DCE MICROCOMPUTER SYSTEMS DESIGNER'S HANDBOOK

DAI THE MICROCOMPUTER ENGINEERING COMPANY

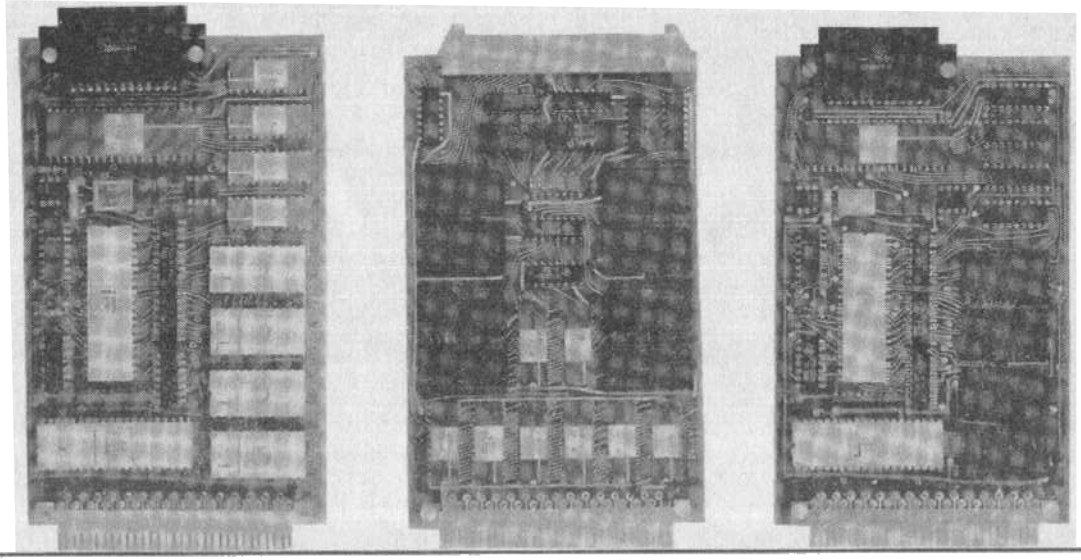
DATA APPLICATIONS INTERNATIONAL Drève des Renards 6 Bte 8. 1180 Brussels Belgium
Tel (02) 375 11 14 - Telex 61 067 daibru b

July 1980.

DAI brings Microcomputers...

COMPLETE

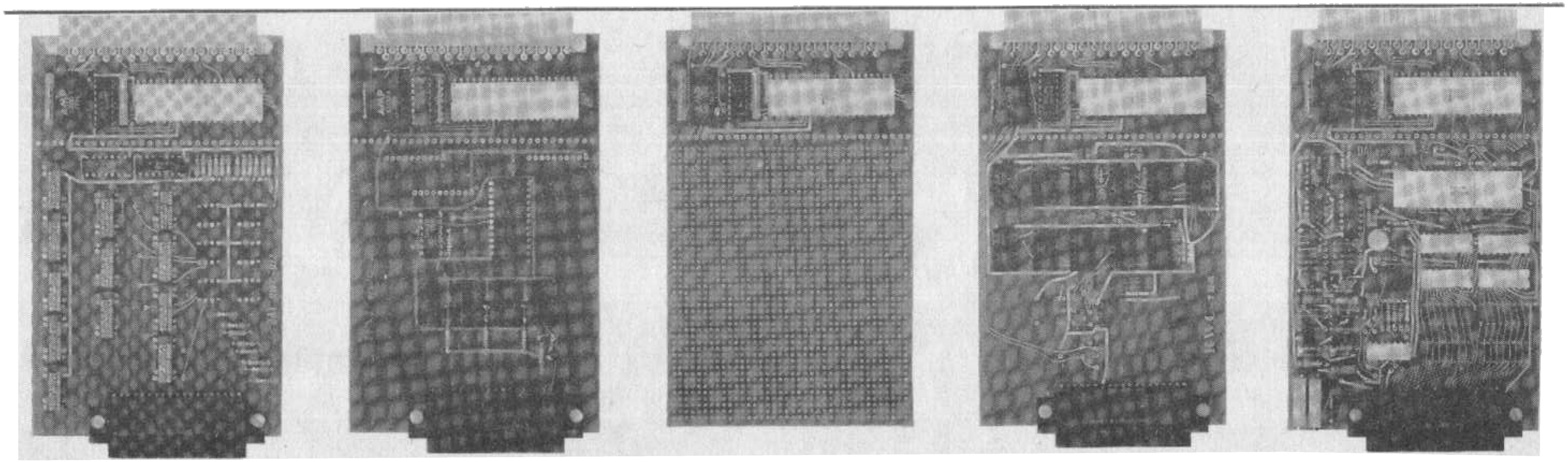
- microcomputer modules
- memory expansion modules
- real-world interface modules



COMPATIBLE

- bus compatible
- package compatible
- software compatible

DCE-BUS



into the Real World!!

- TABLE OF CONTENTS -

INTRODUCTION

Large scale integration technology has produced microprocessors with an assortment of input-output circuits, peripheral devices and memory components. They form a family of versatile logic elements whose operational characteristics can be defined simply by means of a software program stored in memory. This feature identified microcomputers as programmable logic devices, capable of performing a wide range of measuring, monitoring, controlling and data processing functions. They promise to be a new tool in the cost-effective realization of many dedicated control and process applications, which were not economically realizable prviously using conventional digital computing elements.

The low cost of microprocessors and support devices is very misleading due to the high priced engineering and programming effort needed to integrate them into final operational systems. The design of a special microcomputer for a particular application is completely uneconomic in small volumes. To enable economic usage of microprocessors in low volumes, DAI manufactures a complete family of standard microcomputer and interface modules. With these fully-tested modules a user can assemble a complete operational system in a short time.

DAI microcomputers and 'Real-World' interface cards provide a set of easy-to-use hardware building blocks. They enable every user to benefit from the economic and technical advantages offered by microprocessors, with minimum involvement in their technology.

DAI workshop programs teach all the skills necessary to implement operational systems using DAI microcomputer and interface modules. Complete newcomers to digital computer and programming concepts find they can complete working systems during the course. These 3-day workshops are arranged regularly by each local DAI representative, and include comprehensive hands-on experience.

1.1 THE DCE PHILOSOPHY

The DCE (digital control element) family of microcomputers and "Real-World" interface cards were designed to free the designer as much as possible from the problems of hardware design and development, product testing, packaging requirements and low volume component costs. They enable designers, who may not necessarily be computer design experts, to reap the benefits of this new technology even in low and medium volume applications. Being totally independent of the commercial interests of microprocessor manufacturers while working in close co-operation with them, has enabled DAI to freely select the best LSI components in the market, and design the DCE product family for optimum cost-performance.

The DCE product family provides the designer with a comprehensive set of tested hardware, software and package compatible eurocard modules, which can be easily assembled to realize any real-life system. Since all of the hardware modules are burned in and fully logic tested before leaving the factory, the designer does not have to invest in expensive test instrumentation. The cost of system maintenance and eventual troubleshooting is also reduced simply to module exchanging.

All DCE microcomputers and Real-World interface modules have three features in common:

- single eurocard format
- software compatible
- package compatible

DCE microcomputer compatibility means that once any DCE is designed into a particular application, it can easily be updated by another DCE. When additional requirements or changes are needed, they can usually be satified with software changes alone.

1.1.1 DCE Microcomputer Cards

The DCE single eurocard (100 x 160 mm) microcomputers provide the digital computing power of popular 8 or 16 bit microprocessors enhanced by fully implemented serial and parallel I/O capability, program and data memory, independent interval timers and fully vectored interrupts. The serial communication interface is opto-isolated, has programmable baud rates, and can be interrupt driven. The parallel I/O lines can be software programmed as input, output, bi-directional or handshaking ports, with automatic generation of handshake control signals.

The DCE-1 and DCE-2 microcomputers have sockets for 1K (2708) EPROMs, while DCE-1A and DCE-2A have sockets for 2K (2716) EPROMs. The memory expandable DCE-X along with memory expansion cards provide up to 64K memory capacity. These microcomputers are all based on the 8-bit 8080 microprocessor, and are software compatible. The DCE-X86 microcomputer based on the 16-bit 8086 microprocessor has sockets for 2K (2716) EPROMs, and is memory expandable up to 64K words.

1.1.2 Real-World Card Concept

A "Real-World" card is a connection between the digital world of the DCE microcomputers and the real-world of analogue voltages, heavy currents, noisy industrial signals, data communication, contact closures and other real-life phenomena. The designer simply inserts these RWC cards in any combination up to 15 along with a DCE card and a power supply module into the parallel wired eurocard rack or box. Each RWC has a hexadecimal switch for address setting, and the DCE communicates with them via the DCE-BUS using the Real-World software subroutines provided.

The number of RWC cards available has steadily grown to meet the demands of almost every possible application.

1.1.3 DCE Development Tools

By following a new design philosophy, DAI has eliminated the need for expensive dedicated development systems and emulators for developing and debugging system software and hardware.

By utilizing the Bus Monitor Card, PROM Programmer Card, PROM resident software development assistance modules and a software package, the designer can easily turn the euro-rack with DCE and RWC cards into a low-cost development system with advanced features.
He then has powerful debugging tools for testing the software, as well as the software/hardware interfaces in his system.

With the aid of the Utility package the designer can insert his program and interrupt routines into RAM, display and modify the register and memory contents, trace the program while running, perform I/O operations, program and verify EPROMs.

BASIC and FORTRAN packages extend all the advantages of high-level programming languages into DCE systems at a minimum cost.

1.2 GENERAL DESCRIPTION

The user can consider the DCE as a logical building block which he connects to his peripheral devices and circuits. The control function of this logic block is determined by a sequence of instructions (contained in the PROM memory of the DCE processor card) into which the system function has been encoded. Accordingly, it produces output signals to drive user-connected peripheral devices and circuitry.

Between the input from and the output to a peripheral device, there is the process phase during which logical operations, calculations, comparisons, analysis, and data exchange between memory and peripherals can take place. Due to its interrupt handling capability , the DCE can perform multiple real-time tasks simultaneously; for example, message transfer and main program execution.

1.2.1 DCE I/O Architecture

DCE microcomputers contain two powerful LSI subsystems to permit the designer to handle complicated serial and parallel I/O operations with simple instructions:

* The General Interface Control (GIC) provides 24 programmable parallel I/O data lines.

* Timer Interrupt and Communications Control (TICC) provides 5 interval timers, 8 level vectored priority interrupt control, serial data communications with programmable baud rates and 2 parallel data ports.

All input and output ports of the DCE have memory addresses and can be treated as simple memory locations. The DCE-BUS concept allows simple connection to a wide range of external devices via the Real-World interface cards.

1.2.2 General Interface Control (GIC)

The GIC provides 24 parallel input/output lines through three 8-bit general purpose interface ports (Port 0, Port 1, Port 2) as shown on the block diagram in Figure 1-1. These ports can be software programmed independently to serve as input, output. bi-directional

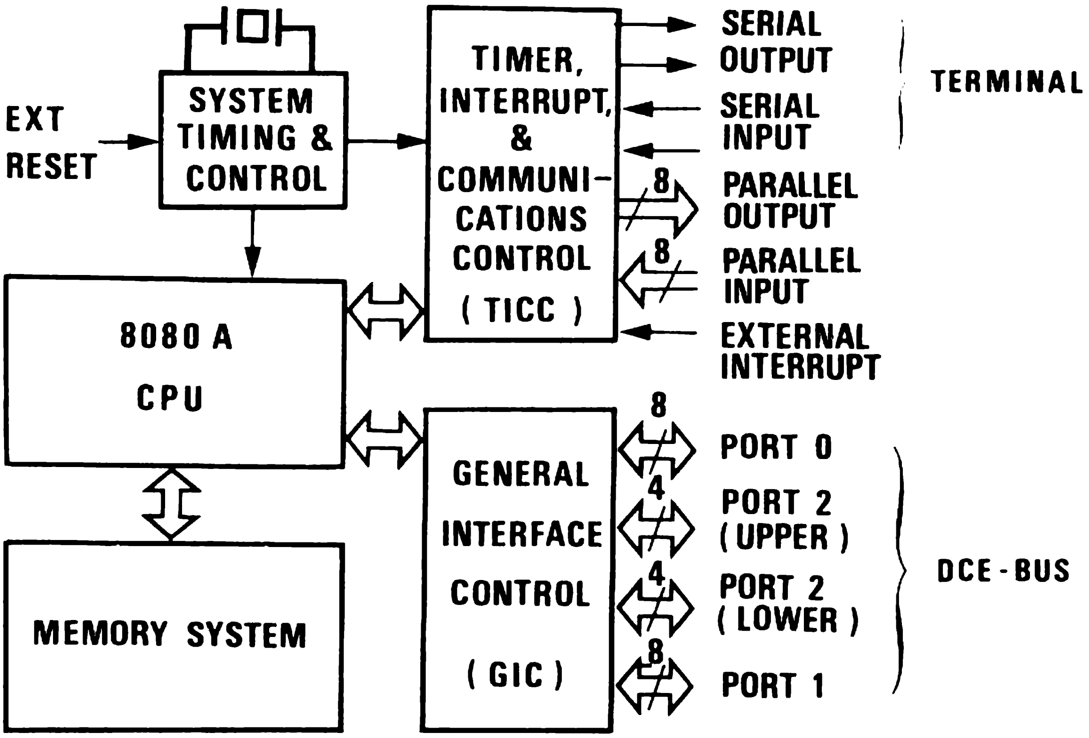


Figure 1-1. General DCE Processor Block Diagram for 8-bit Microcomputers

or handshaking with automatic generation of handshake control signals. Through selected bits or groups of bits of these ports the user can input or output data directly to or from the CPU.

When the DCE microcomputer is used with RWC interface cards, the GIC I/O lines are configured in a special way to implement the DCE-BUS.

There are three modes of I/O operation:

1. Simple Input-Output

   In this mode data is simply written to or read from a specified port; no handshaking is required. While the output in these modes are latched, the inputs are not.

2. Handshaking Input-Output

   The following handshake control signals are automatically generated and processed by the GIC.

   Input Buffer Full
   Output Buffer Full
   Data Strobe
   Data Acknowledge
   Interrupt Request

   While ports 0 and 1 are devoted to Data I/O, selected bits of port 2 pass the handshake control signals. No program steps are necessary to generate, process, or coordinate the handshaking signals.

3. Bi-directional Input-Output

This mode, available for port 0 only, provides the facilities for transmitting or receiving information to or from a peripheral device through a single 8-bit bus. Handshaking signals maintain the proper bus discipline. In this mode inputs and outputs are latched independently from each other.

1.2.3 DCE-BUS Concept

The DCE-BUS provides a common transfer medium for the exchange of data and control information between DCE-BUS compatible processor and interface modules.

It is usually driven by the GIC on the master DCE processor, configured to provide 8-bit input/output data transfer, read and write control signals, two external interrupt requests and eight address control lines for selective access of modules on the bus. Each interface module is plugged into the DCE-BUS through the system connector and given a unique address via a hexadecimal switch on the module. Slave DCE processors are given software defineable addresses. The DCE-BUS can also be hardware driven if necessary for faster operation.

Read and Write subroutines are provided to simplify the transfer of data between DCE processor cards and Real-World interface cards via the DCE-BUS. The designer therefore has a simple means of providing information transfer between many logically and physically separated interface functions, via the master processor. For example, data from an anlogue input channel could be used to control an output device connected to a different interface card.

The DCE-BUS provides the facility to simultaneously connect upto 30 bus compatible Real-World modules and a large number of slave DCE processor modules to a single master DCE processor module. Bus usage between the different processor modules is determined by system software controlled by the master. The DCE-BUS supports the normal software driven bus exchanges, as well as the more powerful high-speed bus transfers generated by hardware.

1.2.4. Timer, Interrupt and Communications Control (TICC)

The TICC provides (in addition to the three 8-bit ports of the GIC) an 8-bit parallel input and an 8-bit parallel output port, 5 interval timers, 8 level vectored priority interrupt control and serial data I/O with programmable Baud rates.

The opto-isolated serial channel is software programmable for Baud rates of 110, 150, 300, 1200, 2400, 4800 and 9600, totally under the control of TICC. This implies that during serial data transmission or reception the CPU can continue the control function to which it is dedicated. At the end of transmission or reception of a character, the TICC interrupts the CPU. The program will then branch to a respective vector routine that handles data transfer between I/O buffers and memory. This feature makes the DCE ideal for real-time message transfer and simultaneous control work.

In addition, the five interval timers of the TICC make the DCE adaptable for simultaneous execution of five control functions at various priority levels. These timers are 8-bit counters. The time interval they provide, vary from 64 to 16,320 micro-seconds (for longer intervals the programmer must cascade timers). With a single instruction, the program loads a timer with a number corresponding to the desired time interval. Loading the timer activates it, and it starts counting down in decrements of 64 micro-seconds. When the count reaches zero an interrupt causes the program to branch to a service routine. In the place of TIMER No. 5 an auxiliary external interrupt input can be selected via a software switch.

Another interrupt-generated program branch can be induced via the external interrupt input.

The eight vectored interrupt sources provided by the TICC have different priority levels, and each can be enabled or disabled individually by programming bits in the interrupt mask register.

1.2.5 Programming

The 8-bit DCE microcomputers are software compatible with the instruction set of 8080 microprocessor, while the 16-bit DCE microcomputer is based on the 8086 CPU. Sockets for erasable PROM memory are provided on the microcomputer card or on a separate memory card, depending on the type of DCE in use. The RAM memory, supplied on the cards, allows storage of intermediate results, I/O buffering, and serves also as the CPU stack memory.

A range of software modules are available to assist the development and testing of software for DCE systems.

1.2.6 Testing

The testing of a complex hardware system such as the DCE can represent a major problem even to the most experienced microcomputer system designer. For this reason, DAI has devoted a considerable amount of effort to the development of a dedicated automatic test system that fully tests each module. The fact that every card is burned in and fully logic tested before leaving the factory represents one of the most profitable advantages of the DCE technology.

1.3 CUSTOMER SUPPORT

DAI was founded in 1971 as an engineering and consulting company to exploit microprocessors. It has grown to the position of a world leading company for microcomputer engineering by designing and implementing a large number of microprocessor based systems. Operational DAI systems cover a very wide range of applications all over the world.

DAI developed the concept of microcomputer workshops, and has trained thousands of engineers in Europe. Every major semiconductor manufacturer has successfully used the DAI workshop program.

The accumulated know-how and unique experience of DAI is made available to the complete spectrum of microprocessor users via DAI Standard Modules, Custom Controllers and Minimum-Cost Devices.

DAI Standard Modules provide a set of hardware building blocks ideal for low-volume applications. They provide a short-cut for the electronic engineer, and a packaged solution for engineers from other disciplines. There are described in detail in this manual.

In medium volumes, a special development to combine all the components necessary for a particular application on a single printed circuit card is more economical. DAI Custom Controllers satisfy this need. DAI accepts complete design and engineering responsibility to deliver a programmed, assembled and tested custom controller card performing to user specifications. These Custom Controllers are ideal for users who want to apply microprocessors with no involvement in their technology.

For high volume applications such as consumer products the component count is critical. In such cases DAI will design and program the system to yield a single-chip microcomputer based product with as few components as possible. A major design effort is devoted to eliminating even the smallest components. In spite of these high design costs, the final unit cost will be very low for large production volumes. DAI Minimum-Cost Devices open up many new and unexpected microcomputer applications in high volumes.

Finally, if a project needs a quick and cost-effective solution, DAI can provide a turn-key solution by supplying the complete microcomputer system with all interface logic and necessary software.

All the products and services of DAI are available locally via a worldwide network of representatives. They regularly conduct DAI microcomputer workshops, and provide local customer support.

2. GENERAL INTERFACE CONTROL (GIC)

The two most important features of the DCE are that it makes optimum use of the available I/O pins and the interrupts are software definable. Therefore, it is possible to provide interface logic for almost any I/O device without the need for additional external logic or components. The multifunction input-output terminals allow the designer to tailor the DCE to the interface characteristics of the connected peripheral devices. This section details the GIC features and explains how to use software to "fit" the DCE to applications with parallel data I/O requirements.

2.1 PARALLEL INPUT-OUTPUT

Section 6. 1.4 details the pin assignments of the 31 pin System Connector. 24 pins of this connector are devoted to parallel data communications through three 8-bit general interface ports: PORT 0, PORT 1, and PORT 2. These ports are general-purpose data channels configured under the direction of the General Interface Control. Through selected bits or groups of bits of these ports, the user can input or output data directly to or from the CPU. Software controls the selection of bits and the modes of I/O according to the possibilities listed in Section 2. 7. 2.

