    A        STB   0,U
00653A F2E6 8D   96  F27E       BSR   READR6   SEND 2 CHARACTERS
00654A F2E8 8D   94  F27E       BSR   READR6
00655A F2EA 1F   31    A        TFR   U,X
00656A F2EC 30   1F    A INACIA LEAX  -1,X     COUNT CHAR SEND TIME
00657A F2EE A6   C4    A        LDA   0,U      ACIASC
00658A F2F0 85   02    A        BITA  #2       CHAR SENT?
00659A F2F2 27   F8  F2EC       BEQ   INACIA   NO
00660A F2F4 8C   E700  A        CPX   #$E700   REALLY 2 STOP BITS?
00661A F2F7 2B   04  F2FD       BMI   IACIA1   YES
00662A F2F9 C6   35    A        LDB   #$35
00663A F2FB E7   C4    A        STB   0,U      ACIASC
00664A F2FD CA   40    A IACIA1 ORAB  #$40     SAVE IN SBIT
00665A F2FF E7   49    A        STB   9,U      SBIT
00666A F301 7F   FF1E  A        CLR   PRDPR    SET PSEUDO DPR=0
00667A F304 86   50    A        LDA   #$50     SET PSEUDO I, F MASKS
00668A F306 B7   FF21  A        STA   PRCC

00670                          * GET HERE FROM ENTERING AT $F564
00671           F309    A RENTER EQU   *
00672A F309 10CE FFE5  A        LDS   #XSTACK  INIT SP
00673A F30D 8E   FF90  A        LDX   #STACK   INIT PSEUDO SP
00674A F310 BF   FF22  A        STX   SPSAVE
```

A N N E X U R E 3

D A T A S H E E T O N T H E
P R O G R A M M A B L E T I M E R
M O D U L E M C 6 8 4 Ø

MOTOROLA

MC6840
(1.0 MHz)
MC68A40
(1.5 MHz)
MC68B40
(2.0 MHz)

PROGRAMMABLE TIMER MODULE (PTM)

The MC6840 is a programmable subsystem component of the M6800 family designed to provide variable system time intervals.

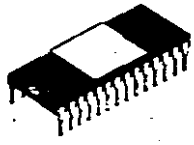
The MC6840 has three 16-bit binary counters, three corresponding control registers, and a status register. These counters are under software control and may be used to cause system interrupts and/or generate output signals. The MC6840 may be utilized for such tasks as frequency measurements, event counting, interval measuring, and similar tasks. The device may be used for square wave generation, gated delay signals, single pulses of controlled duration, and pulse width modulation as well as system interrupts.

- Operates from a Single 5 Volt Power Supply
- Fully TTL Compatible
- Single System Clock Required (Enable)
- Selectable Prescaler on Timer 3 Capable of 4 MHz for the MC6840, 6 MHz for the MC68A40 and 8 MHz for the MC68B40
- Programmable Interrupts ($\overline{IRQ}$) Output to MPU
- Readable Down Counter Indicates Counts to Go Until Time-Out
- Selectable Gating for Frequency or Pulse-Width Comparison
- $\overline{RESET}$ Input
- Three Asynchronous External Clock and Gate/Trigger Inputs Internally Synchronized
- Three Maskable Outputs

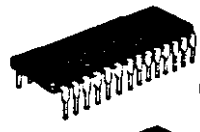
MOS

(N-CHANNEL, SILICON-GATE DEPLETION LOAD)

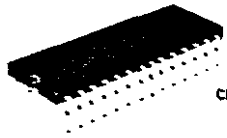
PROGRAMMABLE TIMER



L SUFFIX
CERAMIC PACKAGE
CASE 719



P SUFFIX
PLASTIC PACKAGE
CASE 710



S SUFFIX
CERDIP PACKAGE
CASE 733

MAXIMUM RATINGS

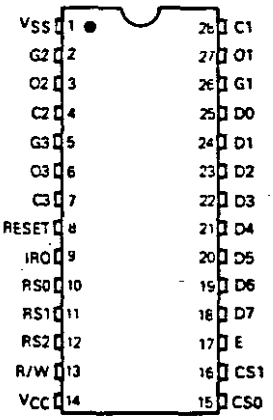
Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range — T_L to T_h MC6840, MC68A40, MC68B40 MC6840C, MC68A40C	T_A	0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Cerdip Plastic Ceramic	θ_{JA}	 65 115 60	°C/W

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

FIGURE 1 — PIN ASSIGNMENT



A N N E X U R E 4

D A T A I N F O R M A T I O N O N T H E
C M O S M I C R O M P U T E R
M C 1 4 6 8 Ø 5 E 2

MOTOROLA

MC146805E2

Advance Information

CMOS
(HIGH PERFORMANCE SILICON GATE)

8-BIT MICROPROCESSOR

8-BIT MICROPROCESSOR UNIT

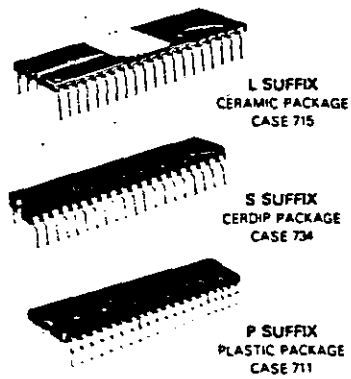
The MC146805E2 Microprocessor Unit (MPU) belongs to the M6805 Family of microcomputers. This 8-bit fully static and expandable microprocessor contains a CPU, on-chip RAM, I/O, and timer. It is a low-power, low cost processor designed for low-end to mid-range applications in the consumer, automotive, industrial, and communications markets where very low power consumption constitutes an important factor. The following are the major features of the MC146805E2 MPU:

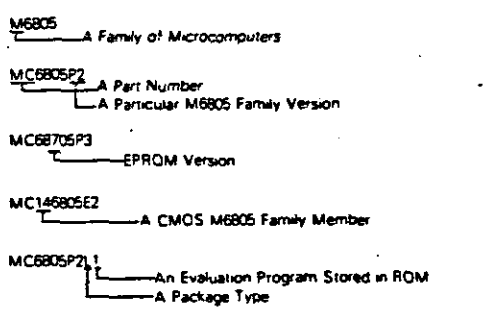
Hardware Features:

- Typical Full Speed Operating Power of 35 mW @ 5V
- Typical WAIT Mode Power of 5 mW
- Typical STOP Mode Power of 25 µW
- 112 Bytes of On-Chip RAM
- 16 Bidirectional I/O Lines
- Internal 8-Bit Timer with Software Programmable 7-Bit Prescaler
- External Timer Input
- Full External and Timer Interrupts
- Multiplexed Address/Data Bus
- Master Reset and Power-On Reset
- Capable of Addressing Up to 8k Bytes of External Memory
- Single 3 to 6 Volt Supply
- On-Chip Oscillator
- 40-Pin Dual-In-Line Package
- Chip-Carrier Also Available

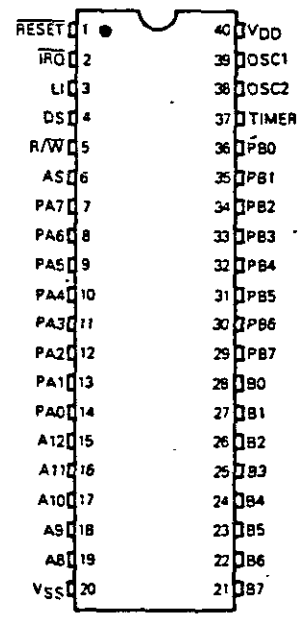
Software Features:

- Similar to the MC6800
- Efficient Use of Program Space
- Versatile Interrupt Handling
- True Bit Manipulation
- Addressing Modes With Indexed Addressing for Tables
- Efficient Instruction Set
- Memory Mapped I/O
- Two Power Saving Standby Modes





PIN ASSIGNMENTS



MAXIMUM RATINGS (voltages referenced to V_{SS})

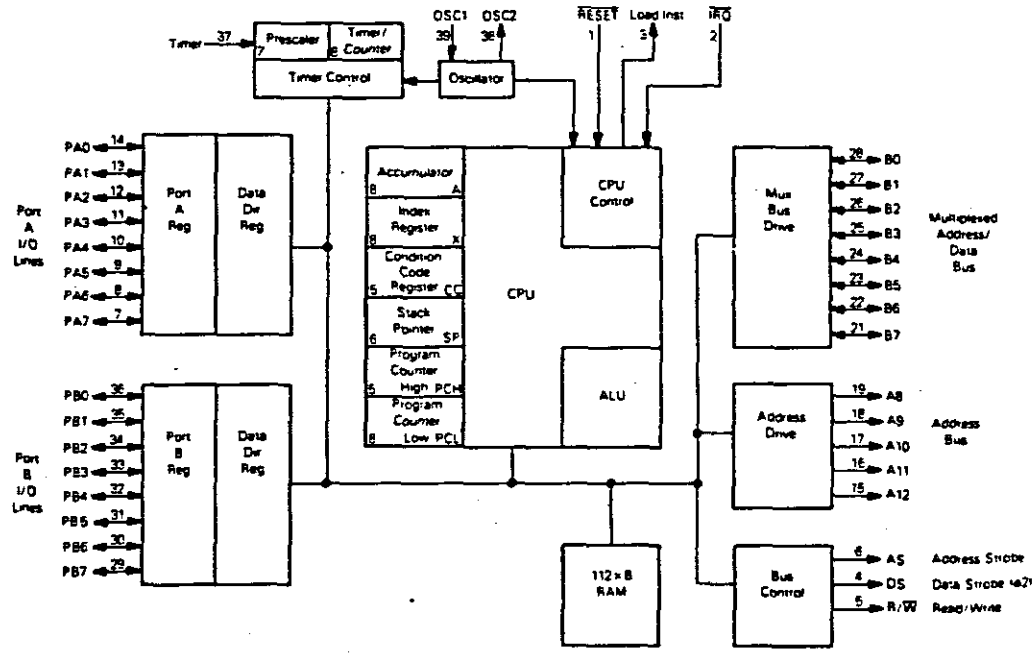
Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8.0	V
All Input Voltages Except OSC1	V_{in}	V_{DD}+0.5 to V_{SS}−0.5	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

Characteristics	Symbol	Value	Unit
Thermal Resistance			
Plastic	θ_{JA}	100	°C/W
Cerdip		60	
Ceramic		50	
Chip-Carrier		TBD	

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

FIGURE 1 — MICROPROCESSOR BLOCK DIAGRAM



DC ELECTRICAL CHARACTERISTICS 3.0 V (V_{DD} = 3.0 Vdc, V_{SS} = 0, T_A = 0° to 70°C, unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Output Voltage $I_{LOAD} \leq 10.0$ µA	V_{OL} V_{OH}	– $V_{DD}-0.1$	0.1 –	V
Total Supply Current (C_L = 50 pF — no DC loads) t_{cyc} = 5 µs				
Run (V_{IL} = 0.2 V, V_{IH} = V_{DD} – 0.2 V)	I_{DD}	–	1.3	mA
Wait (Test Conditions – See Note Below)	I_{DD}	–	200	µA
Stop (Test Conditions – See Note Below)	I_{DD}	–	100	µA
Output High Voltage				
(I_{LOAD} = 0.25 mA) A8-A12, B0-B7	V_{OH}	2.7	–	V
(I_{LOAD} = 0.1 mA) PA0-PA7, PB0-PB7	V_{OH}	2.7	–	V
(I_{LOAD} = 0.25 mA) DS, AS, R/$\overline{W}$	V_{OH}	2.7	–	V
Output Low Voltage				
(I_{LOAD} = 0.25 mA) A8-A12, B0-B7	V_{OL}	–	0.3	V
(I_{LOAD} = 0.25 mA) PA0-PA7, PB0-PB7	V_{OL}	–	0.3	V
(I_{LOAD} = 0.25 mA) DS, AS, R/$\overline{W}$	V_{OL}	–	0.3	V
Input High Voltage				
PA0-PA7, PB0-PB7, B0-B7	V_{IH}	2.1	–	V
TIMER, $\overline{IRQ}$, $\overline{RESET}$	V_{IH}	2.5	–	V
OSC1	V_{IH}	2.1	–	V
Input Low Voltage (All inputs)	V_{IL}	–	0.5	V
Frequency of Operation				
Crystal	f_{OSC}	0.032	1.0	MHz
External Clock	f_{OSC}	DC	1.0	MHz
Input Current $\overline{RESET}$, $\overline{IRQ}$, Timer, OSC1	i_{in}	–	±1	µA
Three-State Output Leakage PA0-PA7, PB0-PB7, B0-B7	I_{TSL}	–	±10	µA
Capacitance $\overline{RESET}$, $\overline{IRQ}$, Timer	C_{in}	–	8.0	pF
Capacitance DS, AS, R/$\overline{W}$, A8-A12, PA0-PA7, PB0-PB7, B0-B7	C_{out}	–	12.0	pF

NOTE: Test conditions for Quiescent Current Values are:

Port A and B programmed as inputs.
V_{IL} = 0.2 V for PA0-PA7, PB0-PB7, and B0-B7.
V_{IH} = V_{DD} – 0.2 V for $\overline{RESET}$, $\overline{IRQ}$, and Timer.
OSC1 input is a squarewave from V_{SS} + 0.2 V to V_{DD} – 0.2 V.
OSC2 output load (including tester) is 35 pF maximum.
Wait mode I_{DD} is affected linearly by this capacitance.

