Library Management System Using Bar Code Reader

PROJECT REPORT

Submitted by

J. MUSTAQ ALI

B. NANDA GOPAL

P. PADMA PRIYA

R. VENKATESH

Guided by

Mr. R. HARIHARAN, M.E.,
Lecturer, EEE Dept.

Sponsored by

PREMIER POLYTRONICS LTD, Coimbatore

IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF
BACHELOR OF ENGINEERING IN
ELECTRICAL & ELECTRONICS ENGINEERING
OF THE BHARATHIAR UNIVERSITY

1997 - 98

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

Kumaraguru College of Technology

COIMBATORE - 641 006.

CERTIFICATE

ACKNOWLEDGEMENT

First and foremost we express our sincere thanks to our beloved guide **Mr.Hariharan, M.E.,** Lecturer in Electrical and Electronics Engineering Department for his valuable guidance and immense support at all stages of this project work.

We wish to express our sincere gratitude to our Head of the Department of Electrical and Electronics Engineering **Dr.K.A. Palaniswamy, B.E., M.Sc., (Engg.) Ph.D., MISTE., C. Eng. (I), F.I.E.,** for extending us all help for the successful completion of the project.

We would like to thank our Principal **Dr.S.Subramanian, B.E., M.Sc., (Engg.), Ph,D., S.M. IEEE.,** for extending his valuable support towards our project.

We whole heartedly express our deep gratitude and sincere thanks to **Mr.K.K. Venkataraman, B.E., MBA.,** Vice President (R & D) and **Ms.A.Rajeshwari, B.E.,** Premier Polytronics Ltd.,Coimbatore for their valuable guidance, benevolent suggestion and for providing us with some of our requirement and remarkable help.

We extend our heartiest thanks to Mr.C.Radhakrishnan developing the software.

Last but not the least, we extend our thanks to our faculty members of R & D [PPL], beloved parents, ever loving friends who have helped us in many ways throughout the course of this project.

SYNOPSIS

SYNOPSIS

Library Transaction system is an important part of library management system concerned with issue and return of books from the library.

The present system employed is a manual process which is laborious and time consuming. The entry and preservation of data in a bulk also results in ambiguity.

Hence in order to save time and perform the task efficiently it is proposed to developed a microcontroller based **Automatic transaction system** in this project. The μc accepts input from bar code reader. The Interfacing circuit between bar code reader and the μc is designed and the software for the μc is developed.

The above system has a limited data base and hence to utilise the system for even larger data entries we have further expended it using PC. By using PC, the entry and retrieval of data becomes even simpler and more effective. With the use of PC, the library transaction system is expanded as library management system.

CONTENTS

CERTIFICATE i

ACKNOWLEDGEMENT ii

SYNOPSIS iii

CONTENTS iv

I. INTRODUCTION 1.1

II. BAR CODE READER

2.1 About bar code reader 2.1

2.2 Types of bar code reader 2.1

2.3 Details of ITF 2.3

2.4 Functions / Process 2.7

III. HARDWARE DETAILS

3.1 Microcontroller Details 3.2

3.2 Memory 3.13

3.3 Buffer and Latches. 3.15

3.4 Keyboard Interface 3.17

3.5 Display Interface 3.17

3.6 Timer 3.19

3.7 Decoder 3.22

IV. CIRCUIT DESIGN

4.1 Logic diagram 4.1

4.2 Circuit Operation 4.1

V. SOFTWARE

5.1 About VB5 5.0

5.2 Flow chart 5.0

5.3 Programs 5.1

VI. CONCLUSION

REFERENCE

APPENDIX

PHOTOGRAPH

INTRODUCTION

CHAPTER · I

INTRODUCTION

Library transaction is an important part of library management system. effective transaction is very important inspective of the size of the library, number of books in the library, numbers of persons enrolled in the library. The library transaction system thus designed can be implemented in all fields namely in college libraries, departmental libraries, research and development libraries, Industry libraries, Public libraries and so on.

The present never ending combats with the microprocessor and the evergrowing electronic industry has made the advent of micro controller or fullfledged utility. Even the arduous jobs now appear easy and feasible by implementing micro controllers. Hence the system designed to perform the transaction consists of micro controller, bar code reader, real time clock, key board display interface and memory.

A Micro controller is nothing but a Microprocessor with exra peripheral on-chip facility like inbuilt memory, and I/O lines. It works at a much higher speed and has a user friendly software.

For best utilisation of the Micro controller, we have opted C - 8032 which has the features which an required for our project.

To start with, the Micro controller is the whole Master mind behind all the operations and controls the operations of all other peripheral. The peripherals include.

EPROM	IC 27C512
RAM	IC 62256
RTC	IC 6242
Decoder	74LS138
Key board interface	IC8279
Programmable peripheral interface	IC8155
Buffer	74HCT245
Latch	74HCT573

EPROM and RAM constitute the memory and is used to store programs and data.

The RTC IC6242 enables continuos display of time, date, month. It is made on line by providing a back up for it. it is also utilised for automatic entry of the date and time when a book is returned or issued.

The keyboard and display interface is used to co-ordinate keyboard and LCD display controller.

The keyboard mins used is 4 x 4 matrix keyboard having 16 keys which is interfaced with IC8279.

The LCD 124 x 240 dot matrix display is interfaced with dot matrix display controller and PPI , IC8155.

All the above components are co-ordinated and implemented efficiently by the software. All the components are connected with the Micro controller using memory mapping technique.

The above project can be still simplified and made user friendly with flexible database by using a pc, barcode reader and other accessories. The main advantage in developing the system using pc is availability of large memory, high speed retrieval of data, quick transaction and large database can be created. Here to account the above said advantage, the software has been developed using VB5 as front end and MS access as back end.

The bar code reader forms the innovative part of our project. The bar code present in the identity card of the company staff is scanned using bar code sensor and is decoded by the decoder in the mother board.The presence of barcode reader makes the task of the librarian even more easier and error free.

BAR CODE READER

CHAPTER II

BAR CODE READER.

Bar codes are a pattern of bars and spaces of varying widths that represent digits, letters, other punctuations symbols for identifying an item/object. It is an identification tool that provides a precise and timely data support for Sophisticated management systems. Bar code increases accuracy in data collection and enhances business efficiency. The way the bars and spaces are arranged on the white label are called the symbology. Number of such symbologies have been evolved to suit different applications.

2.1 TYPES OF BAR CODE

There are many different types of bar code but only four are in common use.

a) EAN/UPC

b) Inter leaved Two of Five

c) Code 39

d) Coda bar and

e) Code 128

2.1.1 EAN/UPC:

These codes are commonly seen on grocery items in super market in the UK EAN stands for European Article Number and the American equivalent is the Universal product code. There are of two types EAN 8, EAN 13 and they are usually of numbers.

2.1.2 CODE 39 :

This code is also known as code 3 of 9. This code can be of any length and all the capital letters of the alphabet are encoded. This code is always started and finished with an asterisk. These codes are widely used in stock markets.

2.1.3. CODA BAR :

It is widely used bar code and can be of any length but only numeric and some character like $, ; , / etc can be conceded. It is used in blood banks, libraries, medical industry etc.

2.1.4 INTERLEAVED 2 OF 5 :

The bar codes employed in the identification card issued in our industry is of this type namely interleaved 2 of 5. It is of numeric type of

Hence this code must have an even numbers of

VALUE OF DIGIT	BAR COMBINATIONS
0	00110
1	10001
2	01001
3	11000
4	00101
5	10100
6	01100
7	00011
8	10010
9	01010

2.3 DIAGRAM OF ITF :

In fig (1)

- Flag represent the country in which codes are made.
- Flag is specific to each country.
- The next 5 digits represents the company numbers which items are manufactured.
- The next 5 digits known as case number represents the product/item. Each product has unique case number

variable length. It must have an even number of digits. ITF is used for industrial applications where a numeric only code is sufficient and it is also used in the retail environment for OUTER CASE Marking. It is a continuos counter checking numeric symbology employing 2 element width. Each character has 5 bars and 5 spaces respectively.

2.2 DESCRIPTION OF INTERLEAVED 2 OF 5

- Interleaved 2 of 5 codes has five bars and four embedded and one trailing space to represent an individual digit.
- Two of these five bars are wide and three of them are narrow hence the name 2 of 5
- The ratio of wide to narrow bars is typically 2:5:1 but this range can vary from 2:2:1 to 3:1
- The code represents digits in the following way with a 0 representing a narrow bar and a 1 a wide bar
- The first digit is represented by first five bars
- The second digit is represented by first five spaces
- The third digit is represented by bars 6 to 10
- The fourth digit is represented by spaces 6 to 10

- The last digit represent the arithmetically calculated check digit.
- ITF has also been adopted by shoe and jewellery retailing trades because of its variable length characteristics.

2.4 MERITS OF THE BAR CODE SYSTEMS

1) Bar codes increases productivity, reduce human errors, cut down costs and improve service and quality.

2) Bar coding is the most accurate and least expensive way to incorporate data into the computer.

3) Manual data collection tends to be inaccurate, and on an average 2-3 errors occurs for every 100 to 300 key strokes. But with a high quality bar code systems, occurrence of errors are approximately one for every 300 million characters.

4) Bar code technology is easy to use and is a reliable one. Also there can be no chance of any distortion of coded data.

5) Apart from above, the above systems facilitates the optional use of resources, which is the key to every successful business.

2.5 PRINCIPLE COMPONENTS OF BAR CODE SYSTEMS

A Bar code system typically consists of the following equipment's namely

a) Bar code labels printed in house or outside by a commercial house.

b) Scanner to read the bar code label's.

c) Decoder which translates collected information into binary coded data, housed in mother board.

d) A host computer or a controller can be employed to reduce in the scanned output from decoder.

2.6 BRIEF SUMMARY OF THE PROCESS INVOLVED:

Initially, irrespective of bar code symbologies, bar codes are scanned using a LASER or LED. The sensor used for scanning bar codes uses laser beam, which reflects the light off the label to a photo sensitive receptor which reads the codes by differentiating between the black bars and white spaces.

The light falling on the black bars are not reflected to the detector hence taken as binary while that falling on the white spaces is reflected to the detector hence taken as binary 0. This information is converted into series of voltage variations. The voltage impulses are digitalised using A/D converter the output of which is fed to the decoder which uses a mathematical algorithm to translate the digital impulses into binary codes and transmit the encoded data message to a PC.

2.7 PROCESS FLOW

Scanner/sensor using laser beam

↓ (reflects light off the label)

Photo sensitive Receptor

↓ (differentiates between black bar and white spaces)

Detector

↓ (Voltage impulse)

A/D Converter

↓ (Digital output)

Decoder

↓ (Encoded data usually ASCII)

MC or PC

2.8 SENSOR TYPE USED AND ITS SPECIFICATIONS

HEWLETT PACKARD, a well known electronic company has developed Industrial digital bar code slot reader which provides excellent scanning performance, a wide variety of bar coded cards and badges using LASER beam and produces digital pulses of varying width as output.

This digital bar code slot reader is highly effective alternative to keyboard data entry, greater data accuracy than optical character recognition and is highly to electro magnet fields.

The epoxy coated metal case with damp - proof mounting systems make these slot readers ideal choices for security access control, attendance recording and card recording applications.

The HBCS- 7100 consists of precision optical system, analog amplifier, digitizing circuit and an output transistor. These elements provide digital output from a single 4.5 to 5.5v.d.c power supply. The open collector transister requires a pull up resistor for proper operation.

The non reflecting black has results in logic 1 and that of reflecting white space results in logic 0. After power up, the slot reader will be fully operational after a period of 6 seconds. During operation, the slot reader assumes logic low state after a short period if no bar code is scanned. The slot reader connector provides a shield which should be terminated to logic ground. The recommended logic interface for the slot reader is shown. Fig(2) and Fig(3).

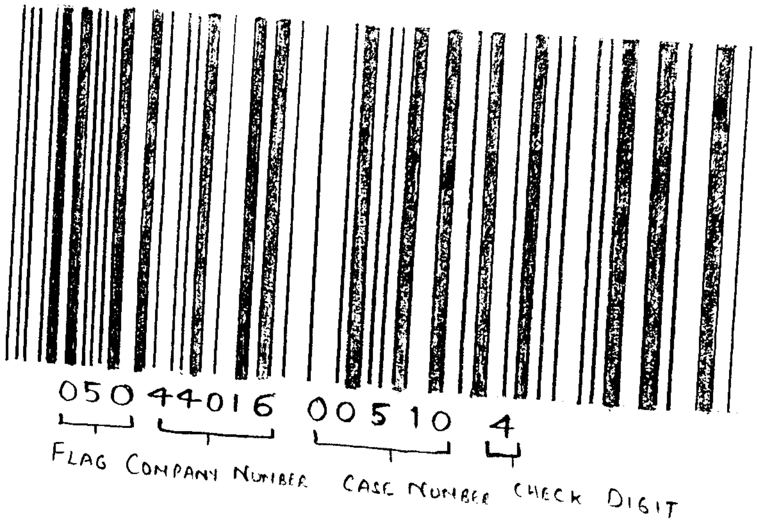
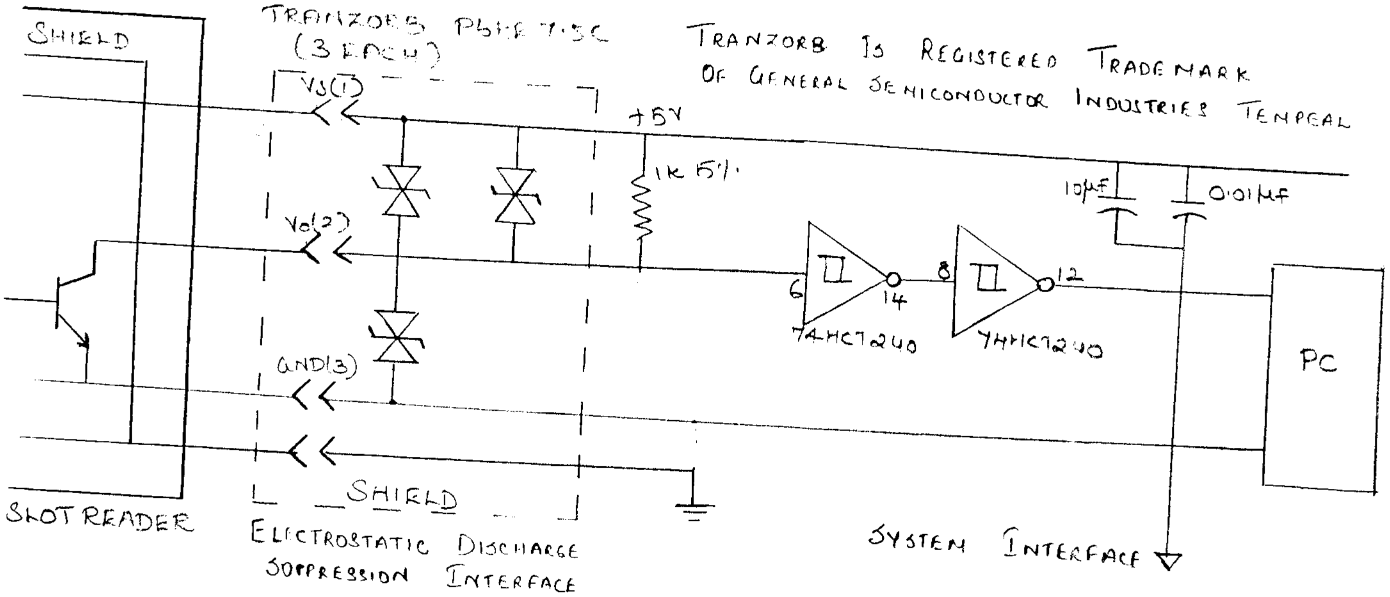


FIG: 1

INTERLEAVED TWO OF FIVE BAR CODES



Tranzorb Is Registered Trademark Of General Semiconductor Industries Tempeal

Fig 2.2

Interface Specifications

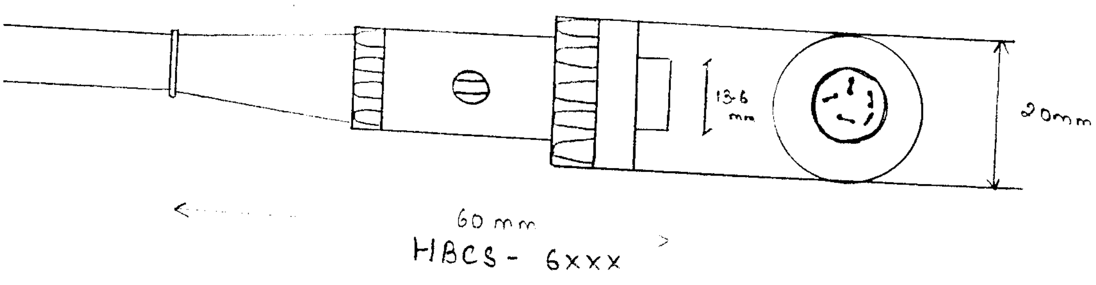


Fig 2.3

Connector Specification

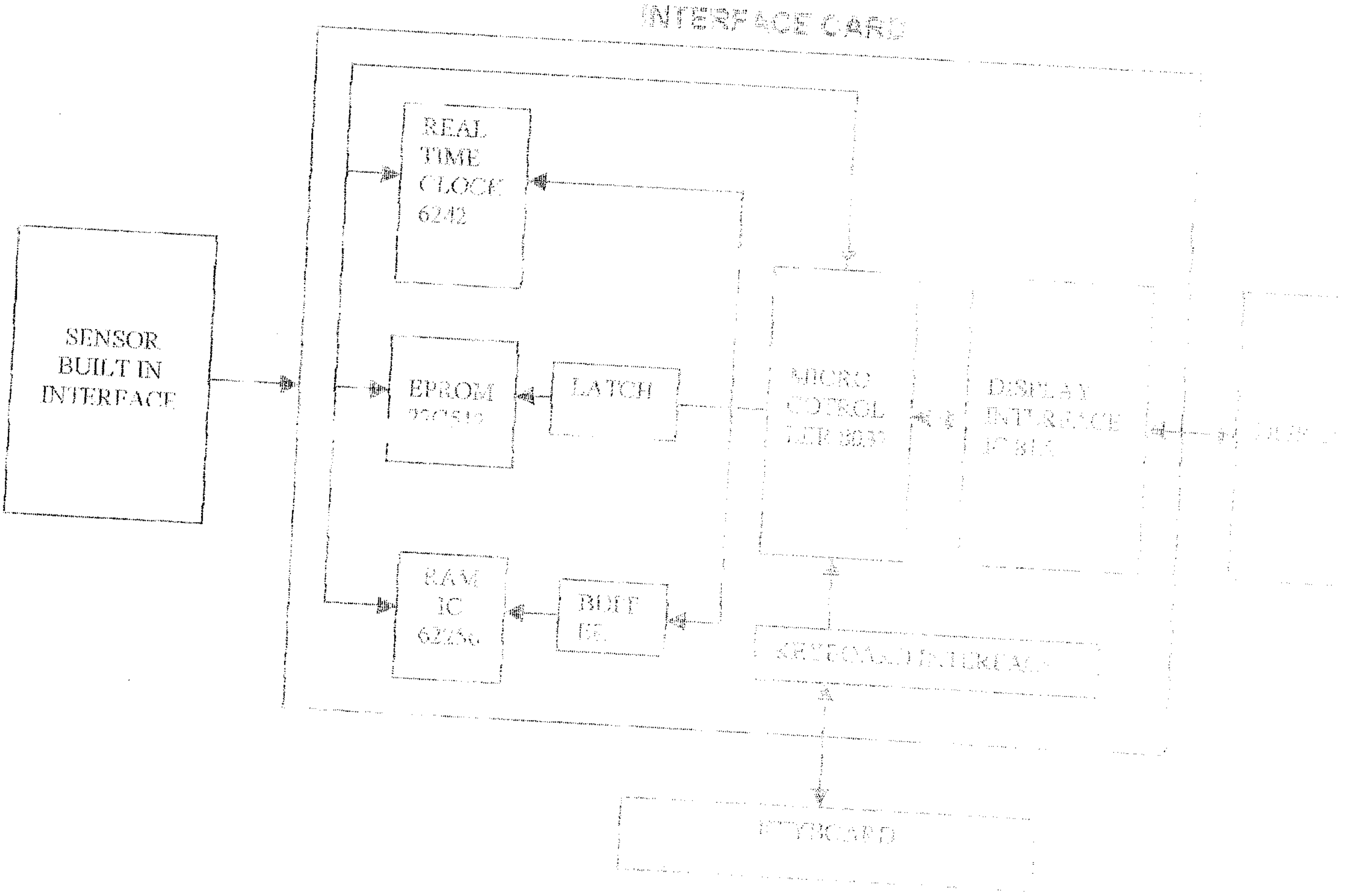


Fig 2.2 DETAIL BLOCK DIAGRAM OF MICROCONTROLLER BASED LIBRARY TRANSACTION SYSTEM

CHAPTER III

HARDWARE DETAILS

3.1 INTRODUCTION

This chapter is concerned with the details of the hardware being employed in developing micro - controller based system design. Hardware being employed are