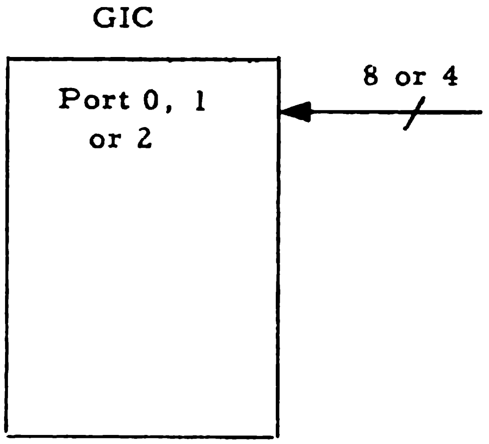
Group A of Section 2.7 . 2 refers to the alternative modes that can be used to input or output data through PORT 0 and the four "upper" bits (4 to 7) of PORT 2. Group B refers to I/O modes through PORT 1 and the four "lower" bits (0 to 3) of PORT 2. A specific configuration from groups A and B can be selected with the instruction of general format: "GICC Amode, Bmode". For example, the instruction "GICC 1,2" sets all 8 bits of PORT 0 in output mode, all 8 bits of PORT 1 in input mode, all 4 "upper" bits of PORT 2 in input mode, and all 4 "lower" bits of PORT 2 in output mode. The "GICC 1,2" in this example is a GIC Configuration Command. The Configuration Commands are usually executed during the initialization of the DCE. GICC is a macro instruction.

A configuration command can be followed by a compatible GIC I/O command listed in Section 2.7.3. For example, if the macro command "STGI 0" follows the "GICC 1,2" instruction, the GIC will write the 8-bit word in the 8080 accumulator to PORT 0. After a Configuration Command several I/O Commands may occur in the program provided they are compatible with the configuration established for the specific ports.

In modes 0 through 3 in both groups the input and output operations are simple: data is simply written to or read from a specified port; no handshaking is required. While the outputs in these modes are latched, the inputs are not. Data written to an output port is latched by the port and may be read back with the "LDGI" instruction (see Section 2.7.3) as if it were stored in RAM memory location. Note also, that in these modes PORT 2 is used simply to transfer data to or from peripheral devices just like ports 0 and 1. Figures 2-1 and 2-2 show the simple input and output modes.

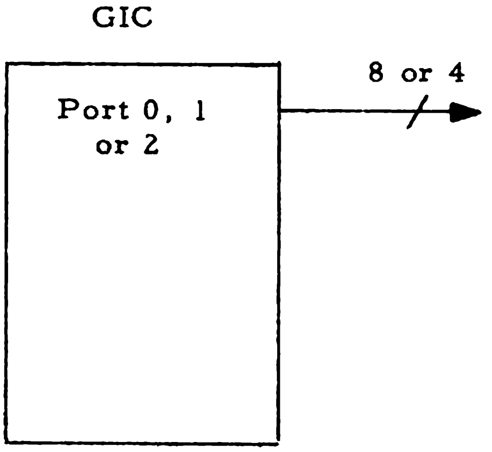
Output data latched in the GIC ports can be handled as if they were stored in memory locations since each port has a specific address as shown on the Memory and Register Diagram in figure 6-2. For example, if the H and Lregister pair of the 8080 holds the address 1C00H of the PORT 0, then the "INR M" instruction will increment the data held in PORT 0. Thus a counter can be set up directly in the GIC port referenced by the H and L register pair.

When a Configuration Command specifies a handshake mode (H.S.), the handshake control signals(H.C.S.) pass through specified bits of PORT 2 as indicated in the following Sections. The input handshake control signals are defined in Figure 2-3 and in Section 2.2, and the output handshake signals are defined in Figure2-4 and in Section 2.3. PORT 0 handshake control signals are provided by PORT 2 upper bits, while PORT 1 handshake control signals are provided by PORT 2 lower bits.



Data at input port transferred to accumulator during LDGI instruction (no latching)

Figure 2-1. GIC Port Function: Input Mode



Data in accumulator latched in output port during STGI instruction.

Figure 2-2. GIC Port Function: Output Mode

2.2 INPUT HANDSHAKE CONTROL SIGNALS (see figure 2.3)

$\overline{\text{STB}}$ (Strobe Input)

Apply to PORT 2 bit 4 for input data at PORT 0, and
to PORT 2 bit 2 for input data at PORT 1.

A "low" on this input loads data into the input latch.

IBF (Input Buffer Full)

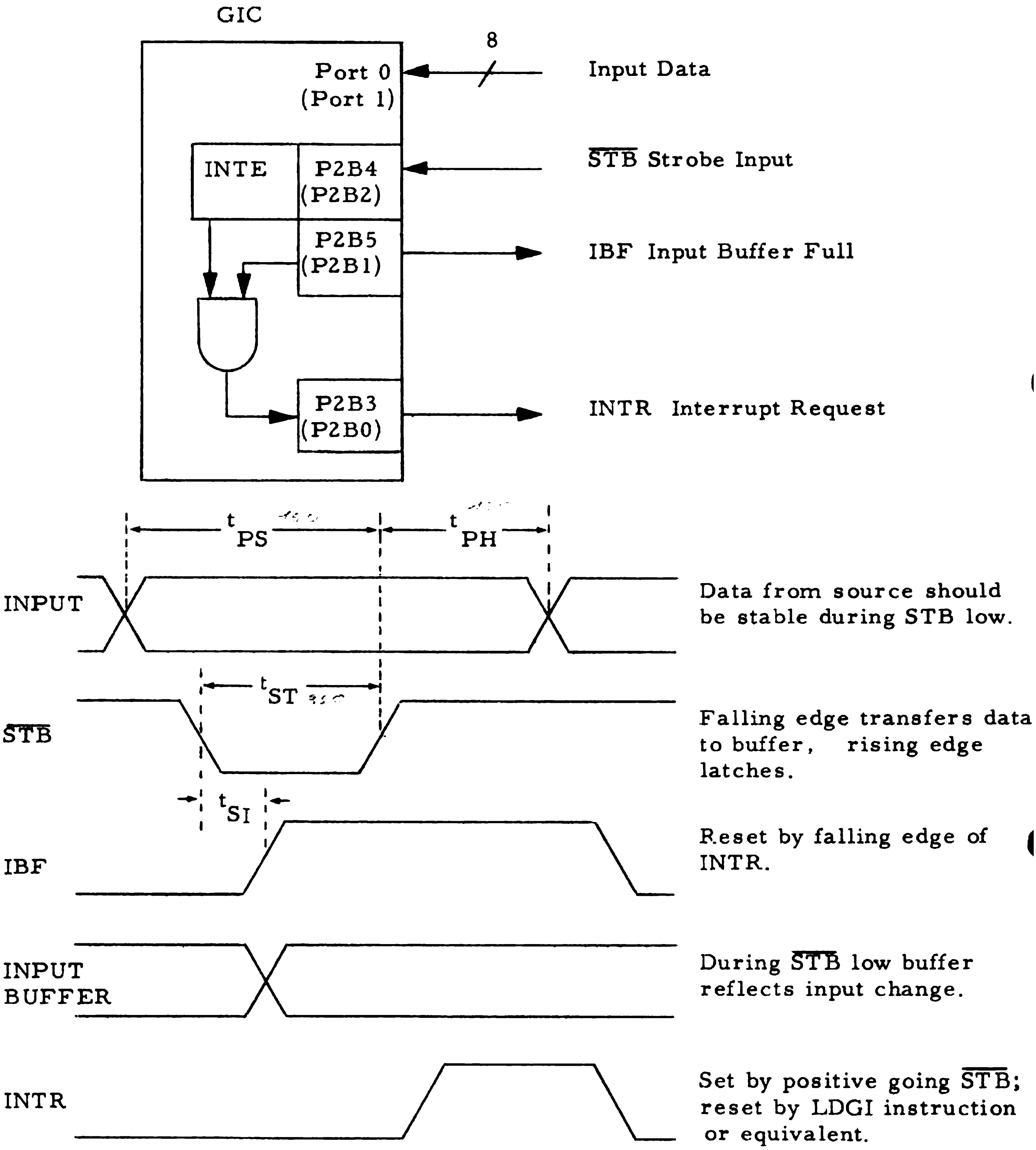
Appears on PORT 2 bit 5 for input data at PORT 0, and
on PORT 2 bit 1 for input data at PORT 1.

A "high" on this output acknowledges to the data source that the data has been loaded into the input latch and no new data should be sent until this output returns to low.

INTR (Interrupt Request)

Appears on PORT 2 bit 3 for input data at PORT 0, and
on PORT 2 bit 0 for input data at PORT 1.

A "high" on this output indicates to program that an input device has strobed its data into the port. Thus, the programmer can allow a specific input device to interrupt the CPU by simply strobing data into the port. To achieve this condition he must connect the INTR output to the EXINTR input, or to the IN7 (auxiliary interrupt) input on the device or system connector. Having the INTR output hard wired to one of the external interrupts, the programmer can still enable or inhibit interrupts with the program. The INTR output indicating data at PORT 0, is blocked if bit 4 of PORT 2 is at logical ZERO, and it is passed if this bit is at logical ONE. Therefore, by setting bit 4 of PORT 2 "high" with the instruction "BSET 4" the interrupt condition is enabled. Clearing this bit with the instruction "BCLR 4" disables the interrupt condition. (The instructions "BSET 2" and "BCLR 2" enable or inhibit interrupts with respect to data at PORT 1). These bits can be written to, even though they are used for the input strobe $\overline{\text{STB}}$. The interrupt modes are also disabled by a new GIC configuration command.



Data from source should be stable during STB low.

Falling edge transfers data to buffer, rising edge latches.

Reset by falling edge of INTR.

During $\overline{STB}$ low buffer reflects input change.

Set by positive going $\overline{STB}$; reset by LDGI instruction or equivalent.

NOTE: For time parameter values see Section 2.6.2

Figure 2-3 Port Function: Handshaking Input Mode (Latching)

2.3 OUTPUT HANDSHAKE CONTROL SIGNALS (see figure 2-4)

$\overline{\text{OBF}}$ (Output Buffer Full)

Appears on PORT 2 bit 7 for output data at PORT 0, and on PORT 2 bit 1 for output data at PORT 1.

A "low" condition on this output indicates that data has been transferred from the 8080 CPU to the output buffer. The falling edge of the $\overline{\text{ACK}}$ input signal resets this output to the high condition.

$\overline{\text{ACK}}$ (Acknowledge Data Received)

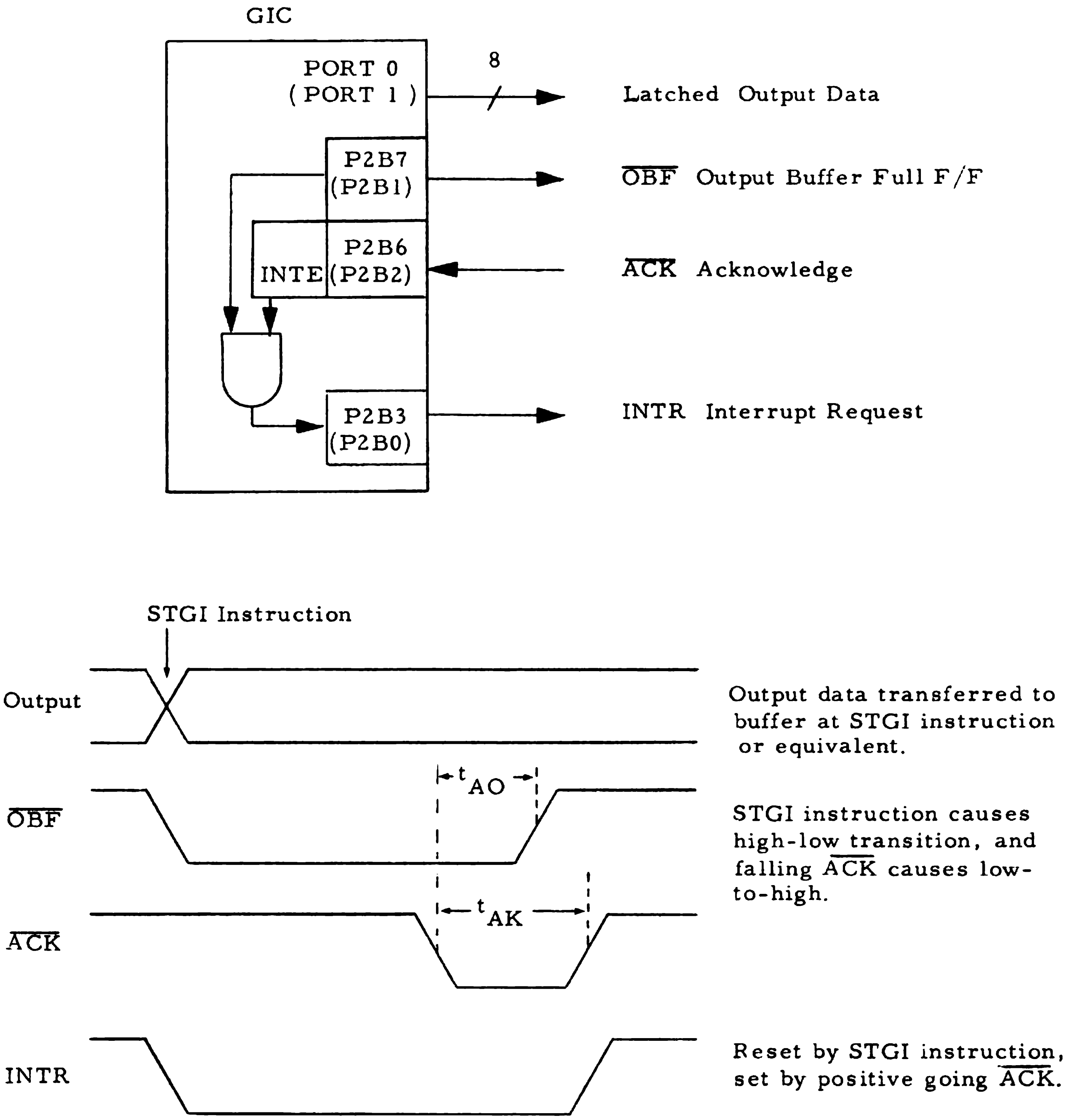
Apply this input to PORT 2 bit 6 for data received from PORT 0 and to PORT 2 bit 2 for data received from PORT 1.

A "low" on this input indicates to the DCE that the peripheral device accepted the data from PORT 0 or PORT 1.

INTR (Interrupt Request)

Appears on PORT 2 bit 3 for output data from PORT 0, and on PORT 2 bit 0 for output data from PORT 1.

A "high" on this output indicates that a peripheral device has accepted data from the port. Thus, the programmer can allow a peripheral device to interrupt the CPU by simply accepting data from the port. To achieve this condition, he must connect the INTR output to the EXINTR input, or to the IN7 (auxiliary interrupt) input on the device or system connector. Having the INTR output hard wired to one of the external interrupts, the programmer can still enable or inhibit interrupts with the program. The INTR output, indicating data taken from PORT 0, is blocked if bit 6 of PORT 2 is at logical ZERO, and it is passed if this bit is at logical ONE. Therefore, by setting bit 6 of PORT 2 "high" with the instruction "BSET 6" the interrupt condition is allowed. Clearing this bit with the instruction "BCLR 6" disables the interrupt condition. (The instructions "BSET 2" and "BCLR 2" enable or inhibit interrupts with respect to data taken from PORT 1). These bits can be written to, even though they are used for input strobe $\overline{\text{ACK}}$. The interrupt modes are also disabled by a new GIC configuration command.


GIC
PORT 0
(PORT 1)
8
Latched Output Data
P2B7
(P2B1)
OBF Output Buffer Full F/F
P2B6
INTE
(P2B2)
ACK Acknowledge
P2B3
(P2B0)
INTR Interrupt Request
STGI Instruction
Output
Output data transferred to buffer at STGI instruction or equivalent.
OBF
t_{AO}
STGI instruction causes high-low transition, and falling ACK causes low-to-high.
ACK
t_{AK}
INTR
Reset by STGI instruction, set by positive going ACK.

NOTE: For time parameter values see Section 2.4.2

Figure 2-4 GIC Port Function: Handshaking Output Mode

There are several combination of modes where not all of the bits in PORT 2 are used for handshake control signals. These bits, if configured as inputs, will transfer data to the CPU when "LDGI 2" or equivalent is executed. However, if they are programmed as outputs, the 4 "upper" bits must be individually accessed using the single-bit set/clear instructions (see Section 2. 7. 1); while the four lower bits (0 to 3) can be set or reset individually or accessed together by the "STGI" command or equivalent.

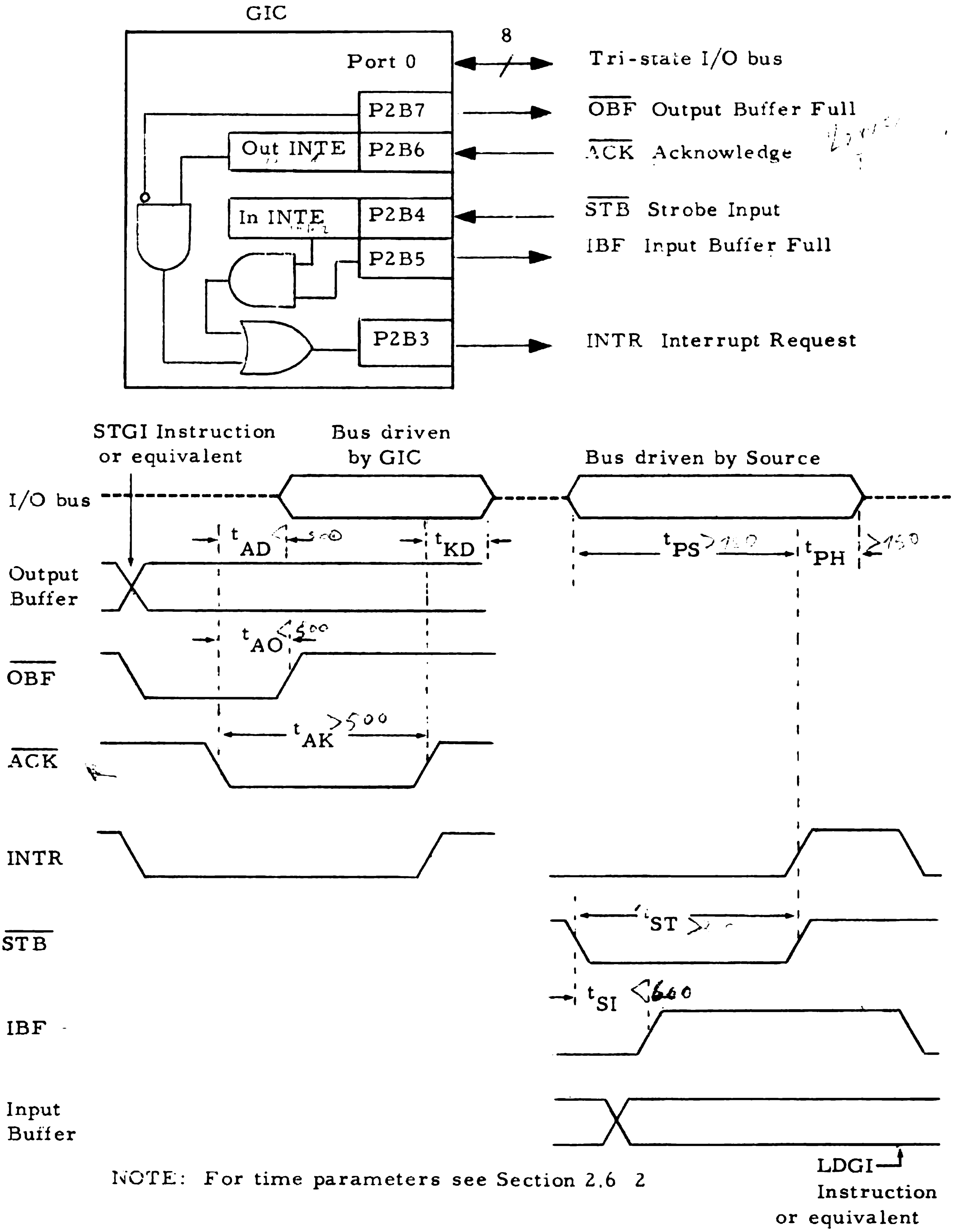
The individual set/clear capability of PORT 2 can control simple switch closing or display computational results. The DCE can also drive directly darlingtons for high voltage displays through any set of eight output buffers selected from PORTS 1 and 2; these buffers can source 1mA at 1. 5 volt as required by this type of load.