DC ELECTRICAL CHARACTERISTICS 5.0 V (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0, T_A = 0° to 70°, unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Output Voltage $I_{LOAD} \leq 10.0\ \mu A$	V_{OL} V_{OH}	– $V_{DD}-0.1$	0.1 –	V V
Total Supply Current (C_L = 130 pF – On Bus, C_L = 50 pF – On Ports, No DC Loads, t_{CYC} = 1.0 µs Run (V_{IL} = 0.2 V, V_{IH} = V_{DD} – 0.2 V)	I_{DD}	–	10	mA
Wait (Test Conditions – See Note Below)	I_{DD}	–	1.5	mA
Stop (Test Conditions – See Note Below)	I_{DD}	–	200	µA
Output High Voltage (I_{LOAD} = 1.6 mA) A8-A12, B0-B7	V_{OH}	4.1	–	V
(I_{LOAD} = 0.36 mA) PA0-PA7, PB0-PB7	V_{OH}	4.1	–	V
(I_{LOAD} = 1.6 mA) DS, AS, R/$\overline{W}$	V_{OH}	4.1	–	V
Output Low Voltage (I_{LOAD} = 1.6 mA) A8-A12, B0-B7	V_{OL}	–	0.4	V
(I_{LOAD} = 1.6 mA) PA0-PA7, PB0-PB7	V_{OL}	–	0.4	V
(I_{LOAD} = 1.6 mA) DS, AS, R/$\overline{W}$	V_{OL}	–	0.4	V
Input High Voltage PA0-PA7, PB0-PB7	V_{IH}	$V_{DD}-2.0$	–	V
TIMER, $\overline{IRQ}$, $\overline{RESET}$	V_{IH}	$V_{DD}-0.8$	–	V
OSC1	V_{IH}	$V_{DD}-1.5$	–	V
Input Low Voltage (All Inputs)	V_{IL}	–	0.8	V
Frequency of Operation Crystal	f_{OSC}	0.032	5.0	MHz
External Clock	f_{OSC}	DC	5.0	MHz
Input Current $\overline{RESET}$, $\overline{IRQ}$, Timer, OSC1	I_{in}	–	±1	µA
Three-State Output Leakage PA0-PA7, PB0-PB7, B0-B7	I_{TSI}	–	±10	µA
Capacitance $\overline{RESET}$, $\overline{IRQ}$, Timer	C_{in}	–	9.0	pF
Capacitance DS, AS, R/$\overline{W}$, A8-A12, PA0-PA7, PB0-PB7, B0-B7	C_{out}	–	12.0	pF

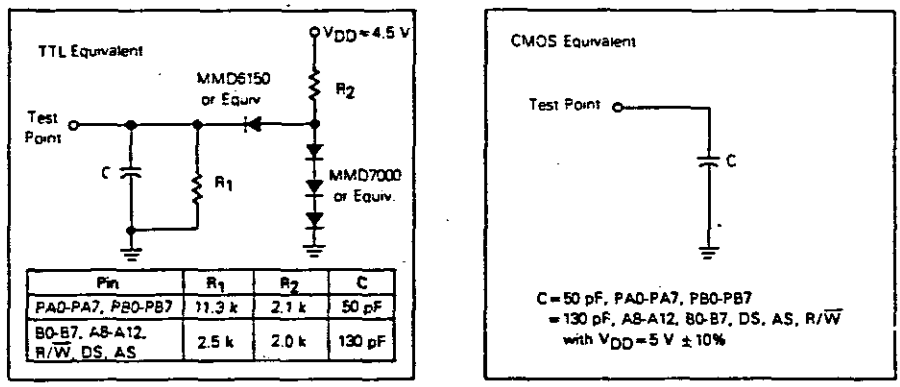
NOTE: Test conditions for Quiescent Current Values are:
Port A and B programmed as inputs.
V_{IL} = 0.2 V for PA0-PA7, PB0-PB7, and B0-B7.
V_{IH} = V_{DD} – 0.2 V for $\overline{RESET}$, $\overline{IRQ}$, and Timer.
OSC1 input is a squarewave from V_{SS} + 0.2 V to V_{DD} – 0.2 V.
OSC2 output load (including tester) is 35 pF maximum.
Wait mode (I_{DD}) is affected linearly by this capacitance.

TABLE 1 — CONTROL TIMING (V_{SS} = 0, T_A = 0° to 70°C)

		V_{DD} = 3.0 V, f_{OSC} = 1 MHz			V_{DD} = 5.0 V ± 10%, f_{OSC} = 5.0 MHz			
Characteristics	Symbol	Min	Typ	Max	Min	Typ	Max	Unit
I/O Port Timing — Input Setup Time (Figure 3)	t_{PVASL}	500	–	–	250	–	–	ns
Input Hold Time (Figure 3)	t_{ASLPX}	100	–	–	100	–	–	ns
Output Delay Time (Figure 3)	t_{ASLPV}	–	–	0	–	–	0	ns
Interrupt Setup Time (Figure 6)	t_{ILASL}	2	–	–	0.4	–	–	µs
Crystal Oscillator Startup Time (Figure 5)	t_{OXOV}	–	30	300	–	15	100	ms
Wait Recovery Startup Time (Figure 7)	t_{IVASH}	–	–	10	–	–	2	µs
Stop Recovery Startup Time (Crystal Oscillator) (Figure 8)	t_{ILASH}	–	30	300	–	15	100	ms
Required Interrupt Release (Figure 6)	t_{DSLIH}	–	–	5	–	–	1.0	µs
Timer Pulse Width (Figure 7)	t_{TH}, t_{TL}	0.5	–	–	0.5	–	–	t_{cyc}
Reset Pulse Width (Figure 5)	t_{RL}	5.2	–	–	1.05	–	–	µs
Timer Period (Figure 7)	t_{TLTL}	1.0	–	–	1.0	–	–	t_{cyc}
Interrupt Pulse Width Low (Figure 16)	t_{ILIH}	1.0	–	–	1.0	–	–	t_{cyc}
Interrupt Pulse Period (Figure 16)	t_{ILIL}	*	–	–	*	–	–	t_{cyc}
Oscillator Cycle Period (1/5 of t_{cyc})	t_{OLOL}	1000	–	–	200	–	–	*ns*
OSC1 Pulse Width High	t_{OH}	350	–	–	75	–	–	ns
OSC1 Pulse Width Low	t_{OL}	350	–	–	75	–	–	ns

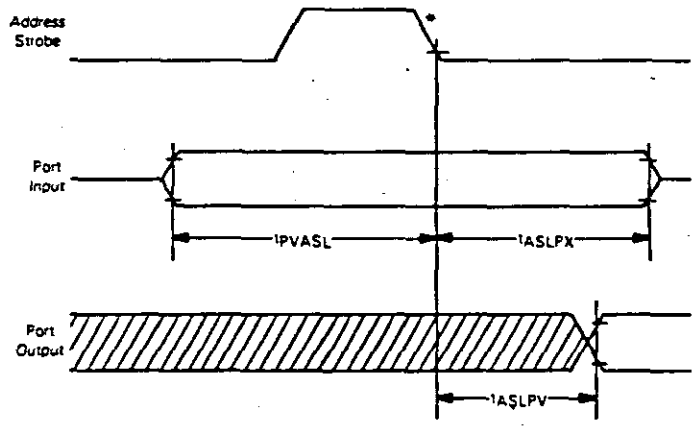
*The minimum period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 20 t_{cyc} cycles.

FIGURE 2 — EQUIVALENT TEST LOADS



Pin	R_1	R_2	C
PA0-PA7, PB0-PB7	11.3 k	2.1 k	50 pF
B0-B7, A8-A12, R/W, DS, AS	2.5 k	2.0 k	130 pF

FIGURE 3 — I/O PORT TIMING
(V_{LOW} = 0.8 V, V_{HIGH} = V_{DD} − 2.0 V, V_{DD} = 5.0 ± 10%, Temp = 0° to 70°C, C_L on Port = 50 pF, f_{OSC} = 5 MHz)

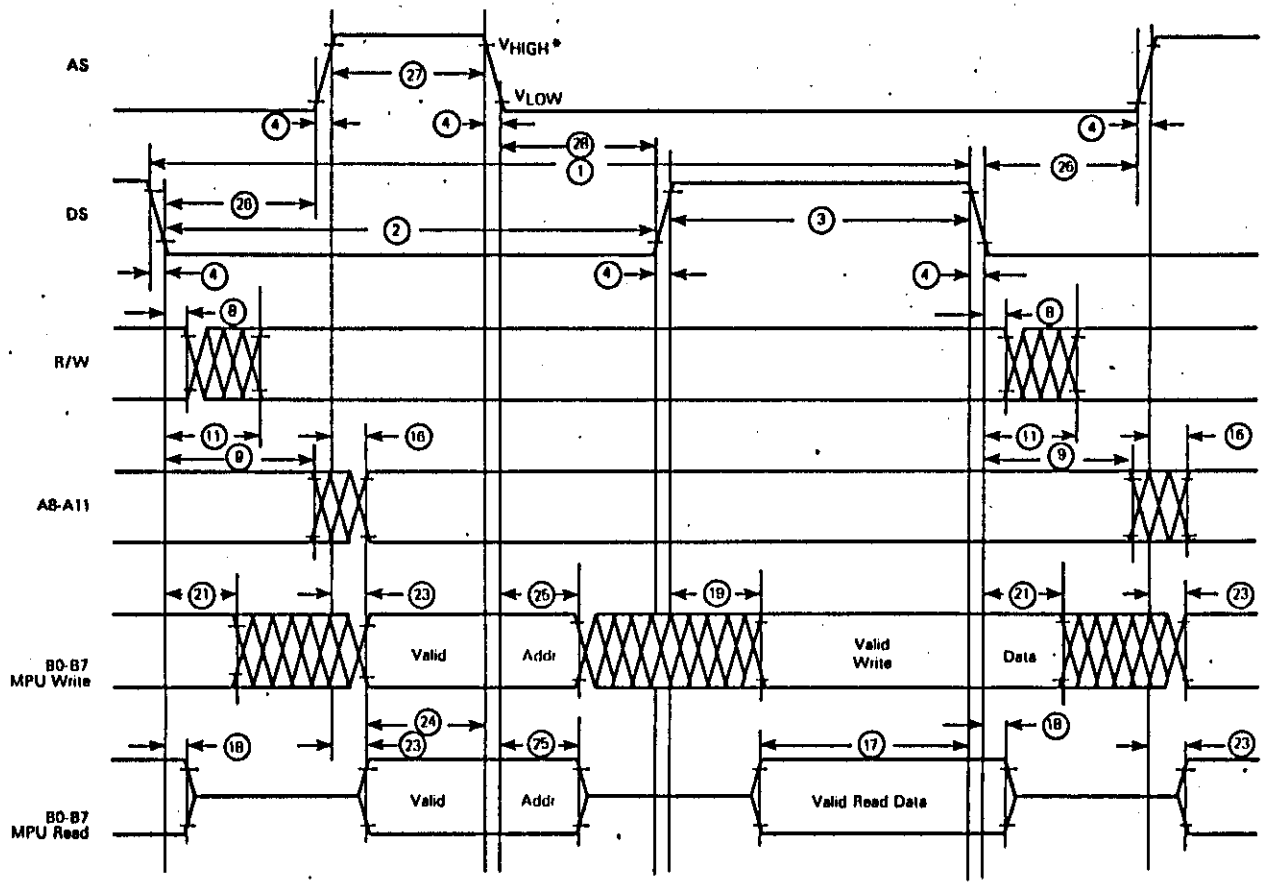


*The address strobe of the first cycle of the next instruction as shown in Table 11.

