DEVELOPMENT OF A DATA BUFFER AND INTERFACE CONVERTER

By Mark Alfred Mack

Thesis submitted in partial fulfilment of the requirements for the Masters Diploma in Technology to the Department of Electrical Engineering (light current) at the Cape Technikon.

Research and Development Centre

Telkom SA Ltd

Cape Town

South Africa

January 1994

To my wife, Venecia

ACKNOWLEDGEMENTS

On completion of this thesis, I would like to extend my thanks to the following individuals for their assistance and support:

- Mr. P.H. Kleinhans and Mr. C. Walser, my supervisors, for their encouragement and assistance throughout the duration of this project.
- Mr. G.V. Williams and his staff at the Telkom S.A. Research and Development Centre, Cape Town for the use of their facilities, their mutual support and with whom many helpfull discussions were held.
- Mr M. Amerika for always being prepared to be of assistance and for his ingenious ideas regarding enclosure design.
- Mrs. J.R. Du Plooy for the manufacture of numerous printed circuit boards.
- Fellow student technologists, engineers and friends for their encouragement and consideration.
- Mr. J.F. Malan and his staff at Bellville E10 OMC for their encouragement and assistance.
- My family and friends for their moral support and understanding.

DECLARATION

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

M. A. MACK

[illegible]

(Signature)

Abstract

The excessive usage of peripheral devices such as TTY's, VDU's and printers, have a significant load on the front-end processor of the Alcatel X83 minicomputer. Because of this, the response time of the front-end processor to the peripherals is excessive. This can also be attributed to the fact that the peripherals have small buffers or no buffers at all. These peripheral devices also operate at slow speeds. A slow response time from the mini-computer results in decreased productivity. The mini-computer does not provide ports for parallel printers even though there is an abundance of these type of printers. The unavailability of a variety of interfaces on the mini-computer makes the reconfiguration of the peripheral network inflexible. Replacing the existing peripheral equipment in order to overcome the above limitations will be costly. This thesis describes the design, development and implementation of a data buffer and interface converter which would overcome the above mentioned limitations.

Opsomming

Die oormatige gebruik van koppeltoerusting soos bv. TTY's, VDU's en drukkers het 'n noemenswaardige las op die voorent verwerker van die Alcatel X83 minikomper. As gevolg hiervan word die reaksietyd van die voorent verwerker na die koppeltoerusting verhoog. Dit kan ook toegeskryf word aan die feit dat die koppeltoerusting klein of geen buffers het nie. Die koppeltoerusting werk ook teen 'n lae spoed. 'n Stadige reaksietyd vanaf die mini-komper het 'n velaging in produktiwieteit tot gevolg. Die minikomper maak ook nie voorsiening vir parallel poorte nie alhoewel daar 'n oorvloed van hierdie tipe drukkers is. Die onbeskikbaarheid van 'n verskeidenheid koppeltoerusting maak die herkonfigurasie van die koppeltoerusting koppelvlak netwerk onbuigsaam. Die vervanging van die huidige koppeltoerusting om sodoende die bogenoemde probleme te oorkom sal kostelik wees. Hierdie tesis beskryf die ontwerp, ontwikkeling en implementering van 'n data buffer en koppelvlak omsetter wat die bogenoemde probleme sal oorkom.

CONTENTS

1. Introduction

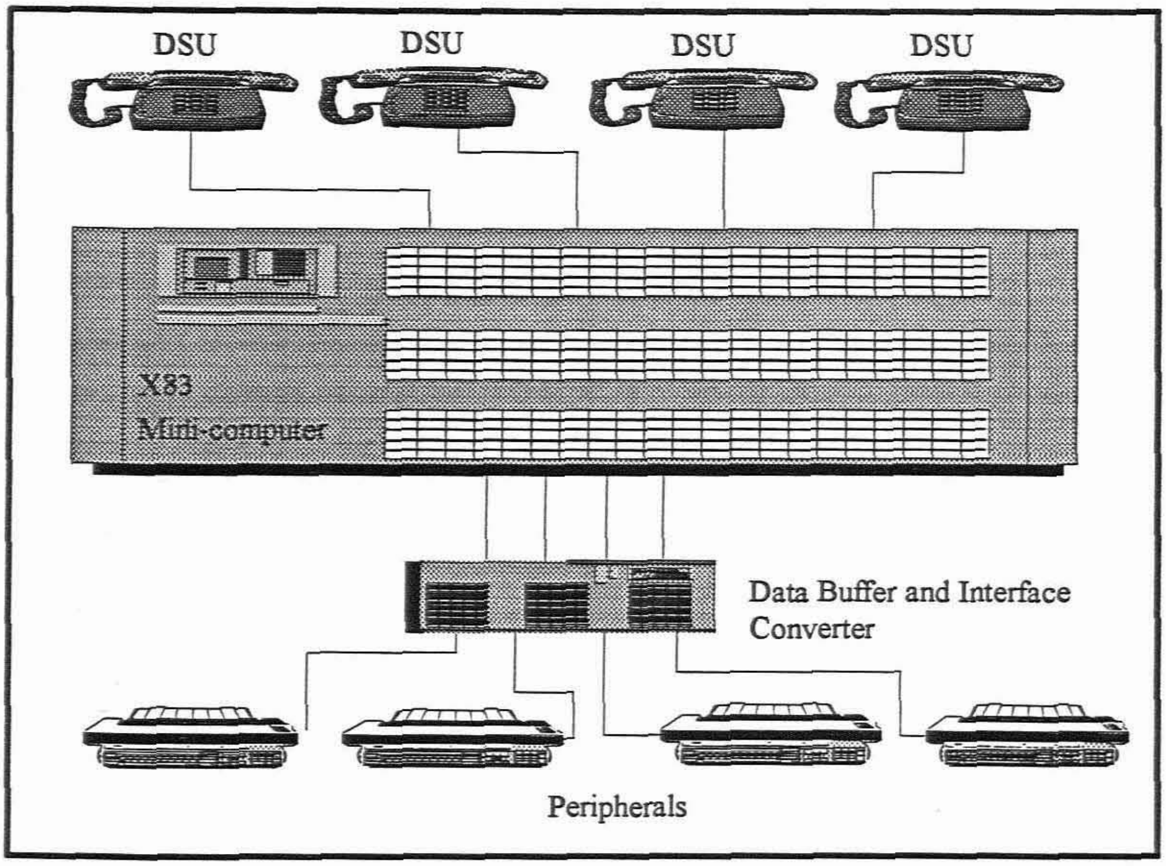


Fig. 1.0

1.1. The mini-computer in the electronic exchange environment

Digital switching units (DSU's) or otherwise known as electronic exchanges are managed by X83 mini-computers.

The X83 mini-computer (OMC83) manages four or more DSU's. A maximum of six DSU's are at present being managed by the X83 at certain sites. The following functions are performed by the X83;

- switching implemented by the E10B exchanges (DSU's)
- operations and maintenance implemented by the Operations

and Maintenance Centre (OMC)

The OMC83 provides the following facilities from a centralised administration site:

- standardisation, centralisation and management of operator terminals for the area;
- execution of commands and collection of processing results;
- accessing and modifying the content of exchange memories;
- collection and output of unsolicited messages from exchanges;
- preparation of results on tape for processing at computer centres.

The hardware organisation of the OMC83 comprises the following:

- 2 x 1.2GByte Hard disks
- 1 x Tape streamer unit
- 2 x Magnetic tape units
- main processor
- front-end processor
- 64kbit exchange links, asynchronous links (V24) and alarm links, all of which form part of the terminal bus.

1.2. The mini-computer peripherals

Peripherals are connected to the asynchronous links on the OMC83.The type of peripherals utilised are TTY's, VDU's, printers and intelligent terminals. All of the mentioned devices are serial devices, except intelligent terminals.

1.3. The Data Buffer and Interface Converter

Fig. 1.0 shows the Data Buffer and Interface Converter (DBIC) in the OMC83 environment. The DBIC is connected between the OMC83 and peripherals that requires interfacing and buffering. The DBIC consists of the following:

- Integrated power supply
- Console card
- 8 x Data Buffer and Interface Converter cards

1.3.1. The Console Card

The Console card consists of a microprocessor-controlled liquid crystal display (LCD) (two rows of 16 alphanumeric characters) and two keys. The LCD and keys make selection of the various parameters, that are sent to the data buffer and interface (DBIC) cards, possible. A serial bus between the console card and the DBIC cards provides for the communication between these cards.

1.3.2. The Data Buffer and Interface (DBIC) Cards

There are eight DBIC cards per Data Buffer and Interface Converter. Each DBIC card provides two serial (V24) ports, one Centronics port, 64kByte of buffering and a serial port for communication with the Console card. The DBIC card accepts, acknowledges, and implements the parameters sent to it by the Console card. These parameters consist of the following:

- the identification code (card number)
- required interfacing ie. serial to serial or serial to parallel
- word configuration of serial ports

The main functions of the DBIC card are to buffer the information which it receives at high speed from the mini-computer and to transmit this information to the peripherals at a slower speed. It also receives information from certain peripherals, such as TTY's, at slow speed and transmits this information to the minicomputer at high speed. The parallel interface that is provided on the DBIC card makes the use of parallel printers in the peripheral network possible.

2. Objective

The mini-computer discussed in chapter 1 has a front-end processor (FTD) whose function is to transfer messages between main memory and the transmission controllers via the terminal bus. The front-end processor (FTD) has 1Mbyte of RAM. The FTD handles the following peripherals via the terminal bus:

- TTY's
- VDU's
- printers
- intelligent terminals
- exchange links
- remote alarm links

The 1 Mbyte of RAM contained in the FTD can overflow due to excessive load from the peripherals. A full buffer indication from the FTD to the main processor will cause a degradation in the processing time of the main processor and consequently the overall response of the OMC83 will be degraded. The processing time of commands entered from terminals under these conditions is so excessive that it causes a reduction in productivity.

TTY's, VDU's and printers utilised have small buffers or no buffers at all. These peripheral buffers overflow when large listings are dumped or made, resulting in the FTD processing time increasing.

Certain peripherals operate at low baud rates (300 to 1200 baud) whereas the interfaces on the terminal bus can operate up to 9600 baud. The processing time of the FTD increases because of the low baud rates at which these peripherals communicate.

The cost to replace these terminals is high. Existing peripherals have been proven to be robust and extremely reliable whereas no record exist, within the OMC83 set-up, about the durability, reliability and robustness of new peripherals.

Parallel printers can not be connected to the interfaces on the terminal bus and furthermore imported data buffers and interface converters are expensive.

The limitations discussed above led to the development of a data buffer and interface converter whose purpose would be to accept data from a computer at the highest possible speed and transfer it to the printer at whatever (slower) speed the printer or terminal is capable of accepting. This saves time as the computer is free to perform other tasks while the buffer continues to send data to the printer or terminal. Serial as well as parallel interfaces are available so as to facilitate the reconfiguration of the peripheral network.

3. Hardware Implementation

Log and alarm terminals used in the OMC83 provide system and alarm information which is of utmost importance in the day to day running of the OMC83. This fact led to a modular design of the data buffer and interface converter. A unitary processor design would be unattractive in that the failure of the processor would result in the failure of multiple ports that are connected to the OMC83.

Further requirements that the data buffer and interface converter (DBIC) had to meet were:

a) provide eight data buffer and interface converter cards each with the following facilities;

   - 2 RS 232 standard interfaces
   - 1 Centronics compatible interface
   - each serial interface should have six selectable baud rates (300, 600, 1200, 2400, 4800, 9600). Full duplex working and hardwire handshaking must be supported. Word configuration : 7 or 8 data bits, 1 or 2 stop bits and parity.
   - leds to display RS232 signals as well as configuration of the DBIC card.
   - 64 kbyte buffer

b) provide a console card with the following facilities;

   - keys for the selection of the DBIC card and the configuration parameters that is to be sent to it.

- LCD display for the menu display.

c) provide an integral power supply unit.

3.1. Console Card

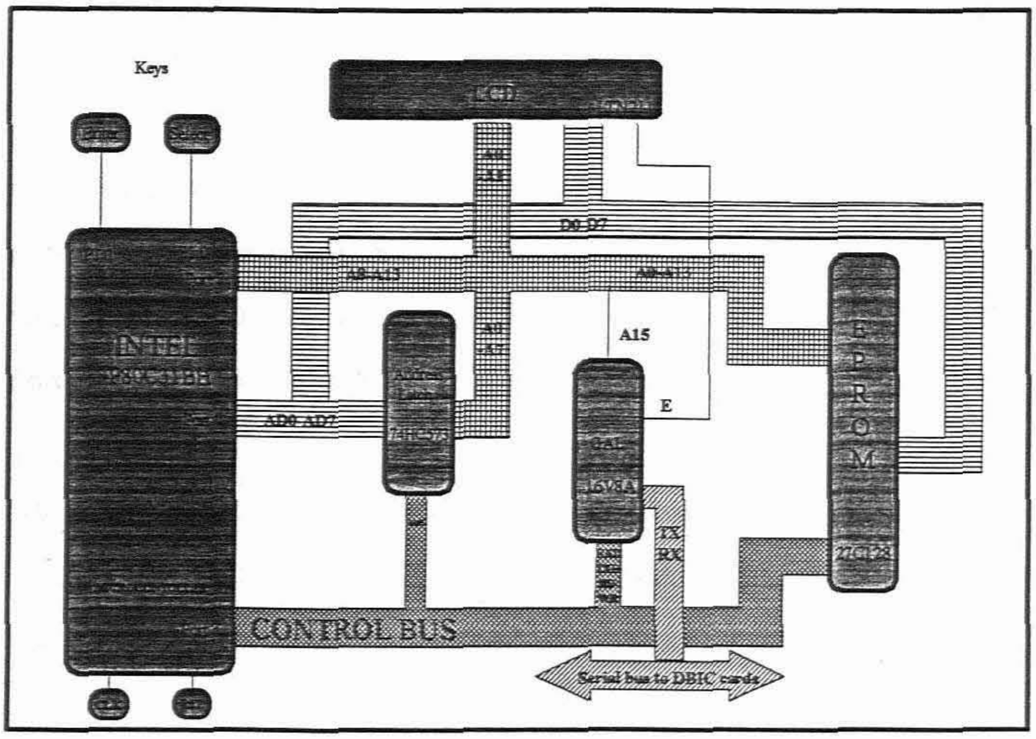


Fig. 2

3.1.1. CPU

The microprocessor utilised in this design is of the INTEL MCS-51 family ie. 80C31BH. The main features of the MCS-51 core are:

- 8-bit CPU
- On-chip oscillator and clock circuitry
- 32 I/O lines
- 64k address space for external data memory

- 64k address space for external program memory
- Two 16-bit timer/counters
- A five-source interrupt with two priority levels
- Full duplex serial port
- Boolean processor
- 128 bytes of on-chip RAM

Figure 2 shows the interconnection of the CPU ports to other chips as used on the Console card. The Control bus as indicated in Fig. 2 is connected to Port 3 of the CPU and consists of the following;

• P3.0	RXD	(serial input port)
• P3.1	TXD	(serial output port)
• P3.2	$\overline{\text{INT0}}$	(external interrupt)
• P3.3	$\overline{\text{INT1}}$	(external interrupt)
• P3.4	T0	(Timer/Counter 0 external input)
• P3.5	T1	(Timer/Counter 1 external input)
• P3.6	$\overline{\text{WR}}$	(external data memory write)
• P3.7	$\overline{\text{RD}}$	(external data memory read)

Only ports P3.0, P3.1, P3.6 and P3.7 were utilised in the Console card design. The onboard full duplex serial port, of which P3.0 and P3.1 form part, is used for the implementation of the serial bus between the Console and DBIC cards. This bus is utilised for DBIC card present detection and for the transmission of the selected parameters to the selected DBIC card. Ports P3.0 and 3.1 form the serial bus.

The CPU address bus is 16 bits wide. This bus is accomplished by an 8-bit multiplexed address/data bus (AD0-AD7) which forms the lower half of the 16-bit bus and Port 2 which forms the upper half. Multiplexing of the address/data bus is achieved with the use of an address latch (74HC573) which latches the address byte with the aid of the Address Latch Enable (ALE) signal onto the address bus. Data bytes appearing on AD0-AD7 after the lower address byte has been latched will not be output by the address latch because at this stage the ALE line would be low.

Port 1.0 and P1.1 have keys connected which are used for the selection of parameters displayed on the LCD display and the validation of these parameters respectively.

3.1.2. Memory

The onboard data memory which is 128 bytes in size was used. This is due to the fact that the amount of variables required in the software did not necessitate external RAM.

A UV erasable 27C128 (16kx8) EPROM IC was used for the program memory. Memory organisation of this IC is 16 384 words of 8 bits each. This IC is ideal for systems requiring low power, high performance and noise immunity. These features are due to the CHMOS process used in the manufacture of these IC's.

The execution of the code programmed into the EPROM by the CPU is facilitated by applying ground (Vss) to the $\overline{EA}$/Vpp pin (U3 pin 1 APPENDIX A page 42 C1) on the CPU. Instructions are read from the external program memory with the aid of the Program Store Enable (PSEN) line (U2 pin 29 APPENDIX A page 42 B1).

3.1.3. GAL.

The use of a Generic Array Logic (GAL) in this design was to firmiliarise myself with Erasable Programmable Logic Devices which are increasingly being used in system designs for the following reasons;

— SMALLER SYSTEM SIZES: Customised components allow for reducing chip count and saving board space, resulting in smaller system physical dimensions.

— LOWER SYSTEM COST: When custom LSI or VLSI components are used instead of standard SSI and MSI logic elements, there is a considerable saving in component cost per system, assembly and manufacturing cost, PCB area and board costs and inventory costs.

— HIGHER PERFORMANCE: Reduced number of IC's contributes to faster system speeds as well as lower power consumption.

- HIGHER RELIABILITY: Since probability of failure is directly related to the number of IC's in the system, a system composed of LSI and VLSI chips is statistically much more reliable than the identical system made up of SSI/MSI devices.

- DESIGN SECURITY: Systems designed with standard components can be replicated relatively easily whereas systems that contain user customised IC's can not be copied because reverse engineering of the customised IC's is extremely difficult. Thus, use of customised IC's allows for the protection of proprietary designs.

- INCREASED FLEXIBILITY: Customised components allow for the tailoring of systems to the end user's specific needs relatively easily. This allows for upgradability and obsolence protection.

The GAL used in the design of the Console card replaces the memory decoder and glue logic chips needed to interface the LCD display to the CPU.

3.2. Data Buffer and Interface Converter Card

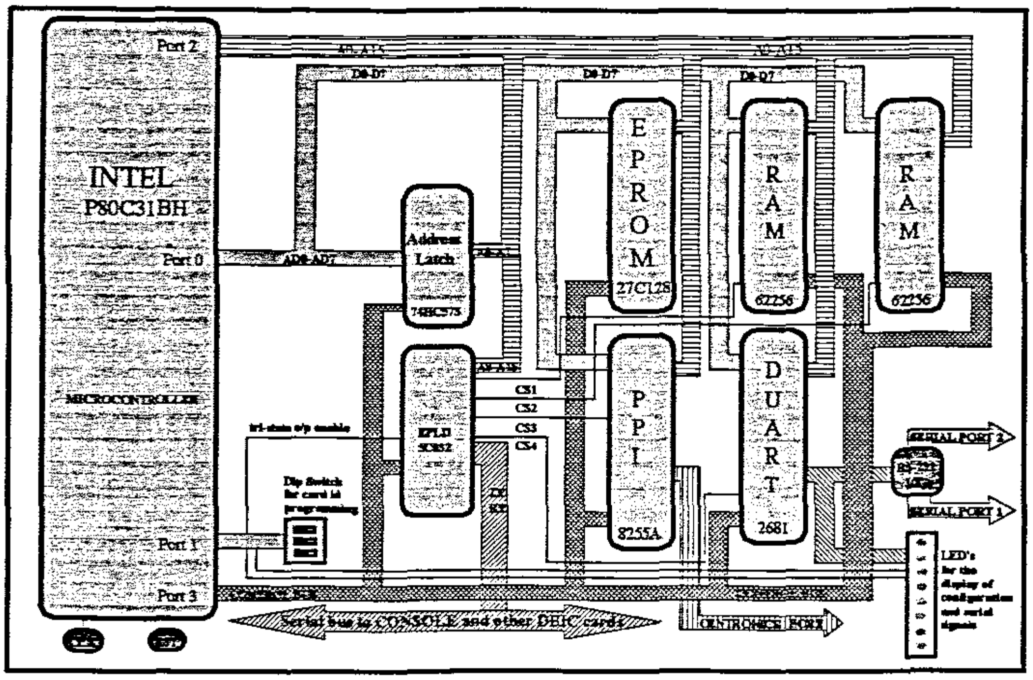


Fig 3

3.2.1. CPU

The same CPU utilised in the Console card design is used in this design. Ports 0 and 2 are used in the same manner as discussed in 3.1.1. Port 3 which forms the control bus consists of;

• P3.0	RXD	(serial input port)
• P3.1	TXD	(serial output port)
• P3.2	$\overline{\text{INT0}}$	(external interrupt) DUART interrupt
• P3.3	$\overline{\text{INT1}}$	(external interrupt) PPI interrupt
• P3.4	T0	(Timer/Counter 0 external input)
• P3.5	T1	(Timer/Counter 1 external input)
• P3.6	$\overline{\text{WR}}$	(external data memory write)

- P3.7 $\overline{\text{RD}}$ (external data memory read)

All Port 3 ports are used except P3.4 and P3.5 . Ports P3.0 and P3.1 are used for the serial bus as described in 3.1.1 .

The port pins of Port 1 are used in the following manner;

• P1.0	id bit 0
• P1.1	id bit 1
• P1.2	id bit 2
• P1.3	tri-state output enable (TR1)
• P1.4	not used
• P1.5	not used
• P1.6	serial to serial communication LED indication
• P1.7	serial to parallel communication LED indication

A dip switch (APPENDIX A page 43 C3) is connected to Port 1 pins P1.0-2 for programming the card identification.

3.2.2. Memory

For the program memory a 27C128 EPROM is used. This is the same type of EPROM that is utilised in the console design.

The data memory, most of which forms the data buffer, consists of two 62256 (32kx8) static RAM chips. The memory is organised as 32 768 words of 8bits each. The maximum amount of bytes which the data memory can store is 65 536. Of the 65 536 bytes, 64 300 bytes are used for buffering and the remainder for variables and registers.

Data memory accessing is affected by strobing the $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$ pins.

3.2.3. Dual Universal Asynchronous Receiver/Transmitter (DUART)

Universal Asynchronous Receivers/Transmitters are devices that are used to convert parallel data into serial data for transmission and convert received serial data to parallel data in communication systems.

The 2681 Dual Asynchronous Receiver/Transmitter (DUART) provides two independent full duplex asynchronous receiver/transmitter channels in a single package. The operating mode of each channel can be programmed independently.

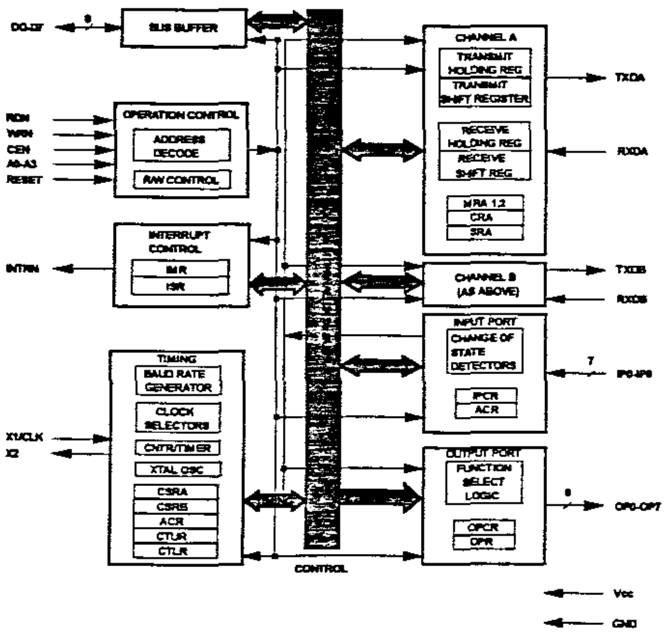


Fig 4

With reference to figure 4 the internal architecture of the DUART consists of the following;

- bus buffer
- operation control section
- interrupt control section
- timing section
- asynchronous channel A
- asynchronous channel B
- input port section
- output port section
- internal bus and signal lines

The data bus buffer provides the interface between the external and internal data busses. It is controlled by the operation control section to allow read and write operations to take place between the controlling CPU and the DUART.

The operation control section receives operation commands from the CPU and generates appropriate signals to internal sections to control device operation.

A single active low output (INTRN) is provided which is activated upon the occurrence of any of eight internal events in the interrupt control section. Associated with the interrupt system are the interrupt mask register (IMR) and the interrupt status register (ISR). The IMR may be programmed to select only certain conditions to cause INTRN to be asserted. The ISR can be read by the CPU to determine all currently active interrupting conditions. For the purpose of this thesis IMR bits 0, 1, 4 and 5 were used. These bits allow the corresponding bits in the ISR to be asserted. An interrupt occurring as a result of these bits will indicate to the CPU that the transmit holding register (THR) of channel A and or channel B is ready to be loaded with a character and that a character has been received in channel A and or channel B and is waiting to be read by the CPU.

The timing section consists of a crystal oscillator, a baud rate generator, a programmable 16-bit counter/timer and four clock selectors. Channel A Clock Select Register (CSRA) in association with the Auxiliary Control Register (ACR) makes the selection of the required baud rate possible.

Each asynchronous channel consists of a transmit holding register (THR), a transmit shift register, a receive holding register (RHR), a receive shift register and associated registers. These associated register are the Channel Mode Registers 1/2 (MR1A,MR2A,MR1B,MR2B), the Channel Command Registers (CRA,CRB) and the Status Registers (SRA,SRB). Selection of the configuration of each channel is accomplished by programming the relevant bits in MR1A, MR2A, MR1B and MR2B. Reception and transmission can be enabled or disabled by programming the relevant bits in CRA and CRB. Resetting the transmitters and receivers is also done in the same fashion. TXD and RXD signal lines are used for the transmission and reception of data respectively.

The input port section consists of an unlatched 7-bit port which can be read by the CPU. Two input port pins were used for both cannel A and B's clear to send (CTS) signals.

The output port section consists of an 8-bit multi-purpose output port which can be used as a general purpose output port. Only two output port pins were used for the implementation of the request to send signal lines of both channels.