2.4 BI-DIRECTIONAL I/O MODE (see figure 2-5)

MODE 8 defined in Section 2. 7. 2 specifies a bi-directional I/O mode. This mode is available for PORT 0 only; it provides the facilities for transmitting and receiving information to and from a peripheral device through a single 8-bit bus. Handshaking signals on bits 3, 4, 5, 6 and 7 of PORT 2 maintain the proper bus discipline. In this mode PORT 0 inputs and outputs are latched independently from each other.

The input and output handshake signals for Mode 8 are the same as described above with the exception of the $\overline{\text{ACK}}$ input. A "low" on this input puts the PORT 0 3-state output buffer in the low impedence state, enabling it to send out the data. Normally, in Mode 8, PORT 0 is in the high impedence state. See Figure 2-5 for the definition of this port function.

There are additional 8-bit parallel I/O ports on the Device Connector. These come from the TICC, and are described in Section 4.



NOTE: For time parameters see Section 2.6 2

Figure 2-5 GIC Port Function: Bi-Directional I/O Mode

2.5 SYNCHRONIZED DATA TRANSFER

The memory and register diagram of Figure 6-2 shows two addresses for PORT 0. The address 5C00 allows fast, burst-mode data transfer between the DCE and peripheral devices such as floppy disc, magnetic tape, or mass storage devices. Fast data transfer, usually performed with peripheral hardware such as DMA controller, is possible to accomplish in the DCE with the user program.

The following example illustrates how to use the synchronized mode for fast data transfer to a 256 byte memory page.

```
      LXI  H, BUFFER   ;  Point to RAM MEMORY BUFFER
      LXI  D, GICS     ;  Point to PORT 0 sync mode  (5C00)
LOOP: LDAX D           ;  Read PORT 0 sync mode
      MOV  M, A        ;  Transfer data to memory
      INR  L           ;  Increment memory block address
      JNZ  LOOP        ;  Loop until end of memory block
                          (register L content is zero)
```

In this program the LDAX instruction waits for the data to arrive to PORT 0; this condition is indicated by the rising edge of the $\overline{\text{STB}}$ pulse. Without the sync-mode port address in the register pair D, the LDAX instruction would not wait, and the CPU would be required to spend extra time to interrogate the port for the presence of the new data. In such a case the transfer of a byte from PORT 0 to RAM would take 26.5 microseconds. In the synchronous mode the software loop above takes 14.5 microseconds to complete, allowing a transfer rate of 70,000 bytes per second.

2.6 GIC PORT CHARACTERISTICS

2.6.1 D.C. Characteristics T_A = 0°C to 70°C

Symbol	Parameter	Min.	Typ.	Max.	Unit	Cond.
V_{IL}	Input Low Voltage			.8	V	
V_{IH}	Input High Voltage	2.0			V	
V_{OL}	Output Low Voltage			.4	V	Note1
V_{OH}	Output High Voltage	2.4			V	Note2
$I_{OH}(1)$	Darlington Drive Current		2.0		mA	Note3

Note 1: I_{OL} = 1.6mA

Note 2: I_{OH} = 50 μA

Note 3: V_{OH} = 1.5V, R_{EXT} = 390 Ω

2.6.2 A.C. Characteristics T_A = 0°C to 70°C

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{AK}	Width of ACK Pulse	500			ns
t_{ST}	Width of STB Pulse	350			ns
t_{PS}	Set-Up Time for Peripheral	150			ns
t_{PH}	Hold Time for Peripheral	150			ns
t_{RA}	Hold Time for A_1, A_0 after RD = 1	379			ns
t_{RC}	Hold Time for CS after RD = 1	5			ns
t_{AD}	Time from ACK = 0 to Output (Mode 2)			500	ns
t_{KD}	Time from ACK = 1 to Output Floating			300	ns
t_{WO}	Time from WR = 1 to OBF = 0			300	ns
t_{AO}	Time from ACK = 0 to OBF = 1			500	ns
t_{SI}	Time from STB = 0 to IBF			600	ns
t_{RI}	Time from RD = 1 to IBF = 0			300	ns

2.7 GIC COMMANDS

2.7.1 Port 2 Bit Set/Clear Commands

Macro Instruction

BCLR	nbit	clears specified bit in PORT 2
BSET	nbit	sets specified bit in PORT 2

nbit = 0, 1, 2, 3, 4, 5, 6, 7

2.7.2 GIC Configuration Command

GICC amode, bmode

GROUP A MODE	PORT 0	PORT 2 (Bits affected)
0	Output	Output (4-7)
1	Output	Input (4-7)
2	Input	Output (4-7)
3	Input	Input (4-7)
4	H. S. Output	H. S. C. (3, 6, 7) Output (4, 5)
5	H. S. Output	H. S. C. (3, 6, 7) Input (4, 5)
6	H. S. Input	H. S. C. (3, 4, 5) Output (6, 7)
7	H. S. Input	H. S. C. (3, 4, 5) Input (6, 7)
8	Bi-directional	H. S. C. (3, 4, 5, 6, 7)

GROUP B MODE	PORT 1	PORT 2 (Bits affected)
0	Output	Output (0-3) +
1	Output	Input (0-3) +
2	Input	Output (0-3) +
3	Input	Input (0-3) +
4	H. S. Output	H. S. C. (0, 1, 2)
6	H. S. Input	H. S. C. (0, 1, 2)

+ Bit 3 not affected if Group A in modes 4 through 8

In the above "Handshake" = H. S.
"Handshake Control" = H. S. C.

2.7.3 GIC I/O Commands

MACRO INSTRUCTION	DESCRIPTION	FUNCTION
LDGI 0	Parallel read of PORT 0	(A)←(PORT 0)
STGI 0	Parallel write to PORT 0	(PORT 0)←(A)
LDGIS 0	Synchronized read of PORT 0	(A)←(PORT 0) *
STGIS 0	Synchronized write to PORT 0	(PORT 0)←(A) *
LDGI 1	Parallel read of PORT 1	(A)←(PORT 1)
STGI 1	Parallel write to PORT 1	(PORT 1)←(A)
LDGI 2	Parallel read of PORT 2	(A)←(PORT 2)
STGI 2	Parallel write to PORT 2	(PORT 2)←(A) **

*Transfer synchronized with the INTR signal low-to-high transition on P2B3 (see section 2.5).

**If Group A is in mode 4 or 6, bits 4-7 can only be changed by direct bit-set or bit-clear commands.

EXAMPLE

To configure Group A in mode 7 and Group B in 0

```
GICC 7, 0
```

After this command PORT 0 will be configured as handshaking input port, with the handshake control signals passing through bits 3, 4, and 5, of PORT 2, as described in Section 2. 2. Bits 6 and 7 of PORT 2 will be configured as simple input. PORT 1 and bits 0 to 3 of PORT 2 will be configured as simple outputs.

NOTE: The Macro instructions BCLR, BSET and GICC are all 5 bytes long and modify the accumulator.

3. TIMER, INTERRUPT AND COMMUNICATIONS CONTROL (TICC)

In addition to the general purpose I/O ports of the system connector, there is an 8-bit parallel input port and a separate 8-bit parallel latching output port available on the 25-pin DEVICE CONNECTOR (see Section 6. 1.5). Data transfer between these ports and the 8080 CPU is under the control of the Timer, Interrupt and Communication Control, TICC; hence these ports are referred to as TICC I/O ports. An 8-bit word present at the TICC input port can be transferred into the accumulator with the "LDIN" instruction, while the content of the accumulator can be transferred to the TICC output port with the "STOUT" instruction.
The TICC output port is inverting.
The serial input-output, the interrupt processing and the interval timers are also under the control of the TICC. This section discusses the TICC features and provides the background information that will enable the programmer to use the DCE as a microcomputer with powerful real-time processing capability.

3. 1 SERIAL INPUT-OUTPUT

Serial output data from the DCE pass through the SOPOS, SONEG output pair, and the input serial data to the DCE pass through the SIPOS ans SINEG input pair located at the DEVICE CONNECTOR. Both pairs are optically isolated from the DCE circuits and are compatible with 20mA current loop system. The serial input is conditioned by a schmidt trigger that switches on at 10mA and off at 5mA, providing thus a high degree of noise immunity. The serial output is capable of switching 30mA.

The serial I/O can be connected in a passive mode or in an active mode. The passive mode is used when the external device supplies the current. In this case connection is shown in Figure 3-1.

The active mode is used when the external device does not supply the current which is usually the case with the teletype. For tms case the connection to the teletype is shown on Figure 3-2. Section 3. 1.2 describes the necessary teletype modifications.

The serial receiver register of the TICC detects the start and stop bits of an incoming character and places it into the receiver buffer. At this time the internal "receiver-buffer-loaded" flag goes high indicating to the CPU that the receiver is loaded with a new character. If the interrupt-mask-register does not mask this particuler input off, and if no higher priority interrupts exist at the time, this flag will interrupt the CPU and cause it to enter the "read-receiver-buffer" routine, at the vector address 0020H, which transfers the buffer contents into the accumulator and processes the data. The "receiver-buffer-loaded" flag is reset automatically when the receiver buffer is read

Before serial transmission of output data, the program must send the control word to specify the BAUD rate and number of stop bits desired (see Section 4. 4. 3). The program provides this facility by transferring an 8-bit code from the accumulator to the TICC rate register by the instruction "STCRR". The code in the TICC register defines the BAUD rate and the number of stop bits. Once the BAUD rate is set, the instruction "STXMT" loads the character to be transmitted into the transmit buffer. The serial transmitter register, upon receiving the character from the buffer, generates the "start" and "stop" bits and shifts the data out at the programmed rate, independently of the CPU.

While the transmitter is shifting out the serial data, the CPU can be executing another program function. At the time that the transmit-buffer goes empty and the TICC is ready to receive the next character from the 8080 CPU, it will set the "transmit-buffer-empty" flag "high" and force an interrupt to occur. The program can then load the next character into the TICC transmit buffer and return to execution of the program function that was in progress before transmit-buffer-empty was signalled.

3.1.1 TICC I/O Conditions and Characteristics

PARAMETER	MIN	TYP	MAX	CONDITION
High level input voltage, V_{IH}	3.3v		6v	
Low level input voltage, V_{IL}	-1v		0.8v	
Input current I_I			±10µA	V_I = 0 - 5v
High level output voltage, V_{OH}	3.7v			I_{OH} = 400µA
Low level output voltage, V_{OL}			0.45v	i_{OL} 1.7mA
Interrupt pulse width, t_W	500ns			
Interrupt pulse spacing	500ns			
SERIAL OUT				
Breakdown voltage, $V(BR)_{CEO}$			30v	
Collector leakage, I_{CEO}			100nA	V_{CE} = 10v
Saturation voltage, $V_{CE}(SAT)$		1.2v		
Rise and fall times, t_R, t_F		80µsec		V_{CE} = 10v
SERIAL IN				I_C = 10mA
Forward current, $I_F(MAX)$			20mA	continuous
Reverse voltage, BV_R	3v			
Forward voltage, V_F			1.5v	I_F = 20mA
Turn on current, $I_F(ON)$		10mA	15mA	
Turn off current, $I_F(OFF)$	2mA	5mA		
Rise and fall times, t_R t_F		400ns		

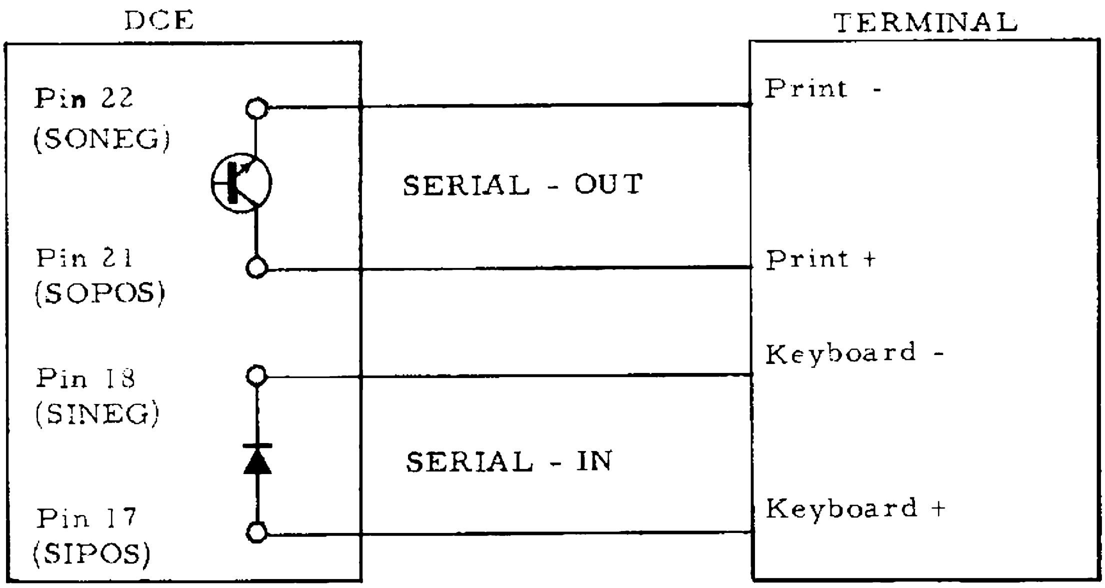


Figure 3-1 Terminal Connection, Passive Mode

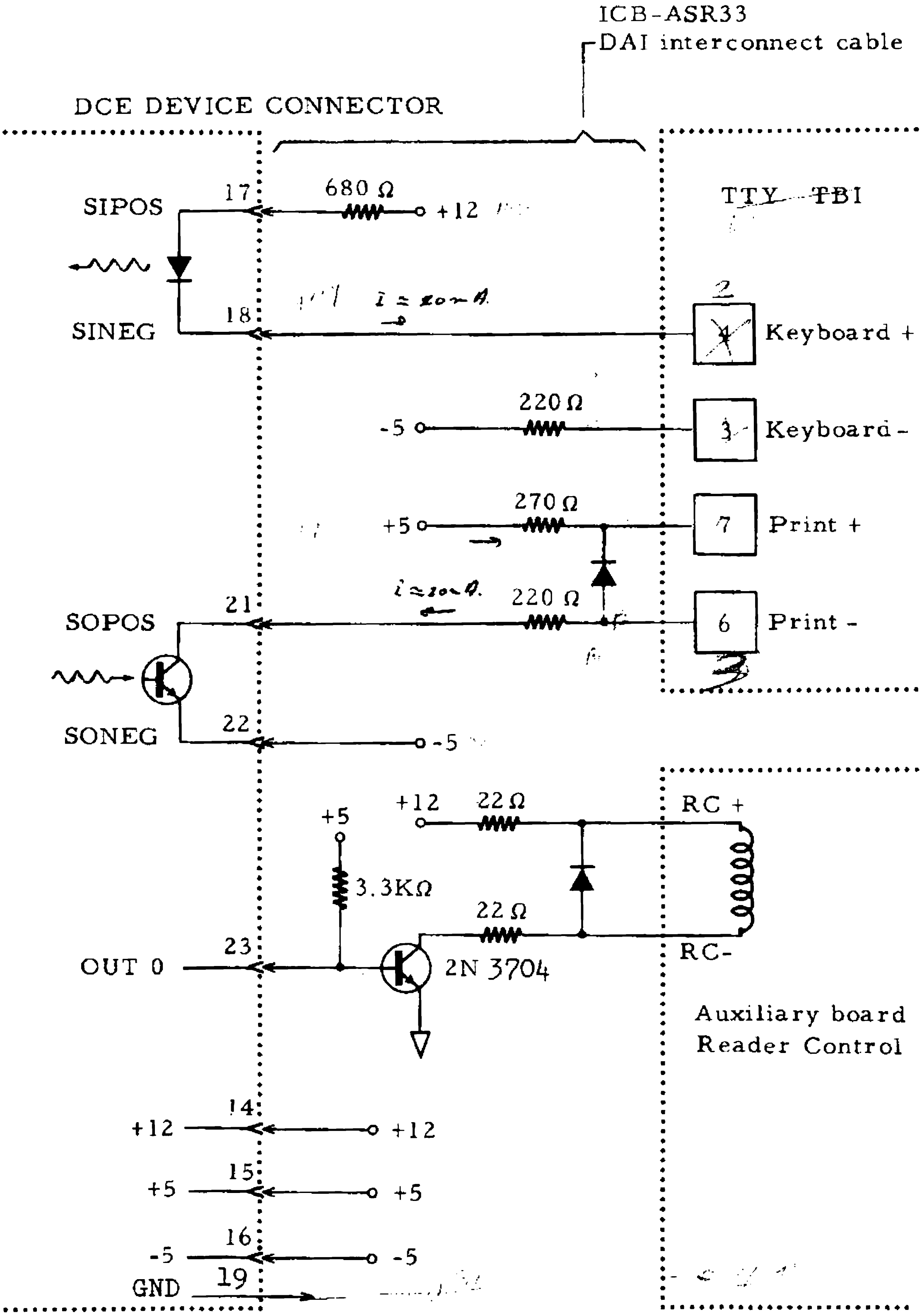


Figure 3-2 Teletype Connection, Active Mode

3.1.2 Teletype Modifications

The 20mA serial I/O of the DCE can be used directly as teletype interface. The user must connect the keyboard and the teleprinter to the DCE, if a 33ASR teletype is used,for communications with the microcomputer. For this connection a slight modification of the MODEL 33ASR is necessary. This modification procedure is detailed in the following two steps:

1, Change the factory-wired 60mA mode to 20mA mode by changing the resistor taps from the existing (factory tapped) at 750ohm position to the end (1450ohm) position in the electrical service unit. Push-on taps are provided, and no soldering is required. Refer to Figure 3-3 on page 3-7 for the location of the resistor. Figure 3-4 shows the location of the taps.

   In addition, a single wire on the barrier terminal strip must be moved at the rear of the electrical service unit. Unscrew the violet wire from terminal 8, and move it to terminal 9. Refer to figures 3-4 and 3-5 on page 3-7.

2, Change the factory-wired half duplex mode to full duplex mode by moving both the blue-white and the brown-yellow wire to terminal number 5. These leads are factory-wired to terminal 4 and 3 respectively. Refer to figures 3-5 and 3-6 on page 3-7.

Certain applications may require the DCE to control the tape reader drive. This can easily be accomplished by installing a simple relay circuit into the electrical service unit of the 33ASR. Realize the relay circuit on a small vector board with a Potter-Brumfield (No. JR-1005) relay, a carbon composition (470 ohm, 0. 5w) resistor, and a (0. 1μF, 200 vdc) capacitor as shown on figure 3-7 on page 3-8. Mount this auxiliary circuit on the vertical metal tab as shown on figure 3-8 on page 3-8.

Figures 3-7 and 3-9 show how to wire the auxiliary board into the electrical service unit. Splice into the brown wire and connect a lead from the splice to one side of the JR-1005 relay. Connect another wire from the splice to the LOCAL position of the MODE switch. Connect the LINE position of the mode switch to the other side of the JR-1005 relay. Identify the LOCAL and LINE terminals referring to figure 3-10. Figure 3-2 on page 3-4 shows how to interconnect the DCE to the modified teletype.