TABLE 2 — BUS TIMING (T_A = 0° to 70°C, V_{SS} = 0 V) See Figure 4

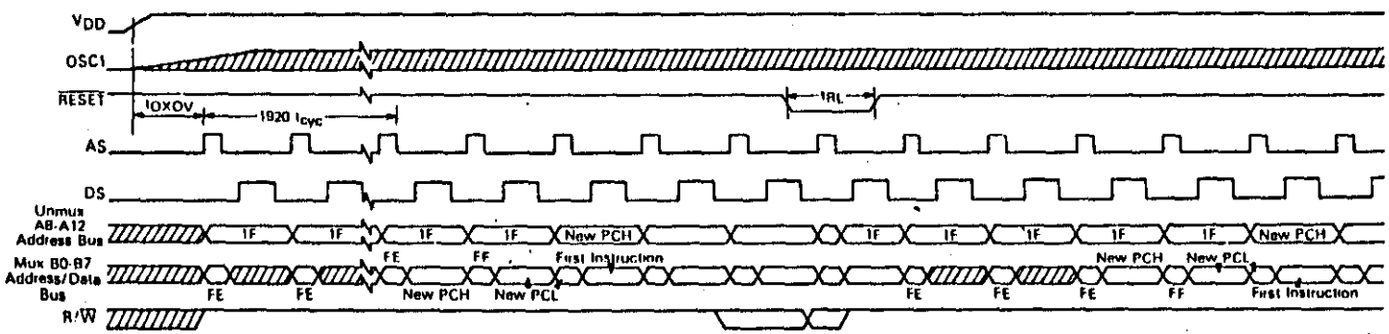
Num	Characteristics	Symbol	f_{OSC} = 1 MHz, V_{DD} = 3.0 V 50 pF Load		f_{OSC} = 5 MHz V_{DD} = 5.0 V ± 10%, 1 TTL and 130 pF Load		Unit
			Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	5000	DC	1000	DC	ns
2	Pulse Width, DS Low	PW_{EL}	2800	–	560	–	ns
3	Pulse Width, DS High or $\overline{RD}$, $\overline{WR}$, Low	PW_{EH}	1800	–	375	–	ns
4	Clock Transition	t_r, t_f	–	100	–	30	ns
8	R/$\overline{W}$ Hold	t_{RWH}	10	–	10	–	ns
9	Non-Muxed Address Hold	t_{AH}	800	–	100	–	ns
11	R/$\overline{W}$ Delay from DS Fall	t_{AD}	–	500	–	300	ns
16	Non-Muxed Address Delay from AS Rise	t_{ADH}	0	200	0	100	ns
17	MPU Read Data Setup	t_{DSR}	200	–	115	–	ns
18	Read Data Hold	t_{DHR}	0	1000	0	160	ns
19	MPU Data Delay, Write	t_{DDW}	–	0	–	120	ns
21	*Write Data Hold*	t_{DHW}	800	–	55	–	ns
23	Muxed Address Delay from AS Rise	t_{BHD}	0	250	0	120	ns
24	Muxed Address Valid to AS Fall	t_{ASL}	600	–	55	–	ns
25	Muxed Address Hold	t_{AHL}	250	750	60	180	ns
26	Delay DS Fall to AS Rise	t_{ASD}	800	–	160	–	ns
27	Pulse Width, AS High	PW_{ASH}	850	–	175	–	ns
28	Delay, AS Fall to DS Rise	t_{ASED}	800	–	160	–	ns

FIGURE 4 — MC146805E2 BUS TIMING

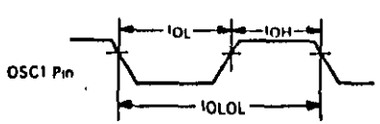


*V_{HIGH} = 2.0 V, V_{LOW} = 0.5 V for V_{DD} = 3 V
V_{HIGH} = V_{DD} – 2.0 V, V_{LOW} = 0.8 V for V_{DD} = 5 V ± 10%

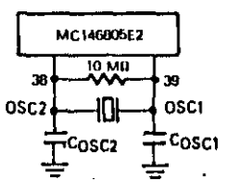
FIGURE 5 – POWER-ON RESET AND $\overline{\text{RESET}}$ TIMING



Oscillator Waveform



Crystal Oscillator Connections



Crystal Parameters Representative Frequencies

	6.0 MHz	4.0 MHz	1.0 MHz
RS max	50Ω	75Ω	400Ω
C0	8 pF	7 pF	5 pF
C1	0.02 pF	0.012 pF	0.008 pF
Q	50 k	40 k	30 k
C_{OSC1}	15-30 pF	15-30 pF	15-40 pF
C_{OSC2}	15-25 pF	15-25 pF	15-30 pF

Crystal Circuit

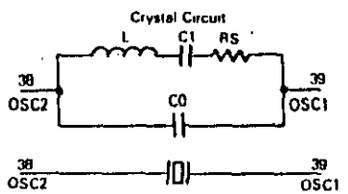
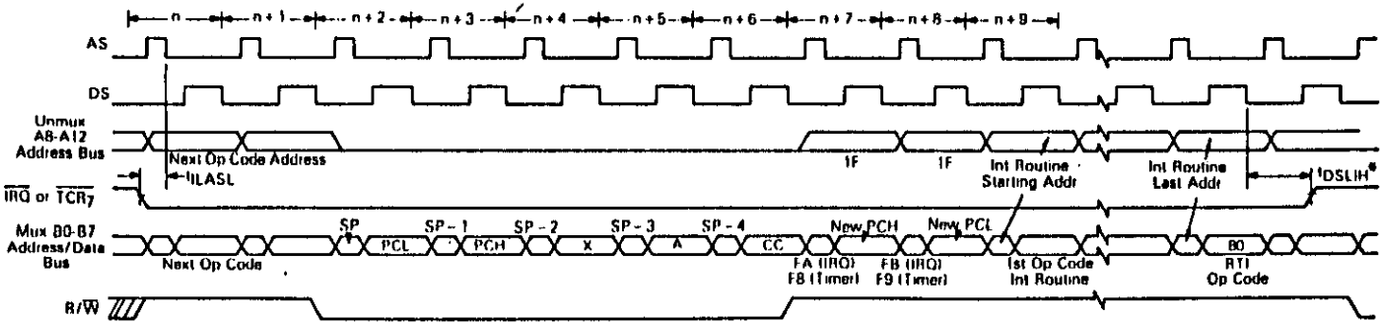


FIGURE 6 — $\overline{IRQ}$ AND $\overline{TCR_7}$ INTERRUPT TIMING



*t_{DSLIH} – The interrupting device must release the IRQ line within this time to prevent subsequent recognition of the same interrupt.

FIGURE 7 — TIMER INTERRUPT AFTER WAIT INSTRUCTION: TIMING

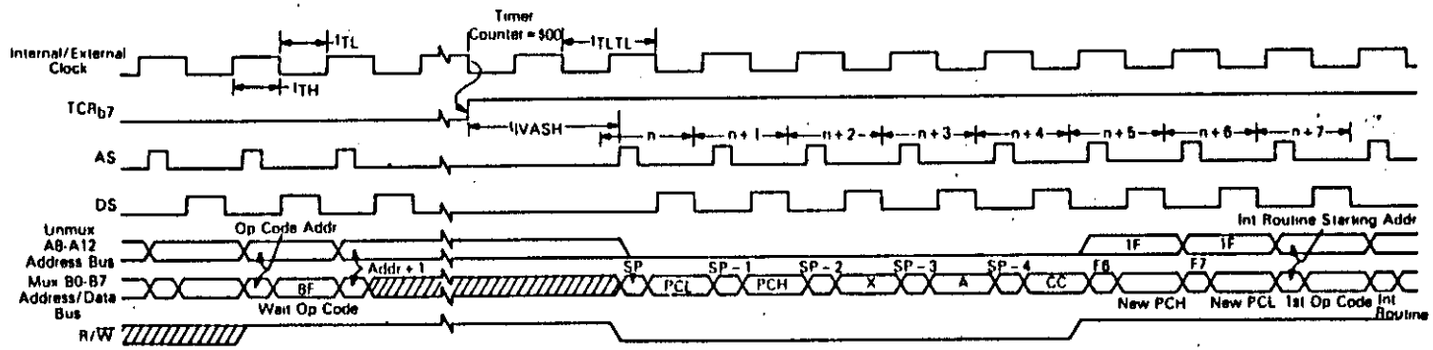
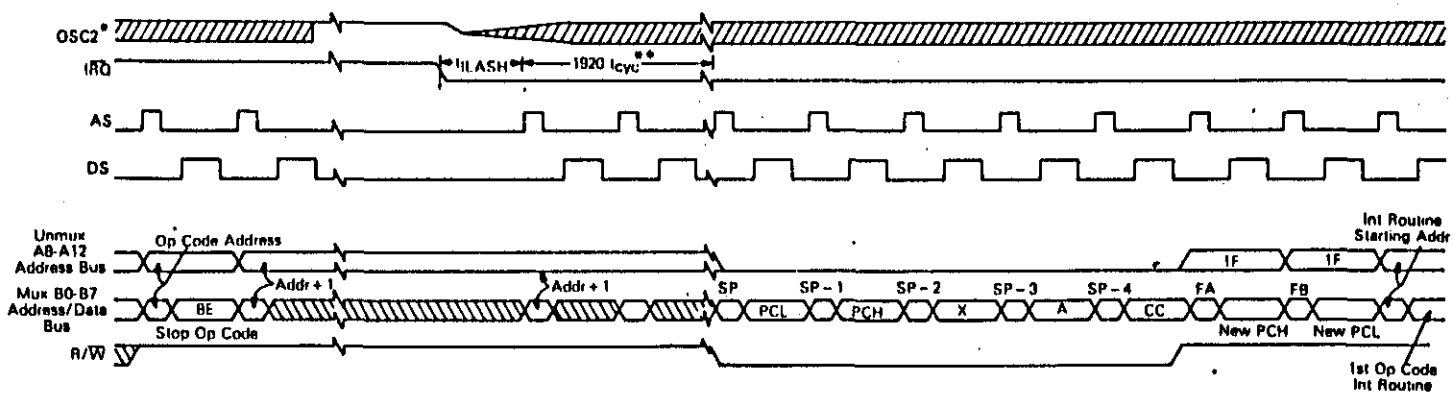


FIGURE 8 — INTERRUPT RECOVERY FROM STOP INSTRUCTION: TIMING



*Represents the internal gating of the OSC1 input pin.
**t_{cyc} is one instruction cycle (for f_{OSC} = 5 MHz, t_{cyc} = 1 µs)

FUNCTIONAL PIN DESCRIPTION

V_{DD} and V_{SS} — V_{DD} and V_{SS} provide power to the chip. V_{DD} provides power and V_{SS} is ground.

$\overline{IRQ}$ (Maskable Interrupt Request) — $\overline{IRQ}$ is a level-sensitive and edge sensitive input which can be used to request an interrupt sequence. The MPU completes the current instruction before it responds to the request. IF $\overline{IRQ}$ is low and the interrupt mask bit (I-bit) in the Condition Code Register is clear, the MPU begins an interrupt sequence at the end of the current instruction. The interrupt circuit recognizes both a "Wire ORed" level as well as pulses on the $\overline{IRQ}$ line (see Interrupt Section for more details). $\overline{IRQ}$ requires an external resistor to V_{DD} for "Wire OR" operation.

$\overline{RESET}$ — The $\overline{RESET}$ input is not required for start-up but can be used to reset the MPU's internal state and provide an orderly software start-up procedure. Refer to the RESET section for a detailed description.

TIMER — The TIMER input is used for clocking the on-chip timer. Refer to TIMER section for a detailed description.

AS (Address Strobe) — Address Strobe (AS) is an output strobe used to indicate the presence of an address on the 8-bit multiplexed bus. The AS line is used to demultiplex the eight least significant address bits from the data bus. A latch controlled by Address Strobe should capture addresses on the negative edge. This output is capable of driving one standard TTL load and 130 pF and is available at $f_{OSC} \div 5$ when the MPU is not in the WAIT or STOP states.

DS (Data Strobe) — This output is used to transfer data to or from a peripheral or memory. DS occurs anytime the MPU does a data read or write. DS also occurs when the MPU does a data transfer to or from the MPU's internal memory. Refer to Table 2 and Figure 4 for timing characteristics. This output is capable of driving one standard TTL load and 130 pF. DS is a continuous signal at $f_{OSC} \div 5$ when the MPU is not in WAIT or STOP state. Some bus cycles are redundant reads of op code bytes.

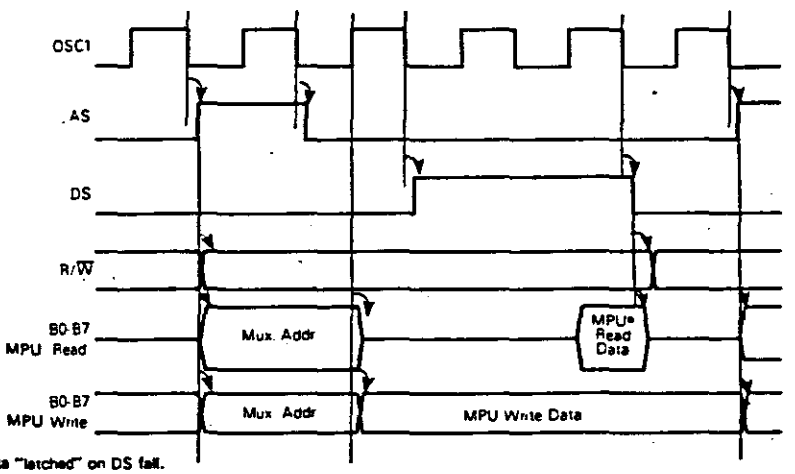
R/$\overline{W}$ (Read/Write) — The R/$\overline{W}$ output is used to indicate the direction of data transfer for both internal memory and I/O registers, and external peripheral devices and memories. This output is used to indicate to a selected peripheral whether the MPU is going to read or write data on the next Data Strobe (R/$\overline{W}$ low = processor write; R/$\overline{W}$ high = processor read). The R/$\overline{W}$ output is capable of driving one standard TTL load and 130 pF. The normal standby state is Read (high).

A8-A12 (High Order Address Lines) — The A8-A12 output lines constitute the higher order non-multiplexed addresses. Each output line is capable of driving one standard TTL load and 130 pF.

B0-B7 (Address/Data Bus) — The B0-B7 bidirectional lines constitute the lower order addresses and data. These lines are multiplexed, with address present at Address Strobe time and data present at Data Strobe time. When in the data mode, these lines are bidirectional, transferring data to and from memory and peripheral devices as indicated by the R/$\overline{W}$ pin. As outputs in either the data or address modes, these lines are capable of driving one standard TTL load and 130 pF.

OSC1, OSC2 — The MC146805E2 provides for two types of oscillator inputs — crystal circuit or external clock. The two oscillator pins are used to interface to a crystal circuit, as shown in Figure 5. If an external clock is used, it must be connected to OSC1. The input at these pins is divided by five to form the cycle rate seen on the AS and DS pins. The frequency range is specified by f_{OSC}. The OSC1 to bus transitions relationships are provided in Figure 9 for system designs using oscillators slower than 5 MHz.

FIGURE 9 — OSC1 TO BUS TRANSITIONS

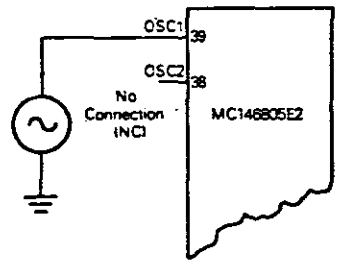


*Read data "latched" on DS fall.

Crystal — The circuit shown in Figure 5 is recommended when using a crystal. The internal oscillator is designed to interface with an AT-cut parallel resonant quartz crystal resonator in the frequency range specified for f_{OSC} in the electrical characteristics table. An external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time.

External Clock — An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 10.