a. Micro controller - Heart of our system
Employed for controlling and processing system

b. Memory - To store programmes and data

c. Buffer and latches - Buffer : To provide sufficient current drive signals
Latch : It is used to interface output devices. It is used to differentiate address and data bus.

d. Keyboard Interface - It includes an LCD dot matrix display driven and PPI chip 8155 and is used to interface the display ad the micro controller.

e. Display Interface - It includes an LCD dot matrix display, display driven and PPI chip 8155 and is used to interface the display and the micro controller.

f. Timer - It is used to continuously display the time.

g. Decoder - It is used to decode the input signal and accordingly select the required hardware chip.

3.2 MICRO CONTROLLER : (8032)

3.2.1. Introduction :

The micro controller 8032 consists of following techniques in its core memory.

a. 8 bit CPU optimised for control applications

b. Extensive boolean processing (signal bit-logical) capabilities.

c. 64K program memory address space.

d. 64K data memory address space.

e. 8K bytes of on chip data memory.

f. 256 bytes of on chip program memory.

g. 32 - bidirectional and individually addressable i/o lines.

h. Three 16 bit timer / counters

i. Full duplex UART

j. 6 source / 5 vector intercept structure with provity levels.

k. On chip clock oscillator.

3.2.2 Hardware Features of 8032

MC-8032 has four I/o parts, which are bi-directional each port consists of a latch, and output driver and an input buffer. The details of each port are as follows.

PORT 0

Port 0 is an 8 bit open drain bi-directional I/O port. As an output port each pin can sink 8 LS TTL inputs. Port 0 is multiplexed low order address and data bus during access to external program and data memory. It also receives the code bytes during programming the EPROM parts and outputs part the code during program verification of the ROM and EPROM parts. Also Port 0 pins that have is written to their can be used as high impedance inputs.

PORT 1 :

Port 1 is an 8 bit bi-directional I/o port with internal pull ups. The port 1 output buffers can sink / source 4 LS TTL input. Port 1 can also receive the low-order address bytes during programming of EPROM parts.

PORT 2 :

Port 2 is an 8 bit bi-directional I/o port with internal pull ups. The port 2 output can sink / source 4 LS TTL inputs. It can also be used to exit higher order address byte and data byte during access to external program and data memory.

PORT 3 :

Port 3 is an 8 bit bi-directional I/o port with internal pull ups. Port 3 output buffer can sink / source 4LS TTL inputs. In can be used as inputs. Port and can serve the various features of the 8032 controller.

PORT PIN	FUNCTION
P3.0	RXD (Serial Input port)
P3.1	TXD (Serial Output port)
P3.2	INTO (External Interrupt 0)
P3.3	INT 1 (External Interrupt 1)
P3.4	TO (Timer 0 external input)
P3.5	T1 (Timer 0 external input)
P3.6	WR (External data memory write strobe)
P3.7	RD (External data memory read strobe)

Registers

Registers are small memories within the CPU. The 16 special function registers can hold frequently used data and intermediate results. Data may be saved in a register until a bus or other unit is ready to receive it or until the program requires it. The registers are distinguished as follows.

Accumulator :

The accumulator is a eight bit register that is a part of the arithmetic logic unit. The results of an arithmetic logic operation is also stored in accumulator which is identified as register A.

Program Counter :

This register is a memory pointer. Memory locations were 16 bit addresses. The microcontroller uses these register to sequence the executive of instructions. The function of the program counter is to point to the memory address from which the next byte is to be fetched. When a byte is being fetched the program counter is incremented by one point and points to the next memory location. The 8032 program counter is 16 bit wide and can access upto 64KB of program or data memory.

FLAG Register :

The process flags are combined in one status word (PSW)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CY	AC	FO	RS1	RSO	OV	X	P

CY	-	Carry flag
AC	-	Auxiliary carry flag
FO	-	Indicator
RSI	-	Register bank
RSO	-	Register bank
OV	-	Overflow flag
X	-	Don't care
P	-	Parity flag

Stack Pointer :

The stack pointer is also a 16 - bit register used as memory pointer. It points to a memory location in the R/W memory, called stack which is defined by loading a 16 - bit address in the stack pointer. The stack pointer is set to fill upwards from specified location. Pushing / popping of register / memory location can be done with one byte of instructions.

Serial Data Buffer :

The Serial data buffer is actually two separate register, a transmit buffer and a receive buffer register. When data is moved to SBUF, it goes to transmit buffer where it is held for serial transmission, when data is moved from SBUF, it comes from the receive buffer.

Time Register :

Register pairs [THO, TLO], [TH1, TL1] and [TH2, TL2] are the 16 bit counting registers for Timer / counter 0, 1 and 2 respectively.

Control and Status Register :

Special function registers IP, IE, TMOD, TCON, SCON nd PCON contain control and status bits for the interrupt system, timers and the serial port.

Stack :

Stack is also a form of temporary register and is located in RAM memory. Data is pushed into or popped out the stack location on the LIFO basis. The stack pointer maintains the address of the last byte entered into the stack and is decremented every time data is pushed into the stack and incremented each time data is popped out of the stack. The stack pointer is always incremented or decremented by two bytes.

Data Pointer Register : (DPTR)

Data pointer register consists of a high byte [DPM] and a low byte [DPL] and can be manipulated as 16 - bit register or as two independent 8 - bit register. It intend function is to hold a 16 - bit address.

Internal Timers / Counters :

There are three Timer / counter register namely Timer 0, Timer 1 and Timer 2. Two 16 - bit register are used. In the 'Timer' function, the register is incremented every machinecycle and in the 'Counter' function, the register is incremented in response to 1 to 0 transition at the external input pin T0, T1. If a 12MHz crystal is used, each tick is microsecond. Longer time delays can be arranged by using the timer repeatedly in a software loop. Timer / Counter can be operated in four models and can be selected by control bits in special function register.

Interrupts :

The 8032 provides six interrupt sources each of which can be programmed to one of the two priority levels. The six interrupt sources are listed below.

SOURCE	DESCRIPTION
INTO	External request from P 3.2 pin.
TIMER 0	Overflow from Timer 0 activates interrupt request flag TF0.
INTI	External request from P 3.3 pin.
TIMER 1	Overflow from Timer 1 activates interrupt request flag TF1.
TIMER 2	Overflow from Timer 2 activates interrupt request flag TF2.
SERIAL PORT	Completion of transmission or reception of one serial frame activates request flag TI [on transmission] RI [on reception]

Each source can be individually enabled or disabled by setting or clearing a bit in special function Register IE.

Serial Interface :

The serial port is full duplex meaning it can transmit and receive simultaneously. It can also receive buffered data meaning it can commence reception of a second byte before a previously received byte has been read from the receive register. The serial port are both accessed at special function register SBUF. A write to SBUF loads the transmit register and a read to SBUF accesses a physically separate receive register.

THE SERIAL PORT CAN OPERATE IN 4 MODES.

MODE 0

Serial data enters and exits through RXO, TXO outputs. 8 bits are transmitted and received. The baud rate is fixed at 1/12 the oscillator frequency.

MODE 1

In mode 1, 10 bits are transmitted and received through RXD, TXD outputs. The 10 bits include a start bit, 8 data bits (LSB first) and a stop bit. The baud rate is variable.

MODE 2

In mode 2, 11 bits are transmitted and received through RXD, TXD output. The 11 bits include a start bit, 8 data bits, a programmable 9th start bit and a stop bit. On transmit the 9th data bit (RB8) can be assigned the value of 0 or 1. On receive, the 9th data bit goes into RB8 in special function register and a stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency.

MODE 3

In mode 3, 11 bits are transmitted and received through RXD, TXD outputs. The 11 bits includes a start bit, 8 data bits, a programmable 9th data bit and a stop bit.

The bar code reader connector pin is interfaced with Micro controller through RXO, TXO pins as shown below. The serial port is operated in MODE 0.

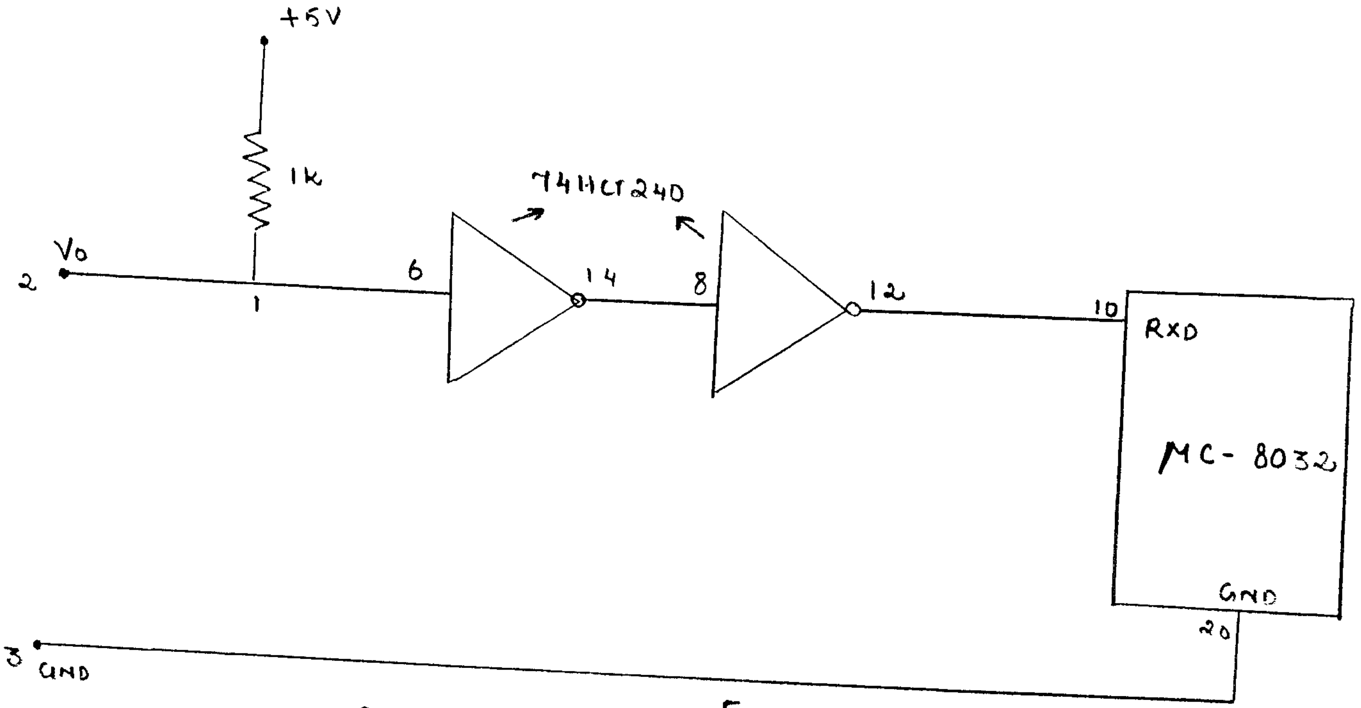


FIG: 3.1

Connector Pin Of BCR

1 - VCC [5V]

2 - Vo

3 - GND

CASE - GND

Serial Interface Of BCR With MC-8032

3.3 MEMORY :

Memory in general is used to store information. There are two types of memory namely RAM (Random Access Memory) and EPROM (Erasable Programmable Read Only Memory). RAM chip, IC 62256 is used to store data namely ID number of staff and their names, access number of the book and corresponding title of the book. The entire program to enable operation of 8032 is stored in EPROM chip.

The details of RAM and EPROM IC chips used in the design are as follows.

3.3.1 EPROM :

The microchip technology IC 27C512 is a CMOS 512K bit EPROM.

The EPROM IC 27C512 is used to store the microcontroller programs. It has the following features namely.

1. High speed performance - 120ms
2. Low power consumption - 40 mA active current

   100μA stand by current
3. Two programming algorithms are available

   a. Fast programming

   b. Rapid pulse programming

4. It is available in

i. 28 pin dual-in-line package

ii. 32 pin chip carrier.

This is a very high speed device which allows the Micro controller to run at full speed without the need for WAIT states.

The block diagram and IC pin details are given in the appendix.

EPROM can be operated in various modes.

1. Read mode :

This mode is accessed when chip enables signal CE pin is low and output enable pin OE / VPP is low. During this mode, data is gated to the output pins.

2. Stand by mode :

When CE goes high, then the mode is called the stand by mode when by no operation is performed.

3. Exase mode :

In this mode, when the EPROM chip is exposed to ultra violet light, the entire memory matrix is erased.

4. Programming mode :

There are two programming modes being employed

1. Fast Programming
2. Rapid pulse algorithm

5. Inhibit Mode :

In this mode, other devices can be programmed. By pulsing CE line low on a particular device, that device will be programmed. All other devices with CE held high will not be programmed.

3.3.2 RAM (Random Access Memory):

The micro chip technology IC62256 is a cmos 256 K bit RAM & is used to store the entire data base. The entire data base stored in RAM can be read / written by 16 address / data lines from μc. The A15 address line of μc is used as chip enable signal whenever it goes low, RAM is selected.

3.4.1 Latch :

The chip used for latch is 74HCT573. It is high speed octal transparent 3 - state latch manufactured with silicon gate CMOS technology. They posses the low power consumption of standard CMOS integrated circuits, our well as the ability to drive 15LSTTL devised.

The outputs are transparent to the inputs when the latch enables IE is high. when the CE goes low the data is latched.

The output enables OE Controls the 3 - state outputs. When the output enable OE is high the output are in the high importance states. The latch operation is independent to the state of the output enable.

Application

Latch is used to connect the bidirectional Address Data bus to the RAM. It converts and provides only Address lines A_0 to A_7 to the RAM & EROM. When the ALE goes high the MSB of the 16 bit Address bus is sent to RAM & EPROM. The pin details & Block diagram are given in the appendix.

3.4.2 Buffer

The chip used as buffer is 74HCT245. It is a high speed octal 3-state buffer intended for 2 way asynchronous communication between data buses. They have high drive current output which enable high speed operation while driving large bus capacitances. They provide low power consumption of standard CMOS circuits with speeds and drive capability comparable to that of LSTTL circuits.

The buffer allow data Transmission from the bus A to bus B or from the bus B to bus A. The logic level of the pin (DIR) determines the direction. When OE pin is high, the data in I/O ports goes to high impedance state.

Application

It is used to connect the Address lines from the latch A_0 to A_7 to the Address lines of EPROM. It is used single direction only. Data transfers from A bus to B bus is only possible. Therefore it is clear that output enable (oe) should be low and directional pin should be high during buffer operations. The pin details & block diagram of IC 74HCT245 is given in appendix

3.5 KEYBOARD INTERFACE :

IC 8279 is the general purpose programmable keyboard interface device designed for use with μc.

Salient features :

- Scanned keyboard matrix
- Scanned sensor mode
- Strobed input entry mode
- 8 character keyboard FIFO
- 2 Key lock out or N key rollover with contact debounce
- Mode programmable scan timing
- Interrupt output on key entry.

3.6 DISPLAY INTERFACE

General Description

8155 is the general purpose programmable display interface device designed for use with micro controllers.

8155 is used for Display Interface. 8155 is RAM & I/O ports implemented in N. channel Depletion mode, silicon gate technology (HMS) to be used in micro controller systems. The RAM portion is designed with 2048 static cells organised as 256x8.

The I/O portion consists of 3 general purpose ports. One of the ports can be programmed to be status pins and other 2 ports to operate in Handshake Mode.

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

Salient features:

1. Completely static operation

2. Internal Address latch

3. 2 programmable 8- bit I/O port

4. 1 programmable 6- bit I/O port

5. Programmable 14- bit Binary Counter/Timer

6. Multiplexed Address <Data Base.

Application:

Out of 3 ports. A port is connected with BCR pins to display the data output from BCR immediately.

C port is connected as Multiplexed Address/databus to the keyboard to display the data that we type.

The B port is connected to the Display controller.

The function of 8155 is it receives the input from the BCD and displays it in the LCD controller display. All the other details to be displayed in the LCD is passed through the 8155.