3.2.4. Programmable Peripheral Interface (PPI)

The 8255A is a programmable peripheral interface (PPI) device. Its function is that of a general purpose I/O component to interface peripheral equipment to a microcomputer system bus. The functional configuration of the 8255A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

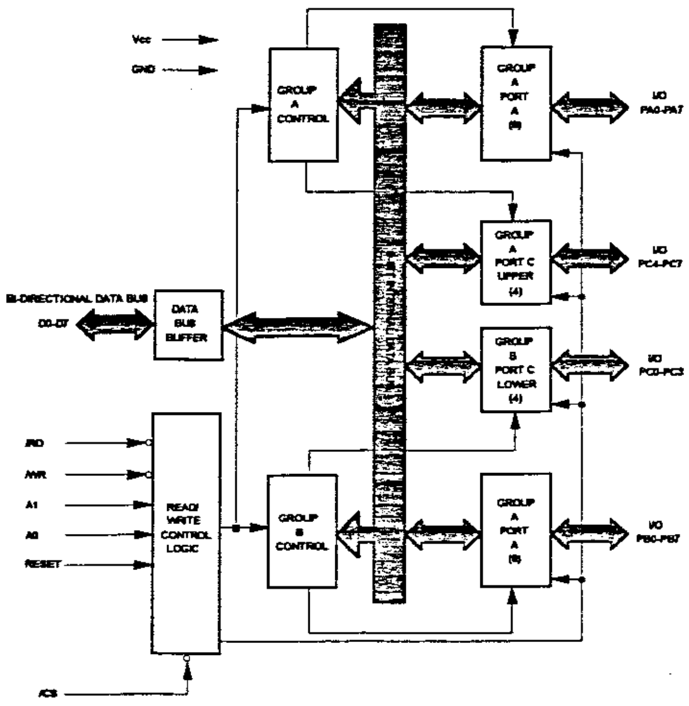


Fig 5

The data bus buffer as indicated in figure 5 is a 3-state 8-bit buffer which is used to interface the 8255A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

The function of the read/write control logic is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the

CPU Address and Control busses and in turn , issues commands to both of the Control Groups.

The 8255A contains three 8-bit ports (A,B and C). All can be configured to a variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255A. Port A consists of one 8-bit data output latch/buffer and one 8-bit data input buffer. Port B consists of one 8-bit data input/output latch/buffer and one data input buffer. Port C on the other hand consists of one 8-bit output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be devided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with Ports A and B. There are three basic modes of operation that can be selected by the system software:

- Mode 0 - Basic Input/Output
- Mode 1 - Strobed Input/Output
- Mode 2 - Bi-directional Bus

The 8255A was configured to Mode 0 , Port A, B and Port C lower as output and Port C higher as input, for the implementation of the parallel printer port. Port A formed the data lines to the printer, port pin PC0 (U11 pin 14 APPENDIX A page 43 B3) the $\overline{\text{STROBE}}$ signal line, port pin PC4 (U11 pin 13 APPENDIX A page 43 B3) the BUSY signal line, port pin PC5 (U11 pin 12 APPENDIX A page 43 B3) the $\overline{\text{ERROR}}$ signal line and port pin PC6 (U11 pin 11 APPENDIX A page 43 B3) the $\overline{\text{ACKNOWLEDGE}}$ signal line.

3.2.5. RS-232

The Maxim 232 line divers/receivers are ideal for applications where a 12V supply is not available as they only require a single 5V supply for operation. The RS-232 drivers/receivers have on-board charge pump voltage converters which convert the 5V input signals to the required RS-232 levels. The two MAX232 (U8/9 APPENDIX A page 43 A2/3) chips used in this instance contains two drivers and two receivers each which are connected between the DUART and the two RS-232 ports.

3.2.6. EPLD

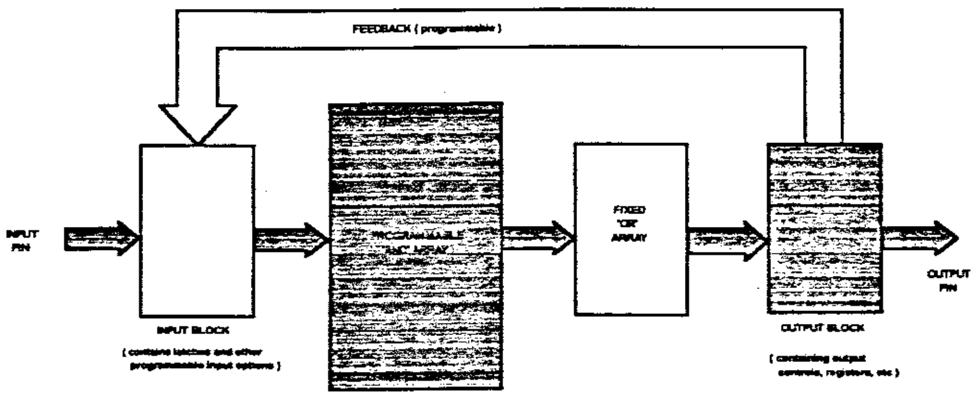

FEEDBACK (programmable)
INPUT PIN
INPUT BLOCK
(contains latches and other programmable input options)
PROGRAMMABLE "AND" ARRAY
FIXED "OR" ARRAY
OUTPUT BLOCK
(containing output controls, registers, etc)
OUTPUT PIN

Fig 6

Erasable Programmable Logic Devices (EPLD's) are similar to Generic Array Logic (GAL's) device as discussed in 3.1.3. Figure 6 gives diagrammatic presentation of the INTEL EPLD architecture. The advantage of EPLD's over GAL's are that EPLD's can operate in stand-by mode which reduces the power consumption considerably. In this design the INTEL 5C032 EPLD (U3 APPENDIX A page 43 C2) was used as a address decoder and buffer for the serial bus between the Console and DBIC cards. The transmit data (TXD) pin of the CPU are fed into the EPLD where it is tri-state buffered. The output of this pin is controlled by an $\overline{\text{OE}}$/TR1 input pin that is in turn controlled by the CPU. The CPU allows the transmission of data from the DBIC card to the Console card only if a valid identification code was sent by the Console card. This also prevents the loading of the input to the Console card by all eight transmit data outputs of the DBIC cards.

3.2.7. LED's

Light Emmitting Diodes (LED's) are used to indicate RS-232 signal conditions, parallel printer error and DBIC card configuration.

3.3. PSU

A linear Power Supply Unit (PSU) consisting of a rectifier bridge-smoothing capacitor-voltage regulator-filter configuration was used. The total current consumption of the completed DBIC unit is ±1,98 amperes. A 5V/3A voltage regulator such as the LM78H05 (U1 APPENDIX A page 44) was used in this design.

4. Software

Initial testing of completed Console and DBIC cards ensured that these cards were electronically correct.

The initial test code for the cards were written in assembler. Writing code in assembler was found to be laborious and I embarked on writing code in the C-language.

4.1. Program Development Tools

The C-language is a general-purpose programming language which combines code-efficiency , structured programming, comfortable data structures and a wide variety of operations. It is a high-level language which is not specialised to certain applications. Presently Cross Compilers are available which offer a way to program in 'C' which truly matches assembly programming in terms of code efficiency and speed.

Use of a high-level language such as C has many advantages over assembly language programming:

- Knowledge of the processor instruction set is not required, rudimentary knowledge of the memory structure of the 8051 CPU is desirable (but not necessary).

- Details like register allocation and the addressing of the various memory and data types is managed by the compiler.

- Programs get a formal structure and can be devided into separate functions. This leads to better program structure.

- The ability to combine variable selection with specific operations improves program readability.

- Keywords and operational functions can be used that more nearly resemble the human thought process.

- Programming and program test time is drastically reduced,this increases efficiency.

- The supplied and supported C libraries contain many standard routines such as:

  formatted output
  numeric conversions
  and floating point arithmetic.

- Existing program parts can be more easily included into new programs, because of the comfortable modular program construction techniques.

- The language C is a very portable language that enjoys wide popular support, and can be quickly adapted to other processors as needed.

A C-compiler for the INTEL MCS-51 microcontrollers was used to compile the C-code and an in-circuit EPROM emulator was used to run the written code. Debugging of the written code was accomplished by setting test bits in the code.

4.2. Memory Mapping

The program and data memory spaces of the INTEL MCS-51 family microcontollers are separate. The 16-bit address bus of the microcontroller is common to both program and data spaces. The difference in reading from these spaces lies in the fact that the program space has a program store enable ($\overline{\text{PSEN}}$) line for reading the code out of program memory space and the data space has a read ($\overline{\text{RD}}$) line for reading data out of data memory space.

The program memory space for both the Console and DBIC cards consists of 16kbyte each. No read only peripheral chips are mapped into these spaces.

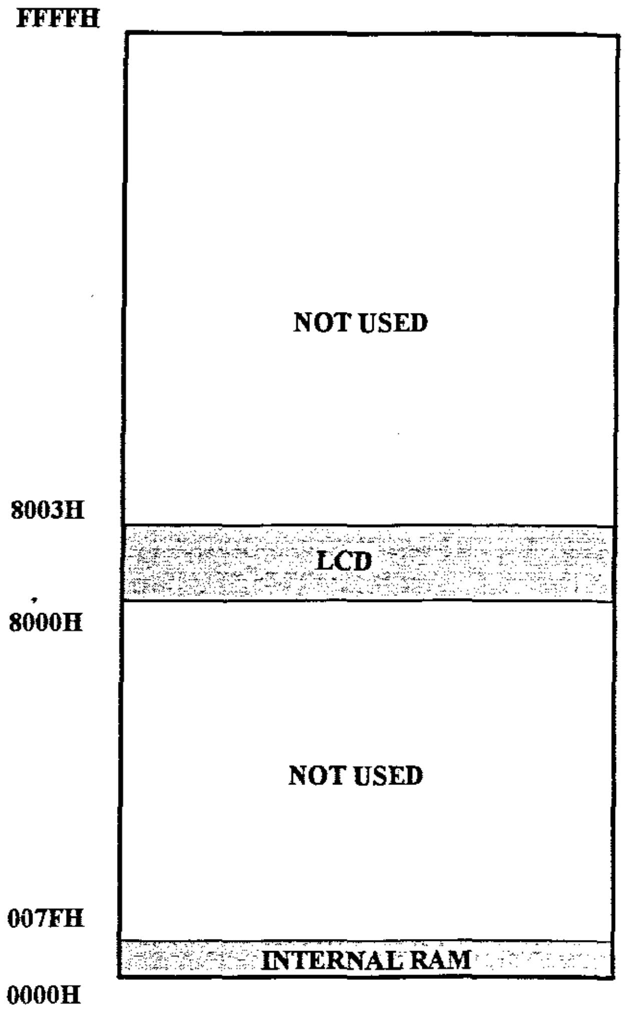


Fig 7

Figure 7 presents a layout of the data memory mapping as used in the Console card design. Data memory ranges from 0000H to 007FH. The LCD display is mapped with address line A15 into address space ranging from 8000H to 8FFFH. Address 8000H represents the data register address of the LCD and addresses 8001H to 8003H the command register address. Address 8004H to 8FFFH are redundant.

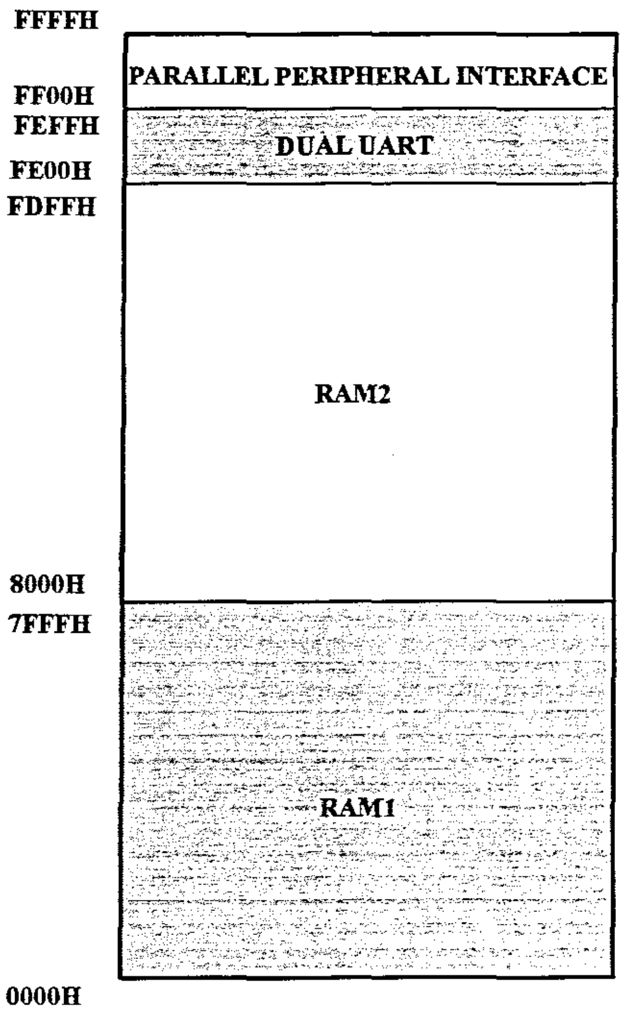


Fig 8

The data memory mapping of the DBIC card is portrayed in Fig. 8. With reference to Fig. 8, address 0000H to 7FFFH represents 32 767 bytes of data space which is contained in RAM1. The next address space is allocated to RAM2 and ranges from address 8000H to FDFFH and represents 32 255 bytes of data memory. A total amount of 65 022 bytes of data memory space is provided for. Of the 65 022 bytes of data memory, 64 000 bytes are used for buffer 1 and 300 bytes for buffer 2. Buffer 1 is used to buffer information coming from the high speed serial port and buffer 2 for information coming from the low speed serial port.

Space for the DUART is allocated from address FE00H to FEFFH. The various registers contained in the DUART are mapped into this address range in the following order;

FE00H	Mode Register A (MR1A,MR2A)
FE01H	Status Reg. A (SRA) & Clk. Select Reg. A (CSRA)
FE02H	Command Register A (CRA)
FE03H	RX Holding Reg. A (RHRA) & TX Hold. Reg. (THRA)
FE04H	I/P Port Change Reg. (IPCR) & Aux. Ctrl Reg. (ACR)
FE05H	Int. Status Reg. (ISR) & Int. Mask Reg. (IMR)
FE06H	Counter/Timer Upper (CTU) C/T Upper Reg. (CTUR)
FE07H	Counter/Timer Lower (CTL) C/T Lower Reg. (CTLR)
FE08H	Mode Reg. B (MR1B,MR2B)
FE09H	Status Reg. B (SRB) & Clock Select Reg. B (CSRB)
FE0AH	Command Reg. B (CRB)
FE0BH	RX Holding Reg. B (RHRB) & TX Hold. Reg. B (THRB)
FE0DH	Input Port & Output Port Configuration Reg. (OPCR)

The addresses for the PPI are allocated at the top of the memory map and ranges from FF00H to FFFFH. The three ports contained in the PPI are located in the mentioned address range at the following addresses;

FF00H	Port A
FF01	Port B
FF02	Port C
FF03	Control Word Reg.

4.3. EPLD and GAL programming

The address decoding on the Console and DBIC cards are implemented with a GAL and EPLD respectively. Equations (APPENDIX C page 51) for the address decoding are programmed into the GAL and EPLD with a universal programmer.

4.4. Console card

On power-up the LCD display, parameter buffer and on-board UART are initialised (APPENDIX B pages 46/7). The user is then prompted on the LCD display to select the DBIC card that needs to be configured. Parameter selection for the selected card is then done. The Console card then transmits the DBIC card identification code via the serial bus to all the DBIC cards. Only the selected DBIC card will respond by sending its identification code to the Console card. If no DBIC card identification is received by the Console card the Console card will display that the selected card is not present. However, if the Console card received the identification code it will then change the mode of the on-board UART and send the configuration parameters to the selected DBIC card. Only the selected DBIC card will receive the parameters because the parameters are sent in a different serial format. The selected DBIC card would at this stage have already changed its CPU's on-board UART to the required mode.

The Console card will then revert to prompting the user to select a DBIC card that needs to be configured.

4.5. Data Buffer and Interface Converter (DBIC) card

The on-board UART, PPI, DUART, buffers and card identification are initialised on power up (APPENDIX B page 48). The DBIC card then tests whether the Console card has sent a identification code. If an identification code is available it then tests if the identification code matches its identification code. Should the identification codes match, the DBIC card will send the same identification code to the Console card. The DBIC card will then change the mode of the on-board UART such that only the this DBIC card will accept the configuration parameters sent by the Console card. The DBIC card will implement the parameters as it receives it. The LED's D9 and D10 (APPENDIX A page 42 D3) will indicate serial to serial communication or serial to parallel communication respectively depending on the configuration selected by the user from the Console card.

If the identification code sent to the DBIC card does not match its identification it will ignore parameters sent to a selected DBIC card. It will then test whether a parallel printer error has occurred and if so it will then flash LED D10 (APPENDIX A page 43 D3). This condition will only occur if the serial to serial communication configuration was not selected.

The DBIC card will thereafter test whether the parallel port needs to be serviced. Should the test result be true it will service the parallel port. After the parallel port has been serviced or the test result not being true the DBIC card will then test whether the DUART needs to be serviced. If so, it will service the DUART and on completion or if the DUART did not require servicing the DBIC card will test whether the Console card has sent an identification code. The DBIC card will then proceed through the loop again.

4.6. Parallel Port

With reference to APPENDIX B page 49, servicing of the parallel port occurs only if the following conditions exist;

an interrupt occurred on interrupt one ($\overline{\text{INT1}}$) of the CPU and no busy condition is present from the printer and the DBIC has been configured for serial to parallel working.

If these conditions are met the parallel port service routine flag is cleared. Should buffer 1 be empty at this stage processing will be passed onto the main routine. If however there is information present in buffer 1 it will be sent to the PPI and then strobed to the printer. A buffer 1 full test ensures that the next reading of data from buffer 1 starts at the beginning of buffer 1 by resetting the read pointer of buffer 1. Further processing is then passed onto the main routine.

4.7. Serial Ports

The configuration parameters received from the Console card will determine whether both or one of the serial ports are used.

For the purpose of this thesis, only one of the serial ports will be described as both are exactly the same except for the fact that they use different buffers to read from and write to.

The configuration parameters sent to the DBIC card configures the serial port to the required baud rate, word configuration, handshaking, by programming the associated registers contained in the DUART.

The DUART is conditioned to transmit data when the transmitter is enabled through the command register (CRA). The DUART indicates to the CPU that it is ready to accept a character by setting the TxRDY bit in the status register (SRA). This condition is programmed to generate an interrupt request at INTRN (U5 pin 21 APPENDIX A page 43 A1). This interrupt condition is cleared when a character is loaded into the transmit holding register (THRA). Data is transferred from THRA to the transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY condition is then asserted again. The transmitter converts the parallel data from the CPU to a serial bit stream on the TXDA (U5 pin 30 APPENDIX A page 43 A1) output pin. It automatically sends a start bit followed by the programmed number of data bits, an optional parity bit and the programmed number of stop bits. The TXDA pin remains high until the next character is sent out. The output on the TXDA is controlled by the CTA (U5 pin 7 APPENDIX A page 43 B1) pin. A clear to send signal is sent on the CTA pin by the receiving device to indicate to the DUART that it is ready to receive data.

The DUART is conditioned to receive data when enabled through the CRA register. The receiver looks for transition of the start bit on the RXDA (U5 pin 31 APPENDIX A page 43 A1) Input pin. If a valid start bit is received the receiver continues sampling the input at one bit time intervals at the theoretical center of the bit, until the proper number of data bits and the parity bits (if any) have been assembled and one stop bit has been detected. The data is then transferred to the receive holding register (RHRA) and the RxRDY bit in the SRA register is set to one. This condition is programmed to give an interrupt at INTRN. After the stop bit has been detected, the receiver will immediately look for the next start bit. The RTA (U5 pin 29 APPENDIX A page 43 B1) output is used by the receiver to control the transmission of data from the transmitting device should a character not be read from the receive holding register by the CPU when the next character arrives.

With reference to APPENDIX B page 50 and APPENDIX D page 88 a description of the buffer handling under serial Port A transmit and receive conditions follows.

A Port A request reception condtion will test the state of buffer 1 before storing the received information. Lines 64 tO 78 of the source code listing (APPENDIX D page 98) the write pointer wraparound is managed, if necerssary. Some information in buffer 1 may have already been read, others not. Since it is not desirable for the write pointer to refer to an unread message ,special care has to be taken. In lines 66 to 70 , this case is handled where the read pointer does not refer to the first element in the array (the information stored there has been read). The byte is then read into the buffer from Port A's receive holding register. We then set the write pointer to refer to the first element in the buffer. What if the write pointer points to the buffers last element and the read pointer points to the first ? This special case is handled in lines 72 to 74. Although the last entry in the array is not being used, we cannot store information there because the write pointer cannot be changed to refer to an unused element. (Remember the first byte of information has not been read yet.) Processing is ten returned to the main rouine. The byte present in Port A's receive holding register will not be read. Bytes not being read from Port A's receive holding register (RHRA) will cause the Port A's FIFO to become full.When this condition occurs the RTA output of Port A will be asserted which cause the transmitting device to stop transmission of information (handshaking).

Lines 79 to 82 are executed if the write pointer refers to an element that is not at the end of the buffer and the buffer is full. We can determine this because no unused elements are between the write pointer and the read pointer

Lines 83 to 87 of the *if-else-if* statement are executed when a byte of information is to be stored in the middle of the buffer after which processing is returned to the main routine.

A Port A request transmission will test the condition of buffer 2 before the transmission commences. Lines 93 to 97 test buffer 2 for buffer empty conditions. The buffer is empty when the read and write pointers of the buffer is equal to each other. Processing is passed to the main routine under buffer empty conditions thus there is no information in the buffer that needs to be transmitted.

If the read pointer is not equal to the write pointer , we pass the information byte to Port A's transmit holding register (THRA). At this stage Port A transmitter would be enabled . Transmission of the information byte is then done under Port A control.Finally, in lines 101 to 105, we advance the read pointer to the next information byte. If advancing the pointer causes it to refer to an element beyond the end of the buffer , we reset the pointer, wrapping the read pointer around to the front of the buffer.

5. Problems encountered

5.1. Software Debugging

Debugging of the code was tedious due to the fact that test bits had to be written into the code for debugging purposes. No full screen debugger was available at the time which would have speeded up the debugging process.

5.2. Power Rail

The Console and DBIC cards are all connected to the power rail with the aid of a backplane. The initial power track thickness as well as decoupling was not sufficient. This resulted in the power rail being noisy and a drop in voltage to certain cards. This was corrected by providing thicker power rails and proper decoupling.

5.3. Serial Bus

Initial testing of the serial bus was done with the Console card and one DBIC card. The test results were satisfactory. The connection of additional DBIC cards to the serial bus degraded the information that was present on the bus. This resulted in intermittent failure of the bus. To overcome this problem driver transistors were provided in the transmit directions of the bus. Further tests proved satisfactory.

6. Conclusion

The Data Buffer and Interface Converter conforms to all the specifications required in the objective at the beginning of this thesis. Future improvements to the DBIC to reduce costs and increase efficiency can be seen in the following chapter.

7. Future Suggestions

The PPI (8225A chip) used on the DBIC card for the implementation of the parallel port can be replaced by a latch such as the 74HC573 and the spare pins of the CPU used for the necessary control signals. The latch being physically smaller than the PPI (8255A) would reduce board size thereby reducing cost.

The keys used on the Console card are connected to pins of Port 1 of the CPU. These keys can be connected to the external interrupt pins of the CPU which would mean that the CPU can go into the idle mode during no key activation periods, thus bringing about a reduction in power.