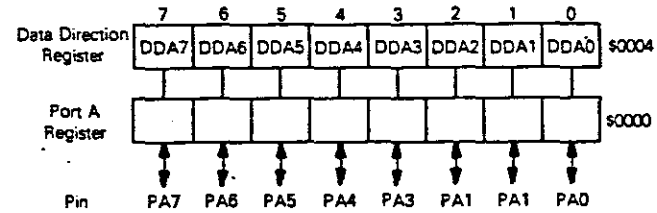
FIGURE 10 — EXTERNAL CLOCK CONNECTION



LI (Load Instruction) — This output is used to indicate that a fetch of the next opcode is in progress. LI remains low during an External or Timer interrupt. The LI output is only used for certain debugging and test systems. For normal operations this pin is not connected. The LI output is capable of driving one standard TTL load and 50 pF. This signal overlaps Data Strobe.

PA0-PA7 — These eight pins constitute Input/Output Port A. Each line is individually programmed to be either an input or output under software control via its Data Direction Register as shown below. An I/O pin is programmed as an output when the corresponding DDR bit is set to a "1," and as an input when it is set to a "0". In the output mode the bits are latched and appear on the corresponding output pins. An MPU read of the port bits programmed as outputs reflect the last value written to that location. When programmed as an input, the input data bit(s) are not latched. An MPU read of the port bits programmed as inputs reflects the current status of the corresponding input pins. The Read/Write port timing is shown in Figure 3. See typical I/O Port Circuitry in Figure 11. During a Power-On Reset or external $\overline{\text{RESET}}$ all lines are configured as inputs (zero in Data Direction Register). The output port register is not initialized by reset. The TTL compatible three-state output buffers are capable of driving one standard TTL load and 50 pF. The DDR is a read/write register.

PB0-PB7 — These eight pins interface to Input/Output Port B. Refer to PA0-PA7 description for details of operation.



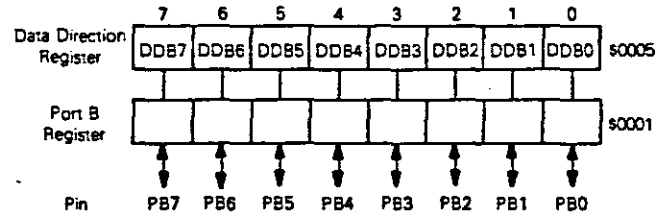


FIGURE 11 — TYPICAL PORT I/O CIRCUITRY

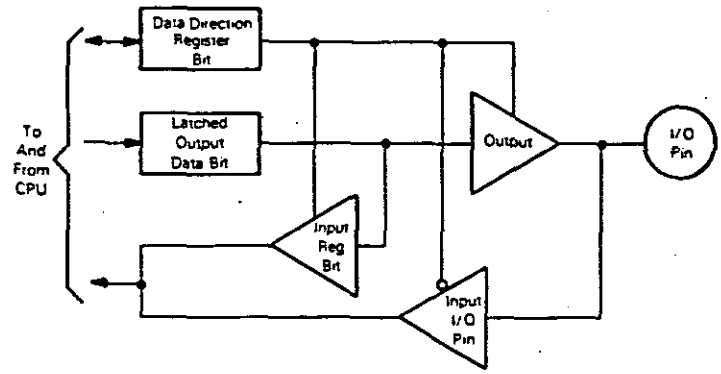


TABLE 3 — I/O PIN FUNCTIONS

R/$\overline{W}$	DDR	I/O Pin Functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

MEMORY ADDRESSING

The MC146805E2 is capable of addressing 8192 bytes of memory and I/O registers. The address space is divided into internal memory space and external memory space, as shown in Figure 12.

The internal memory space is located within the first 128 bytes of memory (first half of page zero) and is comprised of the I/O port locations, timer locations, and 112 bytes of RAM. The MPU can read from or write to any of these locations. A program write to on-chip locations is repeated on the external bus to permit off-chip memory to duplicate the content of on-chip memory. Program reads to on-chip locations also appear on the external bus, but the MPU accepts data only from the addressed on-chip location. Any read data appearing on the input bus is ignored.

The stack pointer is used to address data stored on the stack. Data is stored on the stack during interrupts and subroutine calls. At power up, the stack pointer is set to $7F and it is decremented as data is pushed onto the stack. When data is removed from the stack, the stack pointer is incremented. A maximum of 64 bytes of RAM is available for stack usage. Since most programs use only a small part of the allotted stack locations for interrupts and/or subroutine stacking purposes, the unused bytes are usable for program data storage.

All memory locations above location $007F are part of the external memory map. In addition, ten locations in the I/O portion of the lower 128 bytes of memory space, as shown in Figure 12, are part of the external memory map. All of the external memory space is user definable except the highest 10 locations. Locations $1FF6 to $1FFF of the external address space are reserved for interrupt and reset vectors (see Figure 12).

REGISTERS

The MC146805E2 contains five registers as shown in the programming model in Figure 13. The interrupt stacking order is shown in Figure 14.

ACCUMULATOR (A) — This Accumulator is an 8-bit general purpose register used for arithmetic calculations and data manipulations.

INDEX REGISTER (X) — The X register is an 8-bit register which is used during the indexed modes of addressing. It provides an 8-bit operand which is used to create an effective address. The index register is also used for data manipulations with the Read/Modify/Write type of instructions and as a temporary storage register when not performing addressing operations.

PROGRAM COUNTER (PC) — The program counter is a 13-bit register that contains the address of the next instruction to be executed by the processor.

FIGURE 12 – ADDRESS MAP

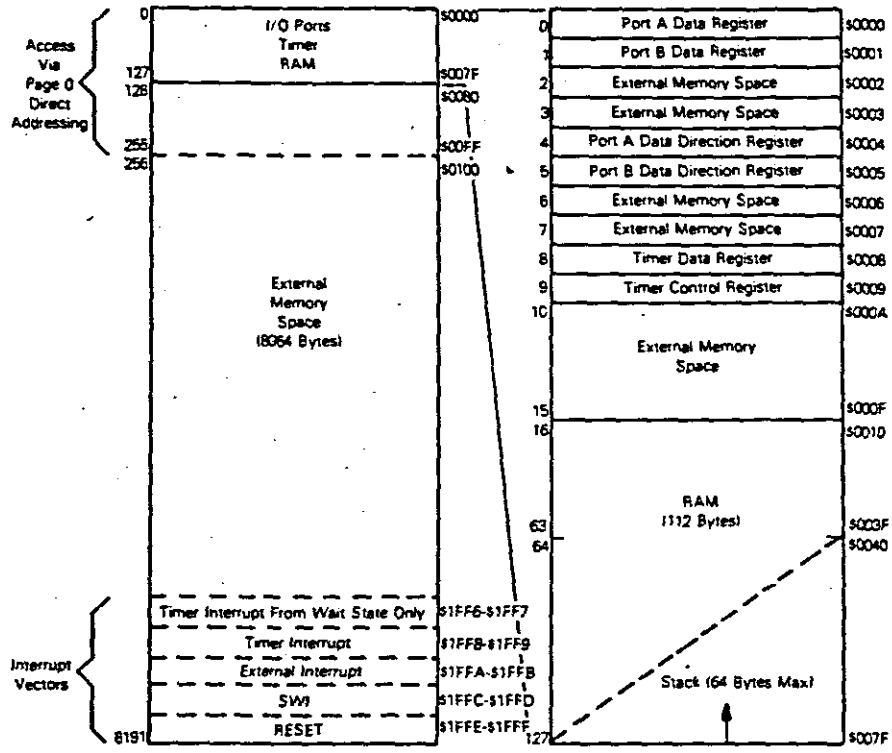


FIGURE 13 — PROGRAMMING MODEL

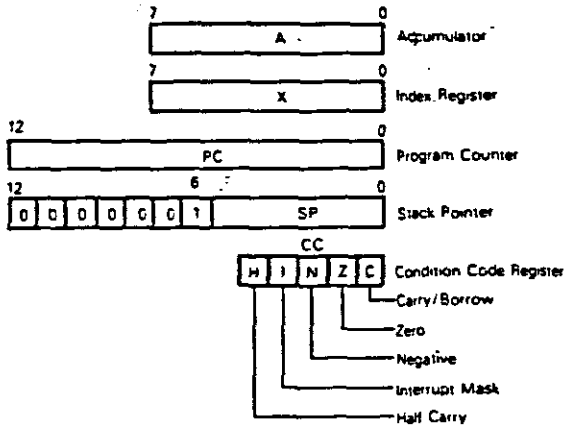
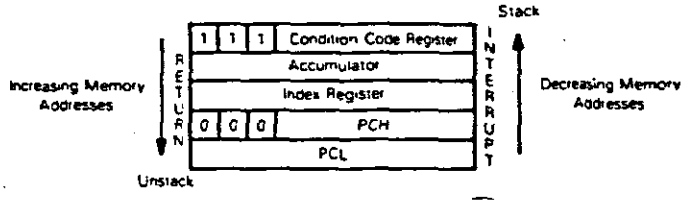


FIGURE 14 — STACKING ORDER



NOTE: Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order

STACK POINTER (SP) — *The stack pointer is a* 13-bit register containing the address of the next free location on the stack. When accessing memory, the seven most-significant bits are permanently set to 0000001. They are appended to the six least-significant register bits to produce an address within the range of $007F to $0040. The stack area of RAM is used to store the return address on subroutine calls *and the machine state during interrupts. During external* or power-on reset, and during a "reset stack pointer" instruction, the stack pointer is set to its upper limit ($007F). Nested interrupts and/or subroutines may use up to 64 (decimal) locations, beyond which the stack pointer "wraps around" and points to its upper limit thereby losing the previously stored information. A subroutine call occupies two RAM *bytes on the stack, while an interrupt uses five bytes.*

CONDITION CODE REGISTER (CC) — The condition code register is a 5-bit register in which each bit is used to indicate the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each of the five bits is explained below.

Half Carry Bit (H) — The H-bit is set to a one when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. The H-bit is useful in Binary Coded Decimal addition subroutines.

Interrupt Mask Bit (I) — When the I-bit is set, both the external interrupt and the timer interrupt are disabled. Clearing this bit enables the above interrupts. If an interrupt occurs while the I-bit is set, the interrupt is latched and will be processed when the I-bit is next cleared.

Negative Bit (N) — When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logical one).

Zero Bit (Z) — When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Carry Bit (C) — The C-bit is set when a carry or a borrow out of the ALU occurs during an arithmetic instruction. The C-bit is also modified during bit test, shift, rotate, and branch types of instruction.

RESETS

The MC146805E2 has two reset modes: an active low external reset pin ($\overline{\text{RESET}}$) and a Power-On Reset function; refer to Figure 5.

$\overline{\text{RESET}}$ (Pin #1) — The $\overline{\text{RESET}}$ input pin is used to reset the MPU and provide an orderly software start-up procedure. When using the external reset mode, the $\overline{\text{RESET}}$ pin must stay low for a minimum of one t_{cyc}. The $\overline{\text{RESET}}$ pin is provided with a Schmitt Trigger to improve its noise immunity capability.

Power-On Reset — The Power-on Reset occurs when a positive transition is detected on V_{DD}. The Power-on Reset is used strictly for power turn-on conditions and should not be used to detect any drops in the power supply voltage. There is no provision for a power-down reset. The power-on circuitry provides for a 1920 t_{cyc} delay from the time of the first oscillator operation. If the external reset pin is low at the end of the 1920 t_{cyc} time out, the processor remains in the reset condition.

Either of the two types of reset conditions causes the following to occur:

- Timer control register interrupt request bit (bit 7) is cleared to a "0".
- Timer control register interrupt mask bit (bit 6) is set to a "1".
- All data direction register bits are cleared to a "0" (inputs).
- Stack pointer is set to $007F.
- The address bus is forced to the reset vector ($1FFE, $1FFF).
- Condition code register interrupt mask bit (I) is set to a "1".
- STOP and WAIT latches are reset.
- External interrupt latch is reset.

All other functions, such as other registers (including output ports) the timer, etc., are not cleared by the reset conditions.

INTERRUPTS

The MC146805E2 is capable of operation with three different interrupts, two hardware (timer interrupt and external interrupt) and one software (SWI). When any of these interrupts occur, normal processing is suspended at the end of the current instruction execution. All of the program registers (the machine state) are pushed onto the stack; refer to Figure 14 for stacking order. The appropriate vector pointing to the starting address of the interrupt service routine is then fetched; refer to Figure 15 for the interrupt sequence.

The priority of the various interrupts from highest to lowest is as follows:

RESET—*—External Interrupt—Timer Interrupt

TIMER INTERRUPT — If the timer mask bit (TCR6) is cleared, then each time the timer decrements to zero (transitions from $01 to $00) an interrupt request is generated. The actual processor interrupt is generated only if the interrupt mask bit of the condition code register is also cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the I-bit in the condition code register is set. This masks further interrupts until the present one is serviced. The processor now vectors to the timer interrupt service routine. The address for this service routine is specified by the contents of $1FF8 and $1FF9 unless the processor is in a WAIT mode in which case users of mask versions BP4XXXX and AW9XXXX should refer to the appendix for additional information regarding exceptions to this function. The contents of $1FF6 and $1FF7 specify the service routine. Also, software must be used to clear the timer interrupt request bit (TCR7). At the end of the timer interrupt service routine, the software normally executes an RTI instruction which restores the machine state and starts executing the interrupted program.