3.7 REAL TIME CLOCK/CALENDER

General description

MSM 6242 is the silicon gate CMOS Real time clock/calendar which is used in direct bus connection of Micro controller in our project. An on-chip 32.768 KHz crystal Oscillator time base is divided to provide addressable 4 bit I/O data for SECONDS, MINUTES, HOURS, DAY OF WEEK, DATE, MONTH AND YEAR

The MSM 6242 normally operate from a 5V+ 10% supply at -30°C to 85°C.

Salient features

- 4 bit data bus
- 4 bit address bus
- 32.768KHz crystal controlled operation.
- 12/24 hour format
- Auto leap year
- ± 30 second error correction

- Single 5V supply
- Battery Backing up to V_{DD} = 2.0V
- Low power dissipation

  20μW Max at V_{DD} = 2V

  150μW Max at V_{DD} = 5V
- 18 pin plastic DIP package

DESCRIPTION OF REGISTERS:

- These are abbreviations for SECOND1, SECOND10, MINUTE1, MINUTE10, HOUR, HOUR10, DAY1, DAY10, MONTH1, MONTH10, YEAR1, YEAR10 AND WEEK. These values are in BCD notations.
- All register are logically positive.
- If data is written which is out of the clock register data limits, it can result in erroneous clock data being read back.
- PM/AM,h_{20},h_{10}.

In the mode setting of 24 hours, PM/AM bit is ignored, while in the setting of 12 hour mode h_{20} is to be set. Otherwise it cause a discrepancy. In reading out the PM/AM bit in the 24 hour mode, it is continuously read out as 0. In reading out h_{20} bit in the 12- hour mode, 0 is written this bit first, then it is continuously read out as 0 unless 1 is being written in to this bit.

- Register Y_1, Y_{10} is used for leap year adjustment
- The register W data limits are 0-6

W_4	W_2	W_1	Day of week
0	0	0	Sunday
0	0	1	Monday
0	1	0	Tuesday
0	1	1	Wednesday
1	0	0	Thursday
1	0	1	Friday
1	1	0	Saturday

* There are also control D Register, control E Register and Control F register which are used for various other operations.

RTC in our project

The RTC or MSM 6242 is used for continuos display of time and calender in our project. This is made feasible by the back up provided which keeps the RTC alive even when the power is switched off. The time and date datas available is readily used by the micro controller to record the timings of the issue or return of a book by the employees. The time and date will flash continuously on the right hand top of the LCD display controller matrix.

3.8 DECODER

The chip used as decoder is 74LS138. It is 1 of 8 line decoder and is often used for address decoding. It is a 16 pin package and consists of 3 inputs A,B,C and 8 output Y0 through Y7. There are three chip select pins G1, G2 and G3. Any one of the outputs Y0 through Y7 will assume low active only when G1 is high, G2 & G2 are low. The particular low output will be selected by the select lines CBA.

Application :

In our project it is used to select various chip based on signals from micro controller. The output lines Y_0 – Y_7 when enabled is used to select IC 8155 and ¼ when enabled is used to select IC8277.

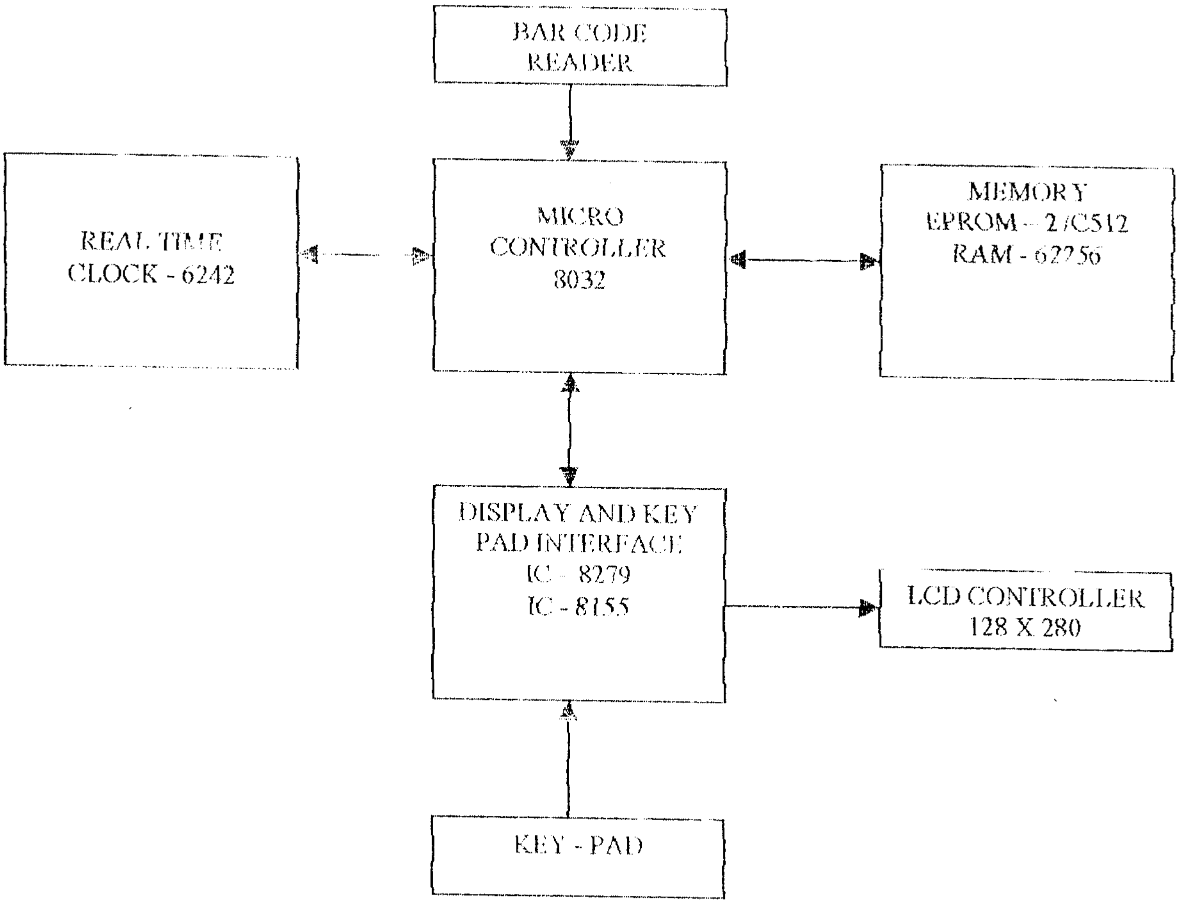


Fig 3.1 SYSTEM BLOCK DIAGRAM OF MICRO – CONTROLLER BASED LIBRARY TRANSACTION

CHAPTER IV

LOGIC DESIGN CIRCUIT

4.1 µC - BASED SYSTEM DESING - PROCESS EXPLANATION

Micro controller is a device which includes processor, memory and I / O signal lines on a single chip fabricated using VLSI technology. It is also called as single chip micro computers. The controller which is employed for processing and control application is µc - 8032.

The process involved in library transaction using µc is as follows. The 8 - bit binary output from BCR is sent to controller through serial interface circuit. The 8 bit data is initially stored in the accumulator. Before processing, the identity number of the persons and then corresponding number is stored in RAM. Also the access number of the book and the title of the book is stored in RAM.

When the controller receives the 8 bit data, is starts its operation of comparing the ACC contents with each of the code stored in RAM location. This task is accomplished by software instruction.

Once the input and that of the code stored in RAM locations are same, the name of the person is displayed in ICD display via display interface.

Next command being displayed is number of books being taken by the person until now.

Following which is the command issue / return is displayed. Now by using the 4 x 4 matrix key board, the letters I or R is typed in and is displayed in LCD (Dot matrix) display.

Next command to be displayed is access number of the book. Once the access number of the book is entered, the title of the book is displayed.

Depending upon whether the book is issued or returned, the number of book being taken by the person is updated.

The above said entire operation of keying in the data, displaying data in the display, retrieving data from RAM location at required time is done by the Micro controller with the aid of the assembly level controller program stored in EPROM. The entire process flow is shown in general block diagram.

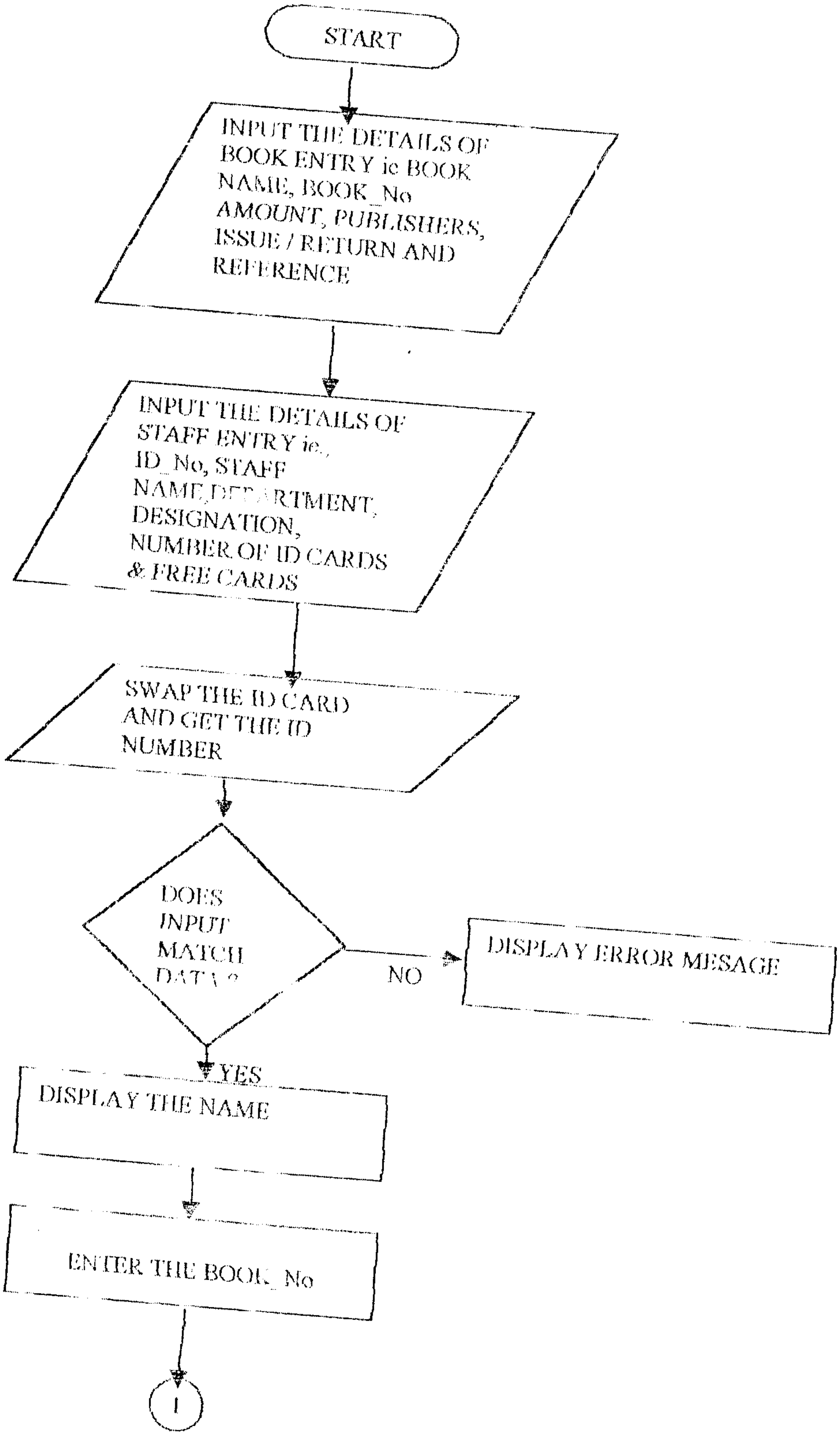
CHAPTER V

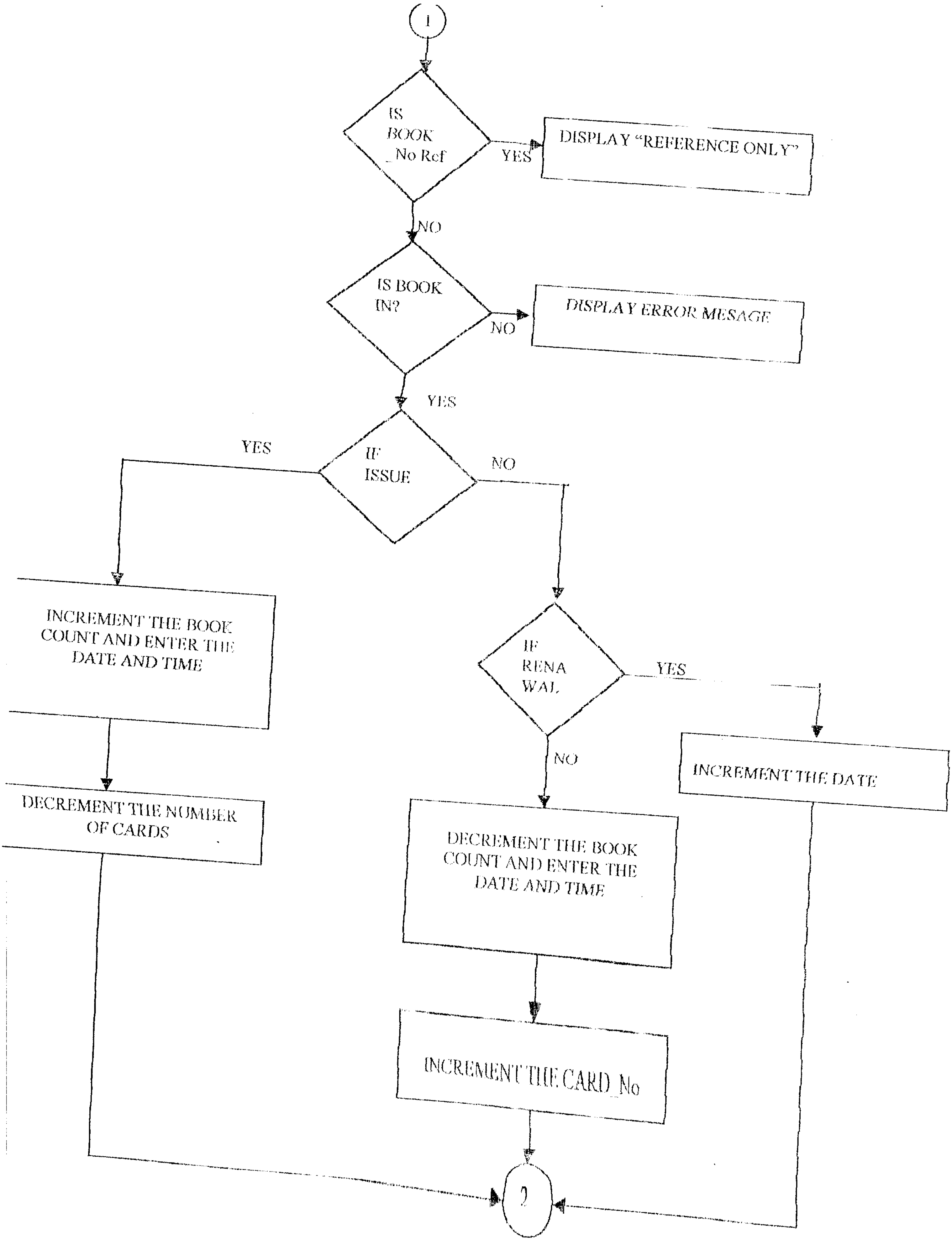
SOFTWARE

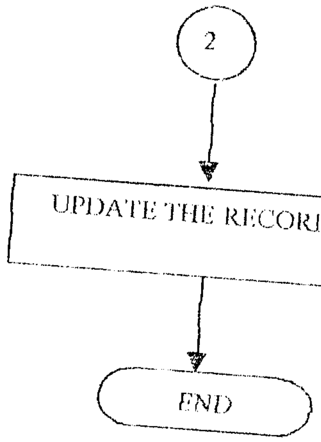
INTRODUCTION

The database for the entire library management system has been developed using visual basic as the front end and MS access as the back end tools. The main purpose of developing the software in high level language is that large data base due to large memory capacity, easy and fast retrieval of data, and is user friendly. The software thus developed has 3 sub heads namely entry details, view & report. Entry details which includes details of books in the library, details of staff working in the firm and the details required during transactions, the second subhead namely view has high flexibility as it helps in knowing the details of books published by a particular company, books written by same author, issue / return of the books in a particular day and simultaneous searching of the title and author of a particular book including transaction details. The third sub head namely report has details regarding books issued, books in the library, book details and staff details. The print out of various details is taken from this section only.