8. Bibliography

Intel Corporation. 1990 **Memory Databook**

Intel Corporation. 1990 **Peripherals Databook**

Intel Corporation. 1989 **Microcomputer Programmable Logic Handbook**

Intel Corporation. 1986 **Microcontoller Handbook**

United Microelectronics Corporation. 1986/7 **Microcomponents and Memory Databook**

Lattice Semiconductor Corporation. 1990 **Gal Databook**

National Semiconductor Corporation. 1984 **Logic Databook**

National Semiconductor Corporation. 1988 **Data Communications Handbook**

Robert C. Hutchinson and Steven B. Just. 1988 **Programming Using the C Language**

Maxim Integrated Products. 1989

Product Specification - RS232

Line Drivers

Kernighan B.W. and Ritchie D.M. 1988

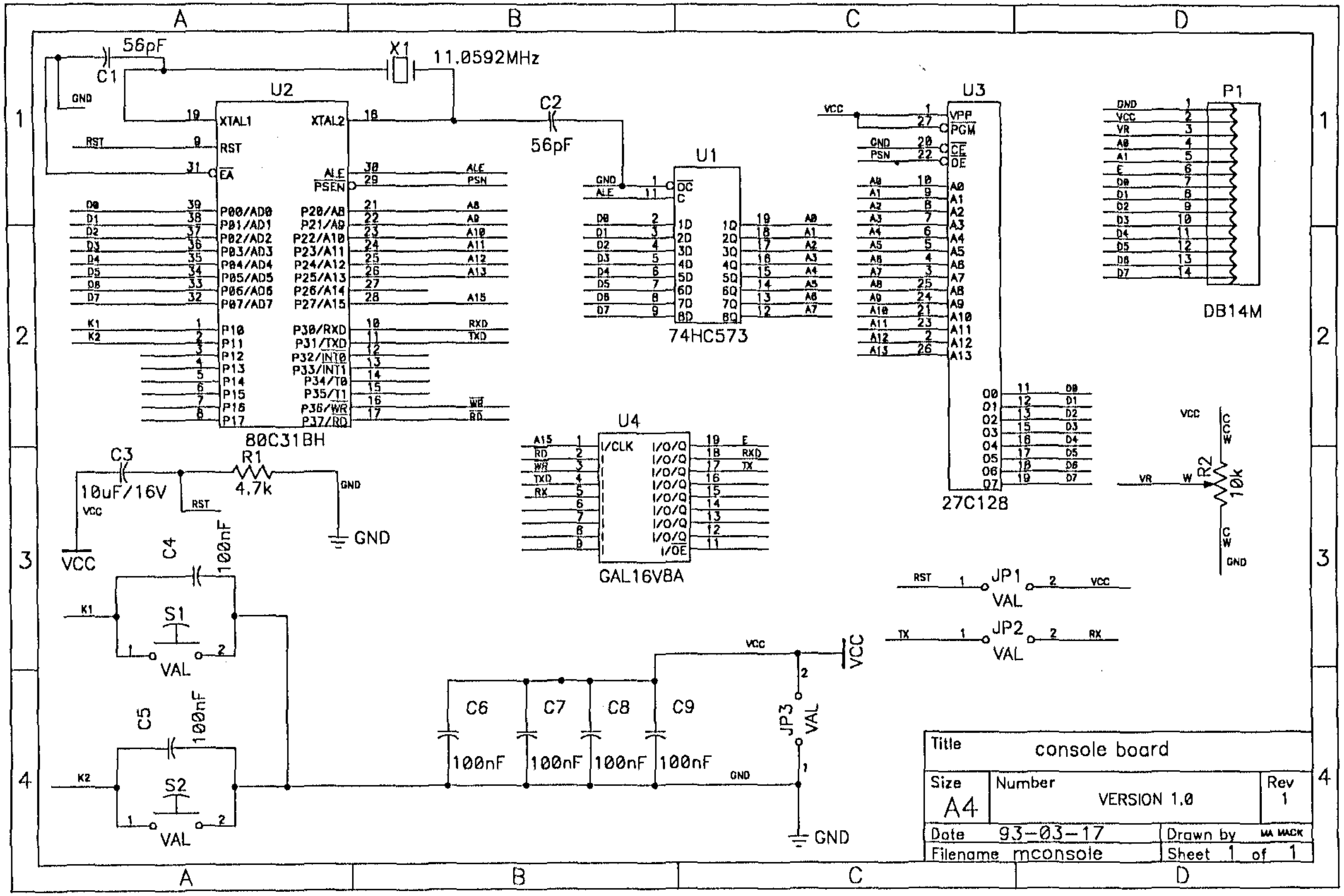
The C Programming Language

2nd Edition

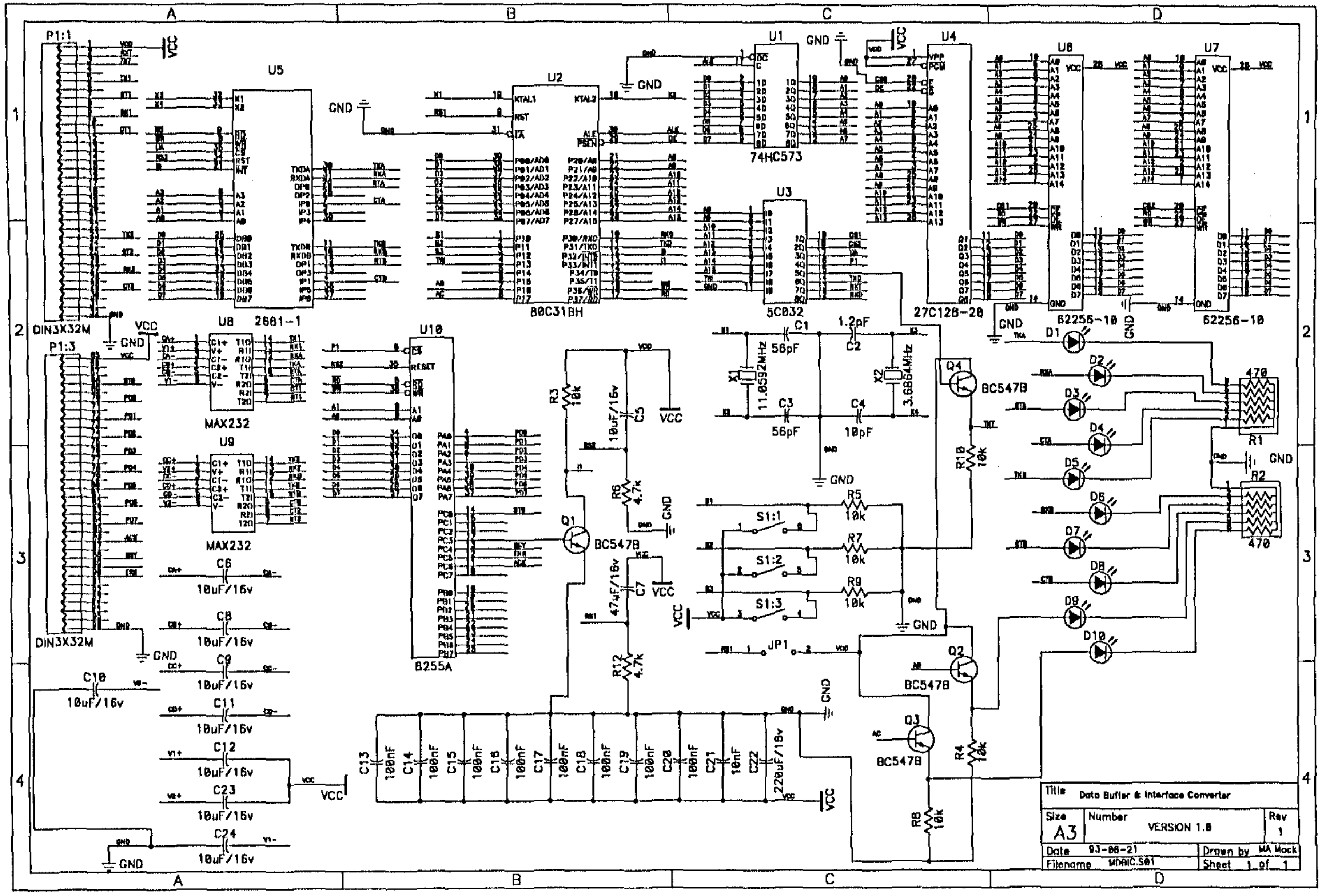
APPENDIX A

Hardware Version 1.0

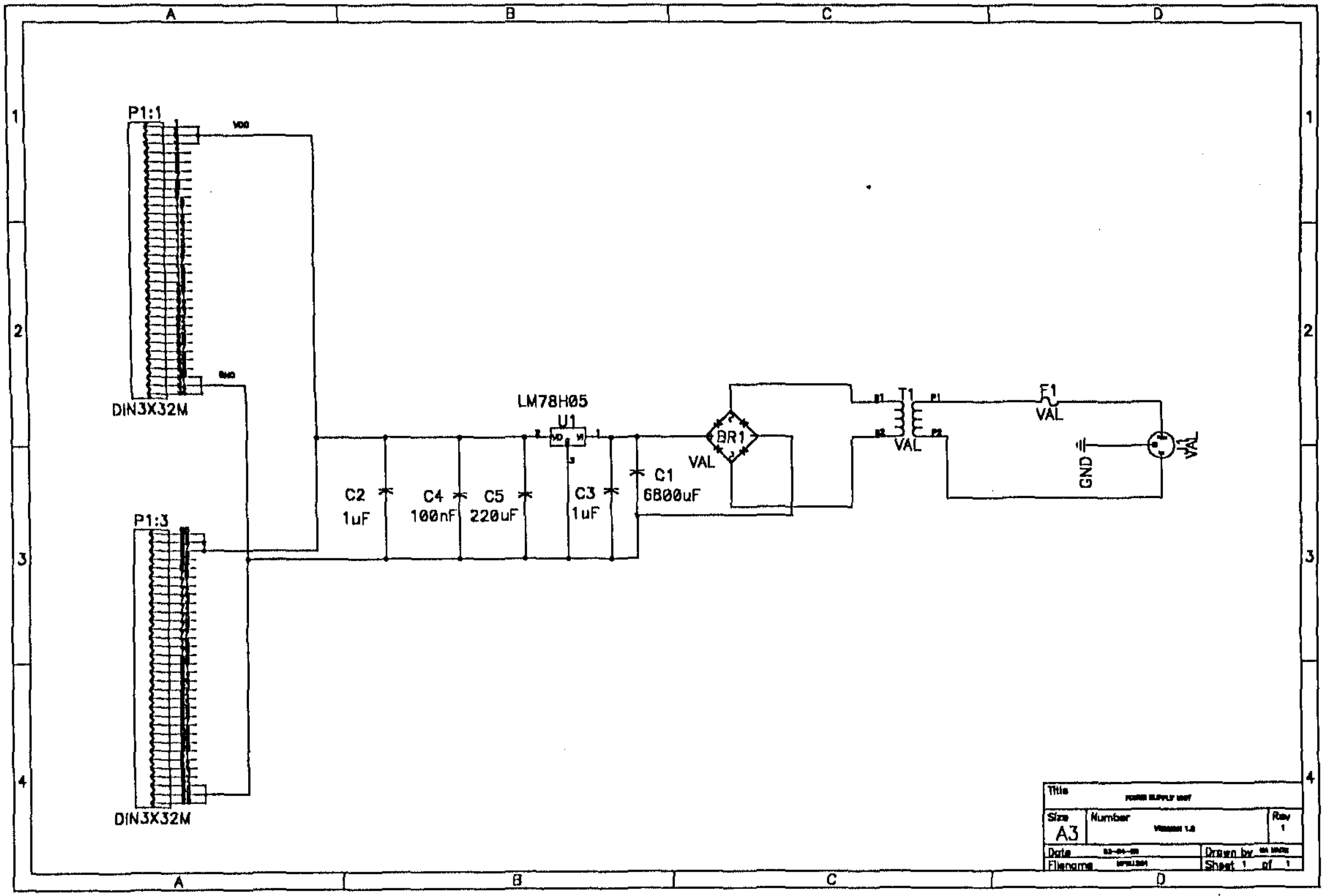
Schematic Diagrams



C1 56pF
X1 11.0592MHz
C2 56pF
U2 80C31BH
XTAL1 XTAL2 RST EA ALE PSEN
P00/AD0 P01/AD1 P02/AD2 P03/AD3 P04/AD4 P05/AD5 P06/AD6 P07/AD7
P20/A8 P21/A9 P22/A10 P23/A11 P24/A12 P25/A13 P26/A14 P27/A15
P10 P11 P12 P13 P14 P15 P16 P17
P30/RXD P31/TXD P32/INT0 P33/INT1 P34/T0 P35/T1 P36/WR P37/RD
U1 74HC573
U3 27C128
VPP PGM CE OE
P1 DB14M
U4 GAL16V8A
I/CLK I/O/Q I/OE
C3 10uF/16V
R1 4.7k
R2 10k
C4 100nF
C5 100nF
S1 VAL
S2 VAL
C6 100nF
C7 100nF
C8 100nF
C9 100nF
JP1 VAL
JP2 VAL
JP3 VAL
VCC
GND
Title console board
Size A4
Number VERSION 1.0
Rev 1
Date 93-03-17
Drawn by MA MACK
Filename mconsole
Sheet 1 of 1



P1:1
DIN3X32M
P1:3
DIN3X32M
U5
2681-1
U2
80C31BH
U1
74HC573
U3
5C032
U4
27C128-20
U6
62256-10
U7
62256-10
U8
MAX232
U9
MAX232
U10
8255A
X1
11.0592MHz
X2
3.6864MHz
C1
56pF
C2
1.2pF
C3
56pF
C4
10pF
C5
10uF/16v
C6
10uF/16v
C7
47uF/16v
C8
10uF/16v
C9
10uF/16v
C10
10uF/16v
C11
10uF/16v
C12
10uF/16v
C23
10uF/16v
C24
10uF/16v
C13 100nF
C14 100nF
C15 100nF
C16 100nF
C17 100nF
C18 100nF
C19 100nF
C20 100nF
C21 10nF
C22 220uF/16v
R1 470
R2 470
R3 10k
R4 10k
R5 10k
R6 4.7k
R7 10k
R8 10k
R9 10k
R10 10k
R12 4.7k
Q1 BC547B
Q2 BC547B
Q3 BC547B
Q4 BC547B
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
S1:1
S1:2
S1:3
JP1
VCC
GND
Title Data Buffer & Interface Converter
Size A3
Number VERSION 1.0
Rev 1
Date 93-06-21
Drawn by MA Mack
Filename MDBIC.S01
Sheet 1 of 1

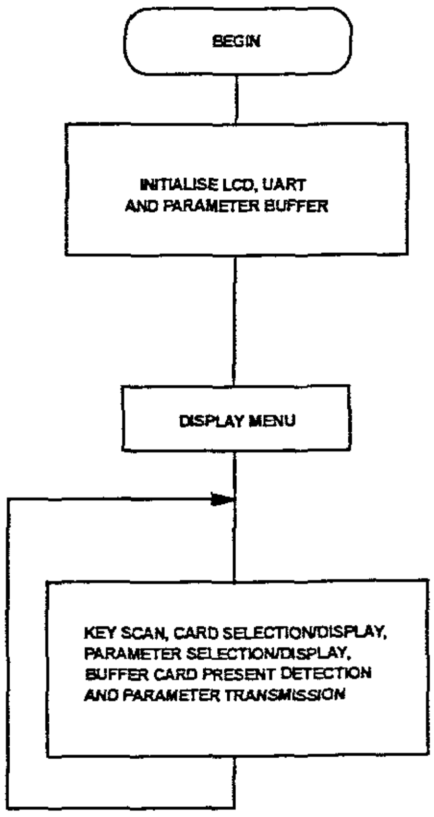

A
B
C
D
1
2
3
4
P1:1
VDD
GND
DIN3X32M
P1:3
DIN3X32M
LM78H05
U1
C2
1uF
C4
100nF
C5
220uF
C3
1uF
C1
6800uF
BR1
VAL
T1
VAL
F1
VAL
GND
J1
VAL
Title
POWER SUPPLY UNIT
Size
A3
Number
Version 1.0
Rev
1
Date
Drawn by
Filename
Sheet 1 of 1

APPENDIX B

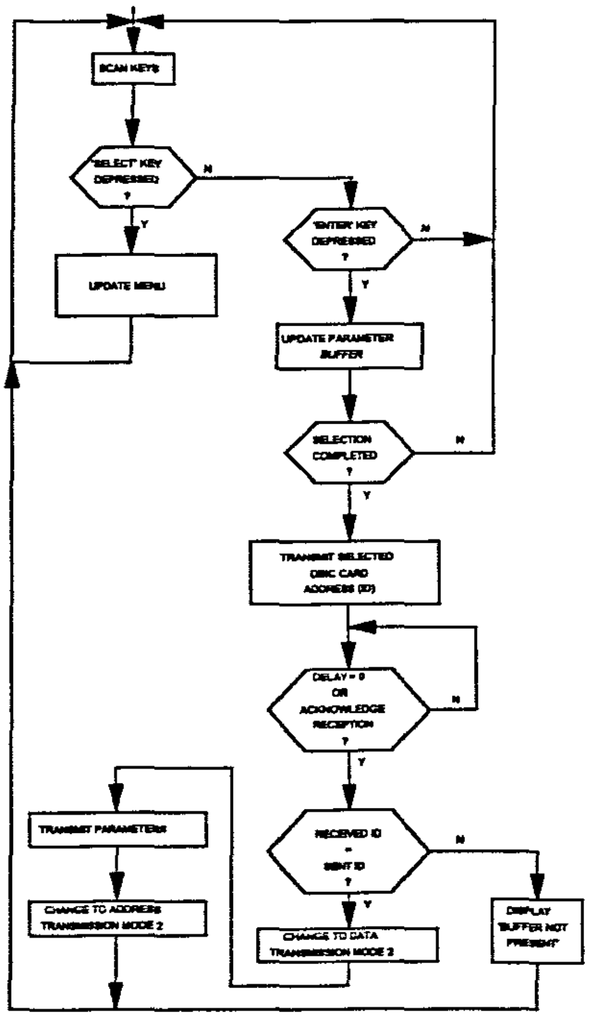
Software Version 1.0

Program Flowcharts

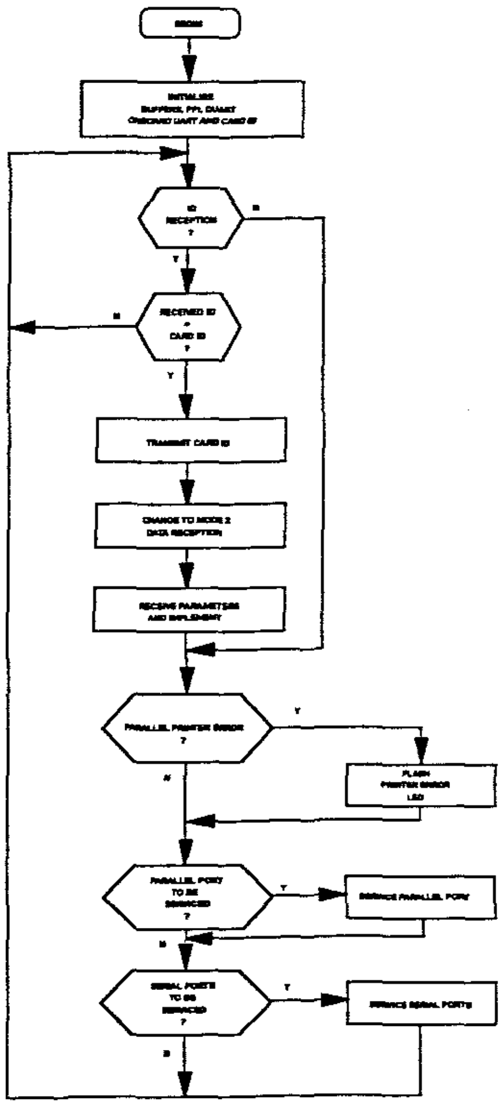
CONSOLE CARD MAIN MENU INTERFACE TO USER


BEGIN
INITIALISE LCD, UART AND PARAMETER BUFFER
DISPLAY MENU
KEY SCAN, CARD SELECTION/DISPLAY, PARAMETER SELECTION/DISPLAY, BUFFER CARD PRESENT DETECTION AND PARAMETER TRANSMISSION

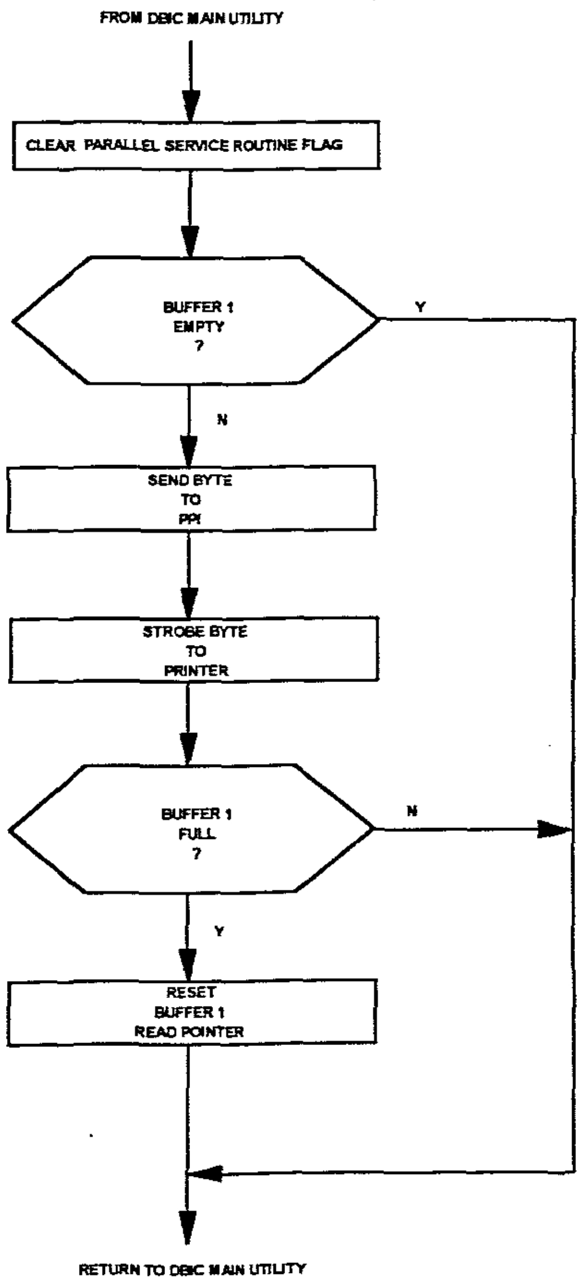
KEY SCAN, CARD SELECTION/DISPLAY, PARAMETER *SELECTION/DISPLAY, BUFFER CARD PRESENT DETECTION AND* PARAMETER TRANSMISSION

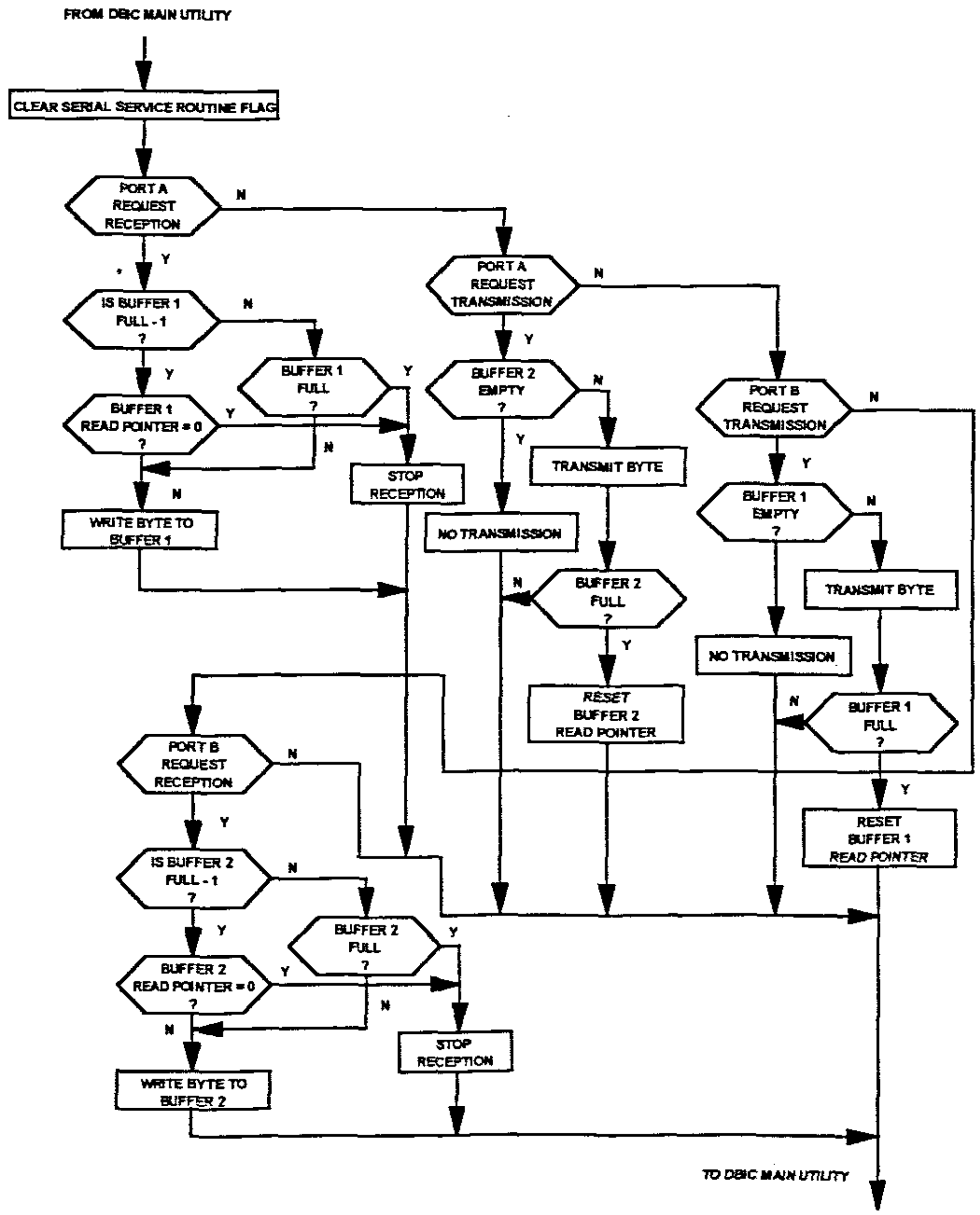

SCAN KEYS
'SELECT' KEY DEPRESSED ?
N
Y
UPDATE MENU
'ENTER' KEY DEPRESSED ?
N
Y
UPDATE PARAMETER BUFFER
SELECTION COMPLETED ?
N
Y
TRANSMIT SELECTED DBIC CARD ADDRESS (ID)
DELAY = 0 OR ACKNOWLEDGE RECEPTION ?
N
Y
RECEIVED ID = SENT ID ?
N
Y
CHANGE TO DATA TRANSMISSION MODE 2
DISPLAY 'BUFFER NOT PRESENT'
TRANSMIT PARAMETERS
CHANGE TO ADDRESS TRANSMISSION MODE 2

DATA BUFFER AND INTERFACE CONVERTER CARD UTILITY



PARALLEL SERVICE ROUTINE UTILITY




FROM DBIC MAIN UTILITY
CLEAR SERIAL SERVICE ROUTINE FLAG
PORT A REQUEST RECEPTION
IS BUFFER 1 FULL - 1 ?
BUFFER 1 FULL ?
BUFFER 1 READ POINTER = 0 ?
WRITE BYTE TO BUFFER 1
STOP RECEPTION
PORT A REQUEST TRANSMISSION
BUFFER 2 EMPTY ?
TRANSMIT BYTE
NO TRANSMISSION
BUFFER 2 FULL ?
RESET BUFFER 2 READ POINTER
PORT B REQUEST TRANSMISSION
BUFFER 1 EMPTY ?
TRANSMIT BYTE
NO TRANSMISSION
BUFFER 1 FULL ?
RESET BUFFER 1 READ POINTER
PORT B REQUEST RECEPTION
IS BUFFER 2 FULL - 1 ?
BUFFER 2 FULL ?
BUFFER 2 READ POINTER = 0 ?
WRITE BYTE TO BUFFER 2
STOP RECEPTION
TO DBIC MAIN UTILITY
Y
N

APPENDIX C

EPLD Programming V1.0

Decoder Equations

```
 module console
 title 'console card mem decode
 Mark A. Mack Telkom SA RDC Cape Town'
 console device 'p16v8as';
'*********************************************************'
 PIN DECLARATIONS
'*********************************************************'
 A15,RD,WR,TX,RX          PIN  1,2,3,4,5;
 TXD,RXD,E                     PIN   17,18,19;
'*********************************************************'
 EQUATIONS
'*********************************************************'
 equations
 TXD  = !TX;
 RXD  = !RX;
 E    = A15 & RD & !WR # A15 & !RD & WR;

END Console
```

```
 module dbic
 title 'dbic card mem decode
 Mark A. Mack Telkom SA RDC Cape Town'
 dbic  device  'E0320';
'*******************************************************'
'PIN DECLARATIONS'
'*******************************************************'
 A8,A9,A10,A11,A12,A13,A14,A15,TR1,RXT,TXD
 PIN
 1,2,3,4,5,6,7,8,9,11,13,14;
 RXD,TXT,P1,UA,CS2,CS1  PIN 12,15,16,17,18,19;
 H,L,X = 1,0,.X.;
 Address = [A15,A14,A13,A12, A11,A10,A9,A8,
 X,X,X,X,X,X];
'*******************************************************'
 'EQUATIONS'
'*******************************************************'
 equations
 TR1    = !TXD;
 TXT.OE = TR1;
 !CS1   = (Address >= ^h0000) & (Address <= ^h7FFF);
 !CS2   = (Address >= ^h8000) & (Address <= ^hFDFF);
 !UA    = (Address >= ^hFE00) & (Address <= ^hFEFF);
 !P1    = (Address >= ^hFF00) & (Address <= ^hFFFF);
 RXD    = !RXT;


 END dbic
```