EXTERNAL INTERRUPT — If the interrupt mask bit of the condition code register is cleared and the external interrupt pin $\overline{\text{IRQ}}$ is "low," then the external interrupt occurs. The action of the external interrupt is identical to the timer interrupt with the exception that the service routine address is specified by the contents of $1FFA and $1FFB. The interrupt logic recognizes both a "wire ORed" level and pulses on the external interrupt line. Figure 16 shows both a functional diagram and timing for the interrupt line. The timing diagram shows two different treatments of the interrupt line ($\overline{\text{IRQ}}$) to the processor. The first configuration shows many interrupt lines "wire ORed" to form the interrupts at the processor. Thus, if after servicing an interrupt the $\overline{\text{IRQ}}$ remains low, then the next interrupt is recognized. The second method is single pulses on the interrupt line spaced far enough apart to be serviced. Users of mask versions BP4XXXX and AW9XXXX should refer to the appendix regarding exceptions to this function. The minimum time between pulses is a function of the length of the interrupt service routine. Once a pulse occurs, the next pulse should not occur until the MPU software has exited the routine (an RTI occurs). This time (t_{ILIL}) is obtained by adding 20 instruction cycles (one cycle $t_{cyc} = 5/f_{OSC}$) to the total number of cycles it takes to complete the service routine including the RTI instruction; refer to Figure 6.

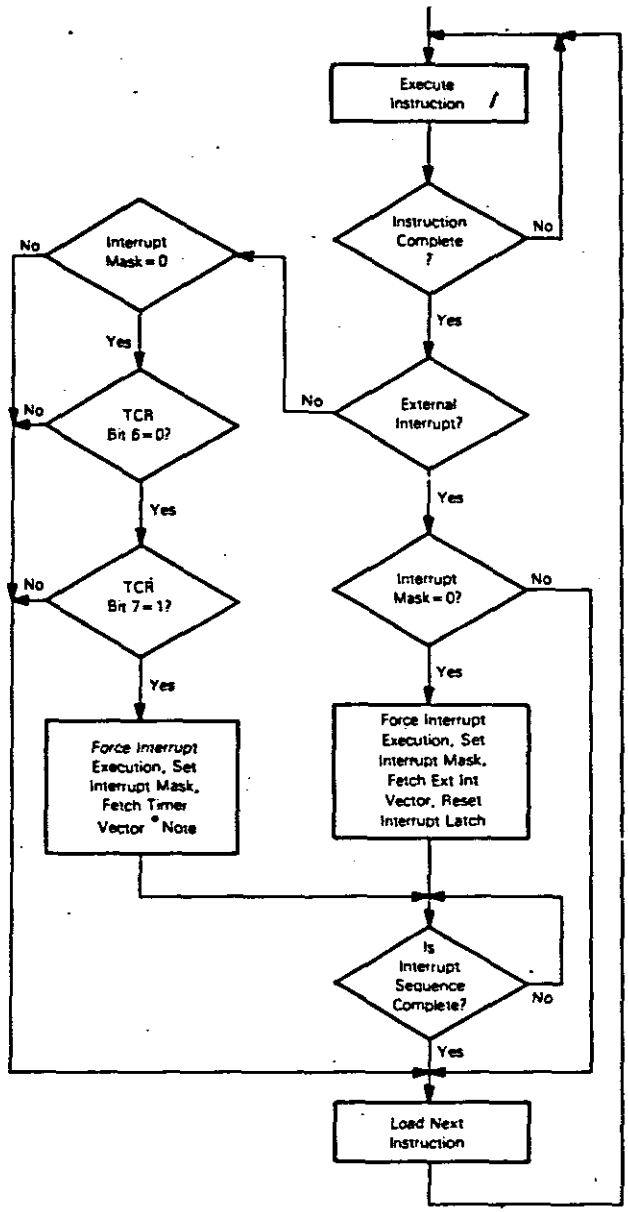
SOFTWARE INTERRUPT (SWI) — The software interrupt is an executable instruction. The action of the SWI instruction is similar to the hardware interrupts. The SWI is executed regardless of the state of the interrupt mask in the condition code register. The service routine address is specified by the contents of memory locations $1FFC and $1FFD. See Figure 15 for Interrupt and Instruction Processing Flowchart.

The following three functions are not strictly interrupts; however, they are tied very closely to the interrupts. These functions are $\overline{\text{RESET}}$, STOP, WAIT.

$\overline{\text{RESET}}$ — The $\overline{\text{RESET}}$ input pin and the internal Power-on Reset function each cause the program to vector to an initialization program. This vector is specified by the contents of memory locations $1FFE and $1FFF. The interrupt mask of the condition code register is also set. Refer to RESET section for details.

*Any current instruction including SWI.

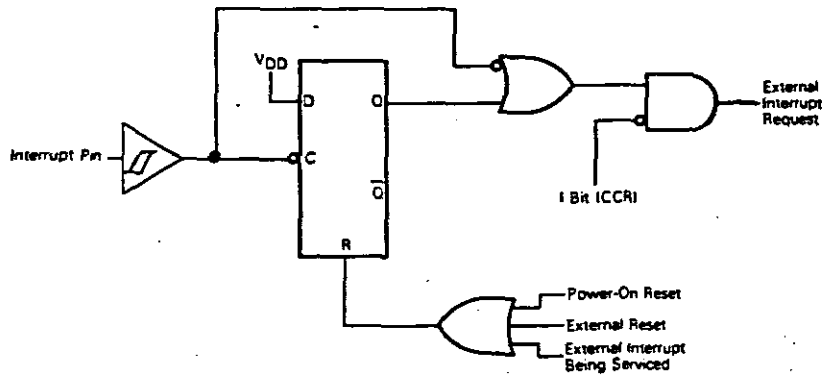
FIGURE 15 — INTERRUPT AND INSTRUCTION PROCESSING FLOWCHART



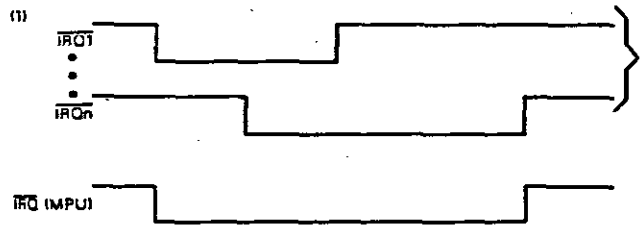
*NOTE: The clear of TCR bit 7 must be accomplished with software.

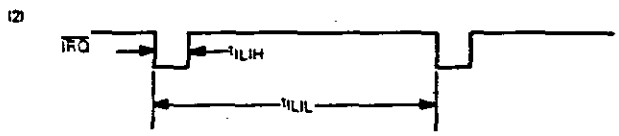
FIGURE 16 — EXTERNAL INTERRUPT

(a) Interrupt Functional Diagram



(b) Interrupt Mode Diagram

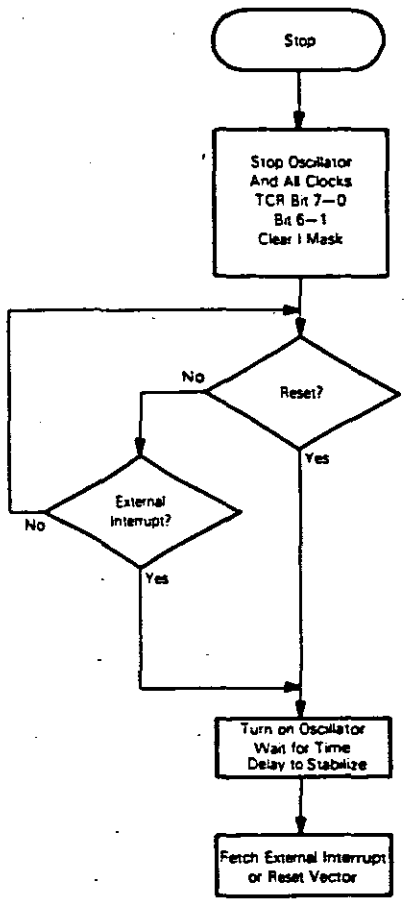




STOP — The STOP instruction places the MC146805E2 in a low power consumption mode. In the STOP function the internal oscillator is turned off, causing all internal processing and the timer to be halted; refer to Figure 17. The DS and AS lines go to a low state and the R/$\overline{W}$ line goes to a high state. The multiplexed address/data bus goes to the data input state. The high order address lines remain at the address of the next instruction. The MPU remains in the STOP mode until an external interrupt or reset occurs; refer to Figure 8 and 17.

During the STOP mode, timer control register (TCR) bits 6 and 7 are altered to remove any pending timer interrupt requests and to disable any further timer interrupts. External interrupts are enabled in the condition code register. All other registers and memory remain unaltered. All I/O lines remain unchanged.

FIGURE 17 — STOP FUNCTION FLOWCHART



WAIT — The WAIT instruction places the MC146805E2 in a low power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode; refer to Table 1. In the WAIT function, the internal clock is disabled from all internal circuitry except the Timer circuit; refer to Figure 18. Thus, all internal processing is halted except the Timer which is allowed to count in a normal sequence. The R/$\overline{W}$ line goes to a high state, the multiplexed address/data bus goes to the data input state, and the DS and AS lines go to the low state. The high order address lines remain at the address of the next instruction. The MPU remains in this state until an external interrupt, timer interrupt, or a reset occurs; refer to Figures 7 and 18.

During the WAIT mode, the I-bit in the condition code register is cleared to enable interrupts. All other registers, memory, and I/O lines remain in their last state. The timer may be enabled to allow a periodic exit from the WAIT mode.If an external and a timer interrupt occur at the same time, the external interrupt is serviced first; then, if the timer interrupt request is not cleared in the external interrupt routine, the normal timer interrupt (not the timer WAIT interrupt) is serviced since the MCU is no longer in the WAIT mode.

TIMER

The MPU timer contains a single 8-bit software programmable counter with 7-bit software selectable prescaler. The counter may be preset under program control and decrements towards zero. When the counter decrements to zero, the timer interrupt request bit, i.e., bit 7 of the Timer Control Register (TCR) is set. Then if the timer interrupt is not masked, i.e., bit 6 of the TCR and the I-bit in the Condition Code Register are both cleared, the processor receives an interrupt. After completion of the current instruction, the processor proceeds to store the appropriate registers on the stack, and then fetches the timer vector address from locations $1FF8 and $1FF9 in order to begin servicing the interrupt, unless it was in locations $1FF6 and $1FF7 the WAIT mode.

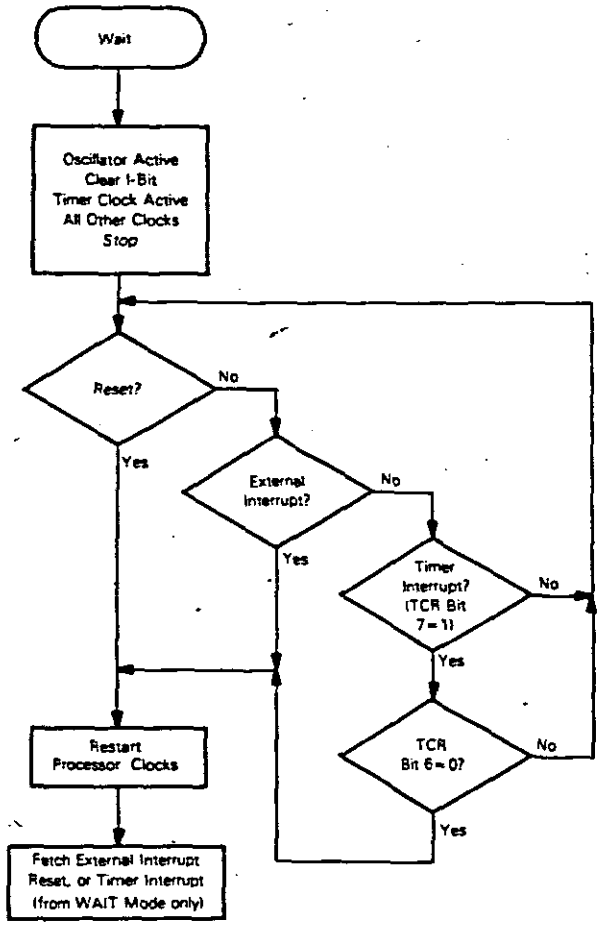
The counter continues to count after it reaches zero, allowing the software to determine the number of internal or external input clocks since the timer interrupt request bit was set. The counter may be read at any time by the processor without disturbing the count. The contents of the counter becomes stable prior to the read portion of a cycle and does not change during the read. The timer interrupt request bit remains set until cleared by the software. If this happens before the timer interrupt is serviced, the interrupt is lost. TCR7 may also be used as a scanned status bit in a non-interrupt mode of operation (TCR6 = 1).

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1, and bit 2 of the TCR are programmed to choose the appropriate prescaler output which is used as the counter input. The processor cannot write into or read from the prescaler; however, its contents are cleared to all "0's" by the write operation into TCR when bit 3 of the written data equals 1, which allows for truncation-free counting.

The Timer input can be configured for three different operating modes, plus a disable mode depending on the value written to the TCR4, TCR5 control bits. Refer to the TIMER CONTROL REGISTER section.

Timer Input Mode 1 — If TCR4 and TCR5 are both programmed to a "0", the input to the Timer is from an internal clock and the Timer input is disabled. The internal clock mode can be used for periodic interrupt generation, as well

FIGURE 18 – WAIT FUNCTION FLOWCHART



as a reference in frequency and event measurement. The internal clock is the instruction cycle clock and is coincident with Address Strobe (AS) except during a WAIT instruction. During a WAIT instruction the AS pin goes to a low state but the internal clock to the Timer continues to run at its normal rate.