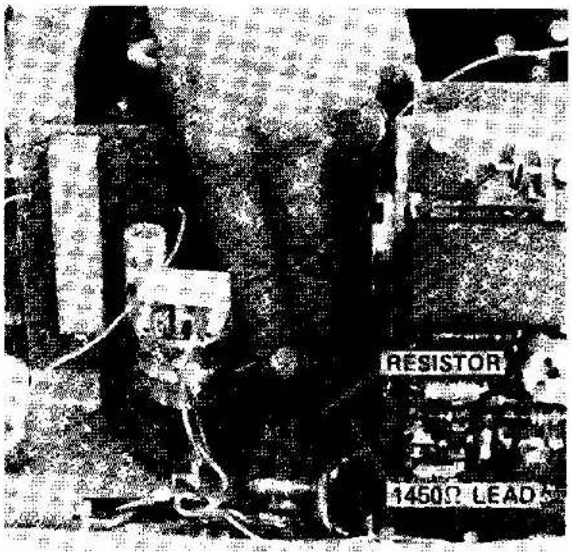


Figure 3-3 Current Source Resistor

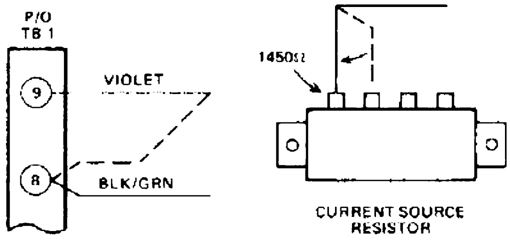


Figure 3-4 TTY: 20mA Wiring

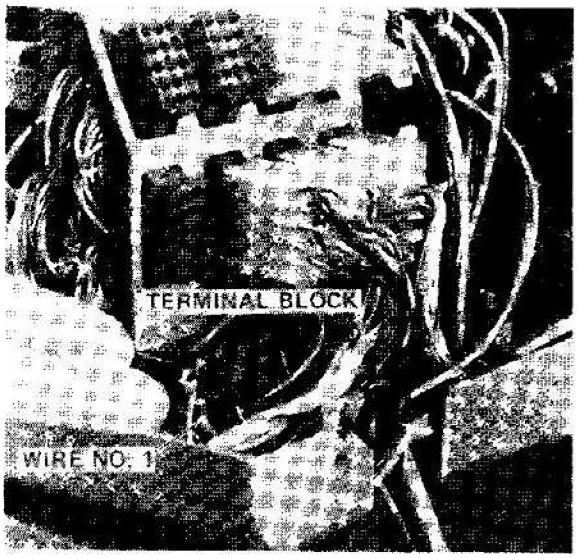


Figure 3-5 Terminal Block Location

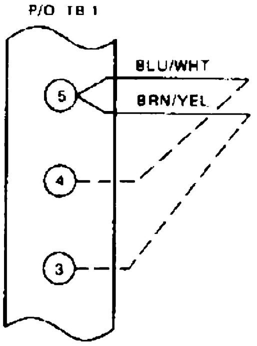


Figure 3-6 TTY: Full-Duplex Wiring

Pictures reprinted from Intel's Intellec 4 reference manual for reference.

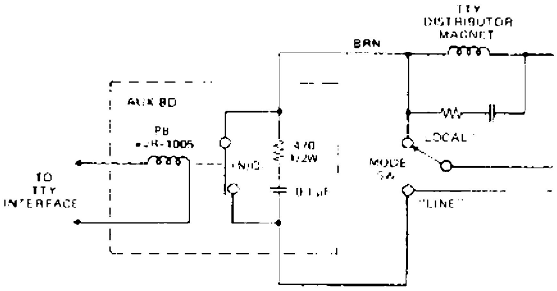


Figure 3-7 Reader Control Auxiliary Board Schematic

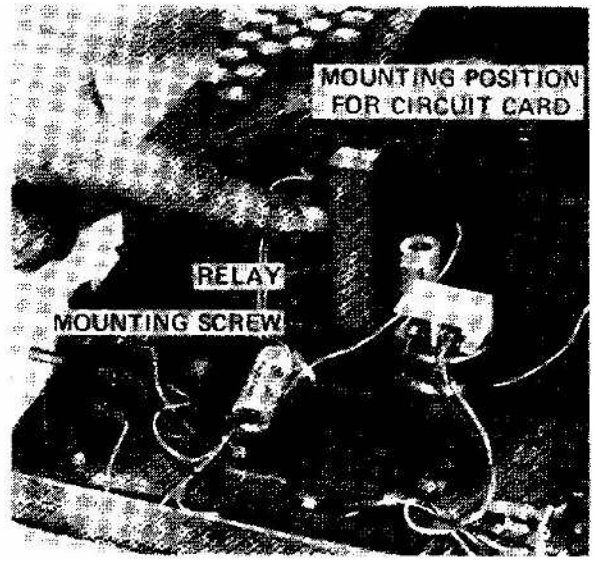


Figure 3-8 Relay Board Location

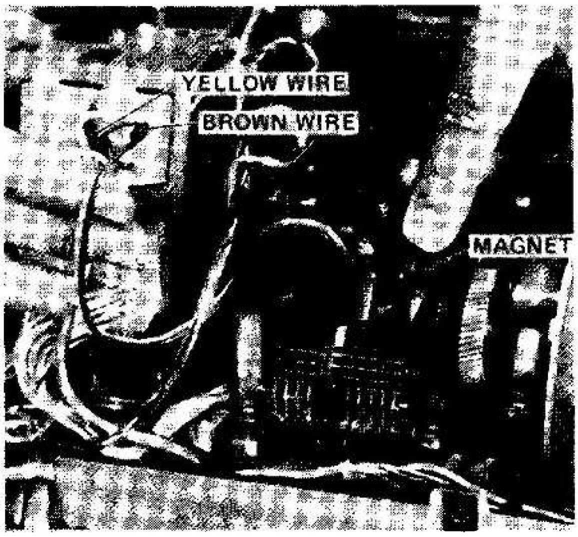


Figure 3-9 ESU Wiring

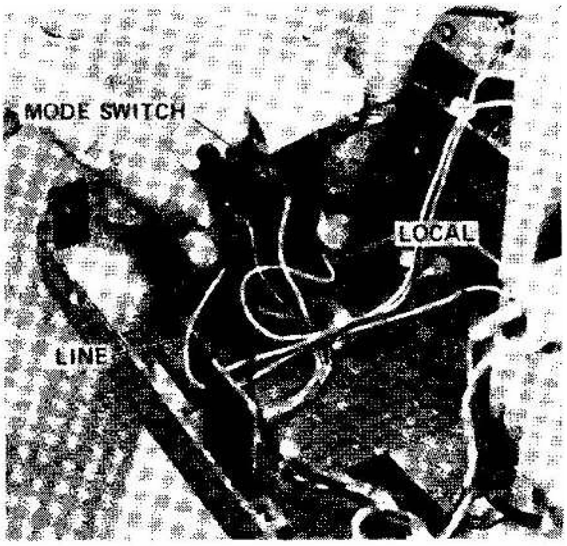


Figure 3-10 Teletype Mode Switch

Pictures reprinted from Intel's Intellec 4 reference manual for reference.

3.2 INTERRUPTS (8-bit DCE)

Interrupt processing ease is one of the most powerful features of the DCE that allows maximum utilization of the CPU. Complex interrupt structures can be established by software that tailors the response of the DCE to customized circumstances under the control of the microcomputer. For example, several peripheral sensors or devices having different priority levels can share DCE execution time. The programmer has the flexibility to use the interrupt control functions to establish priority, mask out undesired interrupts, or generate interrupt requests to deal with high priority conditions. For example, a vectored interrupt is generated when any of the five interval timers count down to zero. If any of the peripheral devices can accept data only at certain time intervals, the program can load one of the timers with the number corresponding to the time interval. While the CPU is processing the main program, the timer counts down. At zero, the CPU is made to branch to the routine that outputs the next data byte, and then it returns to the execution of the main program. Re-loading the timer can be part of the output routine to ensure continuous data output at the desired time intervals.

The programmer can selectively enable desired interrupts by loading a control word into the Interrupt Mask Register from the accumulator with the "STIMR" macro instruction. A logical ONE at bit position 'n' of this register allows interrupt 'n' pass through to interrupt the executing program and branch to the vector address specified in Section 3.2.3. The PROM (or RAM) memory at the address specified must contain the start of the interrupt service routine which saves the CPU registers (if necessary), performs the required processing functions, restores CPU registers, enables CPU interrupts and returns to the previous program. Interrupt requests disabled via the interrupt mask register are also latched in the TICC interrupt register. If the corresponding bit in the interrupt mask register is set subsequently, the interrupt request will become active. All the 8 interrupts are handled by the TICC.

The programmer can choose two different ways to configure the DCE for servicing an interrupt; he can set up an interrupt driven system or a polled interrupt system.

3.2.1 Interrupt Driven System (see Figure 6-1, 6-2)

Conditions: a) CPU interrupts are enabled
b) Bit 3 of TICC Command Register is set
c) Interrupt Mask Register corresponding to interrupt is set.

The following is the sequence of events when an interrupt occurs:

1. The Interrupt Register latches the interrupt condition and the corresponding bit remains set until the interrupt is serviced (i. e. the TICC transfers the corresponding RST instruction to the CPU).
2. TICC sets bit 5 of TICC Status Register high to indicate interrupt pending.
3. TICC sends an INT signal to CPU.
4. CPU acknowledges interrupt with an INTA signal.
5. TICC transfers the RST instruction of the highest pending interrupt to the CPU and clears the corresponding bit in the Interrupt Register.
6. CPU loads the program counter with the RST address and program execution starts at the interrupt service routine.

3.2.2 Polled Interrupt System (see Figure 6-1, 6-2)

Conditions: a) CPU interrupts are disabled
b) Bit 3 of TICC Command Register is cleared
c) Interrupt Mask Register bit corresponding to interrupt is set.

In this system an interrupt condition will not interrupt the CPU.

The following is the sequence of events:

1. The Interrupt Register latches the interrupt condition and the corresponding bit remains set until the CPU reads the interrupt address register.
2. The TICC sets bit 5 of TICC status register high to indicate interrupt pending.
3. TICC sends an INT signal to CPU. The CPU does not acknowledge the INT signal since its interrupt is disabled. (see note)*
4. When the program is ready to accept an interrupt, it must poll the interrupt pending bit of the TICC Status Register (bit 5).
5. If the interrupt pending bit is set, program must read the interrupt address register, where the address of the highest priority pending interrupt is stored. This reading clears the interrupt register bit for that interrupt. When the RST instruction in the interrupt address register is executed, the corresponding service routine is entered (can be done by loading program counter with 9802H via H, L registers).

*NOTE:

If the CPU interrupts are enabled in this mode the CPU will acknowledge the interrupt but the TICC will not release the contents of the interrupt address register. Since the CPU bus in this condition is not driven, it will float high, and the CPU sees an RST 7 (FF) instruction.

An important feature of the Interrupt Register is that it is edge triggered. A low to high transition on the external interrupt, for example, sets bit 2. Servicing the interrupt clears this bit even if the external interrupt input stays at logical ONE. It is, therefore, not necessary to employ external hardware to remove the request.

Regarding point 5 above, it must be noted that the CPU can only read the interrupt Address Register if bit 3 of the TICC Command Register is zero. If this bit is set, any attempt to read the Interrupt Address Register results in a reading of FFH, and pending interrupts are not cleared.

Disable CPU interrupts whenever changing the contents of the TICC interrupt mask register to prevent spurious interrupts due to transients.

The lowest priority interrupt, number 7 (see Figure 6-1) can be generated by either the fifth interval timer or by the auxiliary interrupt IN7 (bit 7 of the TICC parallel input port). The programmer can select Timer 5 or IN7 by setting Bit 2 of the TICC Command Register to zero or one respectively.

The power-on reset and the Timer 1 interrupt both set the program counter to zero. When Timer 1 is being used, the program can determine the source of interrupt 0 by looking at an unused bit of GIC Port 2, for example Bit 4. If this bit is connected to +5v through a resistor, after power-on it will be read as a "one" since all GIC ports will go into the input mode. If the initialization program configures GIC Port 2 Bit 4 as output and sets it to zero, it will remain at zero until the next power-on reset. The program starting at address 0000 can therefore read P2B4 and jump to the initialization routine or Timer 1 service routine accordingly. (note: this technique cannot be used in multiprocessor systems since they use P2B4)

3.2.3 Interrupt Vector Assignments (see Figure 6-2)

INTERRUPT NUMBER	SOURCE	VECTOR ADDRESS (HEX)
0	Reset or Timer 1 has expired	0000
1	Timer 2 has expired	0008
2	EXINTR low to high detected	0010
3	Timer 3 has expired	0018
4	Serial receiver has received a full character	0020
5	Serial transmitter is ready to accept next character	0028
6	Timer 4 has expired	0030
7	Timer 5 has expired or IN7 low to high detected	0038

3.3 INTERVAL TIMERS

The five programmable timers of the DCE provide program controlled time intervals that are used to invoke a subroutine that performs a servicing function for a peripheral device or event that requires periodic recognition. These timers are 8-bit counters, which provide intervals that vary in duration from 64 to 16,320 microseconds. Much longer intervals can be generated by cascading the timers in software.

To start a time count using any of the five timers, the program can load the required interval by an instruction with general format "STTIM n". For example, the instruction "STTIM 2" loads the contents of the accumulator into Timer 2. Loading the timer activates it, and it starts counting down increments of 64 microseconds at the first TICC clock pulse. The first count period can be any length between 0 and 64 microseconds. When the count reaches zero, the bit in the interrupt register corresponding to the timer is set, and an interrupt is generated if CPU interrupts are enabled and that mask bit is set. The timer then remains inactive until a new interval is loaded. Loading an interval value of zero, causes an immediate interrupt. A new value loaded while the interval timer is counting overrides the previous value and the timer starts counting down the new value.

To maintain continued accuracy, a timer has to be reloaded within 64 microseconds after firing. If a series of timed interrupts with crystal accuracy is required, the timer interrupt service routine must reload that timer within 64 microseconds.

Note: None of the TICC command registers used for giving control information or data to the TICC (eg. interrupt mask register) can be read back. Therefore, copies of the contents of these registers should be maintained, if required.

3.4 TICC STATUS AND COMMANDS

The DCE allows the programmer to poll the status of the TICC with respect to external conditions and issue discrete commands to deal with specific circumstances. These actions can be executed via the TICC Status Register, the TICC Command Register, and the Communications Rate Register described in this section.

3.4.1 TICC Status Register (see Figure 6-2)

To query the status of the TICC, the program must transfer the contents of this register to the 8080 CPU and examine its contents bit by bit. The status conditions associated with each bit are described in the following paragraphs and summarized in Sections 3.5.

Bit 0, Framing Error

A logical ONE in this bit position indicates that one or both stop-bits were in error, which implies that the serial data received and held in the receiver buffer is probably erroneous. If this flag is high and the bits of the receiver-buffer-register are all logical ONE, then a break in the serial communications line has occurred. This flag will remain high until the next valid character is received.

Bit 1, Overrun Error

A logical ONE in this bit position indicates that a new character was loaded into the receiver buffer before a previous character was read out by the program. A new read-receiver-buffer command, "LDRCV" or a reset command clears this flag.

Bit 2, Serial Input, Direct Connection

This bit is directly reading the serial input line and permits bypassing the serial receiver register. This feature can be useful if the BAUD rate of the incoming character is not standard. In this case the programmer can handle the non-standard BAUD rate with software and one of the interval timers. Bit 2 can also be used for test purposes or detecting a break; it is normally "high" when no data is being received and line current is flowing.

Bit 3, Receiver Buffer Loaded

A logical ONE in this bit indicates that the receiver buffer is loaded with a new character. A read-receiver-buffer instruction, "LDRCV" or the reset function clears this flag.

Bit 4, Transmitter Buffer Empty

A logical ONE in this bit indicates that the transmitter-buffer is empty. This condition occurs as soon as the buffer register transferred the data to the serial transmitter (which may still be in process of shifting out a character at the time the CPU polls bit 4). The reset function sets this flag high.

Bit 5, Interrupt Pending

A logical ONE in this bit indicates that one or more of the enabled interrupt conditions has occurred. This is the bit polled by the CPU in the polled-interrupt method described in Section 3.2.2

Bit 6, Full Bit Detected

A logical ONE in this bit indicates that the first data bit of an incoming character has been detected in the serial receiver register. This bit, provided for test purposes, will remain high until the entire character has been received. A reset instruction clears this flag.

Bit 7, Start Bit Detected

A logical ONE in this bit indicates that the start bit of an incoming character has been detected in the serial receiver register. This bit, provided for test purposes, will remain high until the entire character has been received. A reset instruction clears this flag.

3.4.2 TICC Command Register (see Figure 6-2)

The programmer can issue discrete commands to the TICC through the Command Register. Desired conditions can be encoded in the accumulator and transferred into the Command Register with the "STTCM" instruction. Here it will remain latched, with the exception of bit 0, until a new command arrives. The conditions associated with each bit of the Command Register are described in the following paragraphs.

Bit 0, Reset

A logical ONE transferred to this bit position will cause the following:

1. Clear the receiver buffer, the receiver register, the receiver-buffer-loaded flag, the start-bit-detected flag, the full-bit-detected flag, and the overrun-error flag.
2. Set the transmitter-buffer-empty flag high, indicating that the transmitter buffer is ready to accept a character from the CPU.
3. Clear the interrupt register except for the bit corresponding to the transmitter-buffer-interrupt, which is set to high
4 Inhibit the interval timers
5. Set transmitter data output high (marking).

The TICC Command Register does not latch a logical ONE transferred to bit 0, and therefore this does not have to be explicitly cleared; it is normally at logical ZERO

The reset function has no effect on the output ports, the external inputs, the interrupt acknowledge enable, the mask register, the rate register, the transmitter buffer and the transmitter register

Bit 1, Break

A logical ONE in this bit sets the serial output to the high impedence or non-conducting state (spacing). The reset function of bit 0 overrides this function. Normally this should be at logical ZERO.

Bit 2, Interrupt 7 Select

A logical ONE in this bit selects the auxiliary interrupt input at bit 7 of the parallel input port of the DEVICE CONNECTOR as input source. A logical ZERO selects the Interval Timer 5.

Bit 3, Interrupt Acknowledge Enable

A logical ONE in this bit enables the TICC to accept an interrut-acknowledge-decode from the CPU and transfer the RST instruction as a response. This condition sets up an interrupt driven system discussed in Section 3.2.1. A logical ZERO in bit 3 causes the TICC to ignore the interrupt-acknowledge-decode, a condition that results in a polled-interrupt-system, discussed in Section 3.2.2.

Bits 4 - 7

These bits are always ZERO, used only for DCE test procedures.

3.4.3 Communication Rate Register (see Figure 6-2)

This register allows the programmer to select any of the 7 standard BAUD rates for serial data transmission and reception. A logical ONE in any of the bits 0 through 6 of this register will select the BAUD rate for both the transmitter and the receiver as defined below:

Bit 0	110 BAUD
Bit 1	150 BAUD
Bit 2	300 BAUD
Bit 3	1200 BAUD
Bit 4	2400 BAUD
Bit 5	4800 BAUD
Bit 6	9600 BAUD

If more than one bit is high, the highest of the rates indicated will result. If all six bits are low, both the receiver and the transmitter will be inhibited.

Bit 7 allows the selection of stop bits to be generated by the transmitter register. Logical ONE in bit 7 selects one stop bit, while logical ZERO selects two stop bits.

The programmer can encode the desired BAUD rate and number of stop bits in the accumulator and transfer this code into the rate register with the "STCRR" instruction. This register holds the code until it receives a new command.

3.5 TICC REGISTER SUMMARY

REGISTER ADDRESS	REGISTER NAME	DESCRIPTION	MACRO INSTRUCTION	FUNCTION
9806	TRANSMIT BUFFER	Asynchronous serial transmitter buffer	STXMT	(TRANSMIT BUFFER)←(A)
9800	RECEIVE BUFFER	Asynchronous serial receive buffer	LDRCV	(A)←(RECEIVE BUFFER)
9807	OUT PORT	Parallel output port	STOUT	(OUT PORT)←(A)
9801	IN PORT	Parallel input port	LDIN	(A)←(IN PORT)
9809	TIMER 1	Interval timer 1	STTIM 1	(TIMER 1)←(A)
980A	TIMER 2	Interval timer 2	STTIM 2	(TIMER 2)←(A)
980B	TIMER 3	Interval timer 3	STTIM 3	(TIMER 3)←(A)
980C	TIMER 4	Interval timer 4	STTIM 4	(TIMER 4)←(A)
980D	TIMER 5	Interval timer 5	STTIM 5	(TIMER 5)←(A)
9808	INTR MASK	Interrupt mask register "1" in bit n enables interrupt n	STIMR	(INTR MASK)←(A)
9803	TICC STATUS	Status register:	LDSTA	(A)←(TICC STATUS)
	BIT 0:	1: Receiver framing error (stop bit(s) in error) 0: No framing error on last character		
	BIT 1:	1: Receiver overrun error (new character received before last one read) 0: No overrun error		
	BIT 2:	0/1: Serial input - direct connection		
	BIT 3:	1: Receive buffer loaded reset to 0 by LDRCV or system reset		
	BIT 4:	1: Transmit buffer empty reset to 0 by STXMT or system reset		
	BIT 5:	1: Some enabled interrupt is pending 0: No enabled interrupts pending		
	BITS 6, 7	Used for DCE test proceedings		
9804	TICC COMMAND	Command register	STTCM	(TICC COMMAND)←(A)
	BIT 0:	1 to reset TICC		
	BIT 1:	1 to send break on serial output-direct connection		
	BIT 2:	1 to select IN7 as source of interrupt 7 0 to select TIMER 5		
	BIT 3:	1 to enable TICC generation of 8080 interrupts		
	BIT 4 - 7	0 always (used for DCE test procedures)		
9805	RATE REGISTER	Communication rate register	STCRR	(RATE REGISTER)←(A)
	BITS 0 - 6	All zero: inhibit serial communications ONE on: select 110, 150, 300, 1200, 2400 4800, or 9600 BAUD by bit 0,1,2,3,4,5 or 6 on respectively		
	BIT 7	1 for one stop bit 0 for two stop bits		
9802	INTER PEND	Interrupt pending register	LDIRP	(A)←(INTR PENDING)
	BITS 5,4,3	Number of highest pending interrupt		
	BITS 7,6,2,1,0	Always 1		

3.6 ASCII CODE TABLE

USASCII (USA Standard Code for Information Interchange)

Bits b7 b6 b5					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b4 ↓	b3 ↓	b2 ↓	b1 ↓	COLUMN → ROW ↓	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	–	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

4 DCE-BUS SPECIFICATIONS

4.1 DCE-BUS CONCEPT

The DCE-BUS provides a common transfer medium for the exchange of data and control information between DCE-BUS compatible modules and one or more processor modules. It allows data devices or processor modules to be directly plugged into the bus in any physical position. It carries two interrupt request signals, a reset signal, D. C. power lines and control signals for device selection, addressing and data transfer.