APPENDIX D

Software Version 1.0

Source Code Listing

CONSOLE SOURCE CODE

```
C51 COMPILER V3.40,  CONSOLE


DOS C51 COMPILER V3.40, COMPILATION OF MODULE CONSOLE
OBJECT MODULE PLACED IN CONSOLE.OBJ
COMPILER INVOKED BY: C:\ZIP\C51.EXE CONSOLE.C CODE


stmt level    source

   1            #pragma PAGELENGTH (66)
   2            #pragma SYMBOLS
   3            #include  <reg51.h>
   4            #include  <stdio.h>
   5            #define       ARRAY_SIZE      10
   6            #define       LCD_COMMAND 1
   7            #define       LCD_DATA    0
   8            #define       RS232       2
   9            #define       LINE_1      0x80
  10            #define       LINE_2      0xC0
  11            #define       CLEAR       0x01
  12            unsigned int   totchar = 0;
  13            unsigned int   key = 00, idt;
  14            unsigned int   kboard , ab;
  15            unsigned int   no;
  16            unsigned char  io_stream;
  17            unsigned char  id,spb,act,idtemp;
  18            unsigned char  inter[12] ,write ;
  19            unsigned  char    lcd_addr;
  20            long  delay;
  21
  22
  23
  24
  25          /*******************************************/
  26          /* KEY SCAN ROUTINE                        */
  27          /*******************************************/
  28
  29
  30            unsigned int   getkey ()
  31
  32            {
  33   1               if (!P10 )
  34   1            { while ( !P10  );        /* wait for release */
  35   2       for (delay=0;delay<=1000;delay++);
  36   2       while (!P10);
  37   2                   act=1;
  38   2
  39   2        if ( key <= 7 )
  40   2                 {
  41   3                  return ( ++key );
  42   3                     }
  43   2                   if ( key == 8)
  44   2                  {
  45   3         key = 0;
  46   3                     return ( ++key );
  47   3                  }
  48   2              if (key == 13 )
  49   2                 {
  50   3         key = 9;
  51   3                     return ( ++key);
```

```
3              }
2        if ( key == 19 )
2              {
3  return ( 21 );
3              }
2            if ( key > 9 && key <= 24 )
2             {
3  return (++key);
3             }
2        if (key == 25 )
2              {
3  return (20);
3              }
2       if ( key > 25 && key <= 40 )
2            {
3  return (++key);
3            }
2       if ( key == 41 )
2            {
3  return (30);
3
3            }
2      return (0);
2
2
2  }
1   if ( !P11 )
1   {
2      if(act)
2      {
3       while (!P11);
3  for (delay+0;delay<=1200;delay++);
3  while (!P11);
3
3   if ( key <= 8  )
3         {
4            return (10);
4        }
3
3      if ( key == 12 )
3        {
4 ab = 1;
4            ++write;
4           return (20);
4         }
3
3      if ( key  > 10 && key <= 13 )
3         {
4 key = 19;
4           ++write;
4            return (++key);
4         }
3
3      if ( key == 19 )
3         {
4 ++ write;
4           return ( 30);
4         }
3
3         if (key > 20 && key <=25 )
3         {
4 key = 29;
4            ++ write;
4            return (++key);
4                 }
3
3              if ( key >= 30 && key <=41)
3                 {
4       if ( ab == 1 )
4                      {
5                       write = 4;
5                       return (45);
5                      }
4                    return (50);
4                 }
3            return (0);
3
3            }
2         }
1            return (0);
1       }

      /*************************************/
      /* LCD INITIALISATION ROUTINE        */
      /*************************************/

          init_lcd( )
        {
1         lcd_addr= 0x80;
1         io_stream= LCD_COMMAND;
1         putchar ( 0x01 );
1         putchar ( 0x02 );
1         putchar ( 0x38 );
1         putchar ( 0x0C );
1         io_stream= LCD_DATA;
1
1       }

      /*****************************************/
      /* UART INITIALISATION ROUTINE           */
      /*****************************************/

                 init_sio0  ( )
        {
1         RI   = 0;
1         EA   = 0;
1         TMOD = 0x20;               /* SET TIMER 1 & 2 MODE */
1         TH1  = 0xFD;               /* SET TIMER 1 PRELOAD  */
1                           /* SET SMOD=0        */
1         SCON = 0xF8;
1         TR1  = 1;
1                                 /* SET SERIAL PORT MODE */
1
1
1       }

      /***************************************/
      /*   LCD CHARACTER DISPLAY ROUTINE     */
      /***************************************/

          set_lcd (char out )
        {
1         io_stream= LCD_COMMAND;
1         putchar ( out );
1         io_stream= LCD_DATA;
1       }

      /***************************************/
```

```
          /*  MAIN ROUTINE                  */
          /****************************************/

              main    ( )
            {
  1           RI = 0;
  1           TI = 0;
  1           write = 11;
  1           inter[write] = 0x00;
  1           write = 0;
  1           init_lcd( );
  1           set_lcd ( CLEAR );
  1           printf  ( "   Select Unit " );
  1           set_lcd ( LINE_2 );
  1           printf  ( "     Unit ?   " );
  1           init_sio0 ();
  1           while   ( 1 )
  1             {
  2
  2           no=0;
  2
  2
  2       /****************************************/
  2       /*   CARD SELCTION,PARAMETER SELECTION, */
  2       /*   BUFFER CARD PRESENT DETECTION AND  */
  2       /*   PARAMETER TRANSMISSION ROUTINE     */
  2       /****************************************/
  2
  2          kboard = getkey ();
  2          switch ( kboard )
  2            {
  3             case 1:
  3             case 2:
  3             case 3:
  3             case 4:
  3             case 5:
  3             case 6:
  3             case 7:
  3             case 8:
  3                    set_lcd ( CLEAR );
  3                    set_lcd ( LINE_1 );
  3                    printf  ( "   Select Unit ");
  3                    set_lcd ( LINE_2 );
  3                    printf ( "     Unit %d " , kboard );
  3                    id = kboard | 0x30 ;
  3       idt = kboard;
  3                    for (write = 0; write <= 10; ++write)
  3                        {
  4                          inter[write] = 0x00;
  4                        }
  3                    write = 3;
  3                    inter[write] = 0x3F;
  3                    write = 6;
  3                    inter[write] = 0x3F;
  3                    write = 0;
  3                    key = kboard;
  3                    id = kboard | 0x30 ;
  3                    break;
  3            case 10:
  3                    set_lcd ( CLEAR );
  3                    set_lcd ( LINE_1 );
  3                    printf ( " Configuration ?" );
  3                    set_lcd ( LINE_2 );
  3                    printf ( "Port A <> Port B " );
```

```
                key = 12;
                inter[write] = 0x33;     /* IMR */
                write = 7;
                inter[write] = 0x05;     /* CRA */
                write = 8;
                inter[write] = 0x05;    /*  CRB */
                write = 9;
                inter[write] = 0x80;
                write = 0;
                break;
        case 13:
                set_lcd ( LINE_2 );
                printf ( "Port A -> Port C ");
                key = kboard;
                write = 9;
                inter [write] = 0xAE;     /* PARA CONTROL */
                write = 10 ;
                inter [write] = 0x0D;     /* PARA CONTROL - bit set
*/
                write =  7;
                inter [write] = 0x05;
                write ++;
                inter [write] = 0x0A;
                write = 0;
                inter [write] = 0x03;
                write = 0;
                break;
        case 20:
                set_lcd ( LINE_1 );
                if (spb)
                    {
                       printf (" Select Port B  ");
                    }
                else
                    {
                      printf ( " Select Port A  ");
                    }
                set_lcd ( LINE_2 );
                printf ( "Baud rate - 9600");
                key = 19;
                inter[write] =0xBB;
                break;
        case 21:
                set_lcd ( LINE_2 );
                printf ( "Baud rate - 4800");
                key =kboard;
                inter[write] = 0x99;
                break;
        case 22:
                set_lcd ( LINE_2 );
                printf ( "Baud rate - 2400");
                key = kboard;
                inter[write] = 0x88;
                break;
        case 23:
                set_lcd (LINE_2 );
                printf ( "Baud rate - 1200");
                key = kboard;
                inter[write] = 0x66;
                break;
        case 24:
                set_lcd ( LINE_2 );
                printf ("Baud rate -  600");
                key = kboard;
```

```
3                inter[write] = 0x55;
3                break;
3         case 25:
3                set_lcd (LINE_2 );
3                printf ("Baud rate -  300");
3                key = kboard;
3                inter [write] = 0x44;
3                break;
3         case 30:
3                set_lcd ( LINE_2 );
3                        printf ("Word cnf - 8,n,1");
3                key =  kboard;
3                inter[write] = 0x93;
3                break;
3         case 31:
3                set_lcd (LINE_2 );
3                printf ("Word cnf - 8,e,1");
3                key = kboard ;
3                inter[write] = 0x83;
3                break;
3         case 32:
3                set_lcd (LINE_2 );
3                printf ("Word cnf - 8,o,1");
3                key = kboard;
3                inter[write] = 0x87;
3                break;
3         case 33:
3                set_lcd ( LINE_2 );
3                printf ("Word cnf - 8,n,2");
3                inter[write] = 0x93;
3                key =  kboard;
3                break;
3                case 34:
3                set_lcd (LINE_2 );
3                printf ("Word cnf - 8,e,2");
3                inter[write] = 0x83;
3                key = kboard ;
3                break;
3         case 35:
3                set_lcd ( LINE_2 );
3                printf ("Word cnf - 8,o,2");
3                inter [write] = 0x87;
3                key =  kboard;
3                break;
3         case 36:
3                set_lcd (LINE_2 );
3                printf ("Word cnf - 7,n,1");
3                inter[write] = 0x92;
3                key = kboard ;
3                break;
3         case 37:
3                set_lcd ( LINE_2 );
3                printf ("Word cnf - 7,e,1");
3                inter[write] = 0x82;
3                key =  kboard;
3                break;
3         case 38:
3                set_lcd (LINE_2 );
3                printf ("Word cnf - 7,o,1");
3                inter[write] = 0x86;
3                key = kboard ;
3                break;
3         case 39:
3                set_lcd (LINE_2 );
```

```
3             printf ("Word cnf - 7,n,2");
3             inter[write] = 0x92;
3             key = kboard ;
3             break;
3     case 40:
3             set_lcd (LINE_2 );
3             printf ("Word cnf - 7,e,2");
3             inter[write] = 0x82;
3             key = kboard ;
3             break;
3     case 41:
3             set_lcd (LINE_2 );
3             printf ("Word cnf - 7,o,2");
3             inter[write] = 0x86;
3             key = kboard ;
3             break;
3     case 45:
3             set_lcd (LINE_1 );
3             printf ( " Select Port B  " );
3             set_lcd ( LINE_2 );
3             printf  ("Baud rate - 9600");
3             key = 19;
3             ab = 0 ;
3             spb = 1;
3             inter [write] = 0xBB;
3             break;
3     case 50:
3             spb = 0;
3             write = 0;
3             key = 0;
3             TB8 = 1;
3             SBUF = id;    /* ID transmission */
3             while (!TI);
3             TI = 0;
3             RI = 0;
3
3             delay=0;
3
3             while ((delay<1000)&&(!RI))    /* ID acknowledged */
3
3                 {
4                   delay++;
4                   set_lcd (CLEAR);
4                 };
3             idtemp = SBUF;
3
3                if (id == idtemp)
3                        {
4                         SM2 = 0; /* uart mode change */
4                         TB8 = 0;
4                          RI = 0;
4
4   /* parameter transmission to DBIC card */
4                         for (write = 0; write <= 12; ++write)
4                            {
5                              TB8 = 0;
5                       for (delay=0;delay<=10;delay++);
5                              SBUF = inter[write];
5                              while (!TI);
5                              TI = 0;
5                             }
4                       }
3                 else    /* DBIC not present detection */
3                       {
```

```

                    set_lcd (CLEAR);
                    set_lcd (LINE_1);
                    printf ("     Buffer");
                    set_lcd (LINE_2);
                    printf ("not present - %d",idt);
                    for (delay=0;delay <=10000;delay++);
                 no=1;
                  }

          SM2 = 1;  /* change to address MODE 2 */
          TB8 = 1;
          set_lcd (CLEAR);
          set_lcd (LINE_1);
          printf ("   Select Unit");
          set_lcd (LINE_2);
          printf ("     Unit *");
          kboard = 0;
          key = 0;
          write = 0;
          ab = 0;
          act=0;
          break;

           }

  }
}