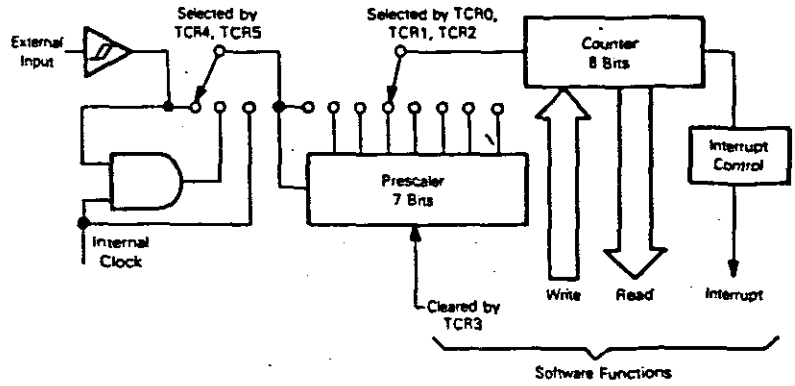
Timer Input Mode 2 – With TCR4 = 1 and TCR5 = 0, the internal clock and the TIMER input pin are ANDed together to form the Timer input signal. This mode can be used to measure external pulse widths. The external pulse simply turns on the internal clock for the duration of the pulse. The resolution of the count in this mode is ±1 clock and therefore accuracy improves with longer input pulse widths.

Timer Input Mode 3 – If TCR4 = 0 and TCR5 = 1, then all inputs to the Timer are disabled.

Timer Input Mode 4 – If TCR4 = 1 and TCR5 = 1, the internal clock input to the Timer is disabled and the TIMER input pin becomes the input to the Timer. The external Timer pin can, in this mode, be used to count external events as well as external frequencies for generating periodic interrupts.

Figure 19 shows a block diagram of the Timer subsystem. Power-on Reset and the STOP instruction cause the counter to be set to $FD.

FIGURE 18 — TIMER BLOCK DIAGRAM



NOTES:
1. Prescaler and 8-bit counter are clocked falling edge of the internal clock (AS) or external input.
2. Counter is written to during Data Strobe (DS) and counts down continuously.

Timer Control Register (TCR)

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0

All bits in this register except bit 3 are Read/Write bits.

TCR7 — Timer interrupt request bit: bit used to indicate the timer interrupt when it is logic "1".
1 — *Set whenever the counter decrements to zero, or under program control.*
0 — Cleared on external reset, power-on reset, STOP instruction, *or program control.*

TCR6 — Timer interrupt mask bit: when this bit is a logic "1" *it inhibits the timer interrupt to the processor.*
1 — Set on external reset, power-on reset, STOP instruction, or program control.
0 — Cleared under program control.

TCR5 — External or internal bit: selects the input clock source to be either the external timer pin or the internal clock. (Unaffected by $\overline{\text{RESET}}$.)
1 — Select external clock source.
0 — Select internal clock source (AS).

TCR4 — External enable bit: control bit used to enable the external timer pin. (Unaffected by $\overline{\text{RESET}}$.)
1 — Enable external timer pin.
0 — Disable external timer pin.

TCR5	TCR4	
0	0	Internal clock (AS) to Timer
0	1	AND of internal clock (AS) and TIMER pin to Timer
1	0	Inputs to Timer disabled
1	1	TIMER pin to Timer

Refer to Figure 19 for Logic Representation.

TCR3 — Timer Prescaler Reset bit: writing a "1" to this bit resets the prescaler to zero. A read of this location always indicates a "0." (Unaffected by $\overline{\text{RESET}}$.)

TCR2, TCR1, TCR0 — Prescaler address bits: decoded to select one of eight taps on the prescaler. (Unaffected by $\overline{\text{RESET}}$.)

Prescaler

TCR2	TCR1	TCR0	Result
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

INSTRUCTION SET

The MPU has a set of 61 basic instructions. They can be divided into five different types: register/memory, read/modify/write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

REGISTER/MEMORY INSTRUCTIONS — Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to Table 4.

READ/MODIFY/WRITE INSTRUCTIONS — These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read/modify/write sequence since it does not modify the value. Refer to Table 5.

BRANCH INSTRUCTIONS — This set of instructions branches if a particular condition is met, otherwise no operation is performed. Branch instructions are two byte instructions. Refer to Table 6.

BIT MANIPULATION INSTRUCTIONS — The MPU is capable of setting or clearing any bit which resides in the first 256 bytes of the memory space, where all port registers, port DDRs, timer, timer control, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test and branch functions are all implemented with a single instruction. For the test and branch instructions the value of the bit tested is also placed in the carry bit of the Condition Code Register. Refer to Table 7 for instruction cycle timing.

CONTROL INSTRUCTIONS — These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 8 for instruction cycle timing.

ALPHABETICAL LISTING — The complete instruction set is given in alphabetical order in Table 9.

OPCODE MAP SUMMARY — Table 10 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MPU uses ten different addressing modes to give the programmer an opportunity to optimize the code to all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions, while the longest instructions (three bytes) permit tables throughout memory. Short and long absolute addressing is also included. Two byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory. Table 9 shows the addressing modes for each instruction, with the effects each instruction has on the Condition Code Register. An opcode map is shown in Table 10.

The term "Effective Address" or EA is used in describing the various addressing modes, which is defined as the address to or from which the argument for an instruction is fetched or stored. The ten addressing modes of the processor are described below. Parentheses are used to indicate "contents of," an arrow indicates "is replaced by" and a colon indicates concatenation of two bytes.

Inherent — In inherent instructions all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, and no other arguments, are included in this mode.

Immediate — In immediate addressing, the operand is contained in the byte immediately following the opcode. Immediate addressing is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

$$EA = PC + 1;\ PC \leftarrow PC + 2$$

Direct — In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two byte instruction. This includes all on-chip RAM and I/O registers and up to 128 bytes of off-chip ROM. Direct addressing is efficient in both memory and speed.

$$EA = (PC + 1);\ PC \leftarrow PC + 2$$

Address Bus High $\leftarrow 0$; Address Bus Low $\leftarrow (PC + 1)$

Extended — In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing modes are capable of referencing arguments anywhere in memory with a single three byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the most efficient addressing mode.

$$EA = (PC + 1):(PC + 2);\ PC \leftarrow PC + 3$$

Address Bus High $\leftarrow (PC + 1)$; Address Bus Low $\leftarrow (PC + 2)$

Indexed, No-Offset — In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is used to move a pointer through a table or to address a frequently referenced RAM or I/O location.

$$EA = X;\ PC \leftarrow PC + 1$$

Address Bus High $\leftarrow 0$; Address Bus Low $\leftarrow X$

TABLE 4 — REGISTER/MEMORY INSTRUCTIONS

		Addressing Modes																		
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5
Store A in Memory	STA	–	–	–	B7	2	4	C7	3	5	F7	1	4	E7	2	5	D7	3	6
Store X in Memory	STX	–	–	–	BF	2	4	CF	3	5	FF	1	4	EF	2	5	DF	3	6
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5
Jump Unconditional	JMP	–	–	–	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4
Jump to Subroutine	JSR	–	–	–	BD	2	5	CD	3	6	FD	1	5	ED	2	6	DD	3	7

TABLE 5 — READ/MODIFY/WRITE INSTRUCTIONS

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	3	5C	1	3	3C	2	5	7C	1	5	6C	2	6
Decrement	DEC	4A	1	3	5A	1	3	3A	2	5	7A	1	5	6A	2	6
Clear	CLR	4F	1	3	5F	1	3	3F	2	5	7F	1	5	6F	2	6
Complement	COM	43	1	3	53	1	3	33	2	5	73	1	5	63	2	6
Negate (2's Complement)	NEG	40	1	3	50	1	3	30	2	5	70	1	5	60	2	6
Rotate Left Thru Carry	ROL	49	1	3	59	1	3	39	2	5	79	1	5	69	2	6
Rotate Right Thru Carry	ROR	46	1	3	56	1	3	36	2	5	76	1	5	66	2	6
Logical Shift Left	LSL	48	1	3	58	1	3	38	2	5	78	1	5	68	2	6
Logical Shift Right	LSR	44	1	3	54	1	3	34	2	5	74	1	5	64	2	6
Arithmetic Shift Right	ASR	47	1	3	57	1	3	37	2	5	77	1	5	67	2	6
Test for Negative or Zero	TST	4D	1	3	5D	1	3	3D	2	4	7D	1	4	6D	2	5

TABLE 6 — BRANCH INSTRUCTIONS

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	3
Branch Never	BRN	21	2	3
Branch IFF Higher	BHI	22	2	3
Branch IFF Lower or Same	BLS	23	2	3
Branch IFF Carry Clear	BCC	24	2	3
(Branch IFF Higher or Same)	(BHS)	24	2	3
Branch IFF Carry Set	BCS	25	2	3
(Branch IFF Lower)	(BLO)	25	2	3
Branch IFF Not Equal	BNE	26	2	3
Branch IFF Equal	BEQ	27	2	3
Branch IFF Half Carry Clear	BHCC	28	2	3
Branch IFF Half Carry Set	BHCS	29	2	3
Branch IFF Plus	BPL	2A	2	3
Branch IFF Minus	BMI	2B	2	3
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	3
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	3
Branch IFF Interrupt Line is Low	BIL	2E	2	3
Branch IFF Interrupt Line is High	BIH	2F	2	3
Branch to Subroutine	BSR	AD	2	6

TABLE 7 — BIT MANIPULATION INSTRUCTIONS

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is Set	BRSET n (n=0...7)	—	—	—	2+n	3	5
Branch IFF Bit n is Clear	BRCLR n (n=0...7)	—	—	—	01+2*n	3	5
Set Bit n	BSET n (n=0...7)	10+2*n	2	5	—	—	—
Clear Bit n	BCLR n (n=0...7)	11+2*n	2	5	—	—	—

TABLE 8 — CONTROL INSTRUCTIONS

Function	Mnemonic	Inherent		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	10
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2
Stop	STOP	8E	1	2
Wait	WAIT	8F	1	2

TABLE 9 — INSTRUCTION SET

Mnemonic	Addressing Modes										Condition Codes				
	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			A	•	A	A	A
ADD		X	X	X		X	X	X			A	•	A	A	A
AND		X	X	X		X	X	X			•	•	A	A	•
ASL	X		X			X	X				•	•	A	A	A
ASR	X		X			X	X				•	•	A	A	A
BCC					X						•	•	•	•	•
BCLR									X		•	•	•	•	•
BCS					X						•	•	•	•	•
BEQ					X						•	•	•	•	•
BHCC					X						•	•	•	•	•
BHCS					X						•	•	•	•	•
BHI					X						•	•	•	•	•
BHS					X						•	•	•	•	•
BIH					X						•	•	•	•	•
BIL					X						•	•	•	•	•
BIT		X	X	X		X	X	X			•	•	A	A	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•
BRN					X						•	•	•	•	•
BRCLR										X	•	•	•	•	A
BRSET										X	•	•	•	•	A
BSET									X		•	•	•	•	•
BSR					X						•	•	•	•	•
CLC	X										•	•	•	•	0
CLI	X										•	0	•	•	•
CLR	X		X			X	X				•	•	0	1	•
CMP		X	X	X		X	X	X			•	•	A	A	A
COM	X		X			X	X				•	•	A	A	1
CPX		X	X	X		X	X	X			•	•	A	A	A
DEC	X		X			X	X				•	•	A	A	•
EOR		X	X	X		X	X	X			•	•	A	A	•
INC	X		X			X	X				•	•	A	A	•
JMP			X	X		X	X	X			•	•	•	•	•
JSR			X	X		X	X	X			•	•	•	•	•
LDA		X	X	X		X	X	X			•	•	A	A	•
LDX		X	X	X		X	X	X			•	•	A	A	•
LSL	X		X			X	X				•	•	A	A	A
LSR	X		X			X	X				•	•	0	A	A
NEG	X		X			X	X				•	•	A	A	A
NOP	X										•	•	•	•	•
ORA		X	X	X		X	X	X			•	•	A	A	•
ROL	X		X			X	X				•	•	A	A	A
ROR	X		X			X	X				•	•	A	A	A
RSP	X										•	•	•	•	•
RTI	X										?	?	?	?	?