The DCE-BUS may be controlled via the GIC on the master processor either in a simple software controlled mode, or in a fast-bus mode with the Large System Adapter (DCE-LSA) in DCE-X based systems TheDCE-BUS can also be driven via the DCE-LSA, to provide a very fast memory-mapped I/O mode of operation.

4.1.1 General Description

Figure 4. 1 illustrates the implementation of the DCE-BUS via the GIC on the master DCE processor module. GIC Port 0 is used as a bidirectional data path for data transfer between the master DCE processor and other modules. GIC Port 1 is used to specify the address of the connecting module, and GIC Port 2 issues the control signals to complete the transfer logic.

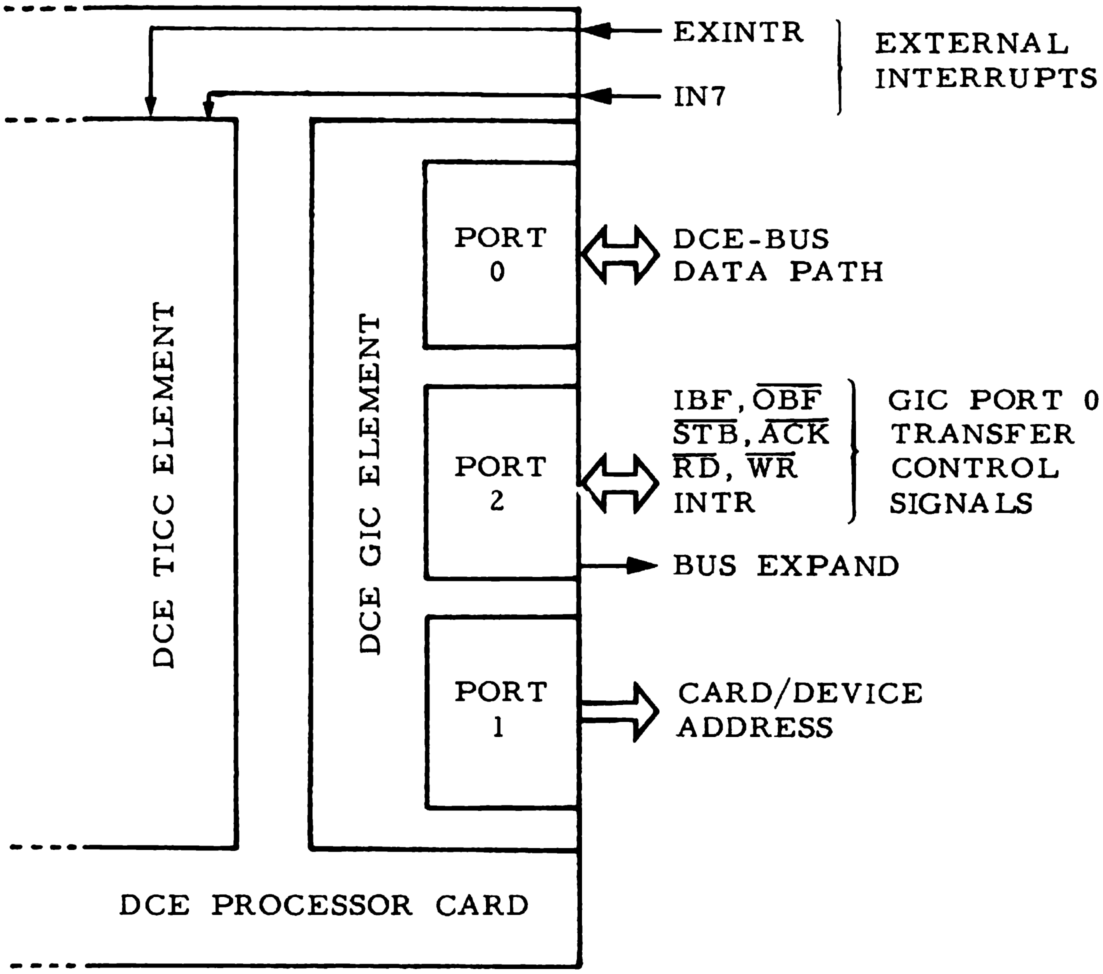


Figure 4.1 : DCE-BUS GIC Port Allocation

In the software controlled transfer modes, Port 0 must be alternated between input and output by software command. The fast-bus mode with DCE-LSA requires Port 0 to be set up as bi-directional. In both modes Port 1 carries the card/device address, and Port 2 carries control signals. Software controlled transfer operations require software activation of the $\overline{RD}$ and $\overline{WR}$ signals with appropriate timing. Fast-bus configured systems generate these $\overline{RD}$ and $\overline{WR}$ signals automatically via the DCE-LSA. The DCE-BUS also enables several DCE processors to operate in master-slave mode under software control.

The memory-mapped I/O mode does not use the GIC at all. The DCE-LSA uses the address, data, read and write signals from the CPU (via the X-BUS) to drive the DCE-BUS.

Two external interrupt request signals are provided on the DCE-BUS. Section 3.2 describes the processing of these interrupts in the 8-bit DCE systems. More interrupt vectors are possible by special logic on the external card to supply interrupt source indentification data.

4.1.2 Software Controlled READ Operation

The Read operation transfers data from the DCE-BUS data path (GIC Port 0) to the selected CPU register of the DCE processor (the accumulator when using the DCE GIC I/O macro commands). Figure 4.2 illustrates the DCE-BUS configured for software controlled read operations.

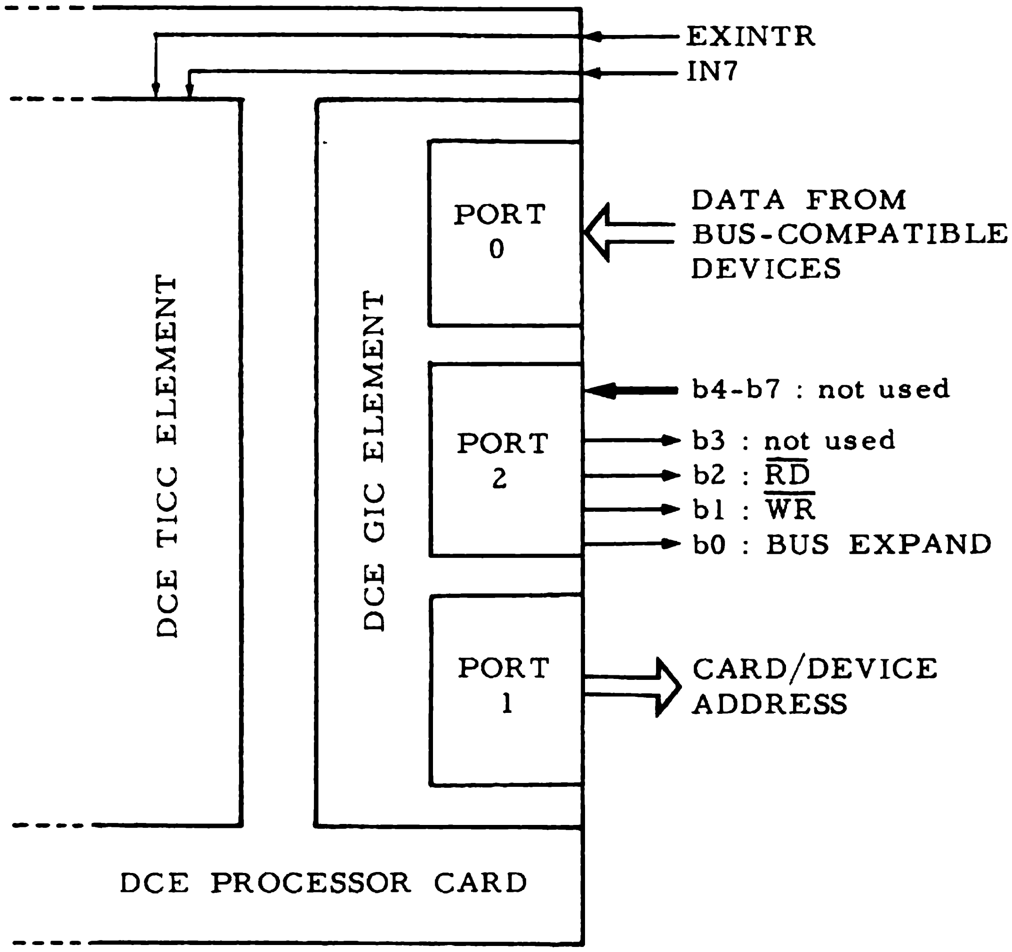
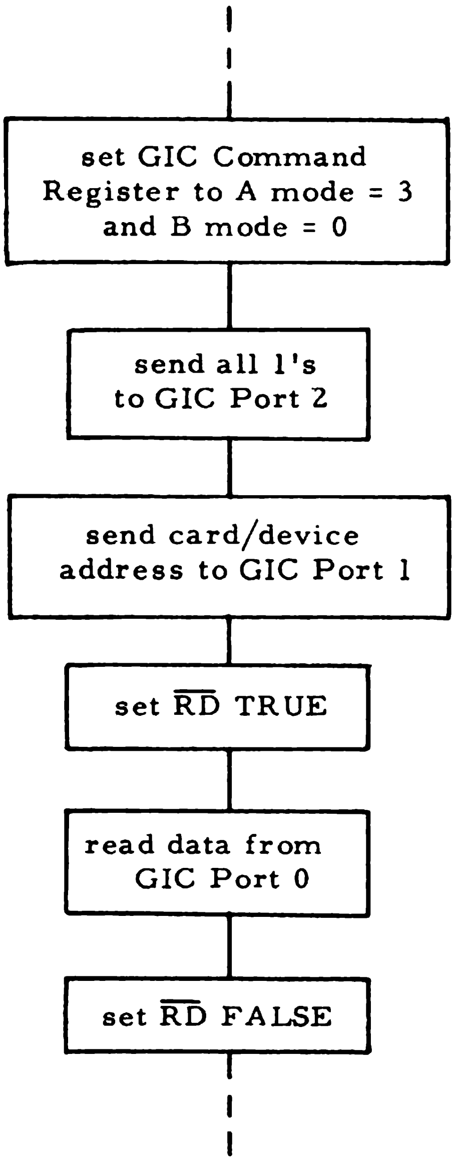


Figure 4.2 : Software Controlled READ Configuration

To place the DCE-BUS in the software controlled Read mode, the GIC should be set up with the GIC configuration command 'GICC 3,0' and GIC Port 2 should be set to all 1's. The complete software controlled read sequence is illustrated below. The stabilization times are accounted for by the execution times of the read command instructions.


set GIC Command Register to A mode = 3 and B mode = 0
send all 1's to GIC Port 2
send card/device address to GIC Port 1
set $\overline{RD}$ TRUE
read data from GIC Port 0
set $\overline{RD}$ FALSE

High-speed read control signal generation by hardware is explained in Section 4. 1. 4.

The following example realizes the software controlled transfer of an 8-bit data word from a device connected to the DCE-BUS into the DCE processor CPU accumulator.

```
GICC    3,0         :  port 0 - input,  port 2 (4-7) - input,
                       port 1 - output, port 2 (0-3) - output.
MVI     A,FF
STGI    2           :  all 1's to GIC port 2
MVI     A,adr       :  load card/device address
STGI    1           :  send to GIC port 1
MVI     A,FB
STGI    2           :  set R̅D̅ true
LDGI    0           :  read data from DCE-BUS into Accumulator
STA     data area      store data
MVI     A,FE
STGI    2           :  set R̅D̅ false
```

Note: interrupts should be disabled during the software read sequence

BUS EXPAND Signal

This signal is provided to allow the realization of large systems by doubling the number of bus-compatible modules that may be driven by a single DCE processor module. It can be used as an extra addressing line to switch between two banks of bus-compatible modules, each of which responds similarly to the Card/Device addresses.

In normal small system usage, this signal may be used to disable the address decode logic on all the bus-compatible modules on the DCE-BUS while they are inactive. Such an arrangement will prevent the accidental activation of modules.

The above example assumes that this signal when high enables all the address decode logic. It is therefore set at the start of the sequence and cleared at the end.

4.1.3 Software Controlled WRITE Operation

The Write operation transfers data from a CPU register of the DCE processor to the DCE-BUS Data Path (GIC Port 0). The accumulator is used by the DCE GIC I/O macro commands. Figure 4.3 illustrates the DCE-BUS configured for software controlled Write operations.

To place the DCE-BUS in the software controlled Write mode, the GIC should be set up with the GIC Configuration Command 'GICC 1,0' and GIC Port 2 should be set to all 1's. The complete software controlled Write sequence is illustrated below. The stabilization times are accounted for by the execution times of the Write command instructions.

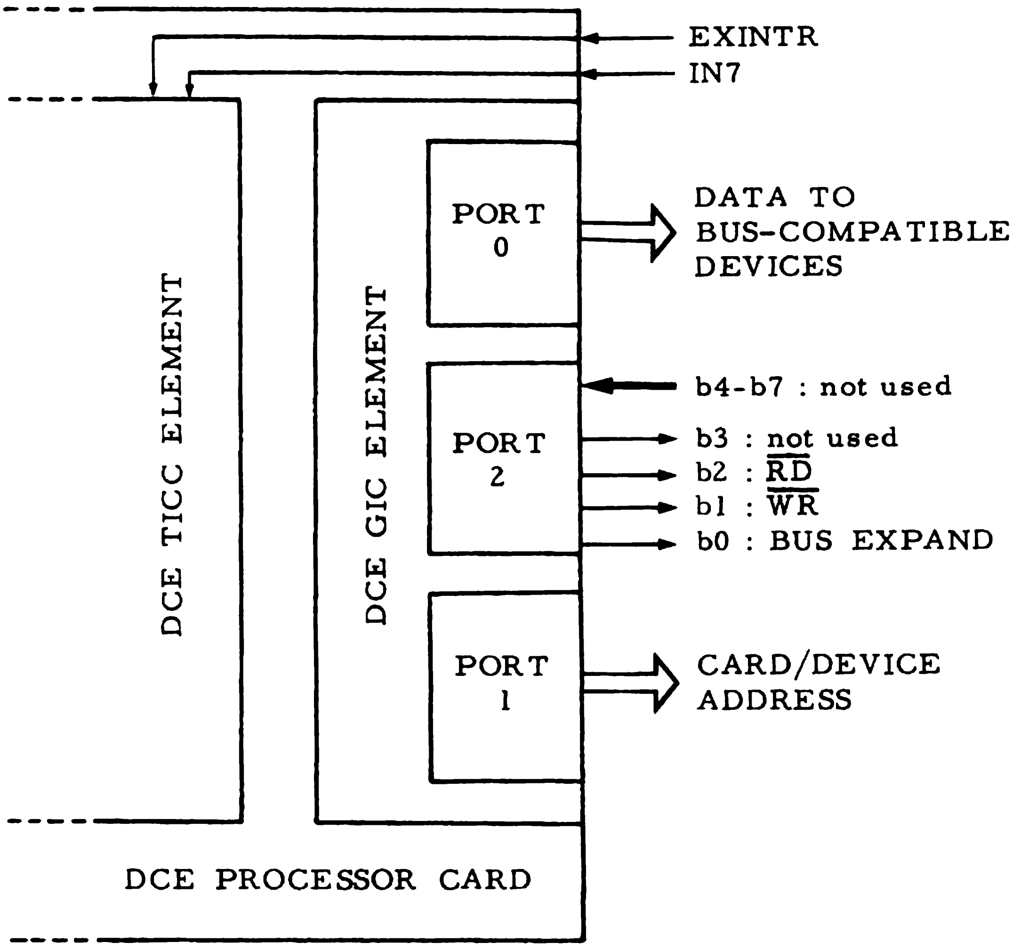
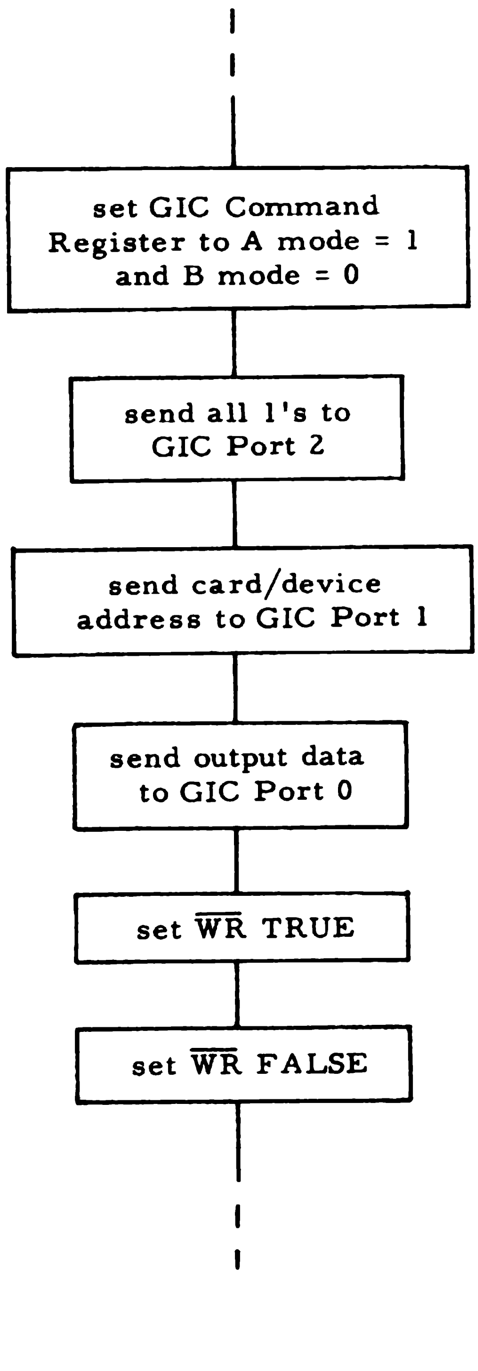


Figure 4.3 : Software Controlled WRITE Configuration


set GIC Command Register to A mode = 1 and B mode = 0
send all 1's to GIC Port 2
send card/device address to GIC Port 1
send output data to GIC Port 0
set $\overline{WR}$ TRUE
set $\overline{WR}$ FALSE

N 80

High-speed write control signal generation by hardware is explained in Section 4. 1. 4.

The following example realizes the software controlled transfer of an 8-bit data word from the DCE processor accumulator to a device connected to the DCE-BUS.

```
GICC    1,0       :  port 0 - output, port 2 (4-7) - input
                     port 1 - output, port 2 (0-3) - output
MVI     A,FF
STGI    2         :  all 1's to GIC Port 2
MVI     A,adr     :  load card/device address
STGI    1         :  send to GIC Port 1
MVI     A,data    :  load data to be output
STGI    0         :  send to GIC Port 0
MVI     A,FD
STGI    2         :  set WR true
MVI     A,FE
STGI    2         :  set WR false
```

!! * Note: interrupts should be disabled during the software write sequence. !!