```

ASSEMBLY LISTING OF GENERATED OBJECT CODE

```
             ; FUNCTION getkey (BEGIN)
                                                   ; SOURCE LINE # 30
                                                   ; SOURCE LINE # 32
                                                   ; SOURCE LINE # 33
0000 309003          JNB      P10,$ + 6H
0003 020000    R     LJMP     ?C0001
                                                   ; SOURCE LINE # 34
0006         ?C0002:
0006 3090FD          JNB      P10,?C0002
0009         ?C0003:
                                                   ; SOURCE LINE # 35
0009 750000    R     MOV      delay+03H,#00H
000C 750000    R     MOV      delay+02H,#00H
000F 750000    R     MOV      delay+01H,#00H
0012 750000    R     MOV      delay,#00H
0015         ?C0004:
0015 7FE8            MOV      R7,#0E8H
0017 7E03            MOV      R6,#03H
0019 7D00            MOV      R5,#00H
001B 7C00            MOV      R4,#00H
001D 120000    E     LCALL    ?C_LPUSH
0020 AF00      R     MOV      R7,delay+03H
0022 AE00      R     MOV      R6,delay+02H
0024 AD00      R     MOV      R5,delay+01H
0026 AC00      R     MOV      R4,delay
0028 120000    E     LCALL    ?C_SLCMP
002B 6002            JZ       $ + 4H
002D 5020            JNC      ?C0007
002F AF00      R     MOV      R7,delay+03H
0031 AE00      R     MOV      R6,delay+02H
0033 AD00      R     MOV      R5,delay+01H
0035 AC00      R     MOV      R4,delay
0037 120000    E     LCALL    ?C_LPUSH
003A 7F01            MOV      R7,#01H
003C 7E00            MOV      R6,#00H
003E 7D00            MOV      R5,#00H
0040 7C00            MOV      R4,#00H
0042 120000    E     LCALL    ?C_LADD
0045 8F00      R     MOV      delay+03H,R7
0047 8E00      R     MOV      delay+02H,R6
0049 8D00      R     MOV      delay+01H,R5
004B 8C00      R     MOV      delay,R4
004D 80C6            SJMP     ?C0004
004F         ?C0007:
                                                   ; SOURCE LINE # 36
004F 3090FD          JNB      P10,?C0007
0052         ?C0008:
                                                   ; SOURCE LINE # 37
0052 750001    R     MOV      act,#01H
                                                   ; SOURCE LINE # 39
0055 D3              SETB     C
0056 E500      R     MOV      A,key+01H
0058 9407            SUBB     A,#07H
005A E500      R     MOV      A,key
005C 9400            SUBB     A,#00H
005E 500C            JNC      ?C0009
                                                   ; SOURCE LINE # 40
                                                   ; SOURCE LINE # 41
0060 0500      R     INC      key+01H
```

```
0062 E500      R      MOV     A,key+01H
0064 7002             JNZ     ?C0093
0066 0500      R      INC     key
0068           ?C0093:
0068 FF               MOV     R7,A
0069 AE00      R      MOV     R6,key
006B 22               RET
                                          ; SOURCE LINE # 42
006C           ?C0009:
                                          ; SOURCE LINE # 43
006C E500      R      MOV     A,key+01H
006E 6408             XRL     A,#08H
0070 4500      R      ORL     A,key
0072 7010             JNZ     ?C0011
                                          ; SOURCE LINE # 44
                                          ; SOURCE LINE # 45
0074 F500      R      MOV     key,A
0076 F500      R      MOV     key+01H,A
                                          ; SOURCE LINE # 46
0078 0500      R      INC     key+01H
007A E500      R      MOV     A,key+01H
007C 7002             JNZ     ?C0094
007E 0500      R      INC     key
0080           ?C0094:
0080 FF               MOV     R7,A
0081 AE00      R      MOV     R6,key
0083 22               RET
                                          ; SOURCE LINE # 47
0084           ?C0011:
                                          ; SOURCE LINE # 48
0084 E500      R      MOV     A,key+01H
0086 640D             XRL     A,#0DH
0088 4500      R      ORL     A,key
008A 7011             JNZ     ?C0012
                                          ; SOURCE LINE # 49
                                          ; SOURCE LINE # 50
008C F500      R      MOV     key,A
008E 750009    R      MOV     key+01H,#09H
                                          ; SOURCE LINE # 51
0091 0500      R      INC     key+01H
0093 E500      R      MOV     A,key+01H
0095 7002             JNZ     ?C0095
0097 0500      R      INC     key
0099           ?C0095:
0099 FF               MOV     R7,A
009A AE00      R      MOV     R6,key
009C 22               RET
                                          ; SOURCE LINE # 52
009D           ?C0012:
                                          ; SOURCE LINE # 53
009D E500      R      MOV     A,key+01H
009F 6413             XRL     A,#013H
00A1 4500      R      ORL     A,key
00A3 7004             JNZ     ?C0013
                                          ; SOURCE LINE # 54
                                          ; SOURCE LINE # 55
00A5 FE               MOV     R6,A
00A6 7F15             MOV     R7,#015H
00A8 22               RET
                                          ; SOURCE LINE # 56
00A9           ?C0013:
                                          ; SOURCE LINE # 57
00A9 D3               SETB    C
00AA E500      R      MOV     A,key+01H
```

```
00AC 9409              SUBB    A,#09H
00AE E500      R       MOV     A,key
00B0 9400              SUBB    A,#00H
00B2 4017              JC      ?C0014
00B4 D3                SETB    C
00B5 E500      R       MOV     A,key+01H
00B7 9418              SUBB    A,#018H
00B9 E500      R       MOV     A,key
00BB 9400              SUBB    A,#00H
00BD 500C              JNC     ?C0014
                                                  ; SOURCE LINE # 58
                                                  ; SOURCE LINE # 59
00BF 0500      R       INC     key+01H
00C1 E500      R       MOV     A,key+01H
00C3 7002              JNZ     ?C0096
00C5 0500      R       INC     key
00C7           ?C0096:
00C7 FF                MOV     R7,A
00C8 AE00      R       MOV     R6,key
00CA 22                RET
                                                  ; SOURCE LINE # 60
00CB           ?C0014:
                                                  ; SOURCE LINE # 61
00CB E500      R       MOV     A,key+01H
00CD 6419              XRL     A,#019H
00CF 4500      R       ORL     A,key
00D1 7004              JNZ     ?C0015
                                                  ; SOURCE LINE # 62
                                                  ; SOURCE LINE # 63
00D3 FE                MOV     R6,A
00D4 7F14              MOV     R7,#014H
00D6 22                RET
                                                  ; SOURCE LINE # 64
00D7           ?C0015:
                                                  ; SOURCE LINE # 65
00D7 D3                SETB    C
00D8 E500      R       MOV     A,key+01H
00DA 9419              SUBB    A,#019H
00DC E500      R       MOV     A,key
00DE 9400              SUBB    A,#00H
00E0 4017              JC      ?C0016
00E2 D3                SETB    C
00E3 E500      R       MOV     A,key+01H
00E5 9428              SUBB    A,#028H
00E7 E500      R       MOV     A,key
00E9 9400              SUBB    A,#00H
00EB 500C              JNC     ?C0016
                                                  ; SOURCE LINE # 66
                                                  ; SOURCE LINE # 67
00ED 0500      R       INC     key+01H
00EF E500      R       MOV     A,key+01H
00F1 7002              JNZ     ?C0097
00F3 0500      R       INC     key
00F5           ?C0097:
00F5 FF                MOV     R7,A
00F6 AE00      R       MOV     R6,key
00F8 22                RET
                                                  ; SOURCE LINE # 68
00F9           ?C0016:
                                                  ; SOURCE LINE # 69
00F9 E500      R       MOV     A,key+01H
00FB 6429              XRL     A,#029H
00FD 4500      R       ORL     A,key
00FF 7004              JNZ     ?C0017
```

```
                                                  ; SOURCE LINE # 70
                                                  ; SOURCE LINE # 71
0101 FE              MOV     R6,A
0102 7F1E            MOV     R7,#01EH
0104 22              RET
                                                  ; SOURCE LINE # 73
0105          ?C0017:
                                                  ; SOURCE LINE # 74
0105 7E00            MOV     R6,#00H
0107 7F00            MOV     R7,#00H
0109 22              RET
                                                  ; SOURCE LINE # 77
010A          ?C0001:
                                                  ; SOURCE LINE # 78
010A 309103          JNB     P11,$ + 6H
010D 020000   R      LJMP    ?C0018
                                                  ; SOURCE LINE # 79
                                                  ; SOURCE LINE # 80
0110 E500     R      MOV     A,act
0112 7003            JNZ     $ + 5H
0114 020000   R    · LJMP    ?C0018
                                                  ; SOURCE LINE # 81
0117          ?C0020:
                                                  ; SOURCE LINE # 82
0117 3091FD          JNB     P11,?C0020
011A          ?C0021:
                                                  ; SOURCE LINE # 83
011A          ?C0022:
011A 7FB0            MOV     R7,#0B0H
011C 7E04            MOV     R6,#04H
011E 7D00            MOV     R5,#00H
0120 7C00            MOV     R4,#00H
0122 120000   E      LCALL   ?C_LPUSH
0125 AF00     R      MOV     R7,delay+03H
0127 AE00     R      MOV     R6,delay+02H
0129 AD00     R      MOV     R5,delay+01H
012B AC00     R      MOV     R4,delay
012D 120000   E      LCALL   ?C_SLCMP
0130 6002            JZ      $ + 4H
0132 5020            JNC     ?C0025
0134 AF00     R      MOV     R7,delay+03H
0136 AE00     R      MOV     R6,delay+02H
0138 AD00     R      MOV     R5,delay+01H
013A AC00     R      MOV     R4,delay
013C 120000   E      LCALL   ?C_LPUSH
013F 7F01            MOV     R7,#01H
0141 7E00            MOV     R6,#00H
0143 7D00            MOV     R5,#00H
0145 7C00            MOV     R4,#00H
0147 120000   E      LCALL   ?C_LADD
014A 8F00     R      MOV     delay+03H,R7
014C 8E00     R      MOV     delay+02H,R6
014E 8D00     R      MOV     delay+01H,R5
0150 8C00     R      MOV     delay,R4
0152 80C6            SJMP    ?C0022
0154          ?C0025:
                                                  ; SOURCE LINE # 84
0154 3091FD          JNB     P11,?C0025
0157          ?C0026:
                                                  ; SOURCE LINE # 86
0157 D3              SETB    C
0158 E500     R      MOV     A,key+01H
015A 9408            SUBB    A,#08H
015C E500     R      MOV     A,key
```

```
015E 9400             SUBB    A,#00H
0160 5005             JNC     ?C0027
                                                 ; SOURCE LINE # 87
                                                 ; SOURCE LINE # 88
0162 7E00             MOV     R6,#00H
0164 7F0A             MOV     R7,#0AH
0166 22               RET
                                                 ; SOURCE LINE # 89
0167          ?C0027:
                                                 ; SOURCE LINE # 91
0167 E500      R      MOV     A,key+01H
0169 640C             XRL     A,#0CH
016B 4500      R      ORL     A,key
016D 700B             JNZ     ?C0028
                                                 ; SOURCE LINE # 92
                                                 ; SOURCE LINE # 93
016F F500      R      MOV     ab,A
0171 750001    R      MOV     ab+01H,#01H
                                                 ; SOURCE LINE # 94
0174 0500      R      INC     write
                                                 ; SOURCE LINE # 95
0176 FE               MOV     R6,A
0177 7F14             MOV     R7,#014H
0179 22               RET
                                                 ; SOURCE LINE # 96
017A          ?C0028:
                                                 ; SOURCE LINE # 98
017A D3               SETB    C
017B E500      R      MOV     A,key+01H
017D 940A             SUBB    A,#0AH
017F E500      R      MOV     A,key
0181 9400             SUBB    A,#00H
0183 401F             JC      ?C0029
0185 D3               SETB    C
0186 E500      R      MOV     A,key+01H
0188 940D             SUBB    A,#0DH
018A E500      R      MOV     A,key
018C 9400             SUBB    A,#00H
018E 5014             JNC     ?C0029
                                                 ; SOURCE LINE # 99
                                                 ; SOURCE LINE # 100
0190 750000    R      MOV     key,#00H
0193 750013    R      MOV     key+01H,#013H
                                                 ; SOURCE LINE # 101
0196 0500      R      INC     write
                                                 ; SOURCE LINE # 102
0198 0500      R      INC     key+01H
019A E500      R      MOV     A,key+01H
019C 7002             JNZ     ?C0098
019E 0500      R      INC     key
01A0          ?C0098:
01A0 FF               MOV     R7,A
01A1 AE00      R      MOV     R6,key
01A3 22               RET
                                                 ; SOURCE LINE # 103
01A4          ?C0029:
                                                 ; SOURCE LINE # 105
01A4 E500      R      MOV     A,key+01H
01A6 6413             XRL     A,#013H
01A8 4500      R      ORL     A,key
01AA 7006             JNZ     ?C0030
                                                 ; SOURCE LINE # 106
                                                 ; SOURCE LINE # 107
01AC 0500      R      INC     write
```

```
                                                   ; SOURCE LINE # 108
01AE FE             MOV     R6,A
01AF 7F1E           MOV     R7,#01EH
01B1 22             RET
                                                   ; SOURCE LINE # 109
01B2          ?C0030:
                                                   ; SOURCE LINE # 111
01B2 D3             SETB    C
01B3 E500     R     MOV     A,key+01H
01B5 9414           SUBB    A,#014H
01B7 E500     R     MOV     A,key
01B9 9400           SUBB    A,#00H
01BB 401F           JC      ?C0031
01BD D3             SETB    C
01BE E500     R     MOV     A,key+01H
01C0 9419           SUBB    A,#019H
01C2 E500     R     MOV     A,key
01C4 9400           SUBB    A,#00H
01C6 5014           JNC     ?C0031
                                                   ; SOURCE LINE # 112
                                                   ; SOURCE LINE # 113
01C8 750000   R     MOV     key,#00H
01CB 75001D   R     MOV     key+01H,#01DH
                                                   ; SOURCE LINE # 114
01CE 0500     R     INC     write
                                                   ; SOURCE LINE # 115
01D0 0500     R     INC     key+01H
01D2 E500     R     MOV     A,key+01H
01D4 7002           JNZ     ?C0099
01D6 0500     R     INC     key
01D8          ?C0099:
01D8 FF             MOV     R7,A
01D9 AE00     R     MOV     R6,key
01DB 22             RET
                                                   ; SOURCE LINE # 116
01DC          ?C0031:
                                                   ; SOURCE LINE # 118
01DC C3             CLR     C
01DD E500     R     MOV     A,key+01H
01DF 941E           SUBB    A,#01EH
01E1 E500     R     MOV     A,key
01E3 9400           SUBB    A,#00H
01E5 401F           JC      ?C0032
01E7 D3             SETB    C
01E8 E500     R     MOV     A,key+01H
01EA 9429           SUBB    A,#029H
01EC E500     R     MOV     A,key
01EE 9400           SUBB    A,#00H
01F0 5014           JNC     ?C0032
                                                   ; SOURCE LINE # 119
                                                   ; SOURCE LINE # 120
01F2 E500     R     MOV     A,ab+01H
01F4 6401           XRL     A,#01H
01F6 4500     R     ORL     A,ab
01F8 7007           JNZ     ?C0033
                                                   ; SOURCE LINE # 121
                                                   ; SOURCE LINE # 122
01FA 750004   R     MOV     write,#04H
                                                   ; SOURCE LINE # 123
01FD FE             MOV     R6,A
01FE 7F2D           MOV     R7,#02DH
0200 22             RET
                                                   ; SOURCE LINE # 124
0201          ?C0033:
```

```
                                                    ; SOURCE LINE # 125
0201 7E00              MOV     R6,#00H
0203 7F32              MOV     R7,#032H
0205 22                RET
                                                    ; SOURCE LINE # 126
0206           ?C0032:
                                                    ; SOURCE LINE # 127
0206 7E00              MOV     R6,#00H
0208 7F00              MOV     R7,#00H
020A 22                RET
                                                    ; SOURCE LINE # 129
                                                    ; SOURCE LINE # 130
020B           ?C0018:
                                                    ; SOURCE LINE # 131
020B 7E00              MOV     R6,#00H
020D 7F00              MOV     R7,#00H
                                                    ; SOURCE LINE # 132
020F           ?C0010:
020F 22                RET
               ; FUNCTION getkey (END)

               ; FUNCTION init_lcd (BEGIN)
                                                    ; SOURCE LINE # 138
                                                    ; SOURCE LINE # 139
                                                    ; SOURCE LINE # 140
0000 750080    R       MOV     lcd_addr,#080H
                                                    ; SOURCE LINE # 141
0003 750001    R       MOV     io_stream,#01H
                                                    ; SOURCE LINE # 142
0006 7F01              MOV     R7,#01H
0008 120000    E       LCALL   _putchar
                                                    ; SOURCE LINE # 143
000B 7F02              MOV     R7,#02H
000D 120000    E       LCALL   _putchar
                                                    ; SOURCE LINE # 144
0010 7F38              MOV     R7,#038H
0012 120000    E       LCALL   _putchar
                                                    ; SOURCE LINE # 145
0015 7F0C              MOV     R7,#0CH
0017 120000    E       LCALL   _putchar
                                                    ; SOURCE LINE # 146
001A E4                CLR     A
001B F500      R       MOV     io_stream,A
                                                    ; SOURCE LINE # 148
001D 22                RET
               ; FUNCTION init_lcd (END)

               ; FUNCTION init_sio0 (BEGIN)
                                                    ; SOURCE LINE # 154
                                                    ; SOURCE LINE # 155
                                                    ; SOURCE LINE # 156
0000 C298              CLR     RI
                                                    ; SOURCE LINE # 157
0002 C2AF              CLR     EA
                                                    ; SOURCE LINE # 158
0004 758920            MOV     TMOD,#020H
                                                    ; SOURCE LINE # 159
0007 758DFD            MOV     TH1,#0FDH
                                                    ; SOURCE LINE # 161
000A 7598F8            MOV     SCON,#0F8H
                                                    ; SOURCE LINE # 162
000D D28E              SETB    TR1
                                                    ; SOURCE LINE # 166
000F 22                RET
```

```
             ; FUNCTION init_sio0 (END)

"C51 COMPILER V3.40,  CONSOLE
28/01/94  12:43:25  PAGE 16

             ; FUNCTION _set_lcd (BEGIN)
;---- Variable 'out' assigned to Register 'R7' ----
                                                    ; SOURCE LINE # 172
                                                    ; SOURCE LINE # 173
                                                    ; SOURCE LINE # 174
0000 750001   R     MOV     io_stream,#01H
                                                    ; SOURCE LINE # 175
0003 120000   E     LCALL   _putchar
                                                    ; SOURCE LINE # 176
0006 E4             CLR     A
0007 F500     R     MOV     io_stream,A
                                                    ; SOURCE LINE # 177
0009 22             RET
             ; FUNCTION _set_lcd (END)

             ; FUNCTION main (BEGIN)
                                                    ; SOURCE LINE # 183
                                                    ; SOURCE LINE # 184
                                                    ; SOURCE LINE # 185
0000 C298           CLR     RI
                                                    ; SOURCE LINE # 186
0002 C299           CLR     TI
                                                    ; SOURCE LINE # 187
0004 75000B   R     MOV     write,#0BH
                                                    ; SOURCE LINE # 188
0007 7400     R     MOV     A,#inter
0009 2500     R     ADD     A,write
000B F8             MOV     R0,A
000C E4             CLR     A
000D F6             MOV     @R0,A
                                                    ; SOURCE LINE # 189
000E F500     R     MOV     write,A
                                                    ; SOURCE LINE # 190
0010 120000   R     LCALL   init_lcd
                                                    ; SOURCE LINE # 191
0013 7F01           MOV     R7,#01H
0015 120000   R     LCALL   _set_lcd
                                                    ; SOURCE LINE # 192
0018 7B05           MOV     R3,#05H
001A 7A00     R     MOV     R2,#HIGH ?SC_0
001C 7900     R     MOV     R1,#LOW ?SC_0
001E 120000   E     LCALL   _printf
                                                    ; SOURCE LINE # 193
0021 7FC0           MOV     R7,#0C0H
0023 120000   R     LCALL   _set_lcd
                                                    ; SOURCE LINE # 194
0026 7B05           MOV     R3,#05H
0028 7A00     R     MOV     R2,#HIGH ?SC_16
002A 7900     R     MOV     R1,#LOW ?SC_16
002C 120000   E     LCALL   _printf
                                                    ; SOURCE LINE # 195
002F 120000   R     LCALL   init_sio0
0032          ?C0037:
                                                    ; SOURCE LINE # 196
                                                    ; SOURCE LINE # 197
                                                    ; SOURCE LINE # 199
0032 E4             CLR     A
0033 F500     R     MOV     no,A
0035 F500     R     MOV     no+01H,A
```

```
                                                  ; SOURCE LINE # 208
0037 120000   R      LCALL    getkey
003A 8E00     R      MOV      kboard,R6
003C 8F00     R      MOV      kboard+01H,R7
                                                  ; SOURCE LINE # 209
003E EE              MOV      A,R6
003F 70F1            JNZ      ?C0037
0041 EF              MOV      A,R7
0042 120000   E      LCALL    ?C_CCASE
0045 0000     R      DW       ?C0047
0047 01              DB       01H
0048 0000     R      DW       ?C0047
004A 02              DB       02H
004B 0000     R      DW       ?C0047
004D 03              DB       03H
004E 0000     R      DW       ?C0047
0050 04              DB       04H
0051 0000     R      DW       ?C0047
0053 05              DB       05H
0054 0000     R      DW       ?C0047
0056 06              DB       06H
0057 0000     R      DW       ?C0047
0059 07              DB       07H
005A 0000     R      DW       ?C0047
005C 08              DB       08H
005D 0000     R      DW       ?C0051
005F 0A              DB       0AH
0060 0000     R      DW       ?C0052
0062 0D              DB       0DH
0063 0000     R      DW       ?C0053
0065 14              DB       014H
0066 0000     R      DW       ?C0056
0068 15              DB       015H
0069 0000     R      DW       ?C0057
006B 16              DB       016H
006C 0000     R      DW       ?C0058
006E 17              DB       017H
006F 0000     R      DW       ?C0059
0071 18              DB       018H
0072 0000     R      DW       ?C0060
0074 19              DB       019H
0075 0000     R      DW       ?C0061
0077 1E              DB       01EH
0078 0000     R      DW       ?C0062
007A 1F              DB       01FH
007B 0000     R      DW       ?C0063
007D 20              DB       020H
007E 0000     R      DW       ?C0064
0080 21              DB       021H
0081 0000     R      DW       ?C0065
0083 22              DB       022H
0084 0000     R      DW       ?C0066
0086 23              DB       023H
0087 0000     R      DW       ?C0067
0089 24              DB       024H
008A 0000     R      DW       ?C0068
008C 25              DB       025H
008D 0000     R      DW       ?C0069
008F 26              DB       026H
0090 0000     R      DW       ?C0070
0092 27              DB       027H
0093 0000     R      DW       ?C0071
0095 28              DB       028H
0096 0000     R      DW       ?C0072
```

```
0098 29              DB       029H
0099 0000      R     DW       ?C0073
009B 2D              DB       02DH
009C 0000      R     DW       ?C0074
009E 32              DB       032H
009F 0000            DW       00H
00A1 0000      R     DW       ?C0037
                                                ; SOURCE LINE # 210
                                                ; SOURCE LINE # 211
                                                ; SOURCE LINE # 212
                                                ; SOURCE LINE # 213
                                                ; SOURCE LINE # 214
                                                ; SOURCE LINE # 215
                                                ; SOURCE LINE # 216
                                                ; SOURCE LINE # 217
                                                ; SOURCE LINE # 218
00A3           ?C0047:
                                                ; SOURCE LINE # 219
00A3 7F01            MOV      R7,#01H
00A5 120000    R     LCALL    _set_lcd
                                                ; SOURCE LINE # 220
00A8 7F80            MOV      R7,#080H
00AA 120000    R     LCALL    _set_lcd
                                                ; SOURCE LINE # 221
00AD 7B05            MOV      R3,#05H
00AF 7A00      R     MOV      R2,#HIGH ?SC_0
00B1 7900      R     MOV      R1,#LOW ?SC_0
00B3 120000    E     LCALL    _printf
                                                ; SOURCE LINE # 222
00B6 7FC0            MOV      R7,#0C0H
00B8 120000    R     LCALL    _set_lcd
                                                ; SOURCE LINE # 223
00BB 7B05            MOV      R3,#05H
00BD 7A00      R     MOV      R2,#HIGH ?SC_31
00BF 7900      R     MOV      R1,#LOW ?SC_31
00C1 850000    E     MOV      ?_printf?BYTE+03H,kboard
00C4 850000    E     MOV      ?_printf?BYTE+04H,kboard+01H
00C7 120000    E     LCALL    _printf
                                                ; SOURCE LINE # 224
00CA E500      R     MOV      A,kboard+01H
00CC 4430            ORL      A,#030H
00CE F500      R     MOV      id,A
                                                ; SOURCE LINE # 225
00D0 850000    R     MOV      idt,kboard
00D3 850000    R     MOV      idt+01H,kboard+01H
                                                ; SOURCE LINE # 226
00D6 E4              CLR      A
00D7 F500      R     MOV      write,A
00D9           ?C0048:
00D9 E500      R     MOV      A,write
00DB D3              SETB     C
00DC 940A            SUBB     A,#0AH
00DE 500B            JNC      ?C0049
                                                ; SOURCE LINE # 227
                                                ; SOURCE LINE # 228
00E0 7400      R     MOV      A,#inter
00E2 2500      R     ADD      A,write
00E4 F8              MOV      R0,A
00E5 E4              CLR      A
00E6 F6              MOV      @R0,A
                                                ; SOURCE LINE # 229
00E7 0500      R     INC      write
00E9 80EE            SJMP     ?C0048
00EB           ?C0049:
```

```
                                                ; SOURCE LINE # 230
00EB 750003   R      MOV      write,#03H
                                                ; SOURCE LINE # 231
00EE 7400     R      MOV      A,#inter
00F0 2500     R      ADD      A,write
00F2 F8              MOV      R0,A
00F3 763F            MOV      @R0,#03FH
                                                ; SOURCE LINE # 232
00F5 750006   R      MOV      write,#06H
                                                ; SOURCE LINE # 233
00F8 7400     R      MOV      A,#inter
00FA 2500     R      ADD      A,write
00FC F8              MOV      R0,A
00FD 763F            MOV      @R0,#03FH
                                                ; SOURCE LINE # 234
00FF E4              CLR      A
0100 F500     R      MOV      write,A
                                                ; SOURCE LINE # 235
0102 850000   R      MOV      key,kboard
0105 850000   R      MOV      key+01H,kboard+01H
                                                ; SOURCE LINE # 236
0108 E500     R      MOV      A,kboard+01H
010A 4430            ORL      A,#030H
010C F500     R      MOV      id,A
                                                ; SOURCE LINE # 237
010E 020000   R      LJMP     ?C0037
                                                ; SOURCE LINE # 238
0111          ?C0051:
                                                ; SOURCE LINE # 239
0111 7F01            MOV      R7,#01H
0113 120000   R      LCALL    _set_lcd
                                                ; SOURCE LINE # 240
0116 7F80            MOV      R7,#080H
0118 120000   R      LCALL    _set_lcd
                                                ; SOURCE LINE # 241
011B 7B05            MOV      R3,#05H
011D 7A00     R      MOV      R2,#HIGH ?SC_45
011F 7900     R      MOV      R1,#LOW ?SC_45
0121 120000   E      LCALL    _printf
                                                ; SOURCE LINE # 242
0124 7FC0            MOV      R7,#0C0H
0126 120000   R      LCALL    _set_lcd
                                                ; SOURCE LINE # 243
0129 7B05            MOV      R3,#05H
012B 7A00     R      MOV      R2,#HIGH ?SC_62
012D 7900     R      MOV      R1,#LOW ?SC_62
012F 120000   E      LCALL    _printf
                                                ; SOURCE LINE # 244
0132 750000   R      MOV      key,#00H
0135 75000C   R      MOV      key+01H,#0CH
                                                ; SOURCE LINE # 245
0138 7400     R      MOV      A,#inter
013A 2500     R      ADD      A,write
013C F8              MOV      R0,A
013D 7633            MOV      @R0,#033H
                                                ; SOURCE LINE # 246
013F 750007   R      MOV      write,#07H
                                                ; SOURCE LINE # 247
0142 7400     R      MOV      A,#inter
0144 2500     R      ADD      A,write
0146 F8              MOV      R0,A
0147 7605            MOV      @R0,#05H
                                                ; SOURCE LINE # 248
0149 750008   R      MOV      write,#08H
```

```
                                                    ; SOURCE LINE # 249
0014C 7400      R      MOV      A,#inter
014E 2500       R      ADD      A,write
0150 F8                MOV      R0,A
0151 7605              MOV      @R0,#05H
                                                    ; SOURCE LINE # 250
0153 750009     R      MOV      write,#09H
                                                    ; SOURCE LINE # 251
0156 7400       R      MOV      A,#inter
0158 2500       R      ADD      A,write
015A F8                MOV      R0,A
015B 7680              MOV      @R0,#080H
                                                    ; SOURCE LINE # 252
015D E4                CLR      A
015E F500       R      MOV      write,A
                                                    ; SOURCE LINE # 253
0160 020000     R      LJMP     ?C0037
                                                    ; SOURCE LINE # 254
0163            ?C0052:
                                                    ; SOURCE LINE # 255
0163 7FC0              MOV      R7,#0C0H
0165 120000     R      LCALL    _set_lcd
                                                    ; SOURCE LINE # 256
0168 7B05              MOV      R3,#05H
016A 7A00       R      MOV      R2,#HIGH ?SC_80
016C 7900       R      MOV      R1,#LOW ?SC_80
016E 120000     E      LCALL    _printf
                                                    ; SOURCE LINE # 257
0171 850000     R      MOV      key,kboard
0174 850000     R      MOV      key+01H,kboard+01H
                                                    ; SOURCE LINE # 258
0177 750009     R      MOV      write,#09H
                                                    ; SOURCE LINE # 259
017A 7400       R      MOV      A,#inter
017C 2500       R      ADD      A,write
017E F8                MOV      R0,A
017F 76AE              MOV      @R0,#0AEH
                                                    ; SOURCE LINE # 260
0181 75000A     R      MOV      write,#0AH
                                                    ; SOURCE LINE # 261
0184 7400       R      MOV      A,#inter
0186 2500       R      ADD      A,write
0188 F8                MOV      R0,A
0189 760D              MOV      @R0,#0DH
                                                    ; SOURCE LINE # 262
018B 750007     R      MOV      write,#07H
                                                    ; SOURCE LINE # 263
018E 7400       R      MOV      A,#inter
0190 2500       R      ADD      A,write
0192 F8                MOV      R0,A
0193 7605              MOV      @R0,#05H
                                                    ; SOURCE LINE # 264
0195 0500       R      INC      write
                                                    ; SOURCE LINE # 265
0197 7400       R      MOV      A,#inter
0199 2500       R      ADD      A,write
019B F8                MOV      R0,A
019C 760A              MOV      @R0,#0AH
                                                    ; SOURCE LINE # 266
019E E4                CLR      A
019F F500       R      MOV      write,A
                                                    ; SOURCE LINE # 267
01A1 7400       R      MOV      A,#inter
01A3 2500       R      ADD      A,write
```

```
01A5 F8              MOV     R0,A
01A6 7603            MOV     @R0,#03H
                                                   ; SOURCE LINE # 268
01A8 E4              CLR     A
01A9 F500     R      MOV     write,A
                                                   ; SOURCE LINE # 269
01AB 020000   R      LJMP    ?C0037
                                                   ; SOURCE LINE # 270
01AE          ?C0053:
                                                   ; SOURCE LINE # 271
01AE 7F80            MOV     R7,#080H
01B0 120000   R      LCALL   _set_lcd
                                                   ; SOURCE LINE # 272
01B3 E500     R      MOV     A,spb
01B5 600B            JZ      ?C0054
                                                   ; SOURCE LINE # 273
                                                   ; SOURCE LINE # 274
01B7 7B05            MOV     R3,#05H
01B9 7A00     R      MOV     R2,#HIGH ?SC_98
01BB 7900     R      MOV     R1,#LOW ?SC_98
01BD 120000   E      LCALL   _printf
                                                   ; SOURCE LINE # 275
01C0 8009            SJMP    ?C0055
01C2          ?C0054:
                                                   ; SOURCE LINE # 277
                                                   ; SOURCE LINE # 278
01C2 7B05            MOV     R3,#05H
01C4 7A00     R      MOV     R2,#HIGH ?SC_115
01C6 7900     R      MOV     R1,#LOW ?SC_115
01C8 120000   E      LCALL   _printf
                                                   ; SOURCE LINE # 279
01CB          ?C0055:
                                                   ; SOURCE LINE # 280
01CB 7FC0            MOV     R7,#0C0H
01CD 120000   R      LCALL   _set_lcd
                                                   ; SOURCE LINE # 281
01D0 7B05            MOV     R3,#05H
01D2 7A00     R      MOV     R2,#HIGH ?SC_132
01D4 7900     R      MOV     R1,#LOW ?SC_132
01D6 120000   E      LCALL   _printf
                                                   ; SOURCE LINE # 282
01D9 750000   R      MOV     key,#00H
01DC 750013   R      MOV     key+01H,#013H
                                                   ; SOURCE LINE # 283
01DF 7400     R      MOV     A,#inter
01E1 2500     R      ADD     A,write
01E3 F8              MOV     R0,A
01E4 76BB            MOV     @R0,#0BBH
                                                   ; SOURCE LINE # 284
01E6 020000   R      LJMP    ?C0037
                                                   ; SOURCE LINE # 285
01E9          ?C0056:
                                                   ; SOURCE LINE # 286
01E9 7FC0            MOV     R7,#0C0H
01EB 120000   R      LCALL   _set_lcd
                                                   ; SOURCE LINE # 287
01EE 7B05            MOV     R3,#05H
01F0 7A00     R      MOV     R2,#HIGH ?SC_149
01F2 7900     R      MOV     R1,#LOW ?SC_149
01F4 120000   E      LCALL   _printf
                                                   ; SOURCE LINE # 288
01F7 850000   R      MOV     key,kboard
01FA 850000   R      MOV     key+01H,kboard+01H
                                                   ; SOURCE LINE # 289
```

```
01FD 7400      R      MOV     A,#inter
01FF 2500      R      ADD     A,write
0201 F8               MOV     R0,A
0202 7699             MOV     @R0,#099H
                                                ; SOURCE LINE # 290
0204 020000    R      LJMP    ?C0037
                                                ; SOURCE LINE # 291
0207           ?C0057:
                                                ; SOURCE LINE # 292
0207 7FC0             MOV     R7,#0C0H
0209 120000    R      LCALL   _set_lcd
                                                ; SOURCE LINE # 293
020C 7B05             MOV     R3,#05H
020E 7A00      R      MOV     R2,#HIGH ?SC_166
0210 7900      R      MOV     R1,#LOW ?SC_166
0212 120000    E      LCALL   _printf
                                                ; SOURCE LINE # 294
0215 850000    R      MOV     key,kboard
0218 850000    R      MOV     key+01H,kboard+01H
                                                ; SOURCE LINE # 295
021B 7400      R      MOV     A,#inter
021D 2500      R      ADD     A,write
021F F8               MOV     R0,A
0220 7688             MOV     @R0,#088H
                                                ; SOURCE LINE # 296
0222 020000    R      LJMP    ?C0037
                                                ; SOURCE LINE # 297
0225           ?C0058:
                                                ; SOURCE LINE # 298
0225 7FC0             MOV     R7,#0C0H
0227 120000    R      LCALL   _set_lcd
                                                ; SOURCE LINE # 299
022A 7B05             MOV     R3,#05H
022C 7A00      R      MOV     R2,#HIGH ?SC_183
022E 7900      R      MOV     R1,#LOW ?SC_183
0230 120000    E      LCALL   _printf
                                                ; SOURCE LINE # 300
0233 850000    R      MOV     key,kboard
0236 850000    R      MOV     key+01H,kboard+01H
                                                ; SOURCE LINE # 301
0239 7400      R      MOV     A,#inter
023B 2500      R      ADD     A,write
023D F8               MOV     R0,A
023E 7666             MOV     @R0,#066H
                                                ; SOURCE LINE # 302
0240 020000    R      LJMP    ?C0037
                                                ; SOURCE LINE # 303
0243           ?C0059:
                                                ; SOURCE LINE # 304
0243 7FC0             MOV     R7,#0C0H
0245 120000    R      LCALL   _set_lcd
                                                ; SOURCE LINE # 305
0248 7B05             MOV     R3,#05H
024A 7A00      R      MOV     R2,#HIGH ?SC_200
024C 7900      R      MOV     R1,#LOW ?SC_200
024E 120000    E      LCALL   _printf
                                                ; SOURCE LINE # 306
0251 850000    R      MOV     key,kboard
0254 850000    R      MOV     key+01H,kboard+01H
                                                ; SOURCE LINE # 307
0257 7400      R      MOV     A,#inter
0259 2500      R      ADD     A,write
025B F8               MOV     R0,A
025C 7655             MOV     @R0,#055H
```

```
                                                  ; SOURCE LINE # 308
025E 020000   R      LJMP     ?C0037
                                                  ; SOURCE LINE # 309
0261          ?C0060:
                                                  ; SOURCE LINE # 310
0261 7FC0            MOV      R7,#0C0H
0263 120000   R      LCALL    _set_lcd
                                                  ; SOURCE LINE # 311
0266 7B05            MOV      R3,#05H
0268 7A00     R      MOV      R2,#HIGH ?SC_217
026A 7900     R      MOV      R1,#LOW ?SC_217
026C 120000   E      LCALL    _printf
                                                  ; SOURCE LINE # 312
026F 850000   R      MOV      key,kboard
0272 850000   R      MOV      key+01H,kboard+01H
                                                  ; SOURCE LINE # 313
0275 7400     R      MOV      A,#inter
0277 2500     R      ADD      A,write
0279 F8              MOV      R0,A
027A 7644            MOV      @R0,#044H
                                                  ; SOURCE LINE # 314
027C 020000   R      LJMP     ?C0037
                                                  ; SOURCE LINE # 315
027F          ?C0061:
                                                  ; SOURCE LINE # 316
027F 7FC0            MOV      R7,#0C0H
0281 120000   R      LCALL    _set_lcd
                                                  ; SOURCE LINE # 317
0284 7B05            MOV      R3,#05H
0286 7A00     R      MOV      R2,#HIGH ?SC_234
0288 7900     R      MOV      R1,#LOW ?SC_234
028A 120000   E      LCALL    _printf
                                                  ; SOURCE LINE # 318
028D 850000   R      MOV      key,kboard
0290 850000   R      MOV      key+01H,kboard+01H
                                                  ; SOURCE LINE # 319
0293 7400     R      MOV      A,#inter
0295 2500     R      ADD      A,write
0297 F8              MOV      R0,A
0298 7693            MOV      @R0,#093H
                                                  ; SOURCE LINE # 320
029A 020000   R      LJMP     ?C0037
                                                  ; SOURCE LINE # 321
029D          ?C0062:
                                                  ; SOURCE LINE # 322
029D 7FC0            MOV      R7,#0C0H
029F 120000   R      LCALL    _set_lcd
                                                  ; SOURCE LINE # 323
02A2 7B05            MOV      R3,#05H
02A4 7A00     R      MOV      R2,#HIGH ?SC_251
02A6 7900     R      MOV      R1,#LOW ?SC_251
02A8 120000   E      LCALL    _printf
                                                  ; SOURCE LINE # 324
02AB 850000   R      MOV      key,kboard
02AE 850000   R      MOV      key+01H,kboard+01H
                                                  ; SOURCE LINE # 325
02B1 7400     R      MOV      A,#inter
02B3 2500     R      ADD      A,write
02B5 F8              MOV      R0,A
02B6 7683            MOV      @R0,#083H
                                                  ; SOURCE LINE # 326
02B8 020000   R      LJMP     ?C0037
                                                  ; SOURCE LINE # 327
02BB          ?C0063:
```

```
                                             ; SOURCE LINE # 328
02BB 7FC0              MOV     R7,#0C0H
02BD 120000   R        LCALL   _set_lcd
                                             ; SOURCE LINE # 329
02C0 7B05              MOV     R3,#05H
02C2 7A00     R        MOV     R2,#HIGH ?SC_268
02C4 7900     R        MOV     R1,#LOW ?SC_268
02C6 120000   E        LCALL   _printf
                                             ; SOURCE LINE # 330
02C9 850000   R        MOV     key,kboard
02CC 850000   R        MOV     key+01H,kboard+01H
                                             ; SOURCE LINE # 331
02CF 7400     R        MOV     A,#inter
02D1 2500     R        ADD     A,write
02D3 F8                MOV     R0,A
02D4 7687              MOV     @R0,#087H
                                             ; SOURCE LINE # 332
02D6 020000   R        LJMP    ?C0037
                                             ; SOURCE LINE # 333
02D9          ?C0064:
                                             ; SOURCE LINE # 334
02D9 7FC0              MOV     R7,#0C0H
02DB 120000   R        LCALL   _set_lcd
                                             ; SOURCE LINE # 335
02DE 7B05              MOV     R3,#05H
02E0 7A00     R        MOV     R2,#HIGH ?SC_285
02E2 7900     R        MOV     R1,#LOW ?SC_285
02E4 120000   E        LCALL   _printf
                                             ; SOURCE LINE # 336
02E7 7400     R        MOV     A,#inter
02E9 2500     R        ADD     A,write
02EB F8                MOV     R0,A
02EC 7693              MOV     @R0,#093H
                                             ; SOURCE LINE # 337
02EE 850000   R        MOV     key,kboard
02F1 850000   R        MOV     key+01H,kboard+01H
                                             ; SOURCE LINE # 338
02F4 020000   R        LJMP    ?C0037
                                             ; SOURCE LINE # 339
02F7          ?C0065:
                                             ; SOURCE LINE # 340
02F7 7FC0              MOV     R7,#0C0H
02F9 120000   R        LCALL   _set_lcd
                                             ; SOURCE LINE # 341
02FC 7B05              MOV     R3,#05H
02FE 7A00     R        MOV     R2,#HIGH ?SC_302
0300 7900     R        MOV     R1,#LOW ?SC_302
0302 120000   E        LCALL   _printf
                                             ; SOURCE LINE # 342
0305 7400     R        MOV     A,#inter
0307 2500     R        ADD     A,write
0309 F8                MOV     R0,A
030A 7683              MOV     @R0,#083H
                                             ; SOURCE LINE # 343
030C 850000   R        MOV     key,kboard
030F 850000   R        MOV     key+01H,kboard+01H
                                             ; SOURCE LINE # 344
0312 020000   R        LJMP    ?C0037
                                             ; SOURCE LINE # 345
0315          ?C0066:
                                             ; SOURCE LINE # 346
0315 7FC0              MOV     R7,#0C0H
0317 120000   R        LCALL   _set_lcd
                                             ; SOURCE LINE # 347
```

```
031A 7B05            MOV     R3,#05H
031C 7A00     R      MOV     R2,#HIGH ?SC_319
031E 7900     R      MOV     R1,#LOW ?SC_319
0320 120000   E      LCALL   _printf
                                                ; SOURCE LINE # 348
0323 7400     R      MOV     A,#inter
0325 2500     R      ADD     A,write
0327 F8              MOV     R0,A
0328 7687            MOV     @R0,#087H
                                                ; SOURCE LINE # 349
032A 850000   R      MOV     key,kboard
032D 850000   R      MOV     key+01H,kboard+01H
                                                ; SOURCE LINE # 350
0330 020000   R      LJMP    ?C0037
                                                ; SOURCE LINE # 351
0333          ?C0067:
                                                ; SOURCE LINE # 352
0333 7FC0            MOV     R7,#0C0H
0335 120000   R      LCALL   _set_lcd
                                                ; SOURCE LINE # 353
0338 7B05            MOV     R3,#05H
033A 7A00     R      MOV     R2,#HIGH ?SC_336
033C 7900     R      MOV     R1,#LOW ?SC_336
033E 120000   E      LCALL   _printf
                                                ; SOURCE LINE # 354
0341 7400     R      MOV     A,#inter
0343 2500     R      ADD     A,write
0345 F8              MOV     R0,A
0346 7692            MOV     @R0,#092H
                                                ; SOURCE LINE # 355
0348 850000   R      MOV     key,kboard
034B 850000   R      MOV     key+01H,kboard+01H
                                                ; SOURCE LINE # 356
034E 020000   R      LJMP    ?C0037
                                                ; SOURCE LINE # 357
0351          ?C0068:
                                                ; SOURCE LINE # 358
0351 7FC0            MOV     R7,#0C0H
0353 120000   R      LCALL   _set_lcd
                                                ; SOURCE LINE # 359
0356 7B05            MOV     R3,#05H
0358 7A00     R      MOV     R2,#HIGH ?SC_353
035A 7900     R      MOV     R1,#LOW ?SC_353
035C 120000   E      LCALL   _printf
                                                ; SOURCE LINE # 360
035F 7400     R      MOV     A,#inter
0361 2500     R      ADD     A,write
0363 F8              MOV     R0,A
0364 7682            MOV     @R0,#082H
                                                ; SOURCE LINE # 361
0366 850000   R      MOV     key,kboard
0369 850000   R      MOV     key+01H,kboard+01H
                                                ; SOURCE LINE # 362
036C 020000   R      LJMP    ?C0037
                                                ; SOURCE LINE # 363
036F          ?C0069:
                                                ; SOURCE LINE # 364
036F 7FC0            MOV     R7,#0C0H
0371 120000   R      LCALL   _set_lcd
                                                ; SOURCE LINE # 365
0374 7B05            MOV     R3,#05H
0376 7A00     R      MOV     R2,#HIGH ?SC_370
0378 7900     R      MOV     R1,#LOW ?SC_370
037A 120000   E      LCALL   _printf
```

```
                                                  ; SOURCE LINE # 366
037D 7400     R      MOV     A,#inter
037F 2500     R      ADD     A,write
0381 F8              MOV     R0,A
0382 7686            MOV     @R0,#086H
                                                  ; SOURCE LINE # 367
0384 850000   R      MOV     key,kboard
0387 850000   R      MOV     key+01H,kboard+01H
                                                  ; SOURCE LINE # 368
038A 020000   R      LJMP    ?C0037
                                                  ; SOURCE LINE # 369
038D          ?C0070:
                                                  ; SOURCE LINE # 370
038D 7FC0            MOV     R7,#0C0H
038F 120000   R      LCALL   _set_lcd
                                                  ; SOURCE LINE # 371
0392 7B05            MOV     R3,#05H
0394 7A00     R      MOV     R2,#HIGH ?SC_387
0396 7900     R      MOV     R1,#LOW ?SC_387
0398 120000   E      LCALL   _printf
                                                  ; SOURCE LINE # 372
039B 7400     R      MOV     A,#inter
039D 2500     R      ADD     A,write
039F F8              MOV     R0,A
03A0 7692            MOV     @R0,#092H
                                                  ; SOURCE LINE # 373
03A2 850000   R      MOV     key,kboard
03A5 850000   R      MOV     key+01H,kboard+01H
                                                  ; SOURCE LINE # 374
03A8 020000   R      LJMP    ?C0037
                                                  ; SOURCE LINE # 375
03AB          ?C0071:
                                                  ; SOURCE LINE # 376
03AB 7FC0            MOV     R7,#0C0H
03AD 120000   R      LCALL   _set_lcd
                                                  ; SOURCE LINE # 377
03B0 7B05            MOV     R3,#05H
03B2 7A00     R      MOV     R2,#HIGH ?SC_404
03B4 7900     R      MOV     R1,#LOW ?SC_404
03B6 120000   E      LCALL   _printf
                                                  ; SOURCE LINE # 378
03B9 7400     R      MOV     A,#inter
03BB 2500     R      ADD     A,write
03BD F8              MOV     R0,A
03BE 7682            MOV     @R0,#082H
                                                  ; SOURCE LINE # 379
03C0 850000   R      MOV     key,kboard
03C3 850000   R      MOV     key+01H,kboard+01H
                                                  ; SOURCE LINE # 380
03C6 020000   R      LJMP    ?C0037
                                                  ; SOURCE LINE # 381
03C9          ?C0072:
                                                  ; SOURCE LINE # 382
03C9 7FC0            MOV     R7,#0C0H
03CB 120000   R      LCALL   _set_lcd
                                                  ; SOURCE LINE # 383
03CE 7B05            MOV     R3,#05H
03D0 7A00     R      MOV     R2,#HIGH ?SC_421
03D2 7900     R      MOV     R1,#LOW ?SC_421
03D4 120000   E      LCALL   _printf
                                                  ; SOURCE LINE # 384
03D7 7400     R      MOV     A,#inter
03D9 2500     R      ADD     A,write
03DB F8              MOV     R0,A
```

```
03DC 7686           MOV     @R0,#086H
                                              ; SOURCE LINE # 385
03DE 850000   R     MOV     key,kboard
03E1 850000   R     MOV     key+01H,kboard+01H
                                              ; SOURCE LINE # 386
03E4 020000   R     LJMP    ?C0037
                                              ; SOURCE LINE # 387
03E7          ?C0073:
                                              ; SOURCE LINE # 388
03E7 7F80           MOV     R7,#080H
03E9 120000   R     LCALL   _set_lcd
                                              ; SOURCE LINE # 389
03EC 7B05           MOV     R3,#05H
03EE 7A00     R     MOV     R2,#HIGH ?SC_98
03F0 7900     R     MOV     R1,#LOW ?SC_98
03F2 120000   E     LCALL   _printf
                                              ; SOURCE LINE # 390
03F5 7FC0           MOV     R7,#0C0H
03F7 120000   R     LCALL   _set_lcd
                                              ; SOURCE LINE # 391
03FA 7B05           MOV     R3,#05H
03FC 7A00     R     MOV     R2,#HIGH ?SC_132
03FE 7900     R     MOV     R1,#LOW ?SC_132
0400 120000   E     LCALL   _printf
                                              ; SOURCE LINE # 392
0403 750000   R     MOV     key,#00H
0406 750013   R     MOV     key+01H,#013H
                                              ; SOURCE LINE # 393
0409 E4             CLR     A
040A F500     R     MOV     ab,A
040C F500     R     MOV     ab+01H,A
                                              ; SOURCE LINE # 394
040E 750001   R     MOV     spb,#01H
                                              ; SOURCE LINE # 395
0411 7400     R     MOV     A,#inter
0413 2500     R     ADD     A,write
0415 F8             MOV     R0,A
0416 76BB           MOV     @R0,#0BBH
                                              ; SOURCE LINE # 396
0418 020000   R     LJMP    ?C0037
                                              ; SOURCE LINE # 397
041B          ?C0074:
                                              ; SOURCE LINE # 398
041B E4             CLR     A
041C F500     R     MOV     spb,A
                                              ; SOURCE LINE # 399
041E F500     R     MOV     write,A
                                              ; SOURCE LINE # 400
0420 F500     R     MOV     key,A
0422 F500     R     MOV     key+01H,A
                                              ; SOURCE LINE # 401
0424 D29B           SETB    TB8
                                              ; SOURCE LINE # 402
0426 850099   R     MOV     SBUF,id
0429          ?C0075:
                                              ; SOURCE LINE # 403
0429 3099FD         JNB     TI,?C0075
042C          ?C0076:
                                              ; SOURCE LINE # 404
042C C299           CLR     TI
                                              ; SOURCE LINE # 405
042E C298           CLR     RI
                                              ; SOURCE LINE # 407
0430 750000   R     MOV     delay+03H,#00H
```

```
0433 750000   R      MOV      delay+02H,#00H
0436 750000   R      MOV      delay+01H,#00H
0439 750000   R      MOV      delay,#00H
043C          ?C0077:
                                                   ; SOURCE LINE # 409
043C 7FE8            MOV      R7,#0E8H
043E 7E03            MOV      R6,#03H
0440 7D00            MOV      R5,#00H
0442 7C00            MOV      R4,#00H
0444 120000   E      LCALL    ?C_LPUSH
0447 AF00     R      MOV      R7,delay+03H
0449 AE00     R      MOV      R6,delay+02H
044B AD00     R      MOV      R5,delay+01H
044D AC00     R      MOV      R4,delay
044F 120000   E      LCALL    ?C_SLCMP
0452 5028            JNC      ?C0078
0454 209825          JB       RI,?C0078
                                                   ; SOURCE LINE # 411
                                                   ; SOURCE LINE # 412
0457 AF00     R      MOV      R7,delay+03H
0459 AE00     R      MOV      R6,delay+02H
045B AD00     R      MOV      R5,delay+01H
045D AC00     R      MOV      R4,delay
045F 120000   E      LCALL    ?C_LPUSH
0462 7F01            MOV      R7,#01H
0464 7E00            MOV      R6,#00H
0466 7D00            MOV      R5,#00H
0468 7C00            MOV      R4,#00H
046A 120000   E      LCALL    ?C_LADD
046D 8F00     R      MOV      delay+03H,R7
046F 8E00     R      MOV      delay+02H,R6
0471 8D00     R      MOV      delay+01H,R5
0473 8C00     R      MOV      delay,R4
                                                   ; SOURCE LINE # 413
0475 7F01            MOV      R7,#01H
0477 120000   R      LCALL    _set_lcd
                                                   ; SOURCE LINE # 414
047A 80C0            SJMP     ?C0077
047C          ?C0078:
                                                   ; SOURCE LINE # 415
047C 859900   R      MOV      idtemp,SBUF
                                                   ; SOURCE LINE # 417
047F E500     R      MOV      A,id
0481 6500     R      XRL      A,idtemp
0483 706B            JNZ      ?C0079
                                                   ; SOURCE LINE # 418
                                                   ; SOURCE LINE # 419
0485 C29D            CLR      SM2
                                                   ; SOURCE LINE # 420
0487 C29B            CLR      TB8
                                                   ; SOURCE LINE # 421
0489 C298            CLR      RI
                                                   ; SOURCE LINE # 424
048B F500     R      MOV      write,A
048D          ?C0080:
048D E500     R      MOV      A,write
048F D3              SETB     C
0490 940C            SUBB     A,#0CH
0492 4003            JC       $ + 5H
0494 020000   R      LJMP     ?C0088
                                                   ; SOURCE LINE # 425
                                                   ; SOURCE LINE # 426
0497 C29B            CLR      TB8
                                                   ; SOURCE LINE # 427
```

```
0499 750000   R      MOV      delay+03H,#00H
049C 750000   R      MOV      delay+02H,#00H
049F 750000   R      MOV      delay+01H,#00H
04A2 750000   R      MOV      delay,#00H
04A5          ?C0083:
04A5 7F0A            MOV      R7,#0AH
04A7 7E00            MOV      R6,#00H
04A9 7D00            MOV      R5,#00H
04AB 7C00            MOV      R4,#00H
04AD 120000   E      LCALL    ?C_LPUSH
04B0 AF00     R      MOV      R7,delay+03H
04B2 AE00     R      MOV      R6,delay+02H
04B4 AD00     R      MOV      R5,delay+01H
04B6 AC00     R      MOV      R4,delay
04B8 120000   E      LCALL    ?C_SLCMP
04BB 6002            JZ       $ + 4H
04BD 5020            JNC      ?C0084
04BF AF00     R      MOV      R7,delay+03H
04C1 AE00     R      MOV      R6,delay+02H
04C3 AD00     R      MOV      R5,delay+01H
04C5 AC00     R      MOV      R4,delay
04C7 120000   E      LCALL    ?C_LPUSH
04CA 7F01            MOV      R7,#01H
04CC 7E00            MOV      R6,#00H
04CE 7D00            MOV      R5,#00H
04D0 7C00            MOV      R4,#00H
04D2 120000   E      LCALL    ?C_LADD
04D5 8F00     R      MOV      delay+03H,R7
04D7 8E00     R      MOV      delay+02H,R6
04D9 8D00     R      MOV      delay+01H,R5
04DB 8C00     R      MOV      delay,R4
04DD 80C6            SJMP     ?C0083
04DF          ?C0084:
                                                ; SOURCE LINE # 428
04DF 7400     R      MOV      A,#inter
04E1 2500     R      ADD      A,write
04E3 F8              MOV      R0,A
04E4 E6              MOV      A,@R0
04E5 F599            MOV      SBUF,A
04E7          ?C0086:
                                                ; SOURCE LINE # 429
04E7 3099FD          JNB      TI,?C0086
04EA          ?C0087:
                                                ; SOURCE LINE # 430
04EA C299            CLR      TI
                                                ; SOURCE LINE # 431
04EC 0500     R      INC      write
04EE 809D            SJMP     ?C0080
                                                ; SOURCE LINE # 432
04F0          ?C0079:
                                                ; SOURCE LINE # 434
                                                ; SOURCE LINE # 436
04F0 7F01            MOV      R7,#01H
04F2 120000   R      LCALL    _set_lcd
                                                ; SOURCE LINE # 437
04F5 7F80            MOV      R7,#080H
04F7 120000   R      LCALL    _set_lcd
                                                ; SOURCE LINE # 438
04FA 7B05            MOV      R3,#05H
04FC 7A00     R      MOV      R2,#HIGH ?SC_438
04FE 7900     R      MOV      R1,#LOW ?SC_438
0500 120000   E      LCALL    _printf
                                                ; SOURCE LINE # 439
0503 7FC0            MOV      R7,#0C0H
```

```
0505 120000   R      LCALL    _set_lcd
                                                   ; SOURCE LINE # 440
0508 7B05            MOV      R3,#05H
050A 7A00     R      MOV      R2,#HIGH ?SC_450
050C 7900     R      MOV      R1,#LOW ?SC_450
050E 850000   E      MOV      ?_printf?BYTE+03H,idt
0511 850000   E      MOV      ?_printf?BYTE+04H,idt+01H
0514 120000   E      LCALL    _printf
                                                   ; SOURCE LINE # 441
0517 750000   R      MOV      delay+03H,#00H
051A 750000   R      MOV      delay+02H,#00H
051D 750000   R      MOV      delay+01H,#00H
0520 750000   R      MOV      delay,#00H
0523          ?C0089:
0523 7F10            MOV      R7,#010H
0525 7E27            MOV      R6,#027H
0527 7D00            MOV      R5,#00H
0529 7C00            MOV      R4,#00H
052B 120000   E      LCALL    ?C_LPUSH
052E AF00     R      MOV      R7,delay+03H
0530 AE00     R      MOV      R6,delay+02H
0532 AD00     R      MOV      R5,delay+01H
0534 AC00     R      MOV      R4,delay
0536 120000   E      LCALL    ?C_SLCMP
0539 6002            JZ       $ + 4H
053B 5020            JNC      ?C0090
053D AF00     R      MOV      R7,delay+03H
053F AE00     R      MOV      R6,delay+02H
0541 AD00     R      MOV      R5,delay+01H
0543 AC00     R      MOV      R4,delay
0545 120000   E      LCALL    ?C_LPUSH
0548 7F01            MOV      R7,#01H
054A 7E00            MOV      R6,#00H
054C 7D00            MOV      R5,#00H
054E 7C00            MOV      R4,#00H
0550 120000   E      LCALL    ?C_LADD
0553 8F00     R      MOV      delay+03H,R7
0555 8E00     R      MOV      delay+02H,R6
0557 8D00     R      MOV      delay+01H,R5
0559 8C00     R      MOV      delay,R4
055B 80C6            SJMP     ?C0089
055D          ?C0090:
                                                   ; SOURCE LINE # 442
055D 750000   R      MOV      no,#00H
0560 750001   R      MOV      no+01H,#01H
                                                   ; SOURCE LINE # 443
0563          ?C0088:
                                                   ; SOURCE LINE # 445
0563 D29D            SETB     SM2
                                                   ; SOURCE LINE # 446
0565 D29B            SETB     TB8
                                                   ; SOURCE LINE # 447
0567 7F01            MOV      R7,#01H
0569 120000   R      LCALL    _set_lcd
                                                   ; SOURCE LINE # 448
056C 7F80            MOV      R7,#080H
056E 120000   R      LCALL    _set_lcd
                                                   ; SOURCE LINE # 449
0571 7B05            MOV      R3,#05H
0573 7A00     R      MOV      R2,#HIGH ?SC_467
0575 7900     R      MOV      R1,#LOW ?SC_467
0577 120000   E      LCALL    _printf
                                                   ; SOURCE LINE # 450
057A 7FC0            MOV      R7,#0C0H
```

```
057C 120000    R       LCALL     _set_lcd
                                                     ; SOURCE LINE # 451
057F 7B05              MOV       R3,#05H
0581 7A00      R       MOV       R2,#HIGH ?SC_482
0583 7900      R       MOV       R1,#LOW ?SC_482
0585 120000    E       LCALL     _printf
                                                     ; SOURCE LINE # 452
0588 E4                CLR       A
0589 F500      R       MOV       kboard,A
058B F500      R       MOV       kboard+01H,A
                                                     ; SOURCE LINE # 453
058D F500      R       MOV       key,A
058F F500      R       MOV       key+01H,A
                                                     ; SOURCE LINE # 454
0591 F500      R       MOV       write,A
                                                     ; SOURCE LINE # 455
0593 F500      R       MOV       ab,A
0595 F500      R       MOV       ab+01H,A
                                                     ; SOURCE LINE # 456
0597 F500      R       MOV       act,A
                                                     ; SOURCE LINE # 457
0599 020000    R       LJMP      ?C0037
                                                     ; SOURCE LINE # 459
                                                     ; SOURCE LINE # 461
                                                     ; SOURCE LINE # 462
059C 22                RET
               ; FUNCTION main (END)


NAME                                         CLASS    MSPACE   TYPE     OFFSET   SIZE
====                                         =====    ======   ====     ======   ====

T0 . . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      00B4H    1
AC . . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      00D6H    1
T1 . . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      00B5H    1
EA . . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      00AFH    1
idtemp . . . . . . . . . . . . . . . . . .   PUBLIC   DATA     U_CHAR   0000H    1
getkey . . . . . . . . . . . . . . . . . .   PUBLIC   CODE     PROC     -----    -----
P1_0 . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      0090H    1
delay. . . . . . . . . . . . . . . . . . .   PUBLIC   DATA     LONG     0001H    4
init_sio0. . . . . . . . . . . . . . . . .   PUBLIC   CODE     PROC     -----    -----
RD . . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      00B7H    1
ES . . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      00ACH    1
RI . . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      0098H    1
INT0 . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      00B2H    1
CY . . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      00D7H    1
TI . . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      0099H    1
INT1 . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      00B3H    1
PS . . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      00BCH    1
inter. . . . . . . . . . . . . . . . . . .   PUBLIC   DATA     ARRAY    0005H    12
OV . . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      00D2H    1
main . . . . . . . . . . . . . . . . . . .   PUBLIC   CODE     PROC     -----    -----
WR . . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      00B6H    1
write. . . . . . . . . . . . . . . . . . .   PUBLIC   DATA     U_CHAR   0011H    1
P10. . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      0090H    1
SBUF . . . . . . . . . . . . . . . . . . .   SFR      DATA     U_CHAR   0099H    1
lcd_addr . . . . . . . . . . . . . . . . .   PUBLIC   DATA     U_CHAR   0012H    1
P11. . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      0091H    1
P12. . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      0092H    1
P13. . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      0093H    1
SCON . . . . . . . . . . . . . . . . . . .   SFR      DATA     U_CHAR   0098H    1
P14. . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      0094H    1
TMOD . . . . . . . . . . . . . . . . . . .   SFR      DATA     U_CHAR   0089H    1
P15. . . . . . . . . . . . . . . . . . . .   ABSBIT   -----    BIT      0095H    1
```

```
P16. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      0096H  1
P17. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      0097H  1
act. . . . . . . . . . . . . . . . . . . . PUBLIC   DATA   U_CHAR   0013H  1
IE0. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      0089H  1
IE1. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      008BH  1
_set_lcd . . . . . . . . . . . . . . . . . PUBLIC   CODE   PROC     -----  -----
  out. . . . . . . . . . . . . . . . . . . * REG *  DATA   CHAR     0007H  1
idt. . . . . . . . . . . . . . . . . . . . PUBLIC   DATA   U_INT    0014H  2
ab . . . . . . . . . . . . . . . . . . . . PUBLIC   DATA   U_INT    0016H  2
spb. . . . . . . . . . . . . . . . . . . . PUBLIC   DATA   U_CHAR   0018H  1
io_stream. . . . . . . . . . . . . . . . . PUBLIC   DATA   U_CHAR   0019H  1
ET0. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      00A9H  1
key. . . . . . . . . . . . . . . . . . . . PUBLIC   DATA   U_INT    001AH  2
ET1. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      00ABH  1
TF0. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      008DH  1
init_lcd . . . . . . . . . . . . . . . . . PUBLIC   CODE   PROC     -----  -----
TF1. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      008FH  1
RB8. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      009AH  1
id . . . . . . . . . . . . . . . . . . . . PUBLIC   DATA   U_CHAR   001CH  1
EX0. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      00A8H  1
IT0. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      0088H  1
TH1. . . . . . . . . . . . . . . . . . . . SFR      DATA   U_CHAR   008DH  1
TB8. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      009BH  1
EX1. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      00AAH  1
IT1. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      008AH  1
P. . . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      00D0H  1
SM0. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      009FH  1

NAME                                       CLASS   MSPACE  TYPE    OFFSET  SIZE
====                                       =====   ======  ====    ======  ====

SM1. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      009EH  1
SM2. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      009DH  1
PT0. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      00B9H  1
PT1. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      00BBH  1
RS0. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      00D3H  1
TR0. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      008CH  1
RS1. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      00D4H  1
TR1. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      008EH  1
PX0. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      00B8H  1
IDL. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      0087H  1
PX1. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      00BAH  1
_putchar . . . . . . . . . . . . . . . . . EXTERN   CODE   PROC     -----  -----
no . . . . . . . . . . . . . . . . . . . . PUBLIC   DATA   U_INT    001DH  2
REN. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      009CH  1
RXD. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      00B0H  1
TXD. . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      00B1H  1
F0 . . . . . . . . . . . . . . . . . . . . ABSBIT   -----  BIT      00D5H  1
kboard . . . . . . . . . . . . . . . . . . PUBLIC   DATA   U_INT    001FH  2
_printf. . . . . . . . . . . . . . . . . . EXTERN   CODE   PROC     -----  -----
totchar. . . . . . . . . . . . . . . . . . PUBLIC   DATA   U_INT    0021H  2

MODULE INFORMATION:   STATIC OVERLAYABLE
   CODE SIZE        =   2021    ----
   CONSTANT SIZE    =    494    ----
   XDATA SIZE       =   ----    ----
   PDATA SIZE       =   ----    ----
   DATA SIZE        =     35    ----
   IDATA SIZE       =   ----    ----
   BIT SIZE         =   ----    ----
END OF MODULE INFORMATION.


C51 COMPILATION COMPLETE.  0 WARNING(S),  0 ERROR(S)
```