FLOW CHART


START
INPUT THE DETAILS OF BOOK ENTRY ie BOOK NAME, BOOK_No AMOUNT, PUBLISHERS, ISSUE / RETURN AND REFERENCE
INPUT THE DETAILS OF STAFF ENTRY ie., ID_No, STAFF NAME,DEPARTMENT, DESIGNATION, NUMBER OF ID CARDS & FREE CARDS
SWAP THE ID CARD AND GET THE ID NUMBER
DOES INPUT MATCH DATA ?
NO
DISPLAY ERROR MESAGE
YES
DISPLAY THE NAME
ENTER THE BOOK_No
1


1
IS BOOK _No Ref
YES
DISPLAY "REFERENCE ONLY"
NO
IS BOOK IN?
NO
DISPLAY ERROR MESAGE
YES
YES
IF ISSUE
NO
INCREMENT THE BOOK COUNT AND ENTER THE DATE AND TIME
DECREMENT THE NUMBER OF CARDS
IF RENA WAL
YES
INCREMENT THE DATE
NO
DECREMENT THE BOOK COUNT AND ENTER THE DATE AND TIME
INCREMENT THE CARD_No
2


2
UPDATE THE RECORDS
END

AUTHOR FORM

```
VERSION 5.00
Object = "{FAEEE763-117E-101B-8933-08002B2F4F5A}#1.1#0"; "DBLIST32.OCX"
Object = "{00028C01-0000-0000-0000-000000000046}#1.0#0"; "DBGRID32.OCX"
Begin VB.Form frmAuthor
        Attribute VB_Name = "frmAuthor"
        Attribute VB_GlobalNameSpace = False
        Attribute VB_Creatable = False
        Attribute VB_PredeclaredId = True
        Attribute VB_Exposed = False
Private Sub Close_Click()
        Unload Me
        Screen.MousePointer = 0
End Sub

Private Sub DBCombo1_Change()
        se$ = "'" + DBCombo1.Text + "'"
        sel$ = "select book_no,book_name,rs,company,field,issue,ref,doe from book where
              author=" & se$
        Data1.RecordSource = sel$
        Data1.Refresh
End Sub

Private Sub DBCombo1_Click(Area As Integer)
        se$ = "'" + DBCombo1.Text + "'"
        sel$ = "select book_no,book_name,rs,company,field,issue,ref,doe from book where
              author=" & se$
        Data1.RecordSource = sel$
        Data1.Refresh
End Sub

Private Sub DBCombo1_KeyPress(KeyAscii As Integer)
        se$ = "'" + DBCombo1.Text + "'"
        sel$ = "select book_no,book_name,rs,company,field,issue,ref,doe from book where
              author=" & se$
        Data1.RecordSource = sel$
        Data1.Refresh
End Sub
```

```
Private Sub Form_Load()
        Data2.RecordSource = "Select author as com From book group by author"
        DBCombo1.DataField = "com"
        DBCombo1.ListField = "com"
        Data2.Refresh
End Sub
```

BOOK FORM

```
VERSION 5.00
Begin VB.Form frmbook
      Attribute VB_Name = "frmbook"
      Attribute VB_GlobalNameSpace = False
      Attribute VB_Creatable = False
      Attribute VB_PredeclaredId = True
      Attribute VB_Exposed = False
Private Sub cmdAdd_Click()
        txtFields(0).Enabled = True
        Data1.Recordset.AddNew
End Sub

Private Sub cmdDelete_Click()
        Dim X As Integer
        Dim db As Database
        Dim rt As Recordset
        X = MsgBox("Are you sure? ", 33, "Delete")
  If X = 1 Then
          Set db = Workspaces(0).OpenDatabase("c:\vb\lib\book.mdb")
          ss$ = "'" + txtFields(0).Text + "'"
          Set rt = db.OpenRecordset("select book_no,id_no from details where book_no=" &
                  ss$)
   If rt.RecordCount <> 0 Then
          MsgBox "Already taken by Id " & rt.fields("id_no").value, 16, "Delete"
          MsgBox "Can't Delete it", 32, "Delete"
          Exit Sub
   Else

          Data1.Recordset.Delete
          Data1.Recordset.MoveNext
   End If
 End If
End Sub
```

```
Private Sub cmdRefresh_Click()
        'this is really only needed for multi user apps
        Data1.Refresh
End Sub

Private Sub cmdUpdate_Click()
        Data1.UpdateRecord
        Data1.Recordset.Bookmark = Data1.Recordset.LastModified
End Sub
book.frm
Private Sub cmdClose_Click()
        Begin VB.Label lblLabels
        Attribute VB_Name = "frmbook"
        Attribute VB_GlobalNameSpace = False
        Attribute VB_Creatable = False
        Attribute VB_PredeclaredId = True
        Attribute VB_Exposed = False
Private Sub cmdAdd_Click()
        txtFields(0).Enabled = True
        Data1.Recordset.AddNew
End Sub

Private Sub cmdDelete_Click()
      ` DimX As Integer
        Dim db As Database
        Dim rt As Recordset
        X = MsgBox("Are you sure? ", 33, "Delete")
 If X = 1 Then
        Set db = Workspaces(0).OpenDatabase("c:\vb\lib\book.mdb")
          ss$ = "'" + txtFields(0).Text + "'"
        Set rt = db.OpenRecordset("select book_no,id_no from details where book_no=" &
                    ss$)
  If rt.RecordCount <> 0 Then
        MsgBox "Already taken by Id " & rt.fields("id_no").value, 16, "Delete"
        MsgBox "Can't Delete it", 32, "Delete"
    Exit Sub
  Else

        Data1.Recordset.Delete
        Data1.Recordset.MoveNext
  End If
 End If
End Sub
```

```
Private Sub cmdRefresh_Click()
        'this is really only needed for multi user apps
        Data1.Refresh
End Sub

Private Sub cmdUpdate_Click()
        Data1.UpdateRecord
        Data1.Recordset.Bookmark = Data1.Recordset.LastModified
End Sub

Private Sub cmdClose_Click()
        Unload Me
        Screen.MousePointer = 0
End Sub

Private Sub Data1_Error(DataErr As Integer, Response As Integer)
        'This is where you would put error handling code
        'If you want to ignore errors, comment out the next line
        'If you want to trap them, add code here to handle them
        MsgBox "Data error event hit err:" & Error$(DataErr)
        Response = 0  'throw away the error
End Sub

Private Sub Data1_Reposition()
        Screen.MousePointer = vbDefault
        On Error Resume Next
        'This will display the current record position
        'for dynasets and snapshots
        Data1.Caption = "Record: " & (Data1.Recordset.AbsolutePosition + 1)
        'for the table object you must set the index property when
        'the recordset gets created and use the following line
        'Data1.Caption = "Record: " & (Data1.Recordset.RecordCount *
     (Data1.Recordset.PercentPosition * 0.01)) + 1
End Sub

Private Sub Data1_Validate(Action As Integer, Save As Integer)
        'This is where you put validation code
        'This event gets called when the following actions occur
     Select Case Action
             Case vbDataActionMoveFirst
             Case vbDataActionMovePrevious
             Case vbDataActionMoveNext
             Case vbDataActionMoveLast
             Case vbDataActionAddNew
             Case vbDataActionUpdate
             Case vbDataActionDelete
             Case vbDataActionFind
             Case vbDataActionBookmark
             Case vbDataActionClose
    End Select
          Screen.MousePointer = vbHourglass
   End Sub

   Private Sub Form_Load()
           Dim db As Database
           Dim rt As Recordset
           Dim qd As QueryDef
           Dim tno As Long
           Set db = Workspaces(0).OpenDatabase("c:\vb\lib\book.mdb")
           Set qd = db.CreateQueryDef("")
           qd.SQL = "update book set issue=0"
           qd.Execute
           qd.close
           Set rt = db.OpenRecordset("select book_no from details")
   Do Until rt.EOF
            Set qd = db.CreateQueryDef("")
            sss$ = "'" + (rt.fields("book_no").value) + "'"
            qd.SQL = "update book set issue=-1 where book_no=" & sss$
             qd.Execute
            qd.close
            rt.MoveNext
   Loop
         rt.close
         db.close
 End Sub

Private Sub txtFields_LostFocus(Index As Integer)
        txtFields(0).Enabled = False
End Sub

        Screen.MousePointer = 0
End Sub
```

```
Private Sub Data1_Error(DataErr As Integer, Response As Integer)
        'This is where you would put error handling code
        'If you want to ignore errors, comment out the next line
        'If you want to trap them, add code here to handle them
        MsgBox "Data error event hit err:" & Error$(DataErr)
        Response = 0  'throw away the error
End Sub

Private Sub Data1_Reposition()
        Screen.MousePointer = vbDefault
        On Error Resume Next
        'This will display the current record position
        'for dynasets and snapshots
        Data1.Caption = "Record: " & (Data1.Recordset.AbsolutePosition + 1)
        'for the table object you must set the index property when
        'the recordset gets created and use the following line
        'Data1.Caption = "Record: " & (Data1.Recordset.RecordCount *
        (Data1.Recordset.PercentPosition * 0.01)) + 1
End Sub
Private Sub Data1_Validate(Action As Integer, Save As Integer)
        'This is where you put validation code
        'This event gets called when the following actions occur
 Select Case Action
        Case vbDataActionMoveFirst
        Case vbDataActionMovePrevious
        Case vbDataActionMoveNext
        Case vbDataActionMoveLast
        Case vbDataActionAddNew
        Case vbDataActionUpdate
        Case vbDataActionDelete
        Case vbDataActionFind
        Case vbDataActionBookmark
        Case vbDataActionClose
 End Select
        Screen.MousePointer = vbHourglass
End Sub

Private Sub Form_Load()
        Dim db As Database
        Dim rt As Recordset
        Dim qd As QueryDef
        Dim tno As Long
        Set db = Workspaces(0).OpenDatabase("c:\vb\lib\book.mdb")
        Set qd = db.CreateQueryDef("")
```

```
            qd.SQL = "update book set issue=0"
            qd.Execute
            qd.close
            Set rt = db.OpenRecordset("select book_no from details")
     Do Until rt.EOF
             Set qd = db.CreateQueryDef("")
             sss$ = "'" + (rt.fields("book_no").value) + "'"
             qd.SQL = "update book set issue=-1 where book_no=" & sss$
             qd.Execute
             qd.close
             rt.MoveNext
    Loop
          rt.close
          db.close
End Sub

Private Sub txtFields_LostFocus(Index As Integer)
          txtFields(0).Enabled = False
End Sub
```

COMPANY FORM

```
VERSION 5.00
Object = "{FAEEE763-117E-101B-8933-08002B2F4F5A}#1.1#0"; "DBLIST32.OCX"
Object = "{00028C01-0000-0000-0000-000000000046}#1.0#0"; "DBGRID32.OCX"
Begin VB.Form frmcom
       Attribute VB_Name = "frmcom"
       Attribute VB_GlobalNameSpace = False
       Attribute VB_Creatable = False
       Attribute VB_PredeclaredId = True
       Attribute VB_Exposed = False
Private Sub Close_Click()
        Unload Me
        Screen.MousePointer = 0
End Sub

Private Sub DBCombo1_Change()
        calls
End Sub

Private Sub DBCombo1_Click(Area As Integer)
        calls
End Sub
```

```
Private Sub DBCombo1_KeyPress(KeyAscii As Integer)
        calls
End Sub

Private Sub Form_Load()
            Data2.RecordSource = "Select company as com From book group by company"
            DBCombo1.DataField = "com"
            DBCombo1.ListField = "com"
            Data2.Refresh
End Sub
Public Sub calls()
            se$ = "'" + DBCombo1.Text + "'"
            sel$ = "select book_no,book_name,author,rs,field,issue,ref,doe from book where
                company=" & se$
            Data1.RecordSource = sel$
            Data1.Refresh
End Sub
```

FIELD FORM

```
VERSION 5.00
Object = "{FAEEE763-117E-101B-8933-08002B2F4F5A}#1.1#0"; "DBLIST32.OCX"
Object = "{00028C01-0000-0000-0000-000000000046}#1.0#0"; "DBGRID32.OCX"
Begin VB.Form frmfield
        Attribute VB_Name = "frmfield"
        Attribute VB_GlobalNameSpace = False
        Attribute VB_Creatable = False
        Attribute VB_PredeclaredId = True
        Attribute VB_Exposed = False
Private Sub Close_Click()
            Unload Me
            Screen.MousePointer = 0
End Sub

Private Sub DBCombo1_Change()
        calls
End Sub

Private Sub DBCombo1_Click(Area As Integer)
        calls
End Sub

Private Sub DBCombo1_KeyPress(KeyAscii As Integer)
        calls      End Sub
```

```
Private Sub Form_Load()
        Data2.RecordSource = "Select field as com From book group by field"
        DBCombo1.DataField = "com"
        DBCombo1.ListField = "com"
        Data2.Refresh
End Sub

Public Sub calls()
        se$ = "'" + DBCombo1.Text + "'"
        sel$ = "select book_no,book_name,author,rs,company,issue,ref,doe from book where
            field=" & se$
        Data1.RecordSource = sel$
        Data1.Refresh
End Sub
```

GRID FORM

```
VERSION 5.00
Object = "{00028C01-0000-0000-0000-000000000046}#1.0#0"; "DBGRID32.OCX"
Begin VB.Form frmGrid
      Attribute VB_Name = "frmGrid"
      Attribute VB_GlobalNameSpace = False
      Attribute VB_Creatable = False
      Attribute VB_PredeclaredId = True
      Attribute VB_Exposed = False
      Dim id As String
Public Sub get_id()
        Dim dt As Database
        Dim rs As Recordset
        Dim re As Recordset
        'Open "c:\access\access.dat" For Input As #1
        Open "c:\vb\lib\DATA" For Input As #1
Do While Not EOF(1) ' Loop until end of file.
        Input #1, my$
Loop
      id = my$
      Clos e #1
      'id = InputBox("Enter Id No: ", "Details", 0)
If id = "" Then
        no.Text = ""
        sname.Text = ""
        sdept.Text = ""
        stok.Text = ""
        sfree.Text = ""
Exit Sub
```

```
    Else
            sq$ = "select book_no,book_name,date_of_issue,renu,ren_date from selection
              where id_no=" & id
            Set dt = Workspaces(0).OpenDatabase("c:\vb\lib\book.mdb")
            Set rs = dt.OpenRecordset(sq$)
            Set Data1.Recordset = rs
            Data1.Refresh
            rs.close
            sl$ = "select id_no,staff_name,dept,no_tok,free_tok from staff where id_no=" & id
            Set re = dt.OpenRecordset(sl$)
     If re.RecordCount = 0 Then
             MsgBox "Staff Id is not found", 16, "Search"
             no.Text = ""
             sname.Text = ""
             sdept.Text = ""
             stok.Text = ""
             sfree.Text = ""
      Exit Sub
    Else
            no.Text = Val(re.fields("id_no").value)
            sname.Text = re.fields("staff_name").value
            sdept.Text = re.fields("dept").value
            stok.Text = Val(re.fields("no_tok").value)
            sfree.Text = Val(re.fields("free_tok").value)
    End If
          re.close
          dt.close
  End If
 End Sub

 Private Sub Close_Click()
          Unload Me
          Screen.MousePointer = 0
 End Sub

 Private Sub Command1_Click()
          get_id
 End Sub

 Private Sub Form_Load()
          get_id
 End Sub
```

SSUE FORM

```
ERSION 5.00
)bject = "{00028C01-0000-0000-0000-000000000046}#1.0#0"; "DBGRID32.OCX"
'egin VB.Form frmIssue
        Attribute VB_Name = "frmIssue"
        Attribute VB_GlobalNameSpace = False
        Attribute VB_Creatable = False
        Attribute VB_PredeclaredId = True
        Attribute VB_Exposed = False

rivate Sub Close_Click()
           Unload Me
           Screen.MousePointer = 0
nd Sub

rivate Sub iss_GotFocus()
          MsgBox "Enter Date with 'MM/DD/YY' Formate", 16, "Date"
nd Sub

rivate Sub iss_KeyPress(KeyAscii As Integer)
 If KeyAscii = 13 Then
           s$ = "#" + iss.Text + "#"
           sel$ = "select book_no,book_name,id_no,staff_name,ren_date from selection
              where date_of_issue=" & s$
              On Error GoTo errorend
           Data1.RecordSource = sel$
           Data1.Refresh
 End If
xit Sub

          errorend:
          MsgBox "Check the date format(mm/dd/yy)", 32, "Date"
   Exit Sub
nd Sub

rivate Sub Select_Click()
       s$ = "#" + iss.Text + "#"
       sel$ = "select book_no,book_name,id_no,staff_name,ren_date from selection where
            date_of_issue=" & s$
       On Error GoTo errorend
       Data1.RecordSource = sel$
       Data1.Refresh
Exit Sub
    errorend:
       ss $ = "Check the date format(mm/dd/yy)"
```

```
            MsgBox ss$, 32, "Date"
        Exit Sub
    End Sub
```

MAIN FORM

```
VERSION 5.00
Object = "{00025600-0000-0000-C000-000000000046}#4.6#0"; "CRYSTL32.OCX"
Begin VB.MDIForm MDImain
        BackColor      =  &H8000000C&
        Caption        =  "Library Management System"
        ClientHeight   =  4536
        ClientLeft     =  1236
        ClientTop      =  1800
        ClientWidth    =  6060
        LinkTopic      =  "MDIForm1"
        MousePointer   =  1  'Arrow
        WindowState    =  2  'Maximized

    Begin Crystal.CrystalReport st
        Left           =  2160
        Top            =  3960
        _ExtentX       =  593
        _ExtentY       =  593
        _Version       =  262150
        ReportFileName =  "c:\vb\lib\staff.rpt"
        WindowControlBox=  -1  'True
        WindowMaxButton =  -1  'True
        WindowMinButton =  -1  'True
        SelectionFormula=  "{staff.id_no}>=0"
    End

Begin Crystal.CrystalReport bk
        Left           =  1680
        Top            =  3960
        _ExtentX       =  593
        _ExtentY       =  593
        _Version       =  262150
        ReportFileName =  "c:\vb\lib\b_detail.rpt"
        WindowControlBox=  -1  'True
        WindowMaxButton =  -1  'True
End     WindowMinButton =  -1  'True
```

```
Begin Crystal.CrystalReport tk
            Left          =  1200
            Top           =  3960
            _ExtentX      =  593
            _ExtentY      =  593
            _Version      =  262150
            ReportFileName  =  "c:\vb\lib\out.rpt"
                    WindowControlBox=  -1  'True
            WindowMaxButton =  -1  'True
            WindowMinButton =  -1  'True
End

Begin Crystal.CrystalReport inputs
            Left          =  720
            Top           =  3960
            _ExtentX      =  593
            _ExtentY      =  593
            _Version      =  262150
            ReportFileName  =  "c:\vb\lib\inside.rpt"
            WindowControlBox=  -1  'True
            WindowMaxButton =  -1  'True
            WindowMinButton =  -1  'True
End

Begin VB.Menu entry
            Caption       =  "&Entry Details"
Begin VB.Menu bo_en
      Caption       =  "&Book Entry"
End
Begin VB.Menu st_entry
      Caption       =  "&Staff Entry"
End

Begin VB.Menu bo_tak
      Caption       =  "Book &Taken"
End
Begin VB.Menu ex
      Caption       =  "&Exit"
End
End
Begin VB.Menu view
      Caption       =  "&View"
Begin VB.Menu search
      Caption       =  "Searching the Books"
Begin VB.Menu title
```

```
            Caption         =   "&Title"
        End
        Begin VB.Menu author
            Caption         =   "Author"
        End
        Begin VB.Menu fields
            Caption         =   "&Related Field
        End
        Begin VB.Menu publiction
            Caption         =   "&Publiction"
        End
        End
        Begin VB.Menu date
            Caption         =   "&Date"
        Begin VB.Menu issue
            Caption         =   "Date of &Issue"
        End
        Begin VB.Menu return
            Caption         =   "Date of &Return"
        End
        End
        Begin VB.Menu taken
            Caption         =   "&Books Taken by Staffs"
        End
      End
      Begin VB.Menu reports
        Caption         =   "&Reports"
      Begin VB.Menu books
          Caption         =   "&Books"
      End
      Begin VB.Menu staff
          Caption         =   "&Staff"
      End
      Begin VB.Menu in
          Caption         =   "Books &In Side"
      End
      Begin VB.Menu out
          Caption         =   "Book &Out Side"
      End
      End
      Begin VB.Menu exit
        Caption         =   "&Exit"
      Begin VB.Menu exits
          Caption         =   "&Exit"
      End
    End
```

```
End
        Attribute VB_Name = "MDImain"
        Attribute VB_GlobalNameSpace = False
        Attribute VB_Creatable = False
        Attribute VB_PredeclaredId = True
        Attribute VB_Exposed = False

Private Sub author_Click()
        Load frmAuthor
        frmAuthor.Show
End Sub

Private Sub bo_en_Click()
        Load frmbook
        frmbook.Show
End Sub

Private Sub bo_tak_Click()
        frmtaken.Show
End Sub

Private Sub books_Click()
        bk.SelectionFormula = "{book.book_no} <> ""
        bk.Action = 1
End Sub

Private Sub ex_Click()
        Dim no As Integer
        no = MsgBox("This will end your Library Session", 65, "Library")
If no = 1 Then
    End
  End If
End Sub

Private Sub exits_Click()
        Dim no As Integer
        no = MsgBox("This will end your Library Session", 65, "Library")
  If no = 1 Then
     End
   End If
End Sub

Private Sub Fields_Click()
        frmfield.Show
End Sub
```

```
Private Sub gene_Click()
        frmReport.Show
End Sub
Private Sub grid_Click()
        frmGrid.Show
End Sub
Private Sub in_Click()
        inputs.SelectionFormula = "not {book.issue}"
        inputs.Action = 1
End Sub
Private Sub issue_Click()
        frmIssue.Show
End Sub
Private Sub MSWordArt1_GotFocus()
End Sub
Private Sub out_Click()
        tk.SelectionFormula = "{selection.book_no} <> """
        tk.Action = 1
End Sub
Private Sub Picture1_Click()
End Sub
Private Sub publiction_Click()
        frmcom.Show
End Sub
Private Sub return_Click()
        frmReturn.Show
End Sub
Private Sub st_entry_Click()
        frmstaff.Show
End Sub
Private Sub staff_Click()
        st.SelectionFormula = "{staff.id_no} >=0"
        st.Action = 1
End Sub
Private Sub taken_Click()
        frmGrid.Show
End Sub
Private Sub title_Click()
        frmsearchb.Show
End Sub
```