RTS	X										•	•	•	•	•
SBC		X	X	X		X	X	X			•	•	A	A	A
SEC	X										•	•	•	•	1
SEI	X										•	1	•	•	•
STA			X	X		X	X	X			•	•	A	A	•
STOP	X										•	0	•	•	•
STX			X	X		X	X	X			•	•	A	A	•
SUB		X	X	X		X	X	X			•	•	A	A	A
SWI	X										•	1	•	•	•
TAX	X										•	•	•	•	•
TST	X		X			X	X				•	•	A	A	•
TXA	X										•	•	•	•	•
WAIT	X										•	0	•	•	•

Condition Code Symbols

- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry/Borrow
- A Test and Set if True, Cleared Otherwise
- • Not Affected
- ? Load CC Register From Stack
- 0 Cleared
- 1 Set

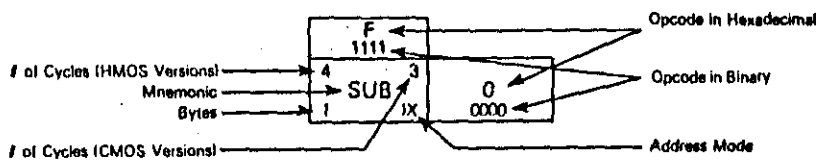
TABLE 10 — MC6805/MC146805 INSTRUCTION SET OPCODE MAP

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH(A)	INH(X)	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	BRSET0	BSET0	BRA	NEG	NEGA	NEGX	NEG	NEG	RTI		SUB	SUB	SUB	SUB	SUB	SUB	0
1	BRCLR0	BCLR0	BRN						RTS		CMP	CMP	CMP	CMP	CMP	CMP	1
2	BRSET1	BSET1	BHI								SBC	SBC	SBC	SBC	SBC	SBC	2
3	BRCLR1	BCLR1	BLS	COM	COMA	COMX	COM	COM	SWI		CPX	CPX	CPX	CPX	CPX	CPX	3
4	BRSET2	BSET2	BCC	LSR	LSRA	LSRX	LSR	LSR			AND	AND	AND	AND	AND	AND	4
5	BRCLR2	BCLR2	BCS								BIT	BIT	BIT	BIT	BIT	BIT	5
6	BRSET3	BSET3	BNE	ROR	RORA	RORX	ROR	ROR			LDA	LDA	LDA	LDA	LDA	LDA	6
7	BRCLR3	BCLR3	BEQ	ASR	ASRA	ASRX	ASR	ASR		TAX		STA	STA	STA	STA	STA	7
8	BRSET4	BSET4	BHCC	LSL	LSLA	LSLX	LSL	LSL		CLC	EOR	EOR	EOR	EOR	EOR	EOR	8
9	BRCLR4	BCLR4	BHCS	ROL	ROLA	ROLX	ROL	ROL		SEC	ADC	ADC	ADC	ADC	ADC	ADC	9
A	BRSET5	BSET5	BPL	DEC	DECA	DECX	DEC	DEC		CLI	ORA	ORA	ORA	ORA	ORA	ORA	A
B	BRCLR5	BCLR5	BMI							SEI	ADD	ADD	ADD	ADD	ADD	ADD	B
C	BRSET6	BSET6	BMC	INC	INCA	INCX	INC	INC		RSP		JMP	JMP	JMP	JMP	JMP	C
D	BRCLR6	BCLR6	BMS	TST	TSTA	TSTX	TST	TST		NOP	BSR	JSR	JSR	JSR	JSR	JSR	D
E	BRSET7	BSET7	BIL						STOP*		LDX	LDX	LDX	LDX	LDX	LDX	E
F	BRCLR7	BCLR7	BIH	CLR	CLRA	CLRX	CLR	CLR	WAIT*	TXA		STX	STX	STX	STX	STX	F
	LSB																MSB

Abbreviations for Address Modes

INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset
*	CMOS Versions Only

LEGEND



Indexed, 8-bit Offset — Here the EA is obtained by adding the contents of the byte following the opcode to that of the index register. The operand is therefore located anywhere within the lowest 511 memory locations. For example, this mode of addressing is useful for selecting the m-th element in an n element table. All instructions are two bytes. The contents of the index register (X) is not changed. The contents of (PC + 1) is an unsigned 8-bit integer. One byte offset indexing permits look-up tables to be easily accessed in either RAM or ROM.

EA = X + (PC + 1); PC ← PC + 2
Address Bus High ← K; Address Bus Low ← X + (PC + 1)
Where: K = The carry from the addition of X + (PC + 1)

Indexed, 16-Bit Offset — In the indexed, 16-bit offset addressing mode the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed 8-bit offset, except that this three byte instruction allows tables to be anywhere in memory (e.g., jump tables in ROM). As with direct and extended, the M6805 assembler determines the most efficient form of indexed offset — 8 or 16 bit. The content of the index register is not changed.

EA = X + [(PC + 1):(PC + 2)]; PC ← PC + 3
Address Bus High ← (PC + 1) + K;
Address Bus Low ← X + (PC + 2)
Where: K = The carry from the addition of X + (PC + 2)

Relative — Relative addressing is only used in branch instructions. In relative addressing the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is limited to the range of −126 to +129 bytes from the branch instruction opcode location. The Motorola assembler calculates the proper offset and checks to see if it is within the span of the branch.

EA = PC + 2 + (PC + 1); PC ← EA if branch taken;
otherwise PC ← PC + 2

Bit Set/Clear — Direct addressing and bit addressing are combined in instructions which set and clear individual memory and I/O bits. In the bit set and clear instructions, the byte is specified as a direct address in the location following the opcode. The first 256 addressable locations are thus accessed. The bit to be modified within that byte is specified with three bits of the opcode. The bit set and clear instructions occupy two bytes, one for the opcode (including the bit number) and the second to address the byte which contains the bit of interest.

EA = (PC + 1); PC ← PC + 2
Address Bus High ← 0; Address Bus Low ← (PC + 1)

Bit Test and Branch — Bit test and branch is a combination of direct addressing, bit addressing and relative addressing. The bit address and condition (set or clear) to be tested is part of the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte (EA1). The signed relative 8-bit offset is in the third byte (EA2) and is added to the PC if the specified bit is set or clear in the specified memory location. This single three byte instruction allows the program to branch based on the condition of any bit in the first 256 locations of memory.

EA1 = (PC + 1)
Address Bus High ← 0; Address Bus Low ← (PC + 1)
EA2 = PC + 3 + (PC + 2); PC ← EA2 if branch taken;
otherwise PC ← PC + 3

SYSTEM CONFIGURATION

Figures 20 through 25 show in general terms how the MC146805E2 bus structure may be utilized. Specified interface details vary with the various peripheral and memory devices employed.

FIGURE 20 — CONNECTION TO CMOS PERIPHERALS

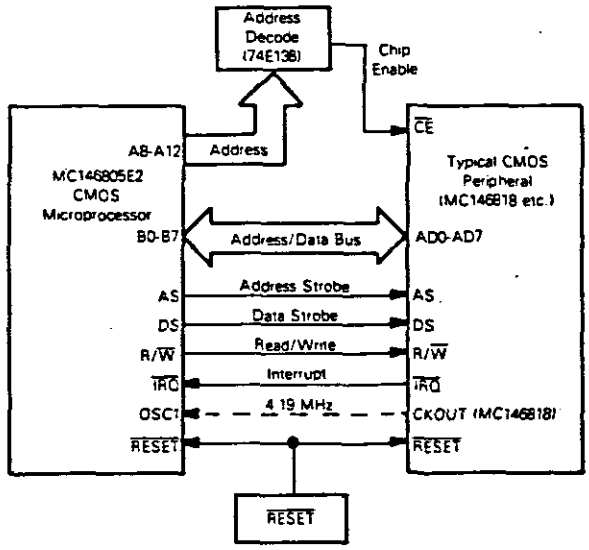


FIGURE 21 — CONNECTION TO CMOS MULTIPLEXED MEMORIES

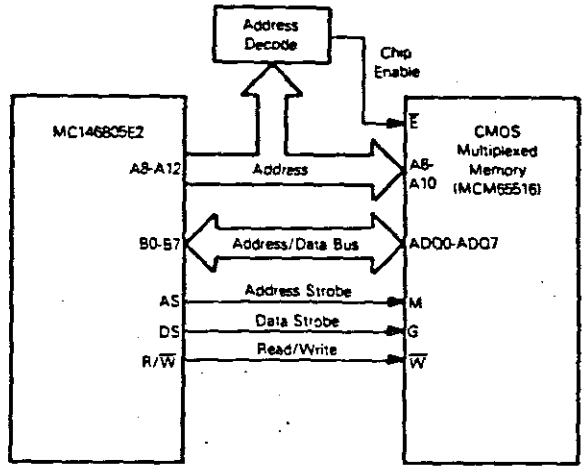
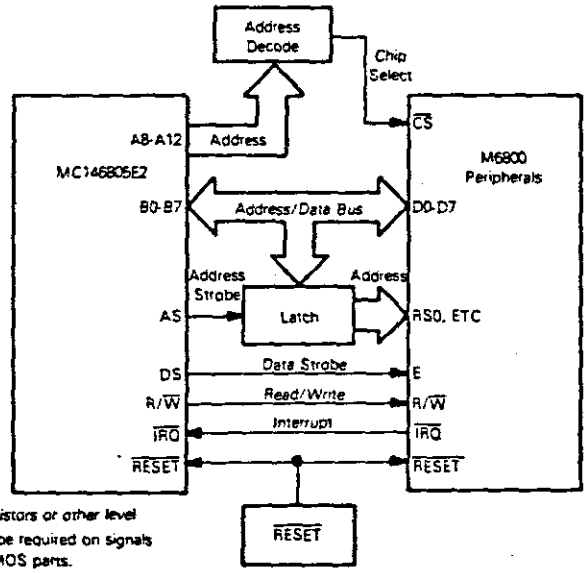


FIGURE 22 — CONNECTION TO M6800 PERIPHERALS



NOTE: In some cases, pullup resistors or other level shifting techniques may be required on signals going from NMOS to CMOS parts.

FIGURE 23 — CONNECTION TO LATCHED NON-MULTIPLEXED CMOS ROM AND EPROM

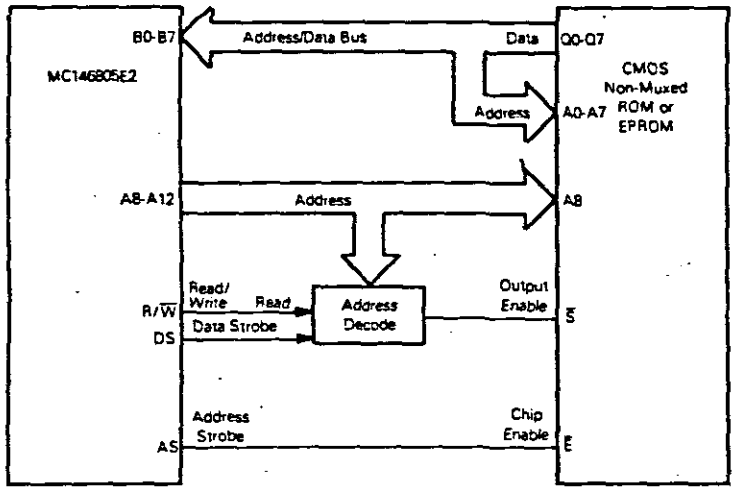


FIGURE 24 — CONNECTION TO STATIC CMOS RAMS

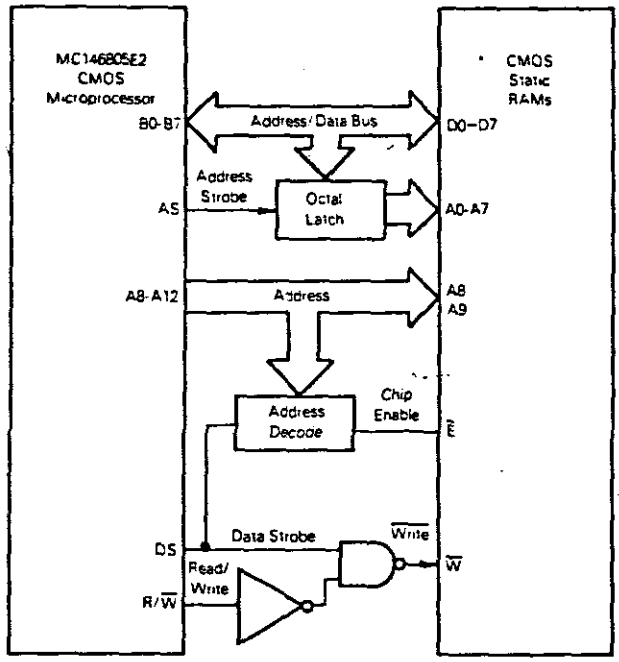


FIGURE 25 — CONNECTION TO LATCHED NON-MULTIPLEXED CMOS RAM

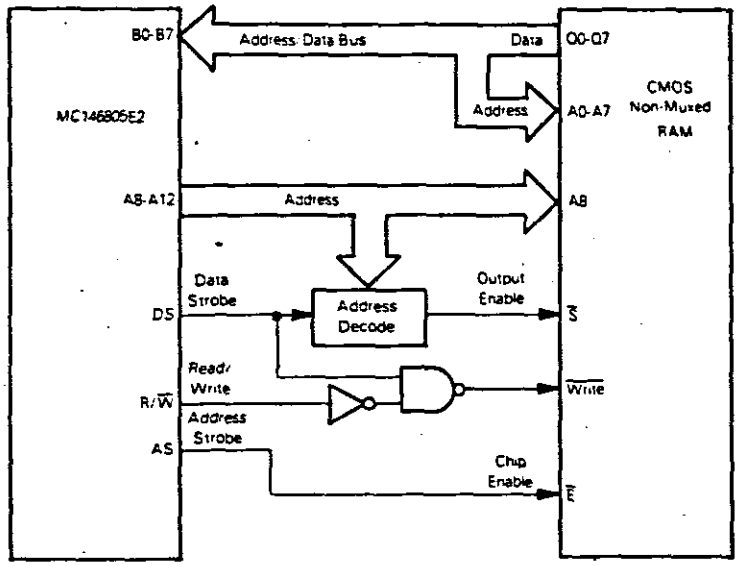


Table 11 provides a detailed description of the information present on the Bus, the Read/Write (R/$\overline{W}$) pin and the Load Instruction (LI) pin during each cycle for each instruction.

This information is useful in comparing actual with expected results during debug of both software and hardware as the control program is executed. The information is categorized in groups according to addressing mode and number of cycles per instruction.