4.1.4 Hardware Controlled READ/WRITE Operation (FAST-BUS Mode)

The Fast-Bus mode of operation assumes that the system is implemented together with the DCE-LSA card (see Section 6.9). The Fast-Bus mode of operation uses the GIC Port 0 in bi-directional mode. Data is moved between the DCE-BUS and the DCE processor under control of the handshake signals from Port 2. The DCE-LSA generates the $\overline{RD}$ and $\overline{WR}$ signals in response to these input and output handshake control signals which are defined in Section 2.4

Figure 4.4 illustrates the DCE-BUS configured for the FAST-BUS mode of operation. To place the DCE-BUS in the FAST-BUS mode the GIC should be set up with the GIC configuration command 'GICC 8,1'.
A complete read and write sequence is illustrated below. The stabilization and transfer control timing is accounted for by the DCE-LSA card logic.

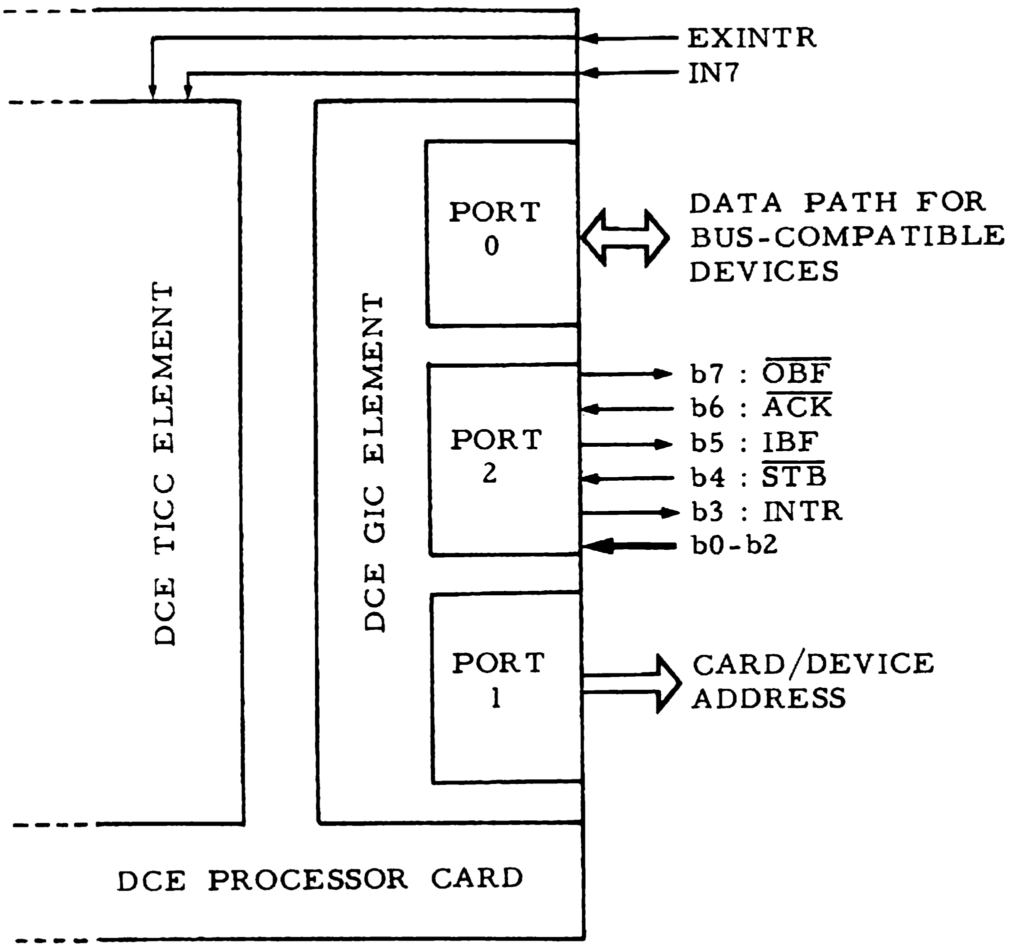
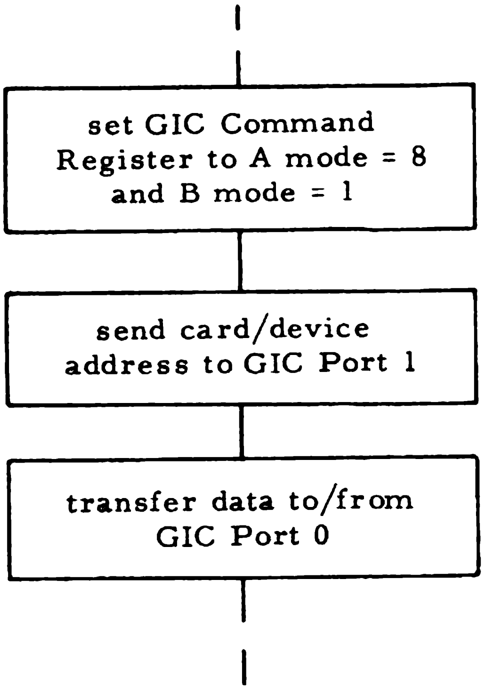


Figure 4.4 : Fast-Bus Configuration


set GIC Command Register to A mode = 8 and B mode = 1
send card/device address to GIC Port 1
transfer data to/from GIC Port 0

Fast-Bus Read/Write Sequence

The following example realizes the hardware controlled transfer of data between the DCE processor accumulator and the device connected to the DCE-BUS

```
GICC    8,1        :  port 0 - bi-directional, port 2 (3-7) - H.S.C
                      port 1 - output, port 2 (0-2) - input
MVI     A,adr      :  load card/device address
STGI    1          :  send to GIC Port 1
MVI     A,data
STGI    0          :  write data to Port 0
or
(LDGI   0          :  read data from Port 0)
```

Note: interrupts should be disabled during these read/write sequences.

To realize interrupt controlled transfers using the Fast-Bus mode, connect the INTR signal of the GIC to one of the interrupt lines. In this case each time the external device has data available for input, or is ready to receive data, an interrupt request will be automatically generated.

4.2 DCE-BUS INTERRUPT FACILITIES

The DCE-BUS carries the two available external vectored interrupt requests to the DCE processor. Connecting modules may drive either of these interrupt request lines asynchronously. See Section 3.2 for a full description of interrupt handling by the DCE processor.

The number of available external interrupt requests may be expanded beyond two by the insertion of priority encoding logic on the bus compatible modules. This should gate interrupt source identification data onto the input data path, upon a read command. This data would then be decoded by software to determine the interrupting device. DAI Real-World cards with multiple-level interrupts may also be used in this respect. See Section 4.3.2 for details.

4.3 DCE-BUS COMPATIBLE MODULES

Figure 4.5 illustrates the basic logical subsets of a DCE compatible module including the logical definitions necessary for multiple interrupt vectors. The DCE-BUS allows some functions to be implemented using software or hardware logic. A good example of this is the realization of Read/Write Control and the multiple interrupt definition vectors. DAI Real-World cards are logically realized for best economic and format compatibility.

4.3.1 Data Devices

The DCE-BUS allocates 8 bits for card and device addressing. Generally, four of the bits are allocated for the specific card address and the other four are allocated for addressing data devices (registers) on that card. Data devices are usually logic elements that have read and write capabilities and are connectable to the DCE-BUS data path. Typically (as in DAI real-world cards) these devices are logical controllers for a particular type of external device. Data devices can require interrupt servicing. It is quite practical and economic to consider a direct link to a data device with dedicated device address lines, that eliminates the necessity for device select decoding. In this case the number of data devices cannot exceed eight. Designers constructing DCE-BUS compatible modules must trade-off the above approach against the added hardware necessary for device address decoding, to fulfill their specific requirements.

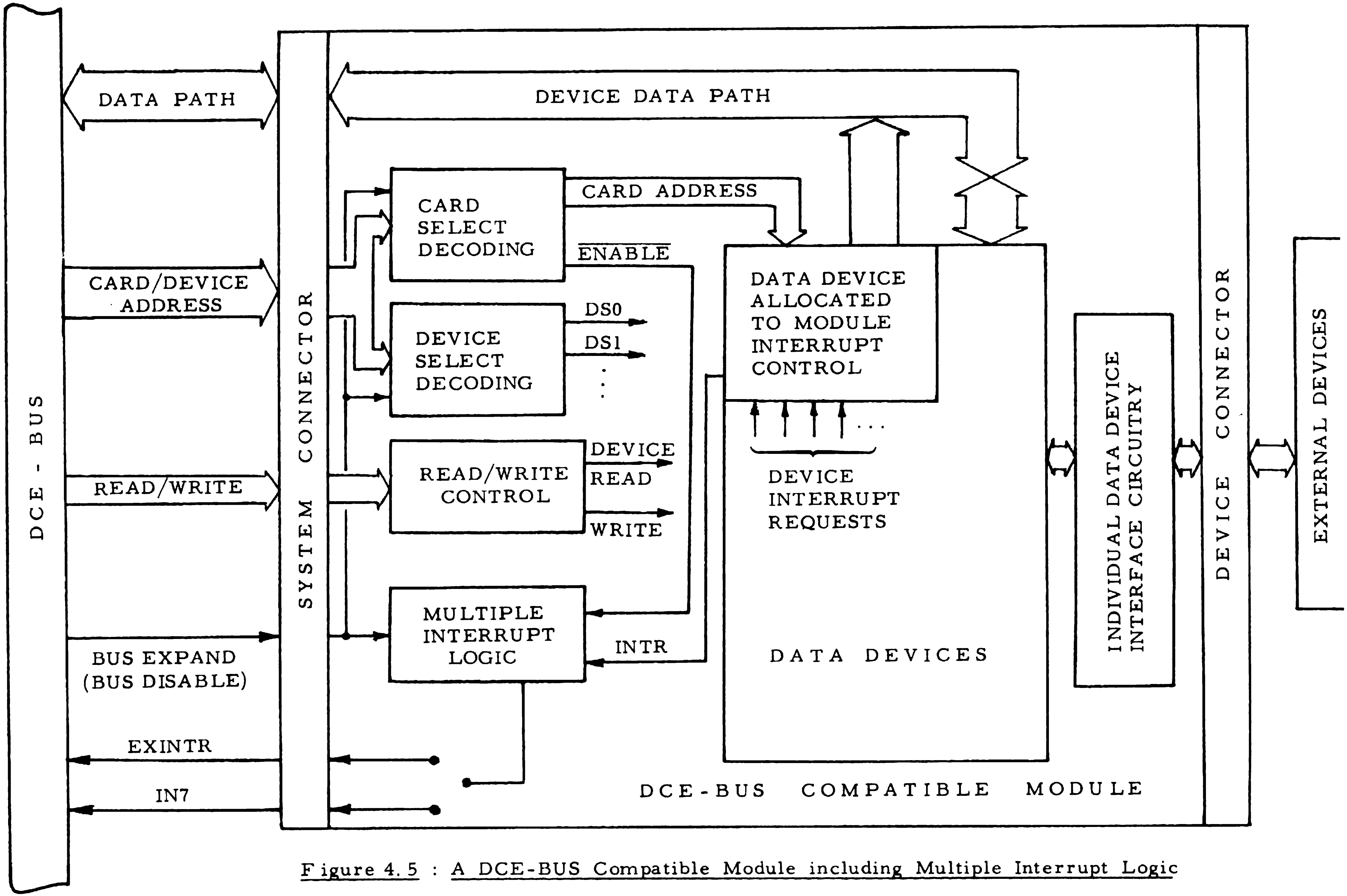


Figure 4.5 : A DCE-BUS Compatible Module including Multiple Interrupt Logic

4.3.2 Multiple Interrupt Control

It can be quite interesting to allow several data devices to interrupt the DCE processor. An effective technique for realizing this would be to allocate one of the data devices as a read-only device that places on the DCE-BUS data path the number of the interrupting device, when addressed by the card/device address lines.

A typical set of logic elements to realize this function would be a priority encoder and some tri-state buffers. Figure 4.6 illustrates this arrangement.

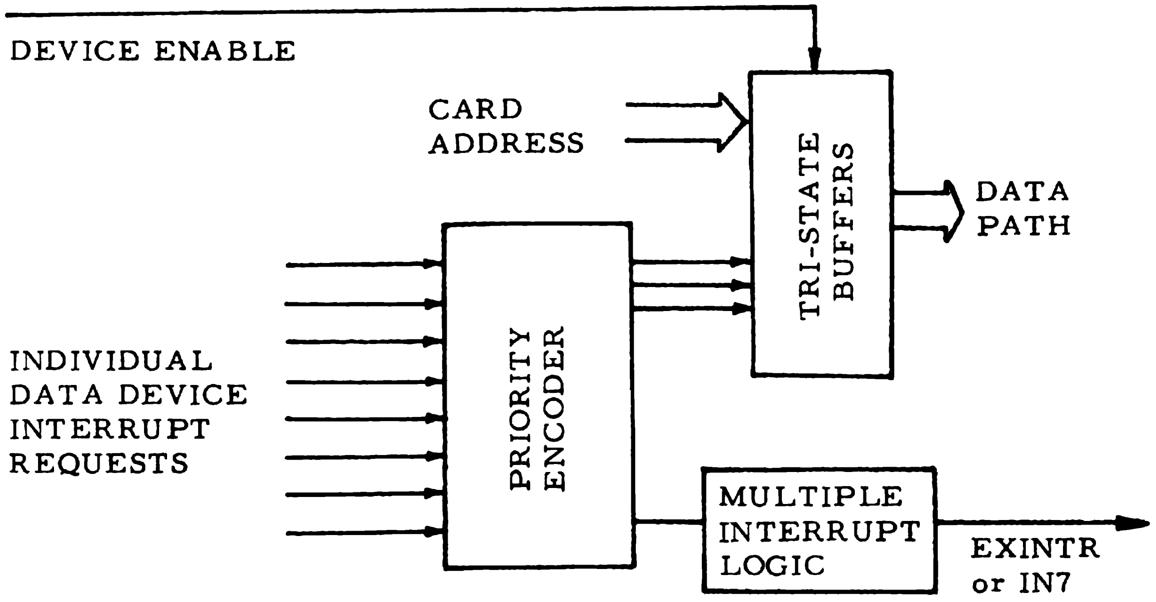


Figure 4.6 : Typical Module Interrupt Controller

To allow more than two bus compatible modules to interrupt the DCE processor, the modules can be equipped with the multi-module interrupt logic. Figures 4.7 and 4.8 illustrate the necessary logic functions to realize multiple vectored interrupts from bus compatible modules. Notice should be taken of the combination of software and hardware logic functions.

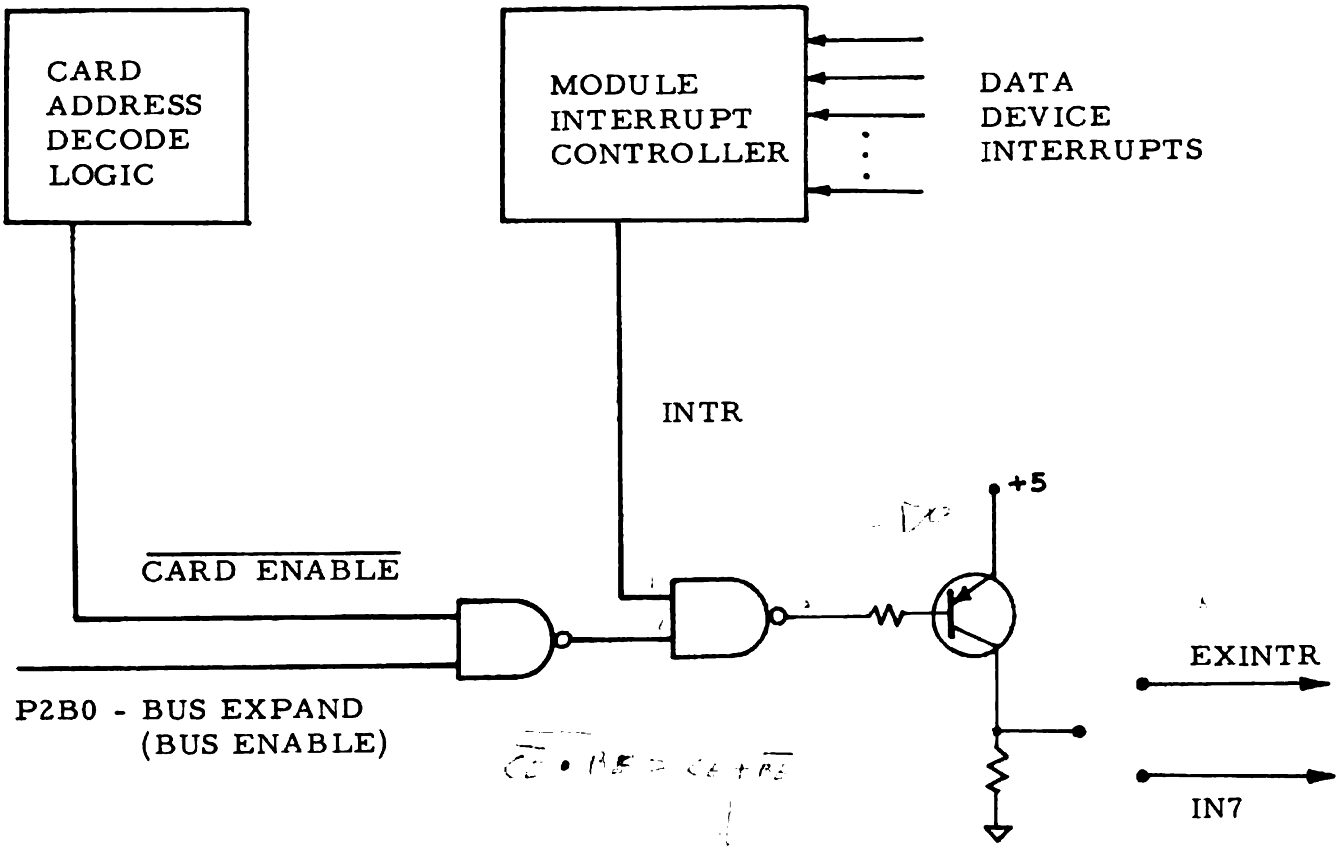

CARD ADDRESS DECODE LOGIC
MODULE INTERRUPT CONTROLLER
DATA DEVICE INTERRUPTS
INTR
+5
CARD ENABLE
P2B0 - BUS EXPAND (BUS ENABLE)
EXINTR
IN7
$\overline{CE} \cdot \overline{BE} = \overline{CE + BE}$

Figure 4.7 : Hardware Logic Necessary For Multiple Module Interrupt Vectoring

By using this technique on each participating bus compatible module, a large number of different interrupt vectors can be connected to each DCE processor. Note the program logic functions that form a part of this scheme. The starting address of this software logic sequence must be at location 0010H or 0038H of the DCE program memory, depending on the usage of interrupt request lines EXINTR or IN7 respectively.