RETURN FORM

```
VERSION 5.00
Object = "{00028C01-0000-0000-0000-000000000046}#1.0#0"; "DBGRID32.OCX"
```

```
\
Begin VB.Form frmReturn
        Attribute VB_Name = "frmReturn"
        Attribute VB_GlobalNameSpace = False
        Attribute VB_Creatable = False
        Attribute VB_PredeclaredId = True
        Attribute VB_Exposed = False
Private Sub Close_Click()
          Unload Me
          Screen.MousePointer = 0
End Sub

Private Sub iss_GotFocus()
          MsgBox "Enter Date with 'MM/DD/YY' Formate", 16, "Date"
End Sub

Private Sub iss_KeyPress(KeyAscii As Integer)
   If KeyAscii = 13 Then
          s$ = "#" + iss.Text + "#"
          sel$ = "select book_no,book_name,id_no,staff_name,date_of_issue from selection
             where ren_date=" & s$
          On Error GoTo errorend
          Data1.RecordSource = sel$
          Data1.Refresh
   End If
   Exit Sub
errorend:
          MsgBox "Check the date format(mm/dd/yy)", 32, "Date"
   Exit Sub
End Sub

Private Sub Select_Click()
          s$ = "#" + iss.Text + "#"
          sel$ = "select book_no,book_name,id_no,staff_name,date_of_issue from selection
                where ren_date=" & s$
          On Error GoTo errorend
          Data1.RecordSource = sel$
          Data1.Refresh
   Exit Sub
errorend:
          MsgBox "Check the date format(mm/dd/yy)", 32, "Date"
   Exit Sub
End Sub
```

S_TITLE FORM

```
VERSION 5.00
Object = "{FAEEE763-117E-101B-8933-08002B2F4F5A}#1.1#0"; "DBLIST32.OCX"
Object = "{00028C01-0000-0000-0000-000000000046}#1.0#0"; "DBGRID32.OCX"
Begin VB.Form frmsearchb
        Attribute VB_Name = "frmsearchb"
        Attribute VB_GlobalNameSpace = False
        Attribute VB_Creatable = False
        Attribute VB_PredeclaredId = True
        Attribute VB_Exposed = False
        Private Sub Close_Click()
           Unload Me
           Screen.MousePointer = 0
End Sub

Private Sub Command1_Click()
        se$ = "'" + db1.Text + "'"
        sel$ = "select book_no,id_no,staff_name,dept,date_of_issue,ren_date from selection
              where book_name=" & se$
        Data3.RecordSource = sel$
        Data3.Refresh
End Sub

Private Sub Command2_Click()
        se$ = "'" + db1.Text + "'"
        sel$ = "select book_no,Author,rs,company,field,doe from book where book_name="
              & se$
        Data1.RecordSource = sel$
        Data1.Refresh
End Sub

Private Sub Form_Load()
        Data2.RecordSource = "Select book_name as bo From book group by book_name"
        db1.DataField = "bo"
        db1.ListField = "bo"
        Data2.Refresh
End Sub
```

STAFF FORM

```
VERSION 5.00
Begin VB.Form frmstaff
        Attribute VB_Name = "frmstaff"
        Attribute VB_GlobalNameSpace = False
        Attribute VB_Creatable = False
        Attribute VB_PredeclaredId = True
        Attribute VB_Exposed = False
        Dim ch As Boolean
   Private Sub cmdAdd_Click()
            'open "c:\access\access.dat" For Input As #1
            Open "c:\vb\lib\DATA" For Input As #1
     Do While Not EOF(1) ' Loop until end of file.
              Input #1, my$
     Loop
           Close #1
     If chk(my$) = 1 Then
              Data1.Recordset.AddNew
              txtFields(0).Text = my$
     Else
                MsgBox "Id already existed", 32, "Add"
     End If
           ch = True
   End Sub
   Private Sub cmdDelete_Click()
           Dim data As Database
           Dim rss As Recordset
           Dim qu As QueryDef
           Dim msg As Integer
           msg = MsgBox("Are you sure? ", 33, "Delete")
              If msg = 1 Then
                 Set data = Workspaces(0).OpenDatabase("c:\vb\lib\book.mdb")
                 Set rss = data.OpenRecordset("select book_no from details where id_no="
                   & txtFields(0).Text)
                   Do Until rss.EOF
                        Set qu = data.CreateQueryDef("")
                        qu.SQL = "update book set issue=0 where book_no=" &
                        Str(rss.fields("book_no").value)
                        qu.Execute
                        qu.close
                        rss.MoveNext
                 Loop
                 Set qu = data.CreateQueryDef("")
                 qu.SQL = "delete * from details where id_no=" & txtFields(0).Text
                 qu.Execute
                 qu.close
                 Data1.Recordset.Delete
                 Data1.Recordset.MoveNext
                Else
                        Exit Sub
                End If
End Sub

Private Sub cmdRefresh_Click()

  Data1.Refresh
End Sub

Private Sub cmdUpdate_Click()
                Data1.UpdateRecord
                Data1.Recordset.Bookmark = Data1.Recordset.LastModified
End Sub

Private Sub cmdClose_Click()
          Unload Me
          Screen.MousePointer = 0
End Sub

Private Sub Data1_Error(DataErr As Integer, Response As Integer)
          'This is where you would put error handling code
          'If you want to ignore errors, comment out the next line
          'If you want to trap them, add code here to handle them
          MsgBox "Data error event hit err:" & Error$(DataErr)
          Response = 0  'throw away the error
End Sub

Private Sub Data1_Reposition()
          Screen.MousePointer = vbDefault
          On Error Resume Next
          'This will display the current record position
          'for dynasets and snapshots
          Data1.Caption = "Record: " & (Data1.Recordset.AbsolutePosition + 1)
          'for the table object you must set the index property when
          'the recordset gets created and use the following line
          'Data1.Caption = "Record: " & (Data1.Recordset.RecordCount *
        (Data1.Recordset.PercentPosition * 0.01)) + 1
End Sub

Private Sub Data1_Validate(Action As Integer, Save As Integer)
          'This is where you put validation code
          'This event gets called when the following actions occur
 Select Case Action
```

```
            Case vbDataActionMoveFirst
            Case vbDataActionMovePrevious
            Case vbDataActionMoveNext
            Case vbDataActionMoveLast
            Case vbDataActionAddNew
            Case vbDataActionUpdate
            Case vbDataActionDelete
            Case vbDataActionFind
            Case vbDataActionBookmark
            Case vbDataActionClose
    End Select
          Screen.MousePointer = vbHourglass
    If Err.Number <> 0 Then
            msg = "There was an error attempting to Add!"
            Err.Clear   ' Clear Err object fields
            Exit Sub
    End If
        End Sub

Private Sub Form_Load()
        Dim db As Database
        Dim rt As Recordset
        Dim qd, qdd As QueryDef
        Dim tno As Long
        Set db = Workspaces(0).OpenDatabase("c:\vb\lib\book.mdb")
        Set rt = db.OpenRecordset("select count(id_no) as co,id_no from details group by
id_no")
        Do Until rt.EOF
            Set qd = db.CreateQueryDef("")
            qd.SQL = "update staff set free_tok=no_tok-" & rt.fields("co").value & Space(2)
            & "where id_no=" & rt.fields("id_no").value
            qd.Execute
            qd.close
            rt.MoveNext
    Loop
    rt.close
    Set qdd = db.CreateQueryDef("")
    qdd.SQL = "update staff set free_tok=no_tok where id_no not in(select id_no from
details group by id_no);"
    qdd.Execute
    qdd.close
    db.close
End Sub
```

```
Private Sub no_Change()
   If ch = True Then
      txtFields(4).Text = no.Text
      ch = False
   End If
End Sub

Public Function chk(tx As String) As Integer
   Dim dd As Database
   Dim rr As Recordset
   Set dd = Workspaces(0).OpenDatabase("c:\vb\lib\book.mdb")
   sa$ = "'" + tx + "'"
   Set rr = dd.OpenRecordset("select id_no from staff where id_no=" & tx)
   If rr.RecordCount = 0 Then
      chk = 1
   Else
      chk = 0
   End If
End Function
```

TEMP DTL FORM

```
VERSION 5.00
Begin VB.Form frmtaken
        Attribute VB_Name = "frmtaken"
        Attribute VB_GlobalNameSpace = False
        Attribute VB_Creatable = False
        Attribute VB_PredeclaredId = True
        Attribute VB_Exposed = False
        Dim da As Database
        Dim rs As Recordset
        Dim chk, fd As Boolean
        Dim finds As String
        Dim ntok, ftok As Integer
        Dim id As Long
        Dim bno As String
        Dim ch As Integer
Private Sub chkFields_GotFocus()
        If chkFields.value = 1 Then
                chk = True
        End If
End Sub
Private Sub chkFields_LostFocus()
        If chk = True Then
```

```
                    chkFields.value = 1
                    txtFields(2).Text = CStr(DateAdd("d", 5, CDate(txtFields(2).Text)))
          End If
End Sub

Private Sub chkFields_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
          If chk = True Then
                    chkFields.value = 1
                    txtFields(2).Text = CStr(DateAdd("d", 5, CDate(txtFields(2).Text)))
          End If
End Sub

Private Sub cmdAdd_Click()
          Dim Dr, dbase As Database
          Dim qd, qdd, dq As QueryDef
          Dim v As Integer
          Dim base As Database
          Dim setr, ro, rcd As Recordset
          bname.Text = ""
          fd = True
          bissue
          stok
          call_tk
          If ftok = 0 Then
                    MsgBox "No free tokens to take book", 32, "Taken"
                    fd = False
                    Exit Sub
          Else
                    bno = InputBox("Enter the book Code: ", "Book Code", 0)
                    If bno = "" Then
                              MsgBox "Enter the Book Code first", 16, "Book Code"
                              fd = False
                              Exit Sub
                    Else
                              Set dbase = Workspaces(0).OpenDatabase("c:\vb\lib\book.mdb")
                              ss$ = "'" + bno + "'"
                              Set ro = dbase.OpenRecordset("Select book_no from book where
                              book_no=" &  ss$)
                              If ro.RecordCo unt = 0 Then
                                        MsgBox "Iligal Book No", 32, "Book"
                                        fd = False
                                        Exit Sub
                              Else
                    Set rcd = dbase.OpenRecordset("select issue from book
                    where book_no=" & ss$)
                    If rcd.fields("issue").value = -1 Then
                        MsgBox "Book is outside", 32, "Add"
                        Exit Sub
                    End If
                    chh
                    If ch = 1 Then
                        ch = 0
                        fd = False
                        Exit Sub
                    Else
                        'Data1.Recordset.Close
                        'Data1.RecordSource = "select * from details
                        where id_no=" & Str(id)
                        Data1.Recordset.AddNew
                        txtFields(0).Text = Str(id)
                        no.Text = bno
                        bo_name
                        Set Dr = workspaces(0). OpenDatabase
                        ("c:\vb\lib\book.mdb")
                        Set qd = Dr.CreateQueryDef("")
                        qd.SQL = "update staff set free_tok=" & Str((ftok
                        - 1)) & Space(2) & "where id_no=" & Str(id)
                        'On Error GoTo errorupfree
                        qd.Execute
                        Set qdd = Dr.CreateQueryDef("")
                        sl$ = "'" + no.Text + "'"
                        qdd.SQL = "update book set issue=-1 where
                        book_no=" & sl$
                        qdd.Execute
                        qdd.close
                        qd.close
                        Dr.close
                    End If
                End If
                ro.close
                dbase.close
            End If
        End If
        fd = False
End Sub

Private Sub cmdAdd_GotFocus()
```

```
        '   txtFields(0).Text = Str(id)
End Sub

Private Sub cmdDelete_Click()
        Dim Drr As Database
        Dim qdd, Qddd As QueryDef
        Dim no1 As String
        Dim sav As Integer
        no1 = "'" + no.Text + "'"
        sav = MsgBox("Are you return the book " & no.Text, 33, "Return")
        If sav = 2 Then
                Exit Sub
        Else

                If Data1.Recordset.RecordCount = 0 Then
                MsgBox "No Books in this id", 64, "Delete"
                Exit Sub
                Else

                        Data1.Recordset.Delete
                        Data1.Recordset.MoveFirst
                        Set Drr = Workspaces(0).OpenDatabase("c:\vb\lib\book.mdb")
                        Set qdd = Drr.CreateQueryDef("")
                        qdd.SQL = "update staff set free_tok=" & Str((ftok + 1)) & Space(2) &
                                "where id_no=" & Str(id)
                        'On Error GoTo errorupfree
                        qdd.Execute
                        Set Qddd = Drr.CreateQueryDef("")
                        Qddd.SQL = "update book set issue=0 where book_no=" & no1
                        On Error GoTo errorupiss
                        Qddd.Execute
                        Qddd.close
                        qdd.close
                        Drr.close
                End If
                bname.Text = ""
        End If
                bo_name
End Sub

Private Sub cmdRefresh_Click()
        Data1.Refresh
End Sub

Private Sub cmdUpdate_Click()
        Data1.UpdateRecord
        Data1.Recordset.Bookmark = Data1.Recordset.LastModified
End Sub

Private Sub cmdClose_Click()
        Unload Me
        Screen.MousePointer = 0
End Sub
Private Sub Data1_Error(DataErr As Integer, Response As Integer)
        MsgBox "Data error event hit err:" & Error$(DataErr)
        Response = 0
End Sub

Private Sub Data1_Reposition()
        Screen.MousePointer = vbDefault
        On Error Resume Next
        Data1.Caption = "Record: " & (Data1.Recordset.AbsolutePosition + 1)
End Sub

Private Sub Data1_Validate(Action As Integer, Save As Integer)
        Select Case Action
                Case vbDataActionMoveFirst
                Case vbDataActionMovePrevious
                Case vbDataActionMoveNext
                Case vbDataActionMoveLast
                Case vbDataActionAddNew
                Case vbDataActionUpdate
                Case vbDataActionDelete
                Case vbDataActionFind
                Case vbDataActionBookmark
                Case vbDataActionClose
        End Select
        Screen.MousePointer = vbHourglass
End Sub

Private Sub doi_Click()
        finds = Screen.ActiveControl.Name
End Sub

Private Sub doi_LostFocus()
   If chkFields.value = 0 And doi.Text <> "" Then
        txtFields(2).Text = CStr(DateAdd("d", 10, CDate(doi.Text)))
   ElseIf chkFields.value = 1 And doi.Text <> "" Then
        txtFields(2).Text = CStr(DateAdd("d", 15, CDate(doi.Text)))
        Else
```

```
           Exit Sub
  End If
 End Sub

 Private Sub find_Click()
      Dim value As String
      bname.Text = ""
      Open "c:\vb\lib\DATA" For Input As #1
  Do While Not EOF(1) ' Loop until end of file.
      Input #1, my$
           loop
               value = my$
               Close #1
               If value = "" Then
               Id = 0
               Else
               id = Val(value)
               End If
  call_data  end Sub

Private Sub Form_Load()
     Dim value As String
     Dim Dr As Database
     Dim re, rs As Recordset
     Dim qd As QueryDef
     Dim too As Integer
     Set Dr = Workspaces(0).OpenDatabase("c:\vb\lib\book.mdb")
     Set re = Dr.OpenRecordset("select count(id_no) as co,id_no from details group by
     id_no;")
     If re.RecordCount = 0 Then
          Set qd = Dr.CreateQueryDef("")
          qd.SQL = "update staff set free_tok=no_tok"
          qd.Execute
          qd.close
    Else
         Do Until re.EOF
              Set rs = Dr.OpenRecordset("select no_tok from staff where id_no=" &
              Str(re.fields("id_no").value))
              If rs.RecordCount = 0 Then
                   Set qd = Dr.CreateQueryDef("")
                   qd.SQL = "update staff set free_tok=no_tok where id_no=" &
                   re.fields("id_no").value
                   qd.Execute
                   qd.close
```

```
                              Else
                                      too = rs.fields("no_tok").value - re.fields("co").value
                                      Set qd = Dr.CreateQueryDef("")
                                      qd.SQL = "update staff set free_tok=" & Str(too) & Space(2) &
                                      "where id_no=" & Str(re.fields("id_no").value)
                                      'On Error GoTo errorupfree
                                      qd.Execute
                                      qd.close
                             End If
                                      rs.close
                                      re.MoveNext
                             Loop
            End If
                             re.close
                             'Open "c:\access\access.dat" For Input As #1
                             Open "c:\vb\lib\DATA" For Input As #1
            Do While Not EOF(1) ' Loop until end of file.
                             Input #1, my$
                    Loop
                             value = my$
                             Close #1
                             'value = InputBox("Enter Id No: ", "Id No", 0)
                    If value = "" Then
                             MsgBox "Press Find to enter Staff Id_No: ", 64, "Details"
                    Else
                             id = Val(value)
                             call_data
                    End If
 'errorr:
   'MsgBox "Error to Load Records", 62, "Error"
   'Exit Sub
 'errorupfree:
 '   MsgBox "Error to Update", 32, "Error"
 End Sub

Private Sub Form_Unload(Cancel As Integer)
        Screen.MousePointer = vbDefault
End Sub

Private Sub no_Change()
        If fd = False Then
                bo_name
        End If
End Sub
```

```
Private Sub no_Click()
        finds = Screen.ActiveControl.Name
End Sub

'Private Sub no_LostFocus()
        'If doi.Text <> "" Then
        '   If chkFields.value = 0 Then
        '       txtFields(2).Text = CStr(DateAdd("d", 10, CDate(doi.Text)))
        '   Else
        '       txtFields(2).Text = CStr(DateAdd("d", 15, CDate(doi.Text)))
        '   End If
   'End If
End Sub

Private Sub txtFields_Change(Index As Integer)
        finds = txtFields(Index).Name + Str(Index)
End Sub

Private Sub txtFields_Click(Index As Integer)
        finds = txtFields(Index).Name + Str(Index)
End Sub

Public Sub call_data()
        Dim count As Long
        ck = 0
        Set da = Workspaces(0).OpenDatabase("c:\vb\lib\book.mdb")
        Set rs = da.OpenRecordset("select * from details where id_no=" & id)
        If rs.RecordCount = 0 Then
                fd = True
        End If
        'On Error GoTo errorset
        Set Data1.Recordset = rs
        'On Error GoTo errorda
         Data1.Refresh
        call_tk
        rs.close
        da.close
'errorset:
'   MsgBox "Error to get Records", 16, "Loading Error"
'   Exit Sub
'errorda:
'   MsgBox "Error to load Datas", 32, "Error"

End Sub
```

```
 Public Sub call_tk()
        Dim dd As Database
        Dim rr As Recordset
        Set dd = Workspaces(0).OpenDatabase("c:\vb\lib\book.mdb")
        Set rr = dd.OpenRecordset("select no_tok,free_tok from staff where id_no=" & id)
         On Error GoTo errorload
        ntok = rr.fields("no_tok").value
        'On Error GoTo errorget
        ftok = rr.fields("free_tok").value
        'On Error GoTo errorget
        rr.close
        dd.close
   errorload:
   'MsgBox "Not Identified", 16, "Identification"
   'MsgBox "Check the Identification Code", 32, "Check"
   Exit Sub
 'errorget:
 '  MsgBox "Error to load records", 64, "Check"
 End Sub

 Public Sub chh()
        Dim ds As Database
         Dim rt As Recordset
        Set ds = Workspaces(0).OpenDatabase("c:\vb\lib\book.mdb")
        Set rt = ds.OpenRecordset("select book_no from book where ref=true;")
        'On Error GoTo errorend
         Do Until rt.EOF
                If bno = rt.fields("book_no").value Then
                        MsgBox "It's only for reference"
                        ch = 1
                End If
                rt.MoveNext
                 Loop
'errorend:
'  MsgBox "Error to add Information:", 32, "Error"
End Sub

Public Sub bo_name()
 1      Dim sd As Database
         Dim tr As Recordset
        If no.Text <> "" Then
                Set sd = Workspaces(0).OpenDatabase("c:\vb\lib\book.mdb")
                'On Error GoTo errorend
                s$ = "'" + no.Text + "'"
                Set tr = sd.OpenRecordset("select book_name from book where book_no=" &
                        s$)
                'On Error GoTo errorend
                        If tr.RecordCount = 0 Then
                        bname.Text = ""
                        Exit Sub
        Else
                bname.Text = tr.fields("book_name").value
      End If
        tr.close
        sd.close
   End If
errorend:
   End Sub

Public Sub bissue()
        Dim db As Database
        Dim rt As Recordset
        Dim qd As QueryDef
        Dim tno As Long
        Set db = Workspaces(0).OpenDatabase("c:\vb\lib\book.mdb")
        Set qd = db.CreateQueryDef("")
        qd.SQL = "update book set issue=0"
        qd.Execute
        qd.close
        Set rt = db.OpenRecordset("select book_no from details")
        Do Until rt.EOF
                Set qd = db.CreateQueryDef("")
                sss$ = "'" + (rt.fields("book_no").value) + "'"
                qd.SQL = "update book set issue=-1 where book_no=" & sss$
                qd.Execute
                qd.close
                rt.MoveNext
        Loop
        rt.close
        db.close
End Sub
```

```
Public Sub stok()
        Dim db As Database
        Dim rt As Recordset
        Dim qd, qdd As QueryDef
        Dim tno As Long
        Set db = Workspaces(0).OpenDatabase("c:\vb\lib\book.mdb")


        Set rt = db.OpenRecordset("select count(id_no) as co,id_no from details group by
                id_no")
        Do Until rt.EOF
                Set qd = db.CreateQueryDef("")
                qd.SQL = "update staff set free_tok=no_tok-" & rt.fields("co").value & Space(2)
                & "where id_no=" & rt.fields("id_no").value
                qd.Execute
                qd.close
                rt.MoveNext
        Loop
        rt.close
         Set qdd = db.CreateQueryDef("")
         qdd.SQL = "update staff set free_tok=no_tok where id_no not in(select id_no from
        details group by id_no);"
        qdd.Execute
        qdd.close
        db.close
End Sub
```