TABLE 11 — SUMMARY OF CYCLE BY CYCLE OPERATION

Address Mode / Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Pin	LI Pin	Data Bus
Inherent						
LSR LSL ASR NEG CLR ROL COM ROR DEC INC TST	3	1 2 3	Op Code Address Op Code Address +1 Op Code Address +1	1 1 1	1 0 0	Op Code Op Code Next Instruction Op Code Next Instruction
TAX CLC SEC STOP CLI SEI RSP WAIT NOP TXA	2	1 2	Op Code Address Op Code Address +1	1 1	1 0	Op Code Op Code Next Instruction
RTS	6	1 2 3 4 5 6	Op Code Address Op Code Address +1 Stack Pointer Stack Pointer +1 Stack Pointer +2 New Op Code Address	1 1 1 1 1 1	1 0 0 0 0 0	Op Code Op Code Next Instruction Irrelevant Data Irrelevant Data Irrelevant Data New Op Code
SWI	10	1 2 3 4 5 6 7 8 9 10	Op Code Address Op Code Address +1 Stack Pointer Stack Pointer −1 Stack Pointer −2 Stack Pointer −3 Stack Pointer −4 Vector Address 1FFC (Hex) Vector Address 1FFD (Hex) Interrupt Routine Starting Address	1 1 0 0 0 0 0 1 1 1	1 0 0 0 0 0 0 0 0 0	Op Code Op Code Next Instruction Return Address (LO Byte) Return Address (HI Byte) Contents of Index Register Contents of Accumulator Contents of CC Register Address of Int. Routine (HI Byte) Address of Int. Routine (LO Byte) Interrupt Routine First Opcode
RTI	9	1 2 3 4 5 6 7 8 9	Op Code Address Op Code Address +1 Stack Pointer Stack Pointer +1 Stack Pointer +2 Stack Pointer +3 Stack Pointer +4 Stack Pointer +5 New Op Code Address	1 1 1 1 1 1 1 1 1	1 0 0 0 0 0 0 0 0	Op Code Op Code Next Instruction Irrelevant Data Irrelevant Data Irrelevant Data Irrelevant Data Irrelevant Data Irrelevant Data New Op Code
Immediate						
ADC EOR CPX ADD LDA LDX AND ORA BIT SBC CMS SUB	2	1 2	Op Code Address Op Code Address +1	1 1	1 0	Op Code Operand Data
Bit Set/Clear						
BSET n BCLR n	5	1 2 3 4 5	Op Code Address Op Code Address +1 Address of Operand Address of Operand Address of Operand	1 1 1 1 0	1 0 0 0 0	Op Code Address of Operand Operand Data Operand Data Manipulated Data
Bit Test and Branch						
BRSET n BRCLR n	5	1 2 3 4 5	Op Code Address Op Code Address +1 Address of Operand Op Code Address +2 Op Code Address +2	1 1 1 1 1	1 0 0 0 0	Op Code Address of Operand Operand Data Branch Offset Branch Offset
Relative						
BCC BHI BNE BEQ BCS BPL BHCC BLS BIL BMC BRN BHCS BIH BMI BMS BRA	3	1 2 3	Op Code Address Op Code Address +1 Op Code Address +1	1 1 1	1 0 0	Op Code Branch Offset Branch Offset
BSR	6	1 2 3 4 5 6	Op Code Address Op Code Address +1 Op Code Address +1 Subroutine Starting Address Stack Pointer Stack Pointer −1	1 1 1 1 0 0	1 0 0 0 0 0	Op Code Branch Offset Branch Offset First Subroutine Op Code Return Address (LO Byte) Return Address (HI Byte)

TABLE 11 — SUMMARY OF CYCLE BY CYCLE OPERATION (CONTINUED)

Address Mode / Instructions	Cycles	Cycles #	Address Bus	R/W Pin	LI Pin	Data Bus
Direct						
JMP	2	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Jump Address
ADC EOR CPX ADD LDA LDX AND ORA BIT SBC CMP SUB	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Operand
		3	Address of Operand	1	0	Operand Data
TST	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Operand
		3	Address of Operand	1	0	Operand Data
		4	Op Code Address + 2	1	0	Op Code Next Instruction
STA STX	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Operand
		3	Op Code Address + 1	1	0	Address of Operand
		4	Address of Operand	0	0	Operand Data
LSL LSR DEC ASR NEG INC CLR ROL COM ROR	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Operand
		3	Operand Address	1	0	Current Operand Data
		4	Operand Address	1	0	Current Operand Data
		5	Operand Address	0	0	New Operand Data
JSR	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Subroutine Address (LO Byte)
		3	Subroutine Starting Address	1	0	1st Subroutine Op Code
		4	Stack Pointer	0	0	Return Address (LO Byte)
		5	Stack Pointer − 1	0	0	Return Address (HI Byte)
Extended						
JMP	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Jump Address (HI Byte)
		3	Op Code Address + 2	1	0	Jump Address (LO Byte)
ADC BIT ORA ADD CMP LDX AND EOR SBC CPX LDA SUB	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address Operand (HI Byte)
		3	Op Code Address + 2	1	0	Address Operand (LO Byte)
		4	Address of Operand	1	0	Operand Data
STA STX	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Operand (HI Byte)
		3	Op Code Address + 2	1	0	Address of Operand (LO Byte)
		4	Op Code Address + 2	1	0	Address of Operand (LO Byte)
		5	Address of Operand	0	0	Operand Data
JSR	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Subroutine (HI Byte)
		3	Op Code Address + 2	1	0	Address of Subroutine (LO Byte)
		4	Subroutine Starting Address	1	0	1st Subroutine Op Code
		5	Stack Pointer	0	0	Return Address (LO Byte)
		6	Stack Pointer − 1	0	0	Return Address (HI Byte)
Indexed, No-Offset						
JMP	2	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
ADC EOR CPX ADD LDA LDX AND ORA BIT SBC CMP SUB	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Index Register	1	0	Operand Data
TST	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Index Register	1	0	Operand Data
		4	Op Code Address + 1	1	0	Op Code Next Instruction
STA STX	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Op Code Address + 1	1	0	Op Code Next Instruction
		4	Index Register	0	0	Operand Data
LSL LSR DEC ASR NEG INC CLR ROL COM ROR	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Index Register	1	0	Current Operand Data
		4	Index Register	1	0	Current Operand Data
		5	Index Register	0	0	New Operand Data
JSR	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Index Register	1	0	1st Subroutine Op Code
		4	Stack Pointer	0	0	Return Address (LO Byte)
		5	Stack Pointer − 1	0	0	Return Address (HI Byte)

TABLE 11 — SUMMARY OF CYCLE BY CYCLE OPERATION (CONTINUED)

Address Mode / Instructions	Cycles	Cycle #	Address Bus	R/W Pin	LI Pin	Data Bus
Indexed 8-Bit Offset						
JMP	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
ADC EOR CPX ADD LDA LDX AND ORA CMP SUB BIT SBC	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	Operand Data
STA STX	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Op Code Address + 1	1	0	Offset
		5	Index Register + Offset	0	0	Operand Data
TST	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	Operand Data
		5	Op Code Address + 2	1	0	Op Code Next Instruction
LSL LSR ASR NEG CLR ROL COM ROR DEC INC	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	Current Operand Data
		5	Index Register + Offset	1	0	Current Operand Data
		6	Index Register + Offset	0	0	New Operand Data
JSR	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	1st Subroutine Op Code
		5	Stack Pointer	0	0	Return Address LO Byte
		6	Stack Pointer − 1	0	0	Return Address HI Byte
Indexed, 16-Bit Offset						
JMP	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
ADC CMP SUB ADD EOR SBC AND ORA CPX LDA BIT LDX	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
		5	Index Register + Offset	1	0	Operand Data
STA STX	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
		5	Op Code Address + 2	1	0	Offset (LO Byte)
		6	Index Register + Offset	0	0	Operand Data
JSR	7	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
		5	Index Register + Offset	1	0	1st Subroutine Op Code
		6	Stack Pointer	0	0	Return Address (LO Byte)
		7	Stack Pointer − 1	0	0	Return Address (HO Byte)

TABLE 11 — SUMMARY OF CYCLE BY CYCLE OPERATION (CONTINUED)

Instructions	Cycles	Cycle #	Address Bus	$\overline{RESET}$ Pin	R/$\overline{W}$ Pin	LI Pin	Data Bus
Other Functions							
Hardware $\overline{RESET}$	5		$1FFE	0	1	0	Irrelevant Data
			$1FFE	0	1	0	Irrelevant Data
		1	$1FFE	1	1	0	Irrelevant Data
		2	$1FFE	1	1	0	Irrelevant Data
		3	$1FFE	1	1	0	Vector High
		4	$1FFE	1	1	0	Vector Low
		5	Reset Vector	1	1	0	Op Code
Power on Reset	1922	1	$1FFE	1	1	0	Irrelevant Data
		•	•	•	•	•	•
		•	•	•	•	•	•
		•	•	•	•	•	•
		1919	$1FFE	1	1	0	Irrelevant Data
		1920	$1FFE	1	1	0	Vector High
		1921	$1FFF	1	1	0	Vector Low
		1922	Reset Vector	1	1	0	Op Code
Instruction	**Cycles**	**Cycle #**	**Address Bus**	$\overline{IRQ}$ Pin	R/$\overline{W}$ Pin	LI Pin	**Data Bus**
$\overline{IRQ}$ Interrupt (Timer Vector $1FF8, $1FF9)	10		Last Cycle of Previous Instruction	0	X	0	X
		1	Next Op Code Address	0	1	0	Irrelevant Data
		2	Next Op Code Address	X	1	0	Irrelevant Data
		3	SP	X	0	0	Return Address (LO Byte)
		4	SP − 1	X	0	0	Return Address (HI Byte)
		5	SP − 2	X	0	0	Contents Index Reg
		6	SP − 3	X	0	0	Contents Accumulator
		7	SP − 4	X	0	0	Contents CC Register
		8	$1FFA	X	1	0	Vector High
		9	$1FFB	X	1	0	Vector Low
		10	$\overline{IRQ}$ Vector	X	1	0	Int Routine First

APPENDIX

MC146805E2 INTERRUPT CLARIFICATION

Under certain circumstances, the MC146805E2 (BP4xxxx & AW9xxxx) 8-bit Microprocessor Unit $\overline{IRQ}$ interrupt does not conform to the operation described in this Advanced Information Sheet (ADI-850R1).

1. The level sensitive $\overline{IRQ}$ mode, which is by far the most frequently used, is **FULLY OPERATIONAL**; thus, most MC146805E2 applications are unaffected. However, the edge-triggered $\overline{IRQ}$ interrupt mode **MIGHT NOT BE SERVICED** under certain programming circumstances; therefore, it is recommended that the edge-triggered mode not be used.

2. An interrupt-vector address **CAN BE** improperly generated in some circumstances. There is a possibility that when an external interrupt ($\overline{IRQ}$) and timer interrupt occur during the Wait mode (following a Wait instruction), address locations $1FF2 and $1FF3 are selected instead of vector locations $1FF6 and $1FF7. There are three specific examples listed below; two of these require no action and the third has a recommended solution.

   a. Those not using the Wait mode need not take any action.

   b. If the Wait mode is used without external interrupt ($\overline{IRQ}$ pin held high), no precautions are required.

   c. When $\overline{IRQ}$ can be active (low) during the Wait mode, the vector in locations $1FF6 and $1FF7 (the Wait mode Timer Interrupt Vector) should be duplicated in $1FF2 and $1FF3. In this way the circumstances that caused selection of the second vector do not disturb normal program execution.

On future MC146805E2 parts, no special actions will be necessary. If you have questions, contact your Motorola distributor or Motorola sales office, or contact Motorola Microprocessor Applications Engineering in Austin, Texas.

A N N E X U R E 5

D A T A S H E E T O N T H E
R E A L T I M E C L O C K
M C 1 4 6 8 1 8

MOTOROLA

MC146818

Advance Information

CMOS

(HIGH-PERFORMANCE SILICON-GATE COMPLEMENTARY MOS)

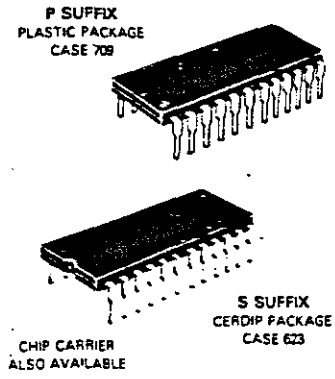
REAL-TIME CLOCK PLUS RAM

REAL-TIME CLOCK PLUS RAM (RTC)

The MC146818 Real-Time Clock plus RAM is a peripheral device which includes the unique MOTEL concept for use with various microprocessors, microcomputers, and larger computers. This part combines three unique features: a complete time-of-day clock with alarm and one hundred year calendar, a programmable periodic interrupt and square-wave generator, and 50 bytes of low-power static RAM. The MC146818 uses high-speed CMOS technology to interface with 1 MHz processor buses, while consuming very little power.

The Real-Time Clock plus RAM has two distinct uses. First, it is designed as a battery powered CMOS part (in an otherwise NMOS/TTL system) including all the common battery backed-up functions such as RAM, time, and calendar. Secondly, the MC146818 may be used with a CMOS microprocessor to relieve the software of the timekeeping workload and to extend the available RAM of an MPU such as the MC146805E2.

- Low-Power, High-Speed, High-Density CMOS
- Internal Time Base and Oscillator
- Counts Seconds, Minutes, and Hours of the Day
- Counts Days of the Week, Date, Month, and Year
- 3 V to 6 V Operation
- Time Base Input Options: 4.194304 MHz, 1.048576 MHz, or 32.768 kHz
- Time Base Oscillator for Parallel Resonant Crystals
- 40 to 200 µW Typical Operating Power at Low Frequency Time Base
- 4.0 to 20 mW Typical Operating Power at High Frequency Time Base
- Binary or BCD Representation of Time, Calendar, and Alarm
- 12- or 24-Hour Clock with AM and PM in 12-Hour Mode
- Daylight Savings Time Option
- Automatic End of Month Recognition
- Automatic Leap Year Compensation
- *Microprocessor Bus Compatible*
- MOTEL Circuit for Bus Universality
- Multiplexed Bus for Pin Efficiency
- Interfaced with Software as 64 RAM Locations
- 14 Bytes of Clock and Control Registers
- 50 Bytes of General Purpose RAM
- Status Bit Indicates Data Integrity
- Bus Compatible Interrupt Signals ($\overline{IRQ}$)
- Three Interrupts are Separately Software Maskable and Testable
  - Time-of-Day Alarm, Once-per-Second to Once-per-Day
  - Periodic Rates from 30.5 µs to 500 ms
  - End-of-Clock Update Cycle
- Programmable Square-Wave Output Signal
- *Clock Output May Be Used as Microprocessor Clock Input*
  - At Time Base Frequency ÷1 or ÷4
- 24-Pin Dual-In-Line Package
- Chip Carrier Also Available



PIN ASSIGNMENTS