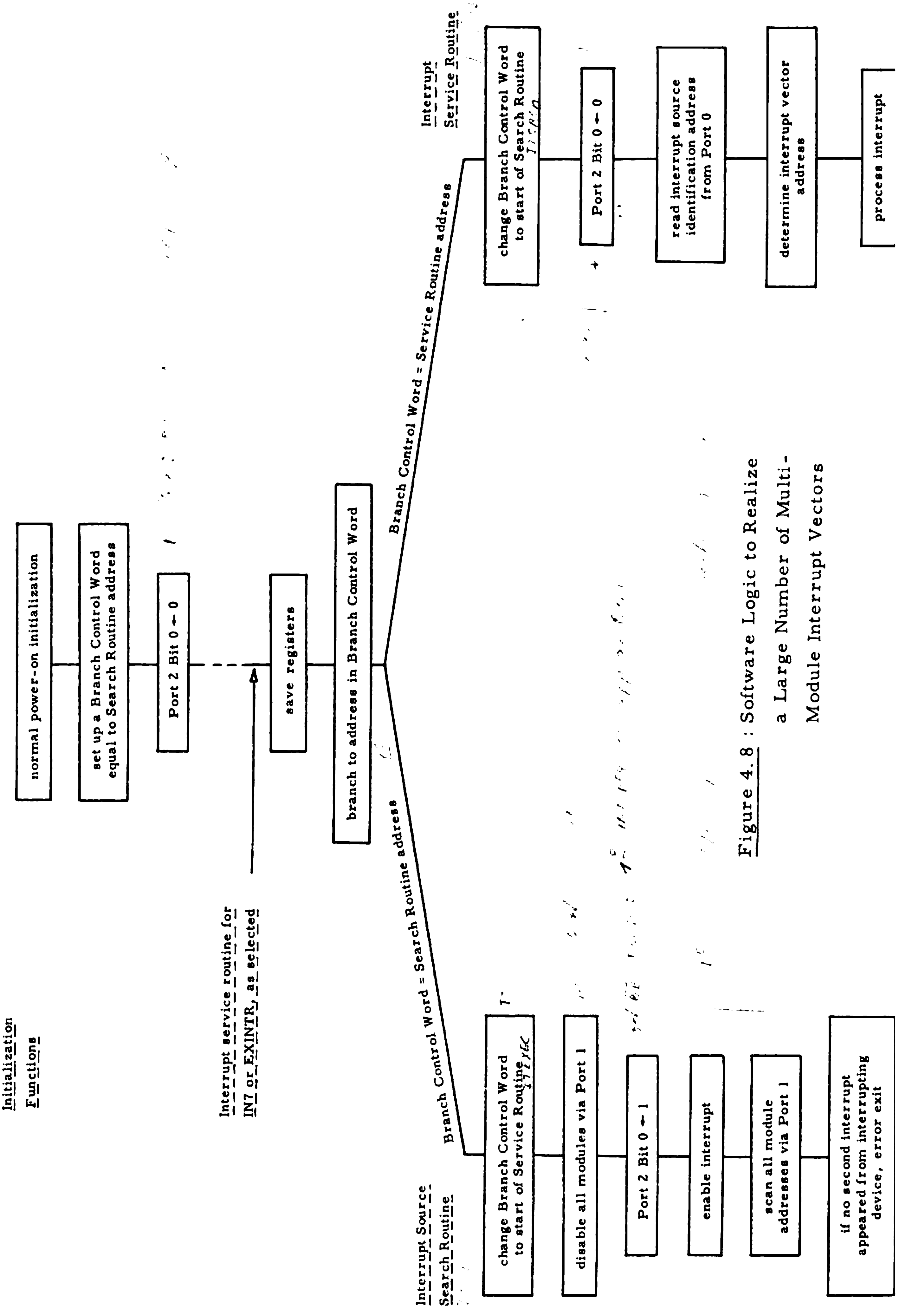

Initialization Functions
normal power-on initialization
set up a Branch Control Word equal to Search Routine address
Port 2 Bit 0 ← 0
Interrupt service routine for IN7 or EXINTR, as selected
save registers
branch to address in Branch Control Word
Branch Control Word = Search Routine address
Branch Control Word = Service Routine address
Interrupt Source Search Routine
change Branch Control Word to start of Service Routine
disable all modules via Port 1
Port 2 Bit 0 ← 1
enable interrupt
scan all module addresses via Port 1
if no second interrupt appeared from interrupting device, error exit
Interrupt Service Routine
change Branch Control Word to start of Search Routine
Port 2 Bit 0 ← 0
read interrupt source identification address from Port 0
determine interrupt vector address
process interrupt

Figure 4.8 : Software Logic to Realize a Large Number of Multi-Module Interrupt Vectors

4.3.3 Read/Write Control

The I/O devices within the DCE processor architecture are structured to make the addressing and data transfer protocol between them and the 8080 CPU logically similar to those for memory locations. This logical order is extended to the bus compatible data devices. Therefore, in addition to the card/device address logic, they also need a read pulse ($\overline{RD}$) and a write pulse ($\overline{WR}$).

Bus compatible modules can be constructed that generate these $\overline{RD}$ and $\overline{WR}$ signals automatically with hardware; or they can be generated simply by software sequences. Figures 3.9 and 3.10 illustrate the basic Read and Write timing sequences.

Systems designed for software controlled Read and Write operations can be enacted as described in Sections 4.1.2 and 4.1.3.

Hardware Read/Write logic must be constructed to satisfy the timing diagrams below. The DCE-LSA card will perform this function.
DAI Real-World cards are designed to operate within systems that are structured for hardware controlled I/O transfer via the DCE-LSA, or for software controlled I/O transfer. Figures 4.11 and 4.13 illustrate the timing sequences for generating $\overline{RD}$ and $\overline{WR}$ pulses from the GIC handshake control signals. Figures 4.12 and 4.14 illustrate the hardware logic elements for realizing such a scheme, for bus compatible modules containing LSI peripheral devices standard to the 8080 (e.g. 8255).
DAI Real-World cards are constructed in this manner, using the 8255 as the RIC (Real-World Interface Control). Section 7 of this manual fully describes all the bus compatible Real-World modules.

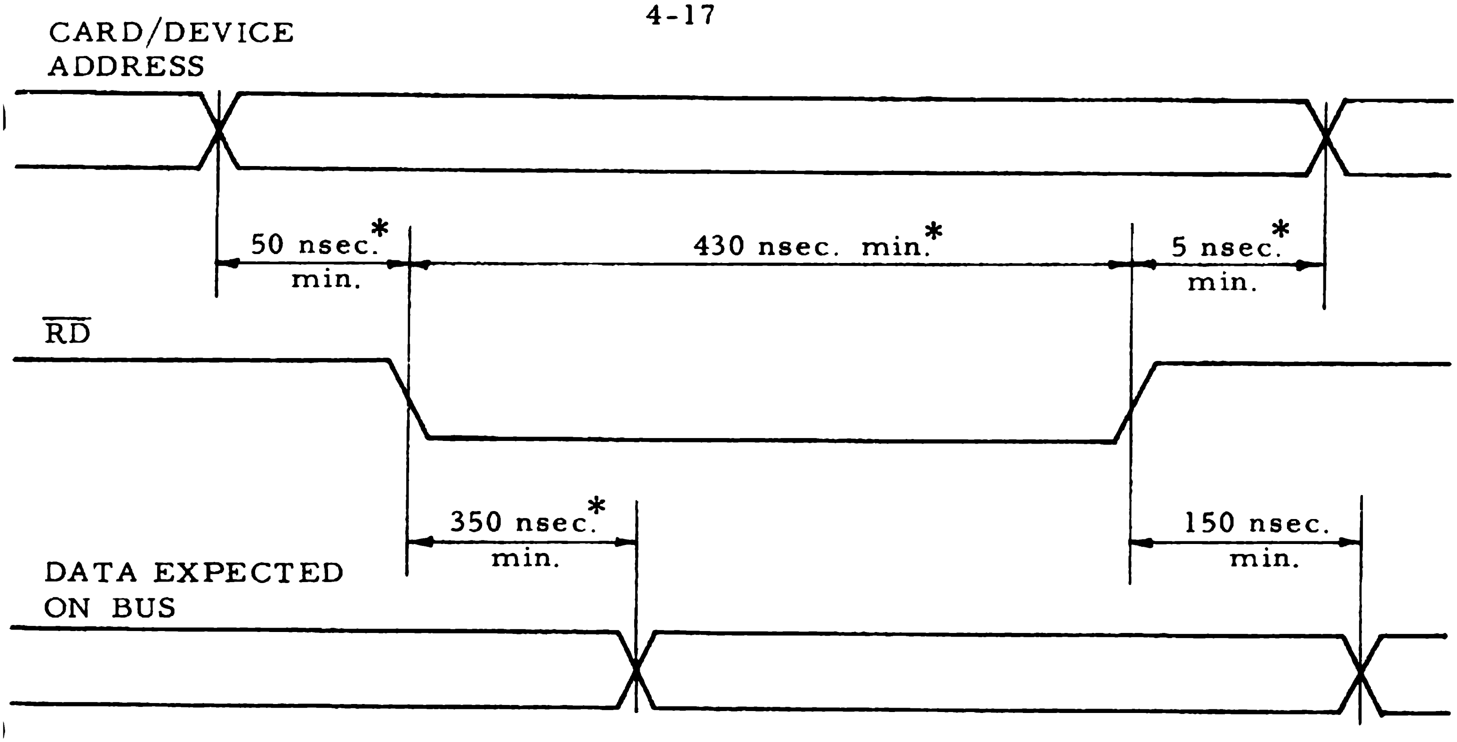


Figure 4.9 : DCE-BUS Basic READ Timing Sequence

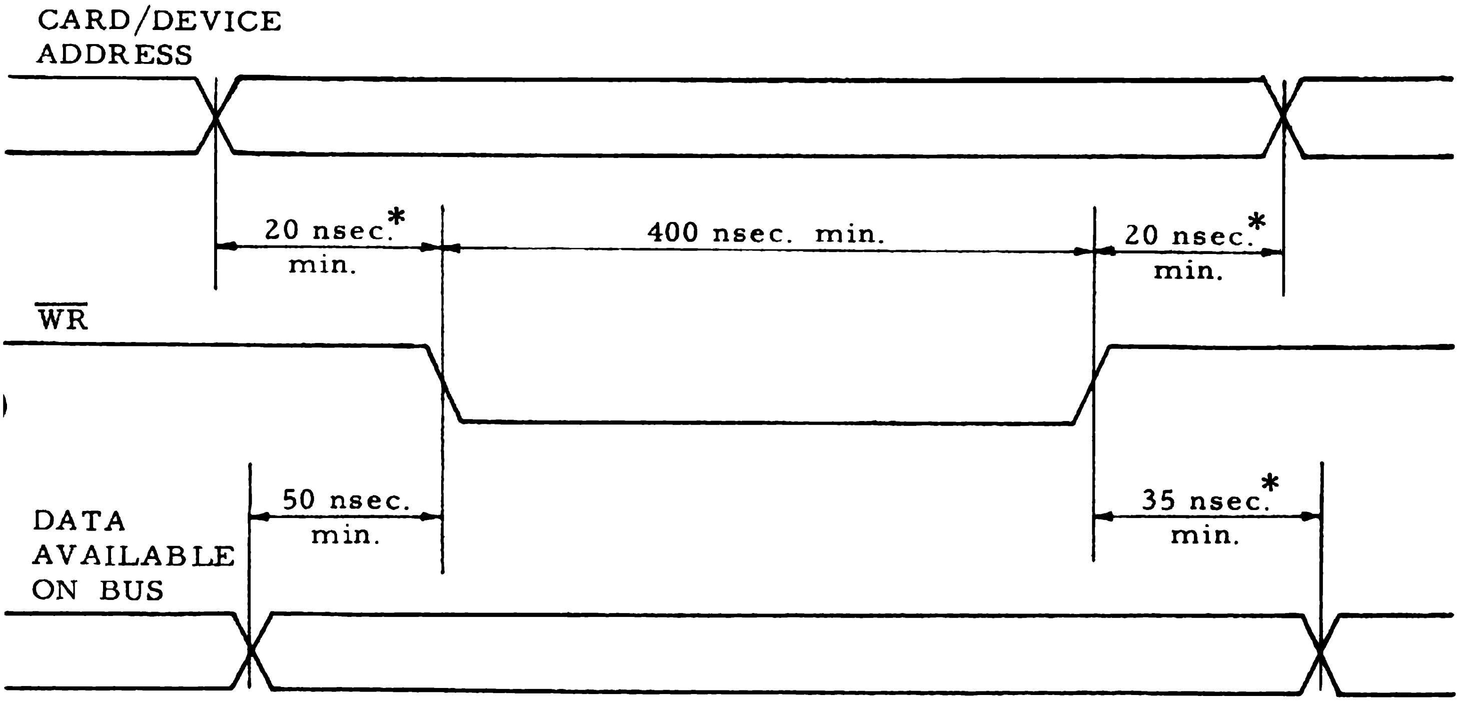
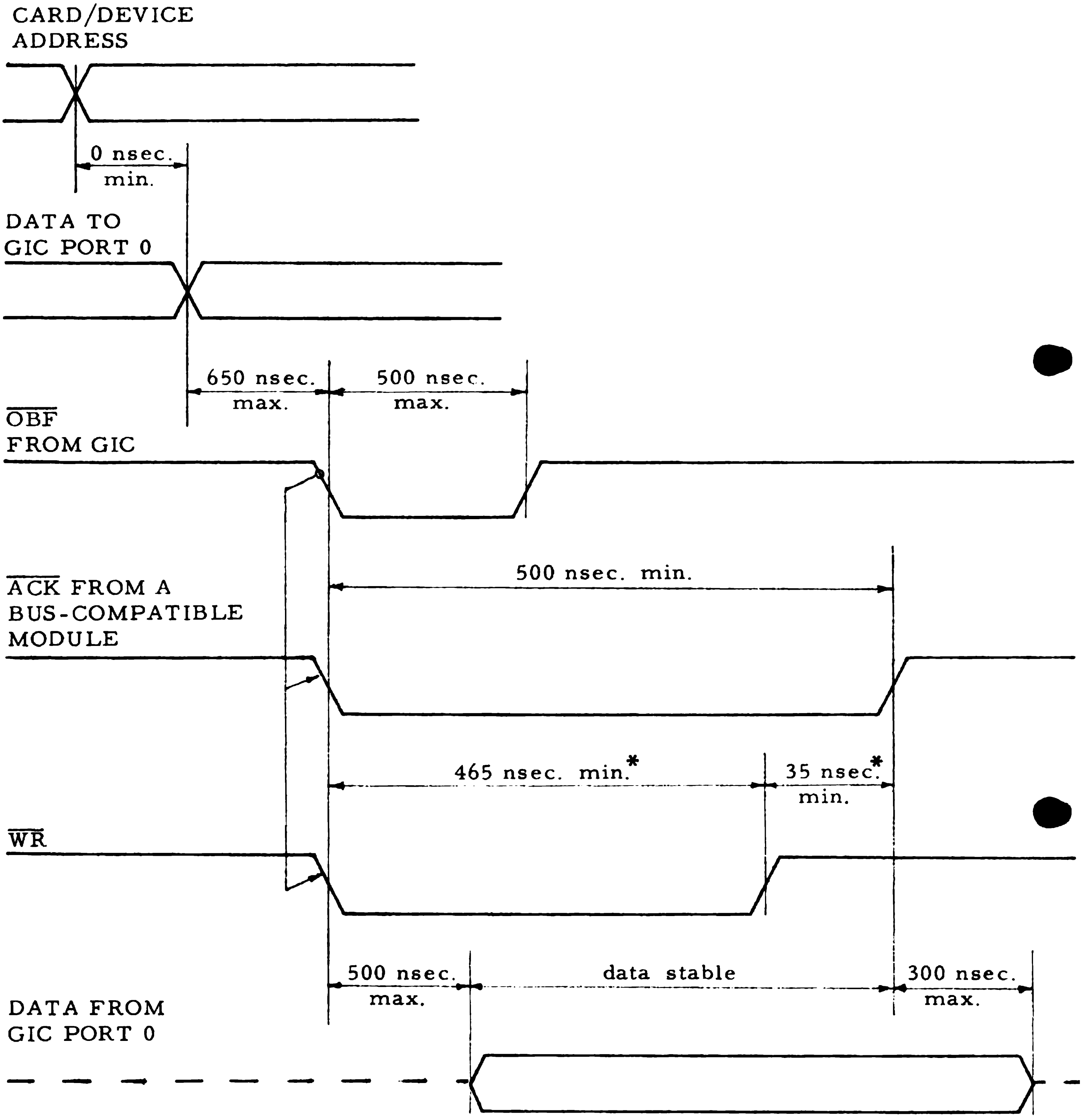


Figure 4.10 : DCE-BUS Basic WRITE Timing Sequence

Time between consecutive READs and/or WRITEs = 850 nsec. min.

* for 8255 type devices as Module Interface Controller



* for 8255 type devices as Module Interface Controller

Figure 4.11 : Using GIC Handshake Signals to Generate $\overline{\text{WR}}$ (Fast-Bus Mode)

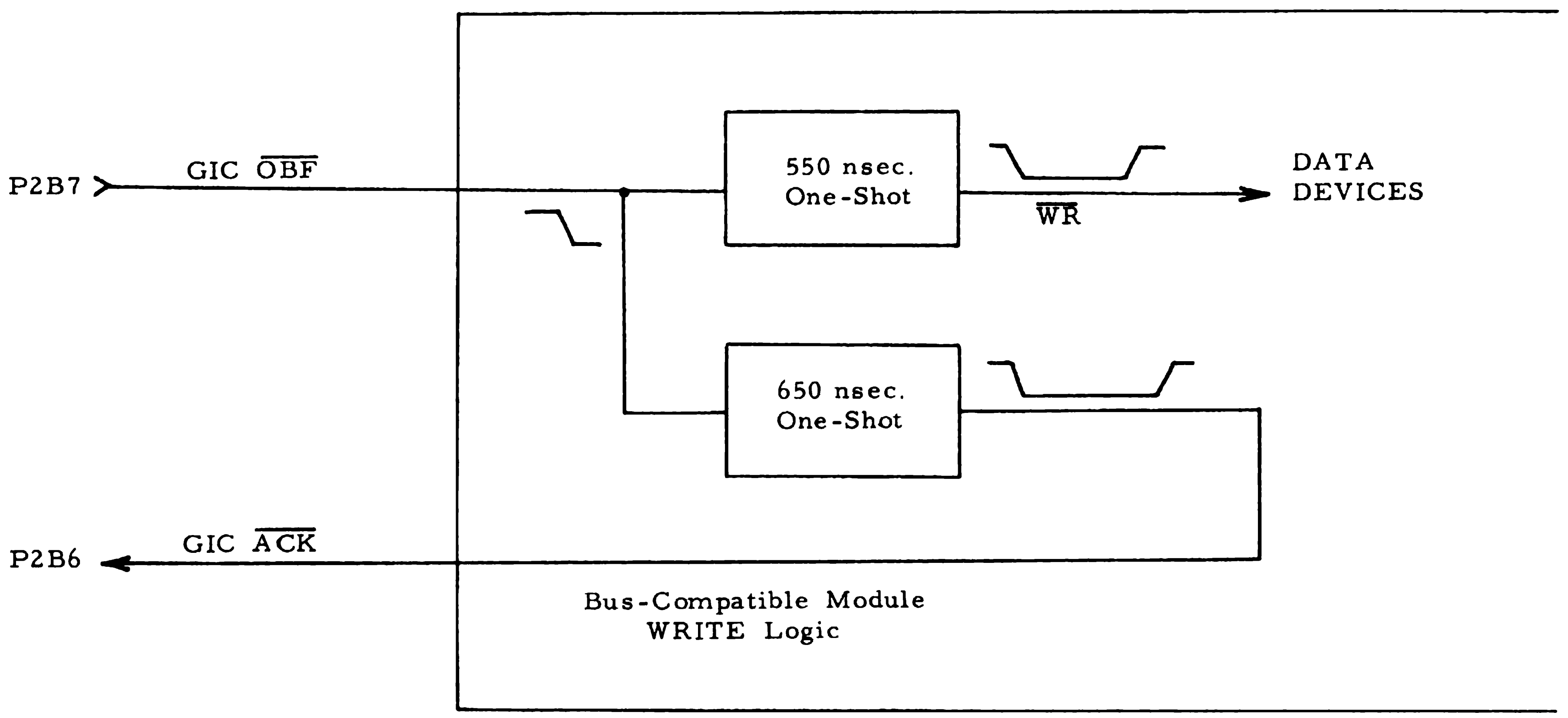
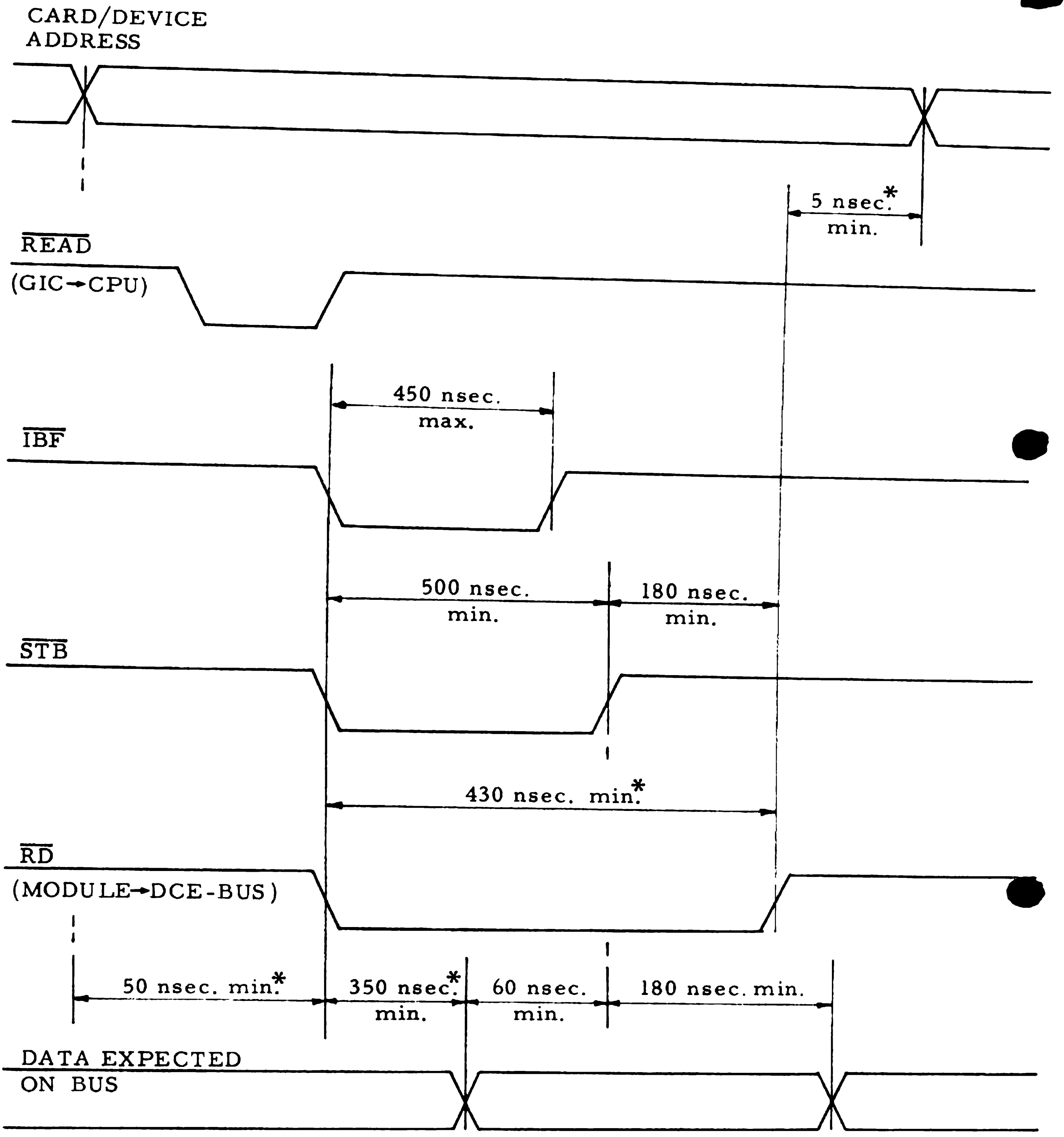