CONCLUSION

A micro controller based library transaction system using bar code reader has been designed. This system can handle only limited database. The database can be increased using PC in the system. The software for PC has been developed in Visual Basic 5. It has high flexibility in accessing the data. The developed system has been successfully tested.

REFERENCES

1. Ramesh S. Gaonkar, "Micro Processor Architecture, Programming And Applications With the 8085 / 8080A", Willy Eastern Limited., 1990.
2. Barry B. Brey, "The Intel Microprocessor 8086/8088, 80186, 80286, 80386 and 80486", Prentice Hall of India Private Limited.
3. Mohammed Rafi Quzzanan, "Micro Processors and Micro Computer Based System Design", Universal Book Stall, New Delhi.
4. "Embedded Microcontrollers", 1996 Edition INTEL
5. "Linear Applications & specific IC's Data Book", National Semi Conductors
6. "Digital Electronics" by Malvino
7. "Opto Electronics devices data book", by Motorola.
8. "A Course in VB5 in 21 days" by Guru Wich

Microchip

27C512

512K (64K x 8) CMOS UV Erasable PROM

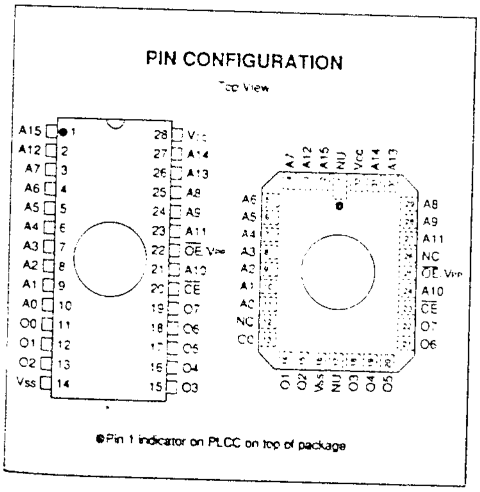
FEATURES

- High speed performance
  - —120ns access time available
- CMOS Technology for low power consumption
  - —40mA Active current
  - —100uA Standby current
- OTP (one-time-programming) available
- Auto-insertion-compatible plastic packages
- Auto ID™ aids automated programming
- * Two programming algorithms allow improved programming times
  - —Fast programming
  - —Rapid-pulse programming
- Organized 64K x 8: JEDEC standard pinouts
  - —28-pin Dual-in-line package
  - —32-pin Chip carrier (leadless or plastic)
- Available for extended temperature ranges:
  - —Commercial: 0° C to 70° C
  - —Industrial: -40° C to 85° C
  - —Military** (B): -55° C to 125° C

DESCRIPTION

The Microchip Technology Inc 27C512 is a CMOS 512K bit (ultraviolet light) Erasable (electrically) Programmable Read Only Memory. The device is organized into 64K words by 8 bits (64K bytes). Accessing individual bytes from an address transition or from power-up (chip enable pin going low) is accomplished in less than 120ns. This very high speed device allows the most sophisticated microprocessors to run at full speed without the need for WAIT states. CMOS design and processing enables this part to be used in systems where reduced power consumption and reliability are requirements.

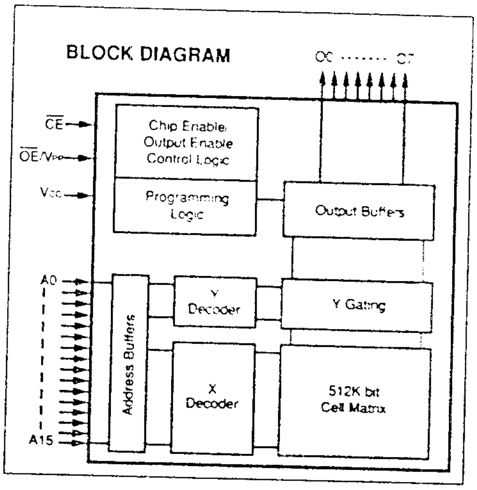
A complete family of packages is offered to provide the most flexability in applications. One-time-programming (OTP) is available for low cost (plastic) applications.



PIN CONFIGURATION

Top View

●Pin 1 indicator on PLCC on top of package



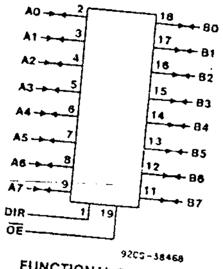
BLOCK DIAGRAM

**See 27C512 Military Data Sheet DS60014

CD54/74HC245
CD54/74HCT245

File Number 1651

High-Speed CMOS Logic



FUNCTIONAL DIAGRAM

Octal-Bus Transceiver, 3-State, Non-Inverting

Type Features:

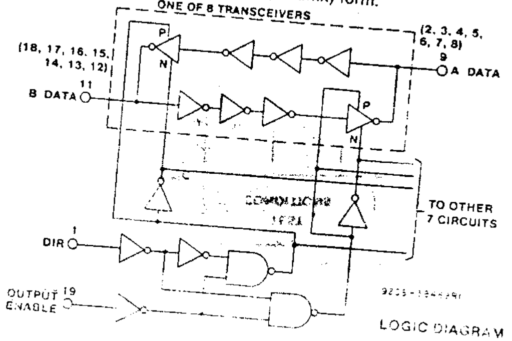
- *Buffered inputs*
- *3-State outputs*
- *Bus line driving capability*
- *Typical propagation delay (A ←→ B)*
  9 ns @ V_{CC} = 5V, C_L = 15 pF, T_A = 25° C

The RCA-CD54/74HC245 and CD54/74HCT245 are high-speed octal 3-state bidirectional transceivers intended for two-way asynchronous communication between data buses. They have high drive current outputs which enable high-speed operation while driving large bus capacitances. They provide the low power consumption of standard CMOS circuits with speeds and drive capabilities comparable to that of LSTTL circuits.

The CD54/74HC245 and CD54/74HCT245 allow data transmission from the A bus to the B bus or from the B bus to the A bus. The logic level at the direction input (DIR) determines the direction. The output enable input ($\overline{OE}$), when high, puts the I/O ports in the high-impedance state.

The HC/HCT245 is similar in operation to the HC/HCT640 and the HC/HCT643.

The CD54HC245 and CD54HCT245 are supplied in 20-lead hermetic dual-in-line ceramic packages (F suffix). The CD74HC245 and CD74HCT245 are supplied in 20-lead dual-in-line plastic packages (E suffix) and in 20-lead in-line surface mount plastic packages (M suffix). Both ,es are also available in chip (H suffix) form.



LOGIC DIAGRAM

Family Features:

- *Fanout (Over Temperature Range):*
  Standard Outputs - 10 LSTTL Loads
  Bus Driver Outputs - 15 LSTTL Loads
- *Wide Operating Temperature Range:*
  CD74HC/HCT: -40 to +85° C
- *Balanced Propagation Delay and Transition Times*
- *Significant Power Reduction Compared to LSTTL Logic ICs*
- *Alternate Source is Philips Signetics*
- *CD54HC/CD74HC Types:*
  2 to 6 V Operation
  High Noise Immunity:
  N_{IL} = 30%, N_{IH} = 30% of V_{CC} @ V_{CC} = 5 V
- *CD54HCT/CD74HCT Types:*
  4.5 to 5.5 V Operation
  Direct LSTTL Input Logic Compatibility
  V_{IL} = 0.8 V Max., V_{IH} = 2 V Min.
  CMOS Input Compatibility
  $I_I \leq 1\ \mu A$ @ V_{OL}, V_{OH}

TRUTH TABLE		
CONTROL INPUTS		OPERATION
$\overline{OE}$	DIR	
L	L	B DATA TO A BUS
L	H	A DATA TO B BUS
H	X	ISOLATION

H = high level, L = [illegible], X = [illegible]

[illegible] modes [illegible] [illegible] nated with [illegible]

CD54/74HC245
CD54/74HCT245

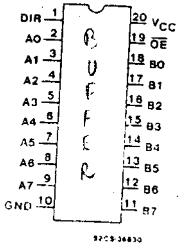
MAXIMUM RATINGS, *Absolute-Maximum Values:*

DC SUPPLY-VOLTAGE, ,V_{CC}):
(Voltages referenced tc ground) ....
DC INPUT DIODE CURFENT, I_{IK} (FOR V_I < -0.5 V OR V_I > V_{CC} +0.5V) .... -0.5 to + 7 V
DC OUTPUT DIODE CURRENT, I_{OK} (FOR V_O < -0.5 V OR V_O > V_{CC} +0.5V) .... ±20mA
DC DRAIN CURRENT, FER OUTPUT (I_O) (FOR -0.5 V < V_O < V_{CC} + 0.5V) .... ±20mA
DC V_{CC} OR GROUND CURRENT (I_{CC}) .... ±35mA
POWER DISSIPATION PER PACKAGE (P_D): .... ±70mA
For T_A = -40 to +60° C (PACKAGE TYPE E) ....
For T_A = +60 to +85° C (PACKAGE TYPE E) .... 500 mW
For T_A = -55 to +100° C (PACKAGE TYPE F, H) .... Derate Linearly at 8 mW/° C to 300 mW
For T_A = +100 to +125° C (PACKAGE TYPE F, H) .... 500 mW
For T_A = -40 to +70° C (PACKAGE TYPE M) .... Derate Linearly at 8 mW/° C to 300 mW
For T_A = +70 to +125° C (PACKAGE TYPE M) .... 400 mW
OPERATING-TEMPERATURE RANGE (T_A): .... Derate Linearly at 6 mW/° C to 70 mW
PACKAGE TYPE F, H ....
PACKAGE TYPE E, M .... -55 to +125° C
STORAGE TEMPERATURE (T_{stg}) .... -40 to +85° C
LEAD TEMPERATURE (DURING SOLDERING): .... -65 to +150° C
At distance 1/16 ± 1/3.! in. (1.59 ± 0.79 mm) from case for 10 s max. .... +265° C
Unit inserted into a PC Board (min. thickness 1/16 in., 1.59 mm)
with solder contacting lead tips only .... +300° C

RECOMMENDED OPERATING CONDITIONS:
For maximum reliability, nominal operating conditions should be selected so that operation is always within the following ranges:

CHARACTERISTIC	LIMITS		UNITS
	MIN.	MAX.	
Supply-Voltage Range (For T_A = Full Package-Temperature Range) V_{CC}:*			
CD54/74HC Types	2	6	V
CD54/74HCT Types	4.5	5.5	V
DC Input or Output Voltage V_I, V_O	0	V_{CC}	V
Operating Temperature T_A:			
CD74 Types	-40	+85	°C
CD54 Types	-55	+125	°C
Input Rise and Fall Times t_r, t_f			
at 2 V	0	1000	ns
at 4.5 V	0	500	ns
at 6 V	0	400	ns

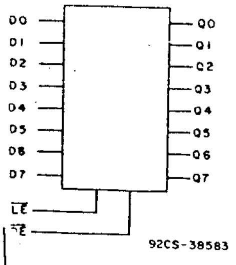
*Unless otherwise specified, all voltages are referenced to Ground.



CD54/74HC373, CD54/74HCT373
CD54/74HC573, CD54/74HCT573

File Number **1679**

High-Speed CMOS Logic



FUNCTIONAL DIAGRAM

Octal Transparent Latch, 3-State Output

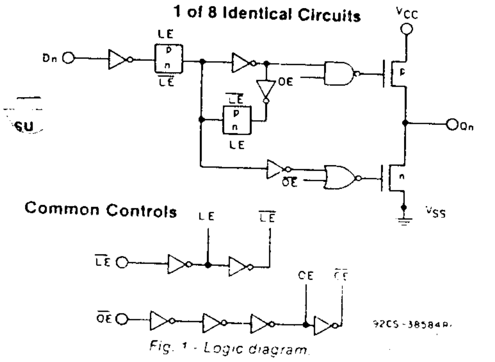
Type Features:

- *Common latch enable control*
- *Common 3-state output enable control*
- *Buffered inputs*
- *3-State outputs*
- *Bus line driving capacity*
- *Typical propagation delay = 12 ns @ V_{CC} = 5V, C_L = 15 pF, T_A = 25° C (Data to Output for HC373)*

The RCA CD54/74HC373/573 and CD54/74HCT373/573 are high speed Octal Transparent Latches manufactured with silicon gate CMOS technology. They possess the low power consumption of standard CMOS integrated circuits, as well as the ability to drive 15 LSTTL devices. The CD54/74HCT373/573 are functionally as well as pin compatible with the standard 54/74LS373 and 573.

The outputs are transparent to the inputs when the latch enable ($\overline{LE}$) is high. When the latch enable ($\overline{LE}$) goes low the data is latched. The output enable ($\overline{OE}$) controls the 3-state outputs. When the output enable ($\overline{OE}$) is high the outputs are in the high impedance state. The latch operation is independent to the state of the output enable. The 373 and 573 are identical in function and differ only in their pinout arrangements.

The CD54HC/HCT373/573 are supplied in 20 lead ceramic dual-in-line packages (F suffix). The CD74HC/HCT373/573 are supplied in a 20-lead plastic dual-in-line plastic package (E suffix) and in 20-lead surface mount plastic packages (M suffix). Both types are also available in chip form (H suffix).