DBIC SOURCE CODE

```
C51 COMPILER V3.40,  DBIC


DOS C51 COMPILER V3.40, COMPILATION OF MODULE DBIC
OBJECT MODULE PLACED IN DBIC.OBJ
COMPILER INVOKED BY: C:\ZIP\C51.EXE DBIC.C CODE

stmt level     source

   1           #pragma  ROM (LARGE)
   2           #pragma  LARGE
   3           #pragma  PAGELENGTH (66)
   4           #pragma  SYMBOLS
   5           #include  <stdio.h>
   6           #include  <intrins.h>
   7           #include  <reg51.h>
   8           #include  <stdlib.h>
   9           #define  XBYTE ((char *) 0x20000L)
  10           #define  QQQ1 64000
  11           #define QQQ2 300
  12           #define  OPRS 0xFE0E
  13           #define OPRR 0xFE0F
  14           #define  MR1A 0xFE00
  15           #define MR2A 0xFE00
  16           #define MR1B 0xFE08
  17           #define MR2B 0xFE08
  18           #define CSRA 0xFE01
  19           #define CSRB 0xFE09
  20           #define CRA 0xFE02
  21           #define CRB 0xFE0A
  22           #define SRA 0xFE01
  23           #define SRB 0xFE09
  24           #define OPCR 0xFE0D
  25           #define ACR 0xFE04
  26           #define IPCR 0xFE04
  27           #define ISR 0xFE05
  28           #define IMR 0xFE05
  29           #define CTUR 0xFE06
  30           #define CTLR 0xFE07
  31           #define THA 0xFE03
  32           #define RHA 0xFE03
  33           #define THB 0xFE0B
  34           #define RHB 0xFE0B
  35           #define PORTA 0xFF00
  36           #define  PORTC 0xFF02
  37           #define CONTROL 0xFF03
  38           long delay;
  39           unsigned char xdata queel[QQQ1], quee2[QQQ2];
  40           unsigned char   id, cntrl, err, pp;
  41           unsigned int    read_end1 ,write_end1 ,read_end2 ,write_end2 ;
  42           unsigned int b1;
  43           unsigned char tempid,tmpp,busy,ack;
  44           bdata    char  inta,intb;
  45           sbit rxa = inta^0;
  46           sbit     txa = inta^2;
  47           sbit     rxb = intb^0;
  48           sbit txb = intb^2;
  49
  50
  51
```

```
/*********************************************/
/* SERIAL PORTS SERVICE ROUTINE              */
/*********************************************/


 uart ( )


    {
IE0 = 0;
         inta = XBYTE[SRA];
         if (rxa )   /* Port A request reception ? */
  {

             if ( write_end1 >= QQQ1-1)
     {
   if ( read_end1 != 0 )
   {
     queel[write_end1] = XBYTE[RHA];
     write_end1 = 0;
   }

   else
   {
   }



     }
           else if ( ( write_end1 + 1) == read_end1 )
        {

        }
  else

         {
          queel[write_end1] = XBYTE[RHA];
          write_end1++;
         }

     }
        if ( txa )   /* Port A request transmission */
  {
    if ( read_end2 == write_end2   )
      {


      }
    else
      {
        XBYTE[THA] = quee2[read_end2];
        read_end2++;
                if ( read_end2 >= QQQ2)
   {
     read_end2 = 0;
   }
      }
  }
intb = XBYTE[SRB];
        if ( txb )   /* Port B request transmission */
{
  if (read_end1==write_end1)
        {

        }
    else
        {
    XBYTE[THB] = queel[read_endl];
                 read_endl++;

   if ( read_endl >= QQQ1)
            {
                     read_endl = 0;
     }

  }
            }
         if (rxb )    /* Port B request reception */
         {
              if ( write_end2 >= QQQ2 - 1 )
    {
                 if ( read_end2 != 0 )
    {
                   quee2[write_end2] = XBYTE[RHB];
    .               write_end2 = 0;
    }
         else
    {

    }
     }
              else if ( ( write_end2 + 1) == read_end2 )
      {

      }
    else
      {
                  quee2[write_end2] = XBYTE[RHB];
                 write_end2++;
      }
           }
P15 = 0;
    }


/*********************************************/
/* PARALLEL PORT SERVICE ROUTINE            */
/*********************************************/


  pport ( )

{ IE1 = 0;
  if (read_endl==write_endl)    /* buffer empty test */
       {
       }
   else
       { XBYTE[CONTROL] = 0x01;
   XBYTE[PORTA] = queel[read_endl];
               read_endl++;
  XBYTE[CONTROL] = 0x00;  /* strobe data to printer */
  XBYTE[CONTROL] = 0x01;

  if ( read_endl >= QQQ1)   /* buffer full test */
          {
                  read_endl = 0;
    }