1 of 8 Identical Circuits

Common Controls

Fig. 1 - Logic diagram

Family Features:

- *Fanout (Over Temperature Range):*
  Standard Outputs - 10 LSTTL Loads
  Bus Driver Outputs - 15 LSTTL Loads
- *Wide Operating Temperature Range:*
  CD74HC/HCT: -40 to +85° C
- *Balanced Propagation Delay and Transition Times*
- *Significant Power Reduction Compared to LSTTL Logic ICs*
- *Alternate Source is Philips/Signetics*
- *CD54HC/CD74HC Types:*
  2 to 6 V Operation
  High Noise Immunity: N_{IL} = 30%, N_{IH} = 30% of V_{CC}; @ V_{CC} = 5 V
- *CD54HCT/CD74HCT Types:*
  4.5 to 5.5 V Operation
  Direct LSTTL Input Logic Compatibility
  V_{IL} = 0.8 V Max., V_{IH} = 2 V Min.
  CMOS Input Compatibility
  $I_I \leq 1\ \mu A$ @ V_{OL}, V_{OH}

TRUTH TABLE

$\overline{\text{Output}}$ Enable	$\overline{\text{Latch}}$ Enable	Data	Output
L	H	H	H
L	H	L	L
L	L	l	L
L	L	h	H
H	X	X	Z

Note:

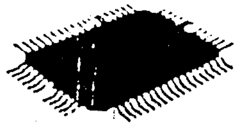
L = Low voltage level | X = Don't Care
H = High voltage level | Z = High Impedance State
l = Low voltage level one set-up time prior to the high to low latch enable transition
h = High voltage level one set-up time prior to the high to low latch enable transition

HD61830B (Dot Matrix Liquid Crystal Graphic Display Controller)

The HD61830B is a dot matrix liquid crystal graphic display controller LSI that stores the display data sent from an 8-bit microcomputer in the external RAM to generate dot matrix liquid crystal driving signals.

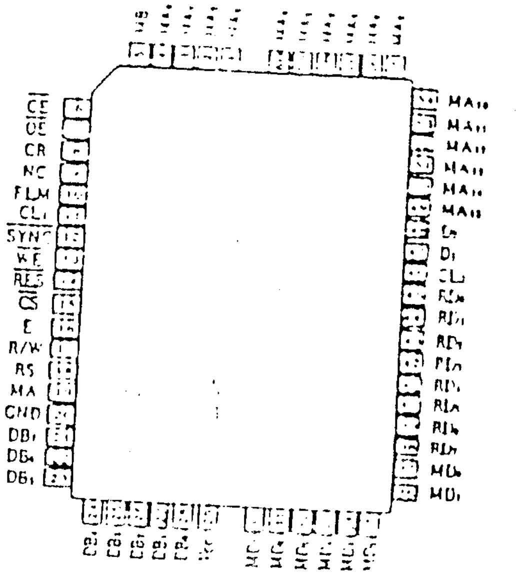
It is possible to select the graphic mode in which the 1-bit data of the external RAM corresponds to the ON/OFF state of 1 dot on liquid crystal display and the character mode in which characters are displayed by storing character codes in the external RAM and developing them into the dot patterns with the internal character generator ROM. Both modes can be provided for various applications.

The HD61830B is produced in the CMOS process Thus, the combination with a CMOS microcomputer can accomplish a liquid crystal display device with lower power dissipation.



(FP-60)

PIN ARRANGEMENT



(Top View)

FEATURES

- Dot matrix liquid crystal graphic display controller
- Display control capacity

  ✓Graphic mode .... 512K dots (2^{16} bytes)

  Character mode ..... 4096 characters (2^{12} characters)
- Internal character generator ROM ..... 7360 bits

  160 types of 5×7 dot character fonts

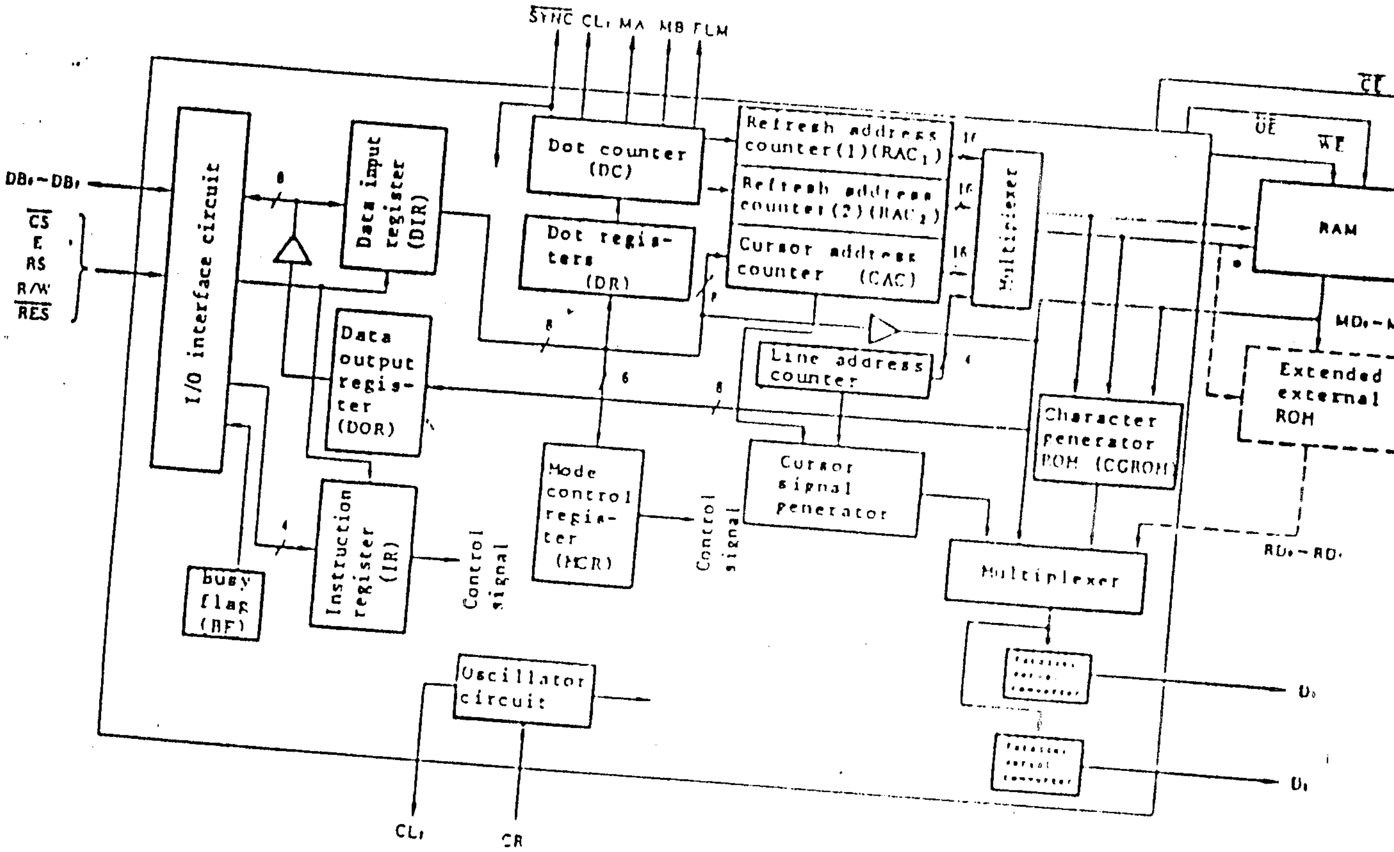
  32 types of 5×11 dot character fonts Total 192 types

  (Can be extended to 256 types (4K bytes max.) by external ROM)
- Interfaceable to 8-bit MPU
- Display duty (Can be selected by a program)

  Static to 1/128 duty selectable ?
- Various instruction functions

  Scroll, Cursor ON/OFF/blink, Character blink, Bit manipulation
- Display method ..... Selectable A or B types
- Operating frequency 2.4MHz
- Low power dissipation
- Power supply: Single +5V
- CMOS process
- 60-pin flat plastic package

BLOCK DIAGRAM



* When extended external ROM is used MA0 ~ MA11 are applied to RAM, MA12 ~ MA15 are applied to extended external ROM.

BLOCK FUNCTIONS

Registers

The HD61830B has the five types of registers: instruction register (IR), data input register (DIR), data output register (DOR), dot registers (DR) and mode control register (MCR).

The IR is a 4-bit register which stores the instruction codes for specifying MCR, DR, a start address register, a cursor address register and so on. The lower order 4 bits DB0 to DB3 of data buses are written in it.

The DIR is an 8-bit register used to temporarily store the data written into the external RAM, DR, MCR and so on.

The DOR is an 8-bit register used to temporarily store the data read from the external RAM. Cursor address information is written into the cursor address counter (CAC) through the DIR. When the memory read instruction is set in the IR (latched at the falling edge of E signal), the data of external RAM is read to DOR by an internal operation. The data is transferred to the MPU by reading the DOR with the next instruction (the contents of DOR are output to the data bus when E is at "High" level).

The DR are registers used to store the dot informations such as character pitches and the number of vertical dots and so on. The information sent from the MPU is written into the DR via the DIR.

The MCR is a 6-bit register used to store the data which specifies states of display such as display ON/OFF and cursor ON/OFF/blink etc. The information sent from the MPU is written in it via the DIR.

Busy Flag (BF)

With "1", the busy flag indicates the HD61830B is performing an internal operation. The next instruction cannot be accepted. As shown in Control Instruction (14), the busy flag is output on DB7 under the conditions of RS=1, R/W=1 and E=1. Make sure the busy flag is "0" before writing the next instruction.

Dot Counters (DC)

The dot counters are counters that generate liquid crystal display timing according to the contents of DR.

- Refresh Address Counters (RAC1/RAC2)

The refresh address counters are counters used to control the addresses of external RAM, character generator ROM (CGROM) and extened external ROM having the two types: RAC1 and RAC2. The RAC1 is used for upper half of screen and the RAC2 for lower half. In the graphic mode, 16-bit data is output and used as the address signal of external RAM. In the character mode, the high order 4 bits (MA12 ~ MA15) are ignored. The 4 bits of line address counter are output instead of it and used as the address of extended ROM.

- Character Generator ROM

The character generator ROM has 7360 bits in total and [illegible] 192 types of character data. A character code (8 bits) from the external RAM and a line code (4 bits) from the line address counter are applied to its address signals, and it outputs 5-bit dot data.

The character font is 5X7 (160 types) or 5X11 ([illegible]) The use of extended ROM allows 8X16 (256 types max.) to be used.

- Cursor Address Counter

The cursor address counter is a 16-bit counter that can be preset by the instruction. It is used to hold an address when the data of external RAM is read or written (when display dot data or a character code is read or written). The value of cursor address counter is automatically increased by 1 after the display data is read or written and after the Set/Clear Bit instruction is executed.

- Cursor Signal Generator

The cursor can be displayed by the instruction in the character mode. The cursor is automatically generated on the display specified by the cursor address and cursor position.

- Parallel/Serial Conversion

The parallel data sent from the external RAM, character generator ROM or extended ROM is converted into serial data by two parallel/serial conversion circuits and transferred to the liquid crystal driver circuits for upper screen and lower screen simultaneously.

Name	Function
$\overline{CE}$	Chip enable $\overline{CE}$=0 ... Chip enable make external RAM in active.
$\overline{OE}$	Output enable $\overline{OE}$=1 ... Output enable informs external RAM that HD61830B requires data bus.
NC	Unused terminal. Don't connect any wires to this terminal.

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit	Note
Supply voltage	V_{CC}	-0.3 ∿ +7.0	V	1,2
Terminal voltage	V_T	-0.3 ∿ V_{CC}+0.3	V	1,2
Operating temperature	Topr	-20 ∿ +75	°C	
Storage temperature	Tstg	-55 ∿ +125	°C	

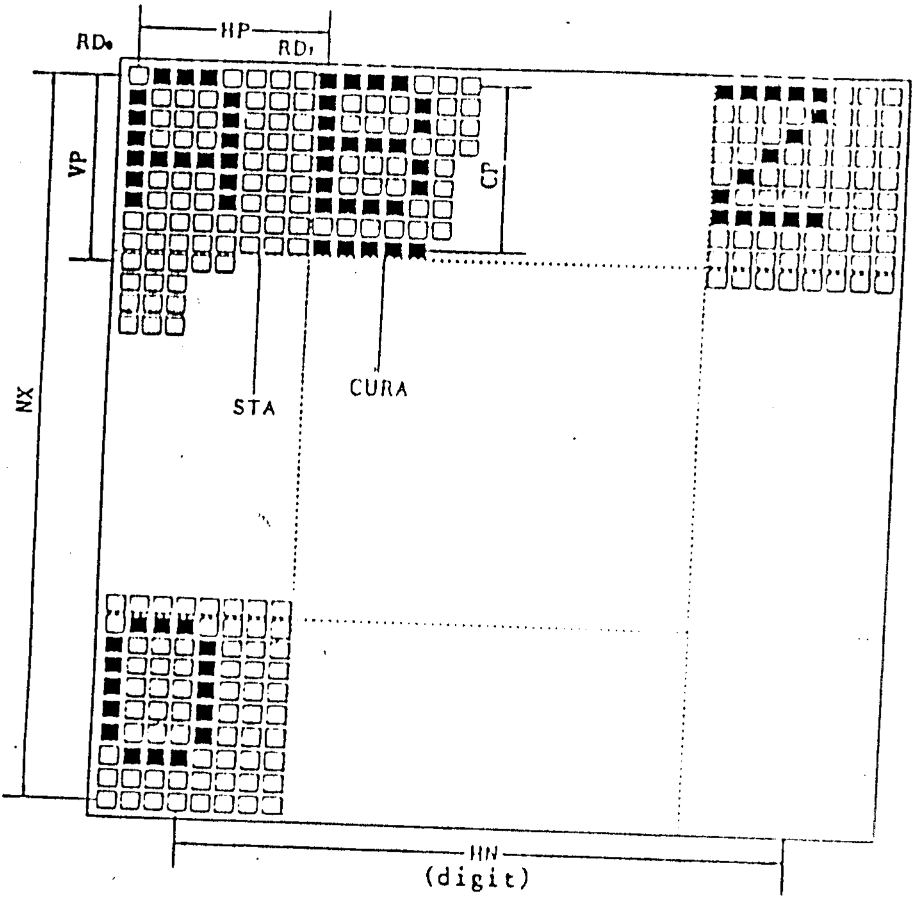
Note 1: All voltage is referred to GND=0V.

Note 2: If LSI's are used beyond absolute maximum ratings, they may be permanently destroyed. We strongly recommend you to use the LSI's within electrical characteristic limits for normal operation, because use beyond these conditions will cause malfunction and poor reliability.

TERMINAL FUNCTIONS

Name	Function
DB0~7	Data bus ... Three-state I/O common terminal Data is transferred to MPU through DB0 to DB7.
$\overline{CS}$	Chip select ... Selected state with $\overline{CS}$=0.
R/W	Read/Write ... R/W=1 ... MPU ← HD61830B R/W=0 ... MPU → HD61830B
RS	Register select ...,RS=1 ... Instruction register RS=0 ... Data register
E	Enable ... Data is written at the fall of E. Data can be read while E is 1.
CR	External clock input.
$\overline{RES}$	Reset ... $\overline{RES}$=0 results in display OFF, slave mode and Hp=6.
MA0~15	External RAM address output In character mode, the line code for external CG is output through MA12 to MA15 ("0": Character 1st line, "F": Character 16th line).
MD0~7	Display data bus ... Three-state I/O common terminal.
RD0~7	ROM data input ... Dot data from external character generator is input.
$\overline{WE}$	Write enable ... Write signal for external RAM.
CL2	Display data shift clock for LCD drivers.
CL1	Display data latch signal for LCD drivers.
FLM	Frame signal for display synchronization.
MA	Signal for converting liquid crystal driving signal into AC, A type.
MB	Signal for converting liquid crystal driving signal into AC, B type.
D1, D2	Display data serial output D1 ... For upper half of screen D2 ... For lower half of screen
$\overline{SYNC}$	Synchronous signal for parallel operation. Three-state I/O common terminal (with pull-up MOS). Master ... Synchronous signal is output. Slave .... Synchronous signal is input.

-to be continued



Symbol	Name	Meaning	Value
Hp	Horizontal character Pitch	Lateral character pitch	6 to 8 dots
H_N	Number of horizontal characters	Number of lateral characters per line (number of digits) in the character mode or number of bytes per line in the graphic mode.	2 to 128 digits (an even number)
Vp	Vertical character pitch	Longitudinal character pitch	1 to 16 dots
Cp	Cursor position	Line number on which the cursor can be displayed	1 to 16 lines
Nx	Number of time division	Inverse of display duty ratio	1 to 128 lines

Note: if the number of vertical dots on screen is taken as m, and the number of horizontal dots as n,

$1/m = 1/Nx$ = display duty ratio

$n = Hp \times H_N$, m/Vp = Number of display lines

$Cp \leq Vp$

OKI semiconductor

MSM6242RS/GS

IRECT BUS CONNECTED CMOS REAL TIME CLOCK/CALENDAR

ENERAL DESCRIPTION

The MSM6242 is a silicon gate CMOS Real Time Clock/Calendar for use in direct bus-connection Micro-ocessor/Microcomputer applications. An on-chip 32.768KHz crystal oscillator time base is divided to provide dressable 4-bit I/O data for SECONDS, MINUTES, HOURS, DAY OF WEEK, DATE, MONTH and YEAR. Data cess is controlled by 4-bit address, chip selects ($\overline{\text{CS0}}$, CS1), $\overline{\text{WRITE}}$, $\overline{\text{READ}}$, and ALE. Control Registers D, E d F provide for 30 SECOND error adjustment, INTERRUPT REQUEST (IRQ FLAG) and BUSY status bits, clock OP, HOLD, and RESET FLAG bits, 4 selectable INTERRUPTS rates are available at the STD.P (STANDARD ULSE) output utilizing Control Register inputs T0, T1 and the ITRPT/STND (INTERRUPT/STANDARD). asking of the interrupt output (STD.P) can be accomplished via the MASK bit. The MSM6242 can operate in 12/24 hour format and Leap Year timing is automatic.

The MSM6242 normally operates from a 5V ± 10% supply at −30 to 85°C. Battery backup operation down 2.0V allows continuation of time keeping when main power is off. The MSM6242 is offered in a 18-pin astic DIP and FLAT package.