```

```
2            }
1
1
1
1
1          P15 = 0 ;
1          }



           /*******************************************/
           /*    ONBOARD SERIAL PORT SERVICE ROUTINE   */
           /*    FOR THE RECEPTION OF CARD ID AND      */
           /*    PARAMETERS AS WELL AS FOR THE         */
           /*    IMPLEMENTATION OF THE PARAMETERS      */
           /*******************************************/


           ser ( )
           {
1             .
1                 if (id == SBUF)
1                   {
2                     P13 = 1;
2                     XBYTE[CRA] = 0x0A;
2                     XBYTE[CRB] = 0x0A;
2                XBYTE[CONTROL] = 0x0C;
2                     RI = 0;
2                TB8 = 1;
2                 SBUF = id;
2                     while (!TI);
2                     TI = 0;
2                     SM2 = 0;
2                     TB8 = 0;
2                     while (!RI);
2                     cntrl = SBUF;
2                     XBYTE[IMR] = cntrl;
2                     XBYTE[ACR] = 0xE0;
2                     RI = 0;
2                     while (!RI);
2                     b1 = SBUF;
2                     RI = 0;
2                     while (!RI);
2                     XBYTE[MR1A] = SBUF;
2                     RI = 0;
2                     while (!RI);
2                     XBYTE[MR2A] = SBUF;
2                     XBYTE[CSRA] = b1;
2                     RI = 0;
2                     while (!RI);
2                     b1 = SBUF;
2                     RI = 0;
2                     while (!RI);
2                     XBYTE[MR1B] = SBUF;
2                     RI = 0;
2                     while (!RI);
2                     XBYTE[MR2B] = SBUF;
2                     XBYTE[CSRB] = b1;
2                     RI = 0;
2                     while (!RI);
2                     XBYTE[CRA] = SBUF;
2                     RI = 0;
2                     while (!RI);
2                     XBYTE[CRB] = SBUF;
```

```
2                RI = 0;
2                while (!RI);
2            tmpp = SBUF;
2                XBYTE[CONTROL] = 0x88;
2              RI = 0;
2             while (!RI);
2             tmpp = SBUF;
2             RI = 0;
2             while(!RI);
2             if (cntrl == 0x33)
2                 {
3               P16 = 1;
3               P17 = 0;
3               pp = 0;
3                 }
2                else
2                {
3           P16 = 0;
3           P17 = 1;
3           pp = 1;
3                }
2          RI = 0;
2          SM2 = 1;
2          TB8 = 1;
2
2             }
1          else
1             {
2           RI = 0;
2             }
1      P13 = 0;
1          }


       /**********************************/
       /*  INITIALISATION ROUTINE FOR    */
       /*  PPI, DUART AND BUFFERS        */
       /**********************************/



       init_all ( )

       { pp = 0;
1      read_end1 = 0;
1      write_end1 = 0;
1      read_end2 = 0;
1      write_end2 = 0;
1      P13 = 0;
1      tempid = P1;
1      id = (((tempid & 0x07) | 0x30) + 1);
1          RI = 0;
1      IT0 = 0;
1      IT1 = 0;
1          EA  = 0;
1      INT1 = 1;
1      INT0 = 1;
1          XBYTE [OPRS] = 0x03;
1          XBYTE [CRA] = 0x10;
1      XBYTE [CRA] = 0x20;
1      XBYTE [CRA] = 0x30;
1      XBYTE [CRB] = 0x10;
1      XBYTE [CRB] = 0x20;
1      XBYTE [CRB] = 0x30;
```

```
1       XBYTE [IMR] = 0x33;
1           XBYTE [ACR] = 0xE0;
1        XBYTE [MR1A] = 0x82;
1        XBYTE [MR1B] = 0x82;
1         XBYTE [MR2A] = 0x17;
1       XBYTE [MR2B] = 0x17;
1           XBYTE [CSRA] = 0xBB;
1        XBYTE [CSRB] = 0x44;
1           XBYTE [OPCR] = 0x00;
1           XBYTE [CONTROL] = 0x88;
1           XBYTE [CRA] = 0x05;
1           XBYTE [CRB] = 0x0A;
1
1
1       }


        /**************************************/
        /* INITIALISATION ROUTINE            */
        /* FOR ONBOARD UART                  */
        /**************************************/


                  init_sio0  ( )
        {
1       RI = 0;
1       EA = 0;
1       TMOD = 0x20;    /* SET TIMER 1 & 2 MODE */
1       TH1 = 0xFD;    /* SET TIMER 1 PRELOAD */
1       REN     = 1;     /* SET SMOD=0  */
1       SCON = 0xF8;
1       TR1 = 1;
1                    /* SET SERIAL PORT MODE */
1
1
1       }

        /****************************************/
        /*  MAIN ROUTINE                 */
        /****************************************/


            main    ( )

        {   init_sio0 ();
1       init_all();
1       XBYTE[CONTROL] = 0x07;
1
1        while (1)
1           {
2       if (RI)
2        {
3         ser ();   /* ID and parameter reception routine*/
3        }
2         XBYTE[CONTROL] = 0x07;
2
2       err = XBYTE[PORTC] & 0x20;
2        if (!err)   /* parallel printer error routine */
2        {
3          if (cntrl != 0x33)
3         {
4                 P17 = !P17;
4                 for (delay=0;delay<=1000;delay++);
4         }
```

```
 370   3         }
 371   2        busy = XBYTE[PORTC] & 0x10;
 372   2        if ((IE1) && (busy!=0x10) && (pp))
 373   2        {
 374   3         pport ( );   /* parallel printer service routine */
 375   3        }
 376   2
 377   2        if (IE0)
 378   2         {
 379   3          uart ( );    /* serial ports service routine */
 380   3         }
 381   2
 382   2          }
 383   1       }
 384
 385
```