EATURES

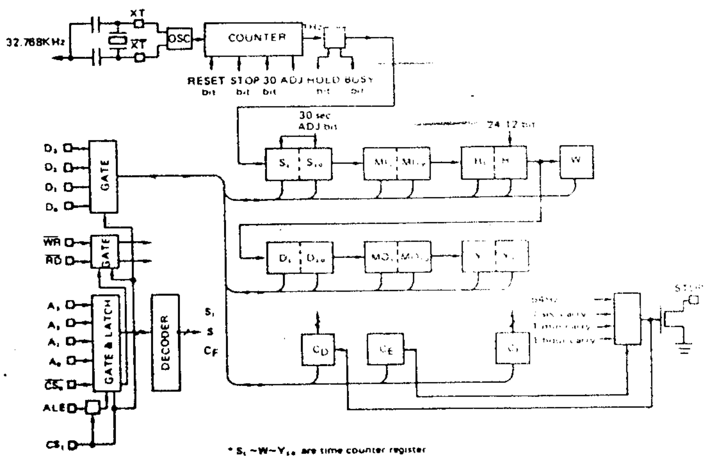
IRECT MICROPROCESSOR/MICROCONTROLLER BUS CONNECTION

ME	MONTH	DATE	YEAR	DAY OF WEEK
:59:59	12	31	80	7

- 4-bit data bus
- 4-bit address bus
- $\overline{\text{READ}}$, $\overline{\text{WRITE}}$, ALE and CHIP SELECT INPUTS
- Status registers — IRQ and BUSY
- Selectable interrupt outputs — 1/64 second, 1 second, 1 minute, 1 hour
- Interrupt masking
- 32.768KHz crystal controlled operation

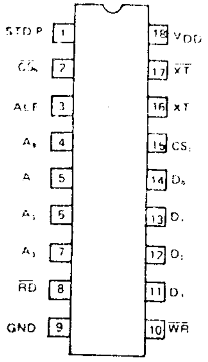
- 12/24 hour format
- Auto leap year
- ±30 second error correction
- Single 5V supply
- Battery backup down to V_{DD} = 2.0V
- Low power dissipation:
  - 20 μW max at V_{DD} = 2V
  - 150 μW max at V_{DD} = 5V
- 18-pin plastic DIP package

UNCTIONAL BLOCK DIAGRAM

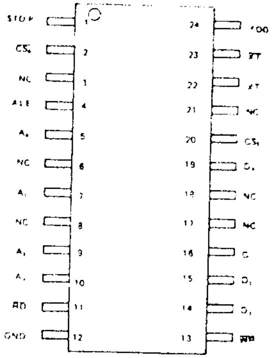


* S_1 ~W~Y_{10} are time counter register
* C_D ~ C_F are control register

PIN CONFIGURATION



18 pin Plastic DIP Package



24 pin Plastic Flat Package

A0-A3:	Address input
D0-D3:	Data input/output
$\overline{CS0}$, CS1	CHIP SELECTS 0, 1
$\overline{RD}$:	READ enable
$\overline{WR}$:	WRITE enable
ALE:	Address latch enable
STD.P:	Standard pulse output
XT, $\overline{XT}$:	XTAL oscillator input/output
VDD:.	+5V supply
VSS:	ground

REGISTER TABLE

Address Input	Address Input				Register Name	Data				Count value	Description
	A_3	A_2	A_1	A_0		D_3	D_2	D_1	D_0		
0	0	0	0	0	S_1	S_8	S_4	S_2	S_1	0 ~ 9	1-second digit register
1	0	0	0	1	S_{10}	•	S_{40}	S_{20}	S_{10}	0 ~ 5	10-second digit register
2	0	0	1	0	MI_1	mi_8	mi_4	mi_2	mi_1	0 ~ 9	1-minute digit register
3	0	0	1	1	MI_{10}	•	mi_{40}	mi_{20}	mi_{10}	0 ~ 5	10-minute digit register
4	0	1	0	0	H_1	h_8	h_4	h_2	h_1	0 ~ 9	1-hour digit register
5	0	1	0	1	H_{10}	•	PM/ AM	h_{20}	h_{10}	0 ~ 2 or 0 ~ 1	PM/AM, 10-hour digit register
6	0	1	1	0	D_1	d_8	d_4	d_2	d_1	0 ~ 9	1-day digit register
7	0	1	1	1	D_{10}	•	•	d_{20}	d_{10}	0 ~ 3	10-day digit register
8	1	0	0	0	MO_1	mo_8	mo_4	mo_2	mo_1	0 ~ 9	1-month digit register
9	1	0	0	1	MO_{10}	•	•	•	MO_{10}	0 ~ 1	10-month digit register
A	1	0	1	0	Y_1	y_8	y_4	y_2	y_1	0 ~ 9	1-year digit register
B	1	0	1	1	Y_{10}	y_{80}	y_{40}	y_{20}	y_{10}	0 ~ 9	10-year digit register
C	1	1	0	0	W	•	w_4	w_2	w_1	0 ~ 6	Week register
D	1	1	0	1	C_D	30 sec. ADJ	IRQ FLAG	BUSY	HOLD	—	Control Register D
E	1	1	1	0	C_E	t_1	t_0	ITRPT /STND	MASK	—	Control Register E
F	1	1	1	1	C_F	TEST	24/12	STOP	REST	—	Control Register F

SCILLATOR FREQUENCY DEVIATIONS

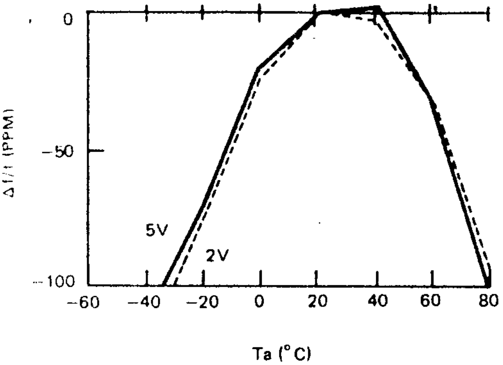


Figure 2. Frequency Deviation (PPM) vs Temperature

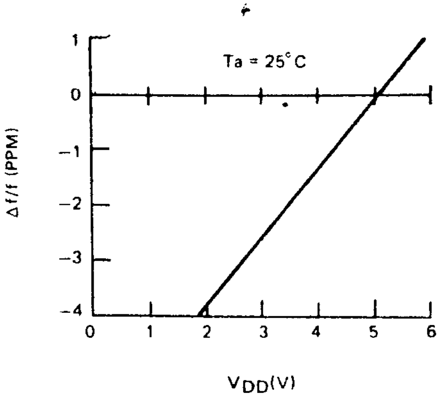
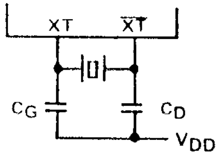


Figure 3. Frequency Deviation (PPM) vs Voltage

lote: 1. The graphs above showing frequency deviation vs temperature/voltage are primarily characteristic of the MSM6242 with the oscillation circuit described below.

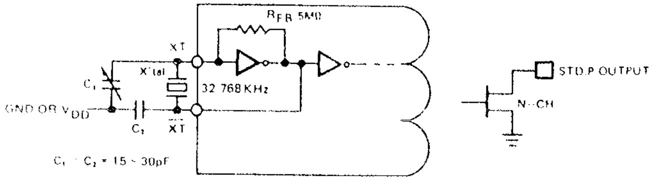


Crystal: Type N_8, P_3 by kinseki (32.768 KHz)

C_G, C_D: 22pF (Temperature Characteristics: 0)

PIN DESCRIPTION

Name	Pin No.		Description
	RS	GS	
D_0	14	19	Data Input/Output pins to be directly connected to a microcontroller bus for reading and writing of the clock/calendar's registers and control registers. D0 = LSB and D3 = MSB.
D_1	13	16	
D_2	12	15	
D_3	11	14	
A_0	4	5	Address input pin for use by a microcomputer to select internal clock/calendar's registers and control registers for Read/Write operations (See Function Table Figure 1). Address input pins A0-A3 are used in combination with ALE for addressing registers.
A_1	5	7	
A_2	6	9	
A_3	7	10	
ALE	3	4	Address Latch Enable pin. This pin enables writing of address data when ALE = 1 and $\overline{CSO}$ = 0; address data is latched when ALE = 0 Microcontroller/Microprocessors having an ALE output should connect to this pin; otherwise it should be connected at V_{DD}.
$\overline{WR}$	10	13	Writing of data is performed by this pin. When CS_1 = 1 and $\overline{CS_0}$ = 0, $D_0 \sim D_3$ data is written into the register at the rising edge of $\overline{WR}$.
$\overline{RD}$	8	11	Reading of register data is accomplished using this pin. When CS_1 = 1, $\overline{CS_0}$ = 0 and $\overline{RD}$ = 0, the data of the register is output to $D_0 \sim D_3$. If both $\overline{RD}$ and $\overline{WR}$ are set at 0 simaltanuously, $\overline{RD}$ is to be inhibited.
$\overline{CS_0}$	2	2	Chip Select Pins. These pins enable/disable ALE, $\overline{RD}$ and $\overline{WR}$ operation. $\overline{CS_0}$ and ALE work in combination with one another, while CS_1 work independent with ALE. CS_1 must be connected to power failure detection as shown in Figure 18.
CS_1	15	20	
STD.P	1	1	Output pin of N-CH OPEN DRAIN type. The output data is controlled by the D_1 data content of C_E register. This pin has a priority to $\overline{CS_0}$ and CS_1. Refer to Figure 9 and FUNCTIONAL DESCRIPTION OF REGISTERS.
XT	16	22	32.768 kHz crystal is to be connected to these pins. When an external clock of 32.768 kHz is to be used for MSM6242's oscillation source, either CMOS output or pull-up TTL output is to be input from XT, while $\overline{XT}$ should be left open.
$\overline{XT}$	17	23	
V_{DD}	18	24	Power supply pin. +2 ~ 6V power is to be applied to this pin.
GND	9	12	Ground pin.



The impedance of the crystal should be less than 30kΩ

FUNCTIONAL DESCRIPTION OF REGISTERS

■ $S_1, S_{10}, MI_1, MI_{10}, H_1, H_{10}, D_1, D_{10}, M\bar{O}_1, M\bar{O}_{10}, Y_1, Y_{10}, W$

a) These are abbreviations for SECOND1, SECOND10, MINUTE1, MINUTE10, HOUR1, HOUR10, DAY1, DAY10, MONTH1, MONTH10, YEAR1, YEAR10, and WEEK. These values are in BCD notation.

b) All registers are logically positive. For example, (S8, S4, S2, S1) = 1001 which means 9 seconds.

c) If data is written which is out of the clock register data limits, it can result in erroneous clock data being read back.

d) PM/AM, h_{20}, h_{10}

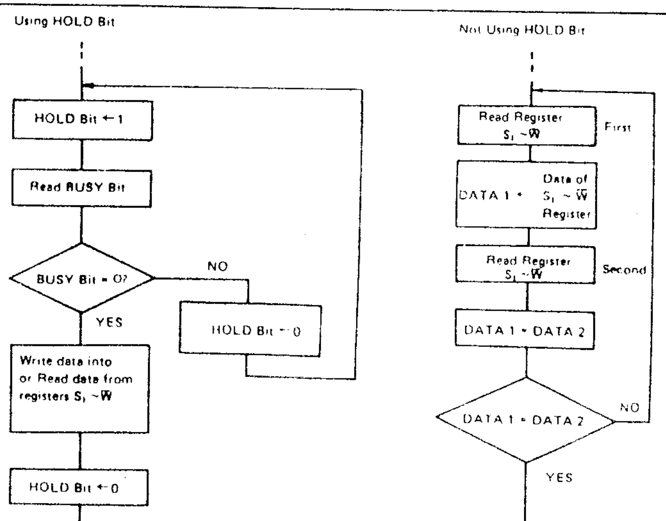
In the mode setting of 24-hour mode, PM/AM bit is ignored, while in the setting of 12-hour mode h_{20} is to be set. Otherwise it causes a discrepancy. In reading out the PM/AM bit in the 24-hour mode, it is continuously read out as 0. In reading out h_{20} bit in the 12-hour mode, 0 is written into this bit first, then it is continuously read out as 0 unless 1 is being written into this bit.

e) Registers Y1, Y10, and Leap Year. The MSM6242 is designed exclusively for the Christian Era and is capable of identifying a leap year automatically. The result of the setting of a non-existant day of the month is shown in the following example: If the date February 29 or November 31, 1985, was written, it would be changed automatically to March 1, or December 1, 1985 at the exact time at which a carry pulse occurs for the day's digit.

f) The Register W data limits are 0–6 (Table 1 shows a possible data definition).

TABLE 1

W_4	W_2	W_1	Day of Week
0	0	0	Sunday
0	0	1	Monday
0	1	0	Tuesday
0	1	1	Wednesday
1	0	0	Thursday
1	0	1	Friday
1	1	0	Saturday



Using HOLD Bit
HOLD Bit ← 1
Read BUSY Bit
BUSY Bit = 0?
NO
YES
HOLD Bit ← 0
Write data into or Read data from registers $S_1 \sim \overline{W}$
HOLD Bit ← 0
Not Using HOLD Bit
Read Register $S_1 \sim \overline{W}$
First
DATA 1 ← Data of $S_1 \sim \overline{W}$ Register
Read Register $S_1 \sim \overline{W}$
Second
DATA 1 ← DATA 2
DATA 1 = DATA 2
NO
YES

TYPICAL APPLICATIONS – INTERFACE WITH MSM80C49

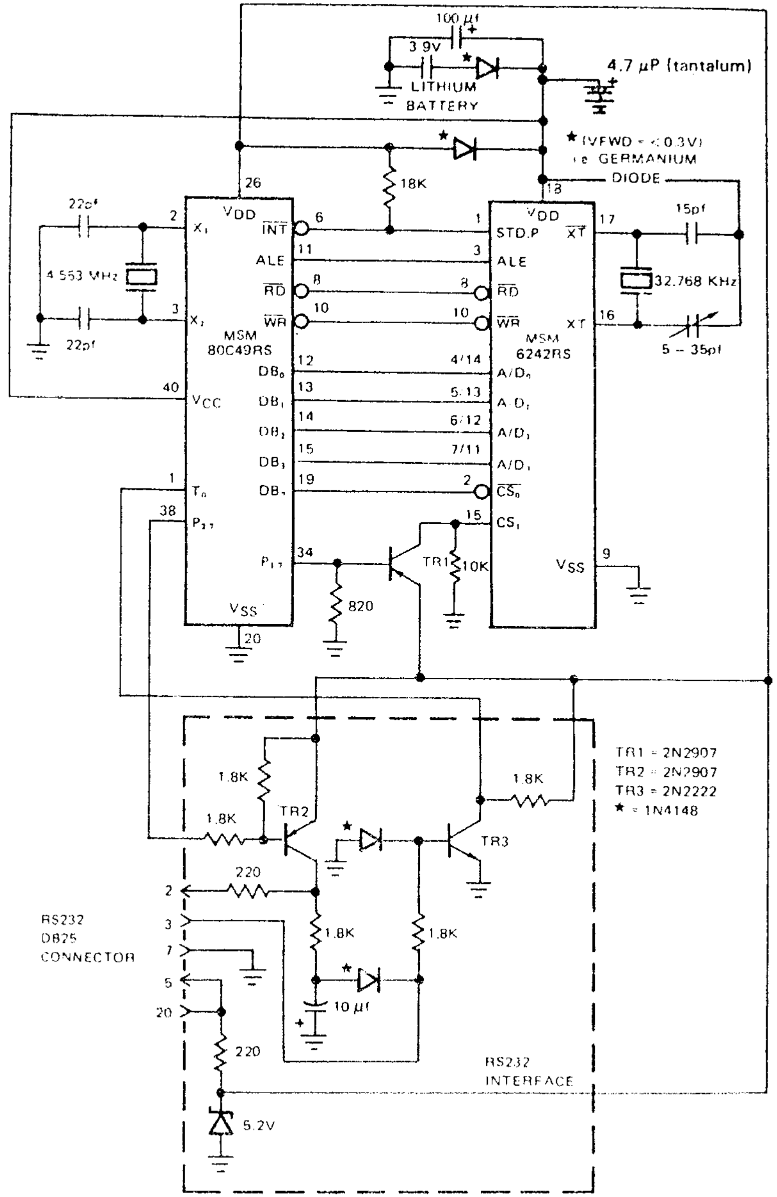
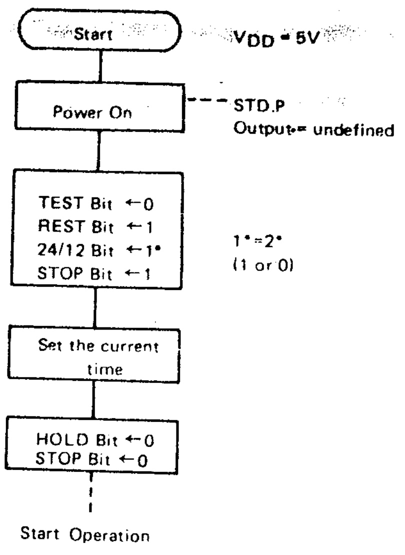


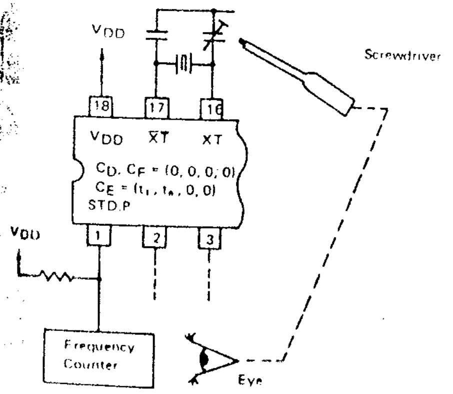
Figure 18.

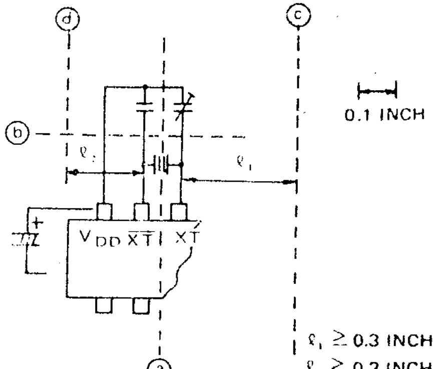
APPLICATION NOTE

1. Power Supply



2. Adjustment of Frequency





CD ~ CF are to be set at as described in the figure and the capacitor is to be adjusted to meet the settle frequency of t0 and t1.
If the right oscillation can not be obtained,

1. Check the waveform of $\overline{XT}$
2. Check CD ~ CF content
3. Check the noise.