ASSEMBLY LISTING OF GENERATED OBJECT CODE

```
             ; FUNCTION uart (BEGIN)
                                                  ; SOURCE LINE # 57
                                                  ; SOURCE LINE # 58
                                                  ; SOURCE LINE # 59
0000 C289              CLR      IE0
                                                  ; SOURCE LINE # 60
0002 90FE01            MOV      DPTR,#0FE01H
0005 E0                MOVX     A,@DPTR
0006 F500        R     MOV      inta,A
                                                  ; SOURCE LINE # 61
0008 200003      R     JB       rxa,$ + 6H
000B 020000      R     LJMP     ?C0001
                                                  ; SOURCE LINE # 62
                                                  ; SOURCE LINE # 64
000E 7FFF              MOV      R7,#0FFH
0010 7EF9              MOV      R6,#0F9H
0012 7D00              MOV      R5,#00H
0014 7C00              MOV      R4,#00H
0016 120000      E     LCALL    ?C_LPUSH
0019 900000      R     MOV      DPTR,#write_endl
001C E0                MOVX     A,@DPTR
001D FA                MOV      R2,A
001E A3                INC      DPTR
001F E0                MOVX     A,@DPTR
0020 FB                MOV      R3,A
0021 FF                MOV      R7,A
0022 AE02              MOV      R6,AR2
0024 E4                CLR      A
0025 FC                MOV      R4,A
0026 FD                MOV      R5,A
0027 120000      E     LCALL    ?C_SLCMP
002A 4024              JC       ?C0002
                                                  ; SOURCE LINE # 65
                                                  ; SOURCE LINE # 66
002C 900000      R     MOV      DPTR,#read_endl
002F E0                MOVX     A,@DPTR
0030 7002              JNZ      ?C0072
0032 A3                INC      DPTR
0033 E0                MOVX     A,@DPTR
0034             ?C0072:
0034 6055              JZ       ?C0001
                                                  ; SOURCE LINE # 67
                                                  ; SOURCE LINE # 68
0036 90FE03            MOV      DPTR,#0FE03H
```

```
0039 E0                MOVX     A,@DPTR
003A FF                MOV      R7,A
003B 7400      R       MOV      A,#LOW queel
003D 2B                ADD      A,R3
003E F582              MOV      DPL,A
0040 EA                MOV      A,R2
0041 3400      R       ADDC     A,#HIGH queel
0043 F583              MOV      DPH,A
0045 EF                MOV      A,R7
0046 F0                MOVX     @DPTR,A
                                                  ; SOURCE LINE # 69
0047 E4                CLR      A
0048 900000    R       MOV      DPTR,#write_endl
004B F0                MOVX     @DPTR,A
004C A3                INC      DPTR
004D F0                MOVX     @DPTR,A
                                                  ; SOURCE LINE # 70
004E 803B              SJMP     ?C0001
                                                  ; SOURCE LINE # 73
                                                  ; SOURCE LINE # 74
                                                  ; SOURCE LINE # 78
0050           ?C0002:
                                                  ; SOURCE LINE # 79
0050 900000    R       MOV      DPTR,#write_endl
0053 E0                MOVX     A,@DPTR
0054 FE                MOV      R6,A
0055 A3                INC      DPTR
0056 E0                MOVX     A,@DPTR
0057 FF                MOV      R7,A
0058 2401              ADD      A,#01H
005A FD                MOV      R5,A
005B E4                CLR      A
005C 3E                ADDC     A,R6
005D FC                MOV      R4,A
005E 900000    R       MOV      DPTR,#read_endl
0061 E0                MOVX     A,@DPTR
0062 FA                MOV      R2,A
0063 A3                INC      DPTR
0064 E0                MOVX     A,@DPTR
0065 6D                XRL      A,R5
0066 7002              JNZ      ?C0073
0068 EC                MOV      A,R4
0069 6A                XRL      A,R2
006A           ?C0073:
006A 601F              JZ       ?C0001
                                                  ; SOURCE LINE # 80
                                                  ; SOURCE LINE # 82
006C           ?C0006:
                                                  ; SOURCE LINE # 85
                                                  ; SOURCE LINE # 86
006C 90FE03            MOV      DPTR,#0FE03H
006F E0                MOVX     A,@DPTR
0070 FD                MOV      R5,A
0071 7400      R       MOV      A,#LOW queel
0073 2F                ADD      A,R7
0074 F582              MOV      DPL,A
0076 EE                MOV      A,R6
0077 3400      R       ADDC     A,#HIGH queel
0079 F583              MOV      DPH,A
007B ED                MOV      A,R5
007C F0                MOVX     @DPTR,A
                                                  ; SOURCE LINE # 87
007D 900000    R       MOV      DPTR,#write_endl+01H
0080 E0                MOVX     A,@DPTR
```

```
0081 04                 INC     A
0082 F0                 MOVX    @DPTR,A
0083 7006               JNZ     ?C0074
0085 900000    R        MOV     DPTR,#write_end1
0088 E0                 MOVX    A,@DPTR
0089 04                 INC     A
008A F0                 MOVX    @DPTR,A
008B           ?C0074:
                                                ; SOURCE LINE # 88
                                                ; SOURCE LINE # 90
008B           ?C0001:
                                                ; SOURCE LINE # 91
008B 300047    R        JNB     txa,?C0008
                                                ; SOURCE LINE # 92
                                                ; SOURCE LINE # 93
008E 900000    R        MOV     DPTR,#write_end2
0091 E0                 MOVX    A,@DPTR
0092 FE                 MOV     R6,A
0093 A3                 INC     DPTR
0094 E0                 MOVX    A,@DPTR
0095 FF                 MOV     R7,A
0096 900000    R        MOV     DPTR,#read_end2
0099 E0                 MOVX    A,@DPTR
009A FC                 MOV     R4,A
009B A3                 INC     DPTR
009C E0                 MOVX    A,@DPTR
009D FD                 MOV     R5,A
009E 6F                 XRL     A,R7
009F 7002               JNZ     ?C0075
00A1 EC                 MOV     A,R4
00A2 6E                 XRL     A,R6
00A3           ?C0075:
00A3 6030               JZ      ?C0008
                                                ; SOURCE LINE # 94
                                                ; SOURCE LINE # 97
00A5           ?C0009:
                                                ; SOURCE LINE # 99
                                                ; SOURCE LINE # 100
00A5 7400      R        MOV     A,#LOW quee2
00A7 2D                 ADD     A,R5
00A8 F582               MOV     DPL,A
00AA EC                 MOV     A,R4
00AB 3400      R        ADDC    A,#HIGH quee2
00AD F583               MOV     DPH,A
00AF E0                 MOVX    A,@DPTR
00B0 90FE03             MOV     DPTR,#0FE03H
00B3 F0                 MOVX    @DPTR,A
                                                ; SOURCE LINE # 101
00B4 900000    R        MOV     DPTR,#read_end2+01H
00B7 E0                 MOVX    A,@DPTR
00B8 04                 INC     A
00B9 F0                 MOVX    @DPTR,A
00BA 7006               JNZ     ?C0076
00BC 900000    R        MOV     DPTR,#read_end2
00BF E0                 MOVX    A,@DPTR
00C0 04                 INC     A
00C1 F0                 MOVX    @DPTR,A
00C2           ?C0076:
                                                ; SOURCE LINE # 102
00C2 C3                 CLR     C
00C3 900000    R        MOV     DPTR,#read_end2+01H
00C6 E0                 MOVX    A,@DPTR
00C7 942C               SUBB    A,#02CH
00C9 900000    R        MOV     DPTR,#read_end2
```

```
00CC E0               MOVX    A,@DPTR
00CD 9401             SUBB    A,#01H
00CF 4004             JC      ?C0008
                                                ; SOURCE LINE # 103
                                                ; SOURCE LINE # 104
00D1 E4               CLR     A
00D2 F0               MOVX    @DPTR,A
00D3 A3               INC     DPTR
00D4 F0               MOVX    @DPTR,A
                                                ; SOURCE LINE # 105
                                                ; SOURCE LINE # 106
                                                ; SOURCE LINE # 107
00D5          ?C0008:
                                                ; SOURCE LINE # 108
00D5 90FE09           MOV     DPTR,#0FE09H
00D8 E0               MOVX    A,@DPTR
00D9 F500      R      MOV     intb,A
                                                ; SOURCE LINE # 109
00DB 300056    R      JNB     txb,?C0012
                                                ; SOURCE LINE # 110
                                                ; SOURCE LINE # 111
00DE 900000    R      MOV     DPTR,#write_endl
00E1 E0               MOVX    A,@DPTR
00E2 FE               MOV     R6,A
00E3 A3               INC     DPTR
00E4 E0               MOVX    A,@DPTR
00E5 FF               MOV     R7,A
00E6 900000    R      MOV     DPTR,#read_endl
00E9 E0               MOVX    A,@DPTR
00EA FC               MOV     R4,A
00EB A3               INC     DPTR
00EC E0               MOVX    A,@DPTR
00ED FD               MOV     R5,A
00EE 6F               XRL     A,R7
00EF 7002             JNZ     ?C0077
00F1 EC               MOV     A,R4
00F2 6E               XRL     A,R6
00F3          ?C0077:
00F3 603F             JZ      ?C0012
                                                ; SOURCE LINE # 112
                                                ; SOURCE LINE # 114
00F5          ?C0013:
                                                ; SOURCE LINE # 116
                                                ; SOURCE LINE # 117
00F5 7400      R      MOV     A,#LOW queel
00F7 2D               ADD     A,R5
00F8 F582             MOV     DPL,A
00FA EC               MOV     A,R4
00FB 3400      R      ADDC    A,#HIGH queel
00FD F583             MOV     DPH,A
00FF E0               MOVX    A,@DPTR
0100 90FE0B           MOV     DPTR,#0FE0BH
0103 F0               MOVX    @DPTR,A
                                                ; SOURCE LINE # 118
0104 900000    R      MOV     DPTR,#read_endl+01H
0107 E0               MOVX    A,@DPTR
0108 04               INC     A
0109 F0               MOVX    @DPTR,A
010A 7006             JNZ     ?C0078
010C 900000    R      MOV     DPTR,#read_endl
010F E0               MOVX    A,@DPTR
0110 04               INC     A
0111 F0               MOVX    @DPTR,A
0112          ?C0078:
```

```
                                                         ; SOURCE LINE # 120
0112 7F00           MOV     R7,#00H
0114 7EFA           MOV     R6,#0FAH
0116 7D00           MOV     R5,#00H
0118 7C00           MOV     R4,#00H
011A 120000   E     LCALL   ?C_LPUSH
011D 900000   R     MOV     DPTR,#read_end1
0120 E0             MOVX    A,@DPTR
0121 FE             MOV     R6,A
0122 A3             INC     DPTR
0123 E0             MOVX    A,@DPTR
0124 FF             MOV     R7,A
0125 E4             CLR     A
0126 FC             MOV     R4,A
0127 FD             MOV     R5,A
0128 120000   E     LCALL   ?C_SLCMP
012B 4007           JC      ?C0012
                                                         ; SOURCE LINE # 121
                                                         ; SOURCE LINE # 122
012D E4             CLR     A
012E 900000   R     MOV     DPTR,#read_end1
0131 F0             MOVX    @DPTR,A
0132 A3             INC     DPTR
0133 F0             MOVX    @DPTR,A
                                                         ; SOURCE LINE # 123
                                                         ; SOURCE LINE # 125
                                                         ; SOURCE LINE # 126
0134          ?C0012:
                                                         ; SOURCE LINE # 127
0134 30006E   R     JNB     rxb,?C0016
                                                         ; SOURCE LINE # 128
                                                         ; SOURCE LINE # 129
0137 900000   R     MOV     DPTR,#write_end2
013A E0             MOVX    A,@DPTR
013B FE             MOV     R6,A
013C A3             INC     DPTR
013D E0             MOVX    A,@DPTR
013E FF             MOV     R7,A
013F C3             CLR     C
0140 942B           SUBB    A,#02BH
0142 EE             MOV     A,R6
0143 9401           SUBB    A,#01H
0145 4024           JC      ?C0017
                                                         ; SOURCE LINE # 130
                                                         ; SOURCE LINE # 131
0147 900000   R     MOV     DPTR,#read_end2
014A E0             MOVX    A,@DPTR
014B 7002           JNZ     ?C0079
014D A3             INC     DPTR
014E E0             MOVX    A,@DPTR
014F          ?C0079:
014F 6054           JZ      ?C0016
                                                         ; SOURCE LINE # 132
                                                         ; SOURCE LINE # 133
0151 90FE0B         MOV     DPTR,#0FE0BH
0154 E0             MOVX    A,@DPTR
0155 FD             MOV     R5,A
0156 7400     R     MOV     A,#LOW quee2
0158 2F             ADD     A,R7
0159 F582           MOV     DPL,A
015B EE             MOV     A,R6
015C 3400     R     ADDC    A,#HIGH quee2
015E F583           MOV     DPH,A
0160 ED             MOV     A,R5
```

```
0161 F0                 MOVX     @DPTR,A
                                                    ; SOURCE LINE # 134
0162 E4                 CLR      A
0163 900000   R         MOV      DPTR,#write_end2
0166 F0                 MOVX     @DPTR,A
0167 A3                 INC      DPTR
0168 F0                 MOVX     @DPTR,A
                                                    ; SOURCE LINE # 135
0169 803A               SJMP     ?C0016
                                                    ; SOURCE LINE # 137
                                                    ; SOURCE LINE # 139
                                                    ; SOURCE LINE # 140
016B          ?C0017:
                                                    ; SOURCE LINE # 141
016B EF                 MOV      A,R7
016C 2401               ADD      A,#01H
016E FF                 MOV      R7,A
016F E4                 CLR      A
0170 3E                 ADDC     A,R6
0171 FE                 MOV      R6,A
0172 900000   R         MOV      DPTR,#read_end2
0175 E0                 MOVX     A,@DPTR
0176 FC                 MOV      R4,A
0177 A3                 INC      DPTR
0178 E0                 MOVX     A,@DPTR
0179 6F                 XRL      A,R7
017A 7002               JNZ      ?C0080
017C EE                 MOV      A,R6
017D 6C                 XRL      A,R4
017E          ?C0080:
017E 6025               JZ       ?C0016
                                                    ; SOURCE LINE # 142
                                                    ; SOURCE LINE # 144
0180          ?C0021:
                                                    ; SOURCE LINE # 146
                                                    ; SOURCE LINE # 147
0180 90FE0B             MOV      DPTR,#0FE0BH
0183 E0                 MOVX     A,@DPTR
0184 FF                 MOV      R7,A
0185 900000   R         MOV      DPTR,#write_end2
0188 E0                 MOVX     A,@DPTR
0189 FC                 MOV      R4,A
018A A3                 INC      DPTR
018B E0                 MOVX     A,@DPTR
018C 2400     R         ADD      A,#LOW quee2
018E F582               MOV      DPL,A
0190 EC                 MOV      A,R4
0191 3400     R         ADDC     A,#HIGH quee2
0193 F583               MOV      DPH,A
0195 EF                 MOV      A,R7
0196 F0                 MOVX     @DPTR,A
                                                    ; SOURCE LINE # 148
0197 900000   R         MOV      DPTR,#write_end2+01H
019A E0                 MOVX     A,@DPTR
019B 04                 INC      A
019C F0                 MOVX     @DPTR,A
019D 7006               JNZ      ?C0081
019F 900000   R         MOV      DPTR,#write_end2
01A2 E0                 MOVX     A,@DPTR
01A3 04                 INC      A
01A4 F0                 MOVX     @DPTR,A
01A5          ?C0081:
                                                    ; SOURCE LINE # 149
                                                    ; SOURCE LINE # 150
```

```
01A5            ?C0016:
                                                    ; SOURCE LINE # 151
01A5 C295              CLR     P15
                                                    ; SOURCE LINE # 152
01A7 22                RET
                ; FUNCTION uart (END)

                ; FUNCTION pport (BEGIN)
                                                    ; SOURCE LINE # 160
                                                    ; SOURCE LINE # 162
0000 C28B              CLR     IE1
                                                    ; SOURCE LINE # 163
0002 900000     R      MOV     DPTR,#write_endl
0005 E0                MOVX    A,@DPTR
0006 FE                MOV     R6,A
0007 A3                INC     DPTR
0008 E0                MOVX    A,@DPTR
0009 FF                MOV     R7,A
000A 900000     R      MOV     DPTR,#read_endl
000D E0                MOVX    A,@DPTR
000E FC                MOV     R4,A
000F A3                INC     DPTR
0010 E0                MOVX    A,@DPTR
0011 FD                MOV     R5,A
0012 6F                XRL     A,R7
0013 7002              JNZ     ?C0082
0015 EC                MOV     A,R4
0016 6E                XRL     A,R6
0017            ?C0082:
0017 604C              JZ      ?C0025
                                                    ; SOURCE LINE # 164
                                                    ; SOURCE LINE # 165
0019            ?C0024:
                                                    ; SOURCE LINE # 167
0019 90FF03            MOV     DPTR,#0FF03H
001C 7401              MOV     A,#01H
001E F0                MOVX    @DPTR,A
                                                    ; SOURCE LINE # 168
001F 7400       R      MOV     A,#LOW queel
0021 2D                ADD     A,R5
0022 F582              MOV     DPL,A
0024 EC                MOV     A,R4
0025 3400       R      ADDC    A,#HIGH queel
0027 F583              MOV     DPH,A
0029 E0                MOVX    A,@DPTR
002A 90FF00            MOV     DPTR,#0FF00H
002D F0                MOVX    @DPTR,A
                                                    ; SOURCE LINE # 169
002E 900000     R      MOV     DPTR,#read_endl+01H
0031 E0                MOVX    A,@DPTR
0032 04                INC     A
0033 F0                MOVX    @DPTR,A
0034 7006              JNZ     ?C0083
0036 900000     R      MOV     DPTR,#read_endl
0039 E0                MOVX    A,@DPTR
003A 04                INC     A
003B F0                MOVX    @DPTR,A
003C            ?C0083:
                                                    ; SOURCE LINE # 170
003C 90FF03            MOV     DPTR,#0FF03H
003F E4                CLR     A
0040 F0                MOVX    @DPTR,A
                                                    ; SOURCE LINE # 171
0041 04                INC     A
```

```
0042 F0                MOVX    @DPTR,A
                                                 ; SOURCE LINE # 173
0043 7F00              MOV     R7,#00H
0045 7EFA              MOV     R6,#0FAH
0047 7D00              MOV     R5,#00H
0049 7C00              MOV     R4,#00H
004B 120000    E       LCALL   ?C_LPUSH
004E 900000    R       MOV     DPTR,#read_end1
0051 E0                MOVX    A,@DPTR
0052 FE                MOV     R6,A
0053 A3                INC     DPTR
0054 E0                MOVX    A,@DPTR
0055 FF                MOV     R7,A
0056 E4                CLR     A
0057 FC                MOV     R4,A
0058 FD                MOV     R5,A
0059 120000    E       LCALL   ?C_SLCMP
005C 4007              JC      ?C0025
                                                 ; SOURCE LINE # 174
                                                 ; SOURCE LINE # 175
005E E4                CLR     A
005F 900000    R       MOV     DPTR,#read_end1
0062 F0                MOVX    @DPTR,A
0063 A3                INC     DPTR
0064 F0                MOVX    @DPTR,A
                                                 ; SOURCE LINE # 176
                                                 ; SOURCE LINE # 178
0065           ?C0025:
                                                 ; SOURCE LINE # 183
0065 C295              CLR     P15
                                                 ; SOURCE LINE # 184
0067 22                RET
               ; FUNCTION pport (END)

               ; FUNCTION ser (BEGIN)
                                                 ; SOURCE LINE # 196
                                                 ; SOURCE LINE # 197
                                                 ; SOURCE LINE # 199
0000 900000    R       MOV     DPTR,#id
0003 E0                MOVX    A,@DPTR
0004 FF                MOV     R7,A
0005 6599              XRL     A,SBUF
0007 6003              JZ      $ + 5H
0009 020000    R       LJMP    ?C0028
                                                 ; SOURCE LINE # 200
                                                 ; SOURCE LINE # 201
000C D293              SETB    P13
                                                 ; SOURCE LINE # 202
000E 90FE02            MOV     DPTR,#0FE02H
0011 740A              MOV     A,#0AH
0013 F0                MOVX    @DPTR,A
                                                 ; SOURCE LINE # 203
0014 90FE0A            MOV     DPTR,#0FE0AH
0017 F0                MOVX    @DPTR,A
                                                 ; SOURCE LINE # 204
0018 90FF03            MOV     DPTR,#0FF03H
001B 740C              MOV     A,#0CH
001D F0                MOVX    @DPTR,A
                                                 ; SOURCE LINE # 205
001E C298              CLR     RI
                                                 ; SOURCE LINE # 206
0020 D29B              SETB    TB8
                                                 ; SOURCE LINE # 207
0022 8F99              MOV     SBUF,R7
```

```
0024          ?C0029:
                                                  ; SOURCE LINE # 208
0024 3099FD        JNB     TI,?C0029
0027          ?C0030:
                                                  ; SOURCE LINE # 209
0027 C299          CLR     TI
                                                  ; SOURCE LINE # 210
0029 C29D          CLR     SM2
                                                  ; SOURCE LINE # 211
002B C29B          CLR     TB8
002D          ?C0031:
                                                  ; SOURCE LINE # 212
002D 3098FD        JNB     RI,?C0031
0030          ?C0032:
                                                  ; SOURCE LINE # 213
0030 900000   R    MOV     DPTR,#cntrl
0033 E599          MOV     A,SBUF
0035 F0            MOVX    @DPTR,A
                                                  ; SOURCE LINE # 214
0036 E0            MOVX    A,@DPTR
0037 90FE05        MOV     DPTR,#0FE05H
003A F0            MOVX    @DPTR,A
                                                  ; SOURCE LINE # 215
003B 90FE04        MOV     DPTR,#0FE04H
003E 74E0          MOV     A,#0E0H
0040 F0            MOVX    @DPTR,A
                                                  ; SOURCE LINE # 216
0041 C298          CLR     RI
0043          ?C0033:
                                                  ; SOURCE LINE # 217
0043 3098FD        JNB     RI,?C0033
0046          ?C0034:
                                                  ; SOURCE LINE # 218
0046 AF99          MOV     R7,SBUF
0048 900000   R    MOV     DPTR,#b1
004B E4            CLR     A
004C F0            MOVX    @DPTR,A
004D A3            INC     DPTR
004E EF            MOV     A,R7
004F F0            MOVX    @DPTR,A
                                                  ; SOURCE LINE # 219
0050 C298          CLR     RI
0052          ?C0035:
                                                  ; SOURCE LINE # 220
0052 3098FD        JNB     RI,?C0035
0055          ?C0036:
                                                  ; SOURCE LINE # 221
0055 90FE00        MOV     DPTR,#0FE00H
0058 E599          MOV     A,SBUF
005A F0            MOVX    @DPTR,A
                                                  ; SOURCE LINE # 222
005B C298          CLR     RI
005D          ?C0037:
                                                  ; SOURCE LINE # 223
005D 3098FD        JNB     RI,?C0037
0060          ?C0038:
                                                  ; SOURCE LINE # 224
0060 90FE00        MOV     DPTR,#0FE00H
0063 E599          MOV     A,SBUF
0065 F0            MOVX    @DPTR,A
                                                  ; SOURCE LINE # 225
0066 900000   R    MOV     DPTR,#b1+01H
0069 E0            MOVX    A,@DPTR
006A 90FE01        MOV     DPTR,#0FE01H
```

```
006D F0              MOVX    @DPTR,A
                                             ; SOURCE LINE # 226
006E C298            CLR     RI
0070         ?C0039:
                                             ; SOURCE LINE # 227
0070 3098FD          JNB     RI,?C0039
0073         ?C0040:
                                             ; SOURCE LINE # 228
0073 AF99            MOV     R7,SBUF
0075 900000  R       MOV     DPTR,#b1
0078 E4              CLR     A
0079 F0              MOVX    @DPTR,A
007A A3              INC     DPTR
007B EF              MOV     A,R7
007C F0              MOVX    @DPTR,A
                                             ; SOURCE LINE # 229
007D C298            CLR     RI
007F         ?C0041:
                                             ; SOURCE LINE # 230
007F 3098FD          JNB     RI,?C0041
0082         ?C0042:
                                             ; SOURCE LINE # 231
0082 90FE08          MOV     DPTR,#0FE08H
0085 E599            MOV     A,SBUF
0087 F0              MOVX    @DPTR,A
                                             ; SOURCE LINE # 232
0088 C298            CLR     RI
008A         ?C0043:
                                             ; SOURCE LINE # 233
008A 3098FD          JNB     RI,?C0043
008D         ?C0044:
                                             ; SOURCE LINE # 234
008D 90FE08          MOV     DPTR,#0FE08H
0090 E599            MOV     A,SBUF
0092 F0              MOVX    @DPTR,A
                                             ; SOURCE LINE # 235
0093 900000  R       MOV     DPTR,#b1+01H
0096 E0              MOVX    A,@DPTR
0097 90FE09          MOV     DPTR,#0FE09H
009A F0              MOVX    @DPTR,A
                                             ; SOURCE LINE # 236
009B C298            CLR     RI
009D         ?C0045:
                                             ; SOURCE LINE # 237
009D 3098FD          JNB     RI,?C0045
00A0         ?C0046:
                                             ; SOURCE LINE # 238
00A0 90FE02          MOV     DPTR,#0FE02H
00A3 E599            MOV     A,SBUF
00A5 F0              MOVX    @DPTR,A
                                             ; SOURCE LINE # 239
00A6 C298            CLR     RI
00A8         ?C0047:
                                             ; SOURCE LINE # 240
00A8 3098FD          JNB     RI,?C0047
00AB         ?C0048:
                                             ; SOURCE LINE # 241
00AB 90FE0A          MOV     DPTR,#0FE0AH
00AE E599            MOV     A,SBUF
00B0 F0              MOVX    @DPTR,A
                                             ; SOURCE LINE # 242
00B1 C298            CLR     RI
00B3         ?C0049:
                                             ; SOURCE LINE # 243
```

```
00B3 3098FD          JNB     RI,?C0049
00B6            ?C0050:
                                          ; SOURCE LINE # 244
00B6 900000     R    MOV     DPTR,#tmpp
00B9 E599            MOV     A,SBUF
00BB F0              MOVX    @DPTR,A
                                          ; SOURCE LINE # 245
00BC 90FF03          MOV     DPTR,#0FF03H
00BF 7488            MOV     A,#088H
00C1 F0              MOVX    @DPTR,A
                                          ; SOURCE LINE # 246
00C2 C298            CLR     RI
00C4            ?C0051:
                                          ; SOURCE LINE # 247
00C4 3098FD          JNB     RI,?C0051
00C7            ?C0052:
                                          ; SOURCE LINE # 248
00C7 900000     R    MOV     DPTR,#tmpp
00CA E599            MOV     A,SBUF
00CC F0              MOVX    @DPTR,A
                                          ; SOURCE LINE # 249
00CD C298            CLR     RI
00CF            ?C0053:
                                          ; SOURCE LINE # 250
00CF 3098FD          JNB     RI,?C0053
00D2            ?C0054:
                                          ; SOURCE LINE # 251
00D2 900000     R    MOV     DPTR,#cntrl
00D5 E0              MOVX    A,@DPTR
00D6 B4330B          CJNE    A,#033H,?C0055
                                          ; SOURCE LINE # 252
                                          ; SOURCE LINE # 253
00D9 D296            SETB    P16
                                          ; SOURCE LINE # 254
00DB C297            CLR     P17
                                          ; SOURCE LINE # 255
00DD E4              CLR     A
00DE 900000     R    MOV     DPTR,#pp
00E1 F0              MOVX    @DPTR,A
                                          ; SOURCE LINE # 256
00E2 800A            SJMP    ?C0056
00E4            ?C0055:
                                          ; SOURCE LINE # 258
                                          ; SOURCE LINE # 259
00E4 C296            CLR     P16
                                          ; SOURCE LINE # 260
00E6 D297            SETB    P17
                                          ; SOURCE LINE # 261
00E8 900000     R    MOV     DPTR,#pp
00EB 7401            MOV     A,#01H
00ED F0              MOVX    @DPTR,A
                                          ; SOURCE LINE # 262
00EE            ?C0056:
                                          ; SOURCE LINE # 263
00EE C298            CLR     RI
                                          ; SOURCE LINE # 264
00F0 D29D            SETB    SM2
                                          ; SOURCE LINE # 265
00F2 D29B            SETB    TB8
                                          ; SOURCE LINE # 267
00F4 8002            SJMP    ?C0057
00F6            ?C0028:
                                          ; SOURCE LINE # 269
                                          ; SOURCE LINE # 270
```

```
00F6 C298           CLR     RI
                                                  ; SOURCE LINE # 271
00F8          ?C0057:
                                                  ; SOURCE LINE # 272
00F8 C293           CLR     P13
                                                  ; SOURCE LINE # 273
00FA 22             RET
              ; FUNCTION ser (END)

              ; FUNCTION init_all (BEGIN)
                                                  ; SOURCE LINE # 283
                                                  ; SOURCE LINE # 285
0000 E4             CLR     A
0001 900000   R     MOV     DPTR,#pp
0004 F0             MOVX    @DPTR,A
                                                  ; SOURCE LINE # 286
0005 900000   R     MOV     DPTR,#read_end1
0008 F0             MOVX    @DPTR,A
0009 A3             INC     DPTR
000A F0             MOVX    @DPTR,A
                                                  ; SOURCE LINE # 287
000B 900000   R     MOV     DPTR,#write_end1
000E F0             MOVX    @DPTR,A
000F A3             INC     DPTR
0010 F0             MOVX    @DPTR,A
                                                  ; SOURCE LINE # 288
0011 900000   R     MOV     DPTR,#read_end2
0014 F0             MOVX    @DPTR,A
0015 A3             INC     DPTR
0016 F0             MOVX    @DPTR,A
                                                  ; SOURCE LINE # 289
0017 900000   R     MOV     DPTR,#write_end2
001A F0             MOVX    @DPTR,A
001B A3             INC     DPTR
001C F0             MOVX    @DPTR,A
                                                  ; SOURCE LINE # 290
001D C293           CLR     P13
                                                  ; SOURCE LINE # 291
001F 900000   R     MOV     DPTR,#tempid
0022 E590           MOV     A,P1
0024 F0             MOVX    @DPTR,A
                                                  ; SOURCE LINE # 292
0025 E0             MOVX    A,@DPTR
0026 5407           ANL     A,#07H
0028 4430           ORL     A,#030H
002A 04             INC     A
002B 900000   R     MOV     DPTR,#id
002E F0             MOVX    @DPTR,A
                                                  ; SOURCE LINE # 293
002F C298           CLR     RI
                                                  ; SOURCE LINE # 294
0031 C288           CLR     IT0
                                                  ; SOURCE LINE # 295
0033 C28A           CLR     IT1
                                                  ; SOURCE LINE # 296
0035 C2AF           CLR     EA
                                                  ; SOURCE LINE # 297
0037 D2B3           SETB    INT1
                                                  ; SOURCE LINE # 298
0039 D2B2           SETB    INT0
                                                  ; SOURCE LINE # 299
003B 90FE0E         MOV     DPTR,#0FE0EH
003E 7403           MOV     A,#03H
0040 F0             MOVX    @DPTR,A
```

```
                                                    ; SOURCE LINE # 300
0041 90FE02          MOV      DPTR,#0FE02H
0044 7410            MOV      A,#010H
0046 F0              MOVX     @DPTR,A
                                                    ; SOURCE LINE # 301
0047 7420            MOV      A,#020H
0049 F0              MOVX     @DPTR,A
                                                    ; SOURCE LINE # 302
004A 7430            MOV      A,#030H
004C F0              MOVX     @DPTR,A
                                                    ; SOURCE LINE # 303
004D 90FE0A          MOV      DPTR,#0FE0AH
0050 7410            MOV      A,#010H
0052 F0              MOVX     @DPTR,A
                                                    ; SOURCE LINE # 304
0053 7420            MOV      A,#020H
0055 F0              MOVX     @DPTR,A
                                                    ; SOURCE LINE # 305
0056 7430            MOV      A,#030H
0058 F0              MOVX     @DPTR,A
                                                    ; SOURCE LINE # 306
0059 90FE05          MOV      DPTR,#0FE05H
005C 7433            MOV      A,#033H
005E F0              MOVX     @DPTR,A
                                                    ; SOURCE LINE # 307
005F 90FE04          MOV      DPTR,#0FE04H
0062 74E0            MOV      A,#0E0H
0064 F0              MOVX     @DPTR,A
                                                    ; SOURCE LINE # 308
0065 90FE00          MOV      DPTR,#0FE00H
0068 7482            MOV      A,#082H
006A F0              MOVX     @DPTR,A
                                                    ; SOURCE LINE # 309
006B 90FE08          MOV      DPTR,#0FE08H
006E F0              MOVX     @DPTR,A
                                                    ; SOURCE LINE # 310
006F 90FE00          MOV      DPTR,#0FE00H
0072 7417            MOV      A,#017H
0074 F0              MOVX     @DPTR,A
                                                    ; SOURCE LINE # 311
0075 90FE08          MOV      DPTR,#0FE08H
0078 F0              MOVX     @DPTR,A
                                                    ; SOURCE LINE # 312
0079 90FE01          MOV      DPTR,#0FE01H
007C 74BB            MOV      A,#0BBH
007E F0              MOVX     @DPTR,A
                                                    ; SOURCE LINE # 313
007F 90FE09          MOV      DPTR,#0FE09H
0082 7444            MOV      A,#044H
0084 F0              MOVX     @DPTR,A
                                                    ; SOURCE LINE # 314
0085 90FE0D          MOV      DPTR,#0FE0DH
0088 E4              CLR      A
0089 F0              MOVX     @DPTR,A
                                                    ; SOURCE LINE # 315
008A 90FF03          MOV      DPTR,#0FF03H
008D 7488            MOV      A,#088H
008F F0              MOVX     @DPTR,A
                                                    ; SOURCE LINE # 316
0090 90FE02          MOV      DPTR,#0FE02H
0093 7405            MOV      A,#05H
0095 F0              MOVX     @DPTR,A
                                                    ; SOURCE LINE # 317
0096 90FE0A          MOV      DPTR,#0FE0AH
```

```
0099 740A            MOV     A,#0AH
009B F0              MOVX    @DPTR,A
                                               ; SOURCE LINE # 320
009C 22              RET
               ; FUNCTION init_all (END)

               ; FUNCTION init_sio0 (BEGIN)
                                               ; SOURCE LINE # 329
                                               ; SOURCE LINE # 330
                                               ; SOURCE LINE # 331
0000 C298            CLR     RI
                                               ; SOURCE LINE # 332
0002 C2AF            CLR     EA
                                               ; SOURCE LINE # 333
0004 758920          MOV     TMOD,#020H
                                               ; SOURCE LINE # 334
0007 758DFD          MOV     TH1,#0FDH

C51 COMPILER V3.40,  DBIC
1

                                               ; SOURCE LINE # 335
000A D29C            SETB    REN
                                               ; SOURCE LINE # 336
000C 7598F8          MOV     SCON,#0F8H
                                               ; SOURCE LINE # 337
000F D28E            SETB    TR1
                                               ; SOURCE LINE # 341
0011 22              RET
               ; FUNCTION init_sio0 (END)

               ; FUNCTION main (BEGIN)
                                               ; SOURCE LINE # 348
                                               ; SOURCE LINE # 350
0000 120000    R     LCALL   init_sio0
                                               ; SOURCE LINE # 351
0003 120000    R     LCALL   init_all
                                               ; SOURCE LINE # 352
0006 90FF03          MOV     DPTR,#0FF03H
0009 7407            MOV     A,#07H
000B F0              MOVX    @DPTR,A
000C           ?C0061:
                                               ; SOURCE LINE # 354
                                               ; SOURCE LINE # 355
                                               ; SOURCE LINE # 356
000C 309803          JNB     RI,?C0063
                                               ; SOURCE LINE # 357
                                               ; SOURCE LINE # 358
000F 120000    R     LCALL   ser
                                               ; SOURCE LINE # 359
0012           ?C0063:
                                               ; SOURCE LINE # 360
0012 90FF03          MOV     DPTR,#0FF03H
0015 7407            MOV     A,#07H
0017 F0              MOVX    @DPTR,A
                                               ; SOURCE LINE # 362
0018 90FF02          MOV     DPTR,#0FF02H
001B E0              MOVX    A,@DPTR
001C 5420            ANL     A,#020H
001E 900000    R     MOV     DPTR,#err
0021 F0              MOVX    @DPTR,A
                                               ; SOURCE LINE # 363
0022 7048            JNZ     ?C0064
                                               ; SOURCE LINE # 364
```

```
                                                    ; SOURCE LINE # 365
0024 900000   R     MOV      DPTR,#cntrl
0027 E0             MOVX     A,@DPTR
0028 6433           XRL      A,#033H
002A 6040           JZ       ?C0064
                                                    ; SOURCE LINE # 366
                                                    ; SOURCE LINE # 367
002C B297           CPL      P17
                                                    ; SOURCE LINE # 368
002E 900000   R     MOV      DPTR,#delay
0031 120000   E     LCALL    ?C_LSTKXDATA
0034 00             DB       #00H
0035 00             DB       #00H
0036 00             DB       #00H
0037 00             DB       #00H
0038          ?C0066:
0038 7FE8           MOV      R7,#0E8H
003A 7E03           MOV      R6,#03H
003C 7D00           MOV      R5,#00H
003E 7C00           MOV      R4,#00H
0040 120000   E     LCALL    ?C_LPUSH
0043 900000   R     MOV      DPTR,#delay
0046 120000   E     LCALL    ?C_LLDXDATA
0049 120000   E     LCALL    ?C_SLCMP
004C 6002           JZ       $ + 4H
004E 501C           JNC      ?C0064
0050 900000   R     MOV      DPTR,#delay
0053 120000   E     LCALL    ?C_LLDXDATA
0056 120000   E     LCALL    ?C_LPUSH
0059 7F01           MOV      R7,#01H
005B 7E00           MOV      R6,#00H
005D 7D00           MOV      R5,#00H
005F 7C00           MOV      R4,#00H
0061 120000   E     LCALL    ?C_LADD
0064 900000   R     MOV      DPTR,#delay
0067 120000   E     LCALL    ?C_LSTXDATA
006A 80CC           SJMP     ?C0066
                                                    ; SOURCE LINE # 369
                                                    ; SOURCE LINE # 370
006C          ?C0064:
                                                    ; SOURCE LINE # 371
006C 90FF02         MOV      DPTR,#0FF02H
006F E0             MOVX     A,@DPTR
0070 5410           ANL      A,#010H
0072 900000   R     MOV      DPTR,#busy
0075 F0             MOVX     @DPTR,A
                                                    ; SOURCE LINE # 372
0076 308B0E         JNB      IE1,?C0069
0079 E0             MOVX     A,@DPTR
007A 6410           XRL      A,#010H
007C 6009           JZ       ?C0069
007E 900000   R     MOV      DPTR,#pp
0081 E0             MOVX     A,@DPTR
0082 6003           JZ       ?C0069
                                                    ; SOURCE LINE # 373
                                                    ; SOURCE LINE # 374
0084 120000   R     LCALL    pport
                                                    ; SOURCE LINE # 375
0087          ?C0069:
                                                    ; SOURCE LINE # 377
0087 308982         JNB      IE0,?C0061
                                                    ; SOURCE LINE # 378
                                                    ; SOURCE LINE # 379
008A 120000   R     LCALL    uart
```

```
                                                  ; SOURCE LINE # 380
                                                  ; SOURCE LINE # 382
0080 020000   R     LJMP    ?C0061
                                                  ; SOURCE LINE # 383
0090 22             RET
              ; FUNCTION main (END)

NAME                                        CLASS    MSPACE   TYPE     OFFSET   SIZE
====                                        =====    ======   ====     ======   ====


P1 . . . . . . . . . . . . . . . . . . .    SFR      DATA     U_CHAR   0090H    1
T0 . . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      00B4H    1
AC . . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      00D6H    1
T1 . . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      00B5H    1
EA . . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      00AFH    1
tempid . . . . . . . . . . . . . . . . .    PUBLIC   XDATA    U_CHAR   0000H    1
P1_0 . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      0090H    1
delay. . . . . . . . . . . . . . . . . .    PUBLIC   XDATA    LONG     0001H    4
b1 . . . . . . . . . . . . . . . . . . .    PUBLIC   XDATA    U_INT    0005H    2
init_sio0. . . . . . . . . . . . . . . .    PUBLIC   CODE     PROC     -----    -----
RD . . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      00B7H    1
ES . . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      00ACH    1
RI . . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      0098H    1
INT0 . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      00B2H    1
CY . . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      00D7H    1
TI . . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      0099H    1
INT1 . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      00B3H    1
PS . . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      00BCH    1
OV . . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      00D2H    1
cntrl. . . . . . . . . . . . . . . . . .    PUBLIC   XDATA    U_CHAR   0007H    1
main . . . . . . . . . . . . . . . . . .    PUBLIC   CODE     PROC     -----    -----
WR . . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      00B6H    1
inta . . . . . . . . . . . . . . . . . .    PUBLIC   DATA     CHAR     0000H    1
intb . . . . . . . . . . . . . . . . . .    PUBLIC   DATA     CHAR     0001H    1
ack. . . . . . . . . . . . . . . . . . .    PUBLIC   XDATA    U_CHAR   0008H    1
P10. . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      0090H    1
SBUF . . . . . . . . . . . . . . . . . .    SFR      DATA     U_CHAR   0099H    1
P11. . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      0091H    1
P12. . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      0092H    1
P13. . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      0093H    1
SCON . . . . . . . . . . . . . . . . . .    SFR      DATA     U_CHAR   0098H    1
P14. . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      0094H    1
TMOD . . . . . . . . . . . . . . . . . .    SFR      DATA     U_CHAR   0089H    1
P15. . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      0095H    1
P16. . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      0096H    1
pport. . . . . . . . . . . . . . . . . .    PUBLIC   CODE     PROC     -----    -----
P17. . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      0097H    1
uart . . . . . . . . . . . . . . . . . .    PUBLIC   CODE     PROC     -----    -----
IE0. . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      0089H    1
IE1. . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      008BH    1
tmpp . . . . . . . . . . . . . . . . . .    PUBLIC   XDATA    U_CHAR   0009H    1
busy . . . . . . . . . . . . . . . . . .    PUBLIC   XDATA    U_CHAR   000AH    1
ET0. . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      00A9H    1
err. . . . . . . . . . . . . . . . . . .    PUBLIC   XDATA    U_CHAR   000BH    1
ET1. . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      00ABH    1
TF0. . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      008DH    1
ser. . . . . . . . . . . . . . . . . . .    PUBLIC   CODE     PROC     -----    -----
TF1. . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      008FH    1
rxa. . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      0000H    1
RB8. . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      009AH    1
rxb. . . . . . . . . . . . . . . . . . .    ABSBIT   -----    BIT      0008H    1
id . . . . . . . . . . . . . . . . . . .    PUBLIC   XDATA    U_CHAR   000CH    1
```

```
EX0. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      00A8H  1
IT0. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      0088H  1
TH1. . . . . . . . . . . . . . . . . . .  SFR      DATA    U_CHAR   008DH  1
txa. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      0002H  1
TB8. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      009BH  1
EX1. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      00AAH  1
IT1. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      008AH  1
                                                                       4

NAME                                      CLASS   MSPACE  TYPE    OFFSET  SIZE
====                                      =====   ======  ====    ======  ====


txb. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      000AH  1
P. . . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      00D0H  1
SM0. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      009FH  1
init_all . . . . . . . . . . . . . . . .  PUBLIC   CODE    PROC     -----  -----
SM1. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      009EH  1
SM2. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      009DH  1
PT0. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      00B9H  1
PT1. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      00BBH  1
RS0. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      00D3H  1
TR0. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      008CH  1
RS1. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      00D4H  1
TR1. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      008EH  1
PX0. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      00B8H  1
IDL. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      0087H  1
PX1. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      00BAH  1
pp . . . . . . . . . . . . . . . . . . .  PUBLIC   XDATA   U_CHAR   000DH  1
quee1. . . . . . . . . . . . . . . . . .  PUBLIC   XDATA   ARRAY    000EH  64000
quee2. . . . . . . . . . . . . . . . . .  PUBLIC   XDATA   ARRAY    FA0EH  300
REN. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      009CH  1
read_end1. . . . . . . . . . . . . . . .  PUBLIC   XDATA   U_INT    FB3AH  2
read_end2. . . . . . . . . . . . . . . .  PUBLIC   XDATA   U_INT    FB3CH  2
RXD. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      00B0H  1
TXD. . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      00B1H  1
F0 . . . . . . . . . . . . . . . . . . .  ABSBIT   -----   BIT      00D5H  1
write_end1 . . . . . . . . . . . . . . .  PUBLIC   XDATA   U_INT    FB3EH  2
write_end2 . . . . . . . . . . . . . . .  PUBLIC   XDATA   U_INT    FB40H  2


MODULE INFORMATION:   STATIC OVERLAYABLE
   CODE SIZE        =   1099    ----
   CONSTANT SIZE    =   ----    ----
   XDATA SIZE       =  64322    ----
   PDATA SIZE       =   ----    ----
   DATA SIZE        =      2    ----
   IDATA SIZE       =   ----    ----
   BIT SIZE         =   ----    ----
END OF MODULE INFORMATION.


C51 COMPILATION COMPLETE.  0 WARNING(S),  0 ERROR(S)
```