Figure 4.12 : A Hardware Logic Configuration to Generate $\overline{WR}$ on Bus-Compatible Modules that use 8255 type Devices (Fast-Bus Mode)



* for 8255 type devices as Module Interface Controller

Figure 4. 13 : Using GIC Handshake Signals to Generate RD (Fast-Bus Mode)

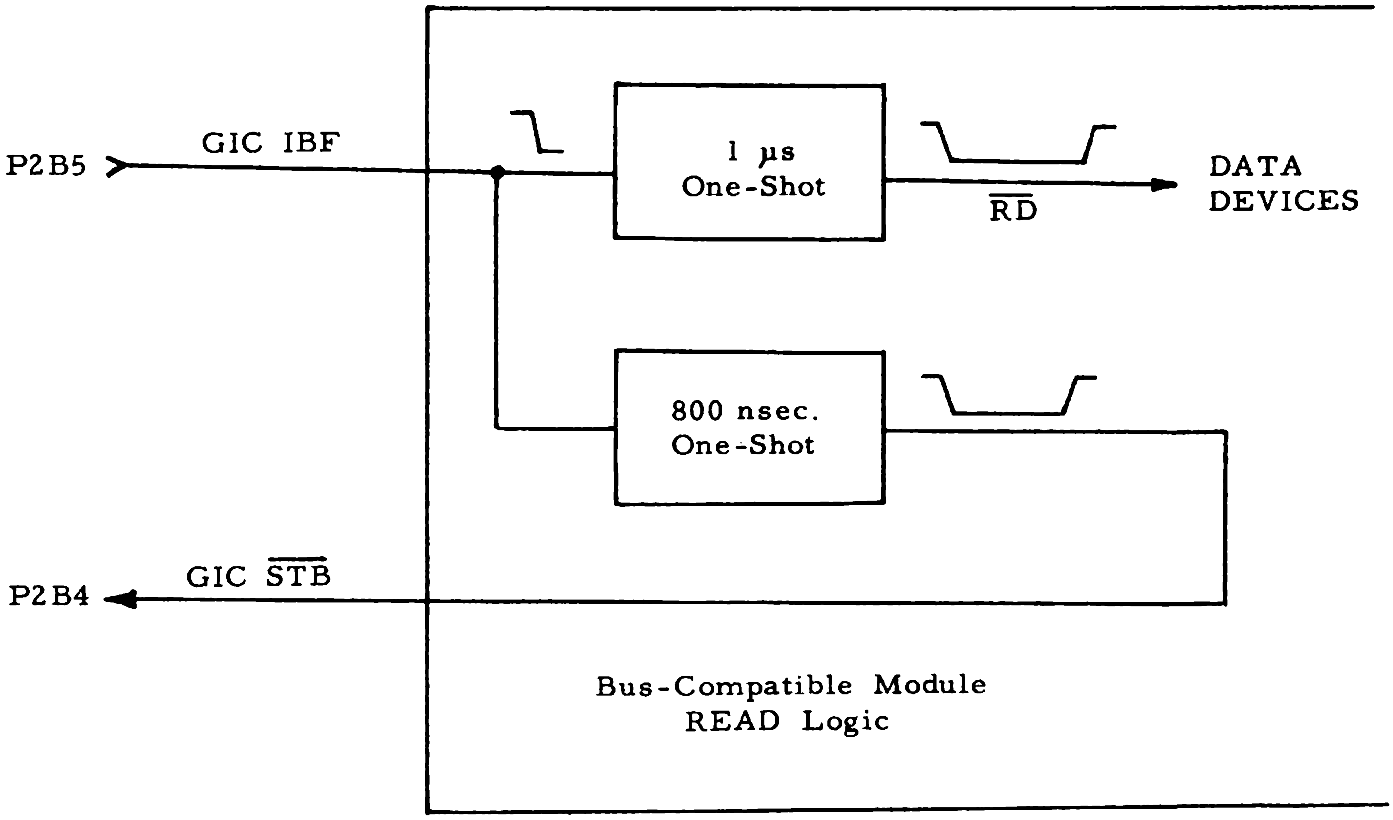


Figure 4.14 : A Hardware Logic Configuration to Generate $\overline{RD}$ on Bus-Compatible Modules that use 8255 type Devices (Fast-Bus Mode)

Note: Using this logic results in a dual transfer of data:

(i) transfer from GIC Input Buffer to DCE processor CPU Register,

(ii) transfer from Peripheral Device to GIC Input Buffer.

Shifting peripheral data into the DCE processor CPU requires two Read operations.

4.3.4 Card/Device Address Decoding

The DCE-BUS provides eight latched bits via GIC Port 1, intended for use as card/device addresses for bus compatible modules. Address decode logic should be provided as needed to select a particular card (module) and then a particular data device on that card.

It is possible to construct systems where bus compatible modules and data devices on them are directly addressed by the bits from GIC Port 1, thus eliminating the need for hardware address decode logic. However, with the insertion of hardware decoding as shown in Figure 4.5, the number of modules connectable to the DCE-BUS, and the number of data devices on each such module can be made large.

DAI Real-World cards use 4 out of the 8 address bits as card select address. The other 4 bits are used for device selection within the selected card.

4.4 DCE-BUS LOADING

The DCE-BUS is driven by the GIC device on a DCE processor module. Each GIC I/O line has a sink capability of 1.6mA at 0.4V.

The family of Real-World modules present a standard interface to the DCE-BUS. This is usually implemented by a Real-World Interface Controller (RIC), which is the same type of device as the GIC on DCE processor modules. Real-World modules with such an interface (eg RWC-F, RWC-T24) present 1 unit load to the DCE-BUS. Modules which do not use a single device RIC interface to the DCE-BUS may present the equivalent of several unit loads to the DCE-BUS (eg. RWC-CCE).

The DCE-BUS can support at least 32 such unit loads without any further buffering.

5 PROGRAMMING THE DCE

5.1 STANDARD 8080 INSTRUCTION SET

93 Instructions which can be grouped as:

* Data transfer group: 13 instructions to move data between registers or between registers and memory.

* Arithmetic group: 20 instructions to add, subtract, increment or decrement data in register or memory.

* Logical group: 19 instructions to and, or, exclusive or, compare, rotate or complement data in registers or memory.

* Branch group: 29 instructions to conditionally, or unconditionally jump, call subroutine, or return from subroutines.

* Stack, I/O, machine control group: 12 instructions for input/output, stack handling, and system control.

For a résumé of the 8080 instruction set see the table of Section 5.5.

5.2 GIC AND TICC MACRO INSTRUCTIONS

A macro instruction is an indication to the assembler that a symbol appearing in the label field of a statement actually stands for a group of instructions. The following set of GIC and TICC commands are macro instructions, and their definition punched on a paper-tape must preceed the user source program during pass 1 of the MAC-80 assembler (it need not be used during passes 2, 3,or 4). Tapes with the GIC and TICC macro instructions are available to users of the DCE. For those users who do not have access to an assembler and wish to program directly with the HEX codes, a GIC and TICC macro expansion table is provided in Sections 5.3 and 5.4 on pages 5-3 and 5-4. The DCE-DM Development System Resident Assembler (UAE) automatically recognizes all the GIC and TICC macros and resolves the register address differences between the different DCE microcomputer modules.

5.2.1 GIC Macro Instructions

GICC	amode, bmode	Configure GIC to one of 63 possible mode combinations.
BSET	nbit	Set specified bit in GIC PORT 2
BCLR	nbit	Clear specified bit in GIC PORT 2
LDGI	nport	Load contents of specified GIC port to accumulator.
STGI	nport	Store contents of accumulator in specified GIC port.
LDGIS	0	Load contents of GIC PORT 0 to accumulator synchronized with handshake signals.
STGIS	0	Store contents of accumulator in GIC PORT 0 synchronized with handshake signals.

5.2.2 TICC Macro Instructions

STXMT		Store contents of accumulator in asynchronous transmitter buffer.
LDRCV		Load contents of asynchronous receiver buffer to accumulator.
STOUT		Store contents of accumulator in TICC output port.
LDIN		Load data present at TICC input port to accumulator.
STTIM	ntimer	Store contents of accumulator in specified timer and start timer.
STIMR		Store contents of accumulator in TICC interrupt mask register.
LDSTA		Load TICC status register to accumulator.
STTCM		Store contents of accumulator in TICC command register.
STCRR		Store contents of accumulator in communications rate register.
LDIRP		Load TICC interrupt pending register to acc.

5.3 GIC MACRO EXPANSIONS *

```
GICC      MACRO     AM, BM
          DB        3EH, AMx8+BM+80H, 32H, 03H, 1CH
          ENDM      MVI A,
                    STA 1C03H
BSET      MACRO     NB
          DB        3EH, NBx2+1, 32H, 03H, 1CH
          ENDM      MVI A, NBx2+1
                    STA 1C03H
BCLR      MACRO     NB
          DB        3EH, NBx2, 32H, 03H, 1CH
          ENDM      MVI A, NBx2
                    STA 1C03H
LDGI      MACRO     NP
          DB        3AH, NP, 1CH          LDA   1C(NP)H
          ENDM

STGI      MACRO     NP
          DB        32H, NP, 1CH          STA   1C(NP)H
          ENDM

LDGIS     MACRO
          DB        3AH, 0, 5CH          LDA   5C00H
          ENDM

STGIS     MACRO
          DB        32H, 0, 5CH         STA   5C00H
          ENDM
```

*copyright 1976, Data Applications International.

This program is provided to users of DAI's DCE as a programming aid. No other use is authorized without DAI's written consent.

Note : The above macro expansions are applicable for DCE-1 and DCE-2.

The GIC register addresses are different for DCE-X.

5.4 TICC MACRO EXPANSIONS*

```
STXMT   MACRO
        DB          32H, 6, 98H
        ENDM

LDRCV   MACRO
        DB          3AH, 0, 98H
        ENDM

STOUT   MACRO
        DB          32H, 7, 98H
        ENDM

LDIN    MACRO
        DB          3AH, 1, 98H
        ENDM

STTIM   MACRO       NT
        DB          32H, NT+8, 98H
        ENDM

STIMR   MACRO
        DB          32H, 8, 98H
        ENDM

LDSTA   MACRO
        DB          3AH, 3, 98H
        ENDM

STTCM   MACRO
        DB          32H, 4, 98H
        ENDM

STCRR   MACRO
        DB          32H, 5, 98H
        ENDM

LDIRP   MACRO
        DB          3AH, 2, 98H
        ENDM
```

Note: The above macro expansions are applicable for DCE-1 and DCE-2. The TICC register addresses are different for DCE-X.

5.5 REPERTOIRE OF DCE INSTRUCTIONS

Z80 Assembler

MOVE GROUP

(hl)

reg A	B	C	D	E	H	L	M	ASSEMBLFR FORM	FUNCTION		S	Z	AC	P	CY
7F	78	79	7A	7B	7C	7D	7E	MOV A, reg	(A)←(reg)	Ld [illegible], reg	-	-	-	-	-
47	40	41	42	43	44	45	46	MOV B, reg	(B)←(reg)		-	-	-	-	-
4F	48	49	4A	4B	4C	4D	4E	MOV C, reg	(C)←(reg)		-	-	-	-	-
57	50	51	52	53	54	55	56	MOV D, reg	(D)←(reg)		-	-	-	-	-
5F	58	59	5A	5B	5C	5D	5E	MOV E, reg	(E)←(reg)		-	-	-	-	-
67	60	61	62	63	64	65	66	MOV H, reg	(H)←(reg)		-	-	-	-	-
6F	68	69	6A	6B	6C	6D	6E	MOV L, reg	(L)←(reg)	l, reg	-	-	-	-	-
77	70	71	72	73	74	75	-	MOV M, reg	(M)←(reg)	→ ld (hl), reg.	-	-	-	-	-

ACCUMULATOR GROUP

87	80	81	82	83	84	85	86	ADD reg	(A)←(A) + (reg)		+	+	+	+	+
8F	88	89	8A	8B	8C	8D	8E	ADC reg adc	(A)←(A) + (reg) + (CY)		+	+	+	+	+
97	90	91	92	93	94	95	96	SUB reg	(A)←(A) - (reg)		+	+	+	+	+
9F	98	99	9A	9B	9C	9D	9E	SBB reg sbc	(A)←(A) - (reg) - (CY)		+	+	+	+	+
A7	A0	A1	A2	A3	A4	A5	A6	ANA reg And.	(A)←(A) ∩ (reg)		+	+	0	+	0
AF	A8	A9	AA	AB	AC	AD	AE	XRA reg xor.	(A)←(A) ⊕ (reg)		+	+	0	+	0
B7	B0	B1	B2	B3	B4	B5	B6	ORA reg or.	(A)←(A) ∪ (reg)		+	+	0	+	0
BF	B8	B9	BA	BB	BC	BD	BE	CMP reg cp	(A) - (reg)		+	+	+	+	+

INCREMENT/DECREMENT REGISTER

3C	04	0C	14	1C	24	2C	34	INR reg	(reg)←(reg) + 1	INC reg	+	+	+	+	-
3D	05	0D	15	1D	25	2D	35	DCR reg	(reg)←(reg) - 1	DEC reg	+	+	+	+	-

REGISTER PAIR GROUP

rp = bc, de, hl, sp

bc B	de D	hl rp H	sp SP	af. PSW	ASSEMBLFR FORM	FUNCTION		S	Z	AC	P	CY
03	13	23	33	-	INX rp	(rp)←(rp) + 1	INC rp	-	-	-	-	-
0B	1B	2B	3B	-	DCX rp	(rp)←(rp) - 1	DEC rp	-	-	-	-	-
0A	1A	-	-	-	LDAX rp	(A)←((rp))	Ld a, (rp)	-	-	-	-	-
02	12	-	-	-	STAX rp	((rp))←(A)	ld (rp), a	-	-	-	-	-
09	19	29	39	-	DAD rp	(H,L)←(H,L)+(rp)		-	-	-	-	+
C5	D5	E5	-	F5	PUSH rp	((SP)-1)←(rh),((SP)-2)←(rl), (SP)←(SP)-2		-	-	-	-	-
C1	D1	E1	-	F1	POP rp	(rl)←((SP)),(rh)←((SP)+1), (SP)←(SP)+2		-	-	-	-	-

IMMEDIATE GROUP

				S	Z	AC	P	CY
3E dd	MVI A, data	(A)← data	ld reg, data.	-	-	-	-	-
06 dd	MVI B, data	(B)←data		-	-	-	-	-
0E dd	MVI C, data	(C)←data		-	-	-	-	-
16 dd	MVI D, data	(D)←data		-	-	-	-	-
1E dd	MVI E, data	(E)←data		-	-	-	-	-
26 dd	MVI H, data	(H)←data		-	-	-	-	-
2E dd	MVI L, data	(L)←data		-	-	-	-	-
36 dd	MVI M, data	(M)←data	ADD a, data or data	-	-	-	-	-
C6 dd	ADI data	(A)←(A) + data		+	+	+	+	+
CE dd	ACI data	(A)←(A) + data + (CY)		+	+	+	+	+
D6 dd	SUI data	(A)←(A) - data		+	+	+	+	+
DE dd	SBI data	(A)←(A) - data - (CY)		+	+	+	+	+
E6 dd	ANI data	(A)←(A) ∩ data		+	+	0	+	0
EE dd	XRI data	(A)←(A) ⊕ data		+	+	0	+	0
F6 dd	ORI data	(A)←(A) ∪ data		+	+	0	+	0
FE dd	CPI data	(A) - data		+	+	+	+	+
01 al ah	LXI B, addr	(B)←ah,(C)←al		-	-	-	-	-
11 al ah	LXI D, addr	(D)←ah,(E)←al		-	-	-	-	-
21 al ah	LXI H, addr	(H)←ah,(L)←al		-	-	-	-	-
31 al ah	LXI SP,addr	(SP_h)←ah,(SP_l)←al		-	-	-	-	-

DIRECT ADDRESS GROUP

				S	Z	AC	P	CY
3A al ah	LDA addr	(A)←(addr)	ld a,(address)	-	-	-	-	-
32 al ah	STA addr	(addr)←(A)	ld (address, a)	-	-	-	-	-
2A al ah	LHLD addr	(L)←(addr), (H)←(addr + 1)		-	-	-	-	-
22 al ah	SHLD addr	(addr)←(L), (addr + 1)←(H)		-	-	-	-	-

HEX CODES		ASSEMBLER FORM	FUNCTION		FLAGS S	Z	AC	P	CY
JUMP GROUP									
C3 al ah		JMP addr	(PC)←addr	JP addrs	-	-	-	-	-
C2 al ah		JNZ addr	If Z=0, (PC)←addr	z, addrs	-	-	-	-	-
CA al ah		JZ addr	If Z=1, (PC)←addr		-	-	-	-	-
D2 al ah		JNC addr	If CY=0, (PC)←addr	nc,	-	-	-	-	-
DA al ah		JC addr	If CY=1, (PC)←addr	c,	-	-	-	-	-
E2 al ah		JPO addr	If P=0, (PC)←addr	p,	-	-	-	-	-
EA al ah		JPE addr	If P=1, (PC)←addr	pe,	-	-	-	-	-
F2 al ah		JP addr	If S=0, (PC)←addr	p,	-	-	-	-	-
FA al ah		JM addr	If S=1, (PC)←addr		-	-	-	-	-
E9		PCHL ↳ jp(hl)	$(PC_h)\leftarrow(H), (PC_l)\leftarrow(L)$	p, m,	-	-	-	-	-
CALL GROUP									
CD al ah	CALL	CALL addr	(TOS)←(PC), (PC)←addr		-	-	-	-	-
C4 al ah		CNZ addr	If Z=0, (TOS)←(PC),(PC)←addr		-	-	-	-	-
CC al ah		CZ addr	If Z=1, (TOS)←(PC),(PC)←addr		-	-	-	-	-
D4 al ah	z.c jp	CNC addr	If CY=0,(TOS)←(PC),(PC)←addr		-	-	-	-	-
DC al ah		CC addr	If CY=1,(TOS)←(PC),(PC)←addr		-	-	-	-	-
E4 al ah		CPO addr	If P=0, (TOS)←(PC),(PC)←addr		-	-	-	-	-
EC al ah		CPE addr	If P=1, (TOS)←(PC),(PC)←addr		-	-	-	-	-
F4 al ah		CP addr	If S=0, (TOS)←(PC),(PC)←addr		-	-	-	-	-
FC al ah		CM addr	If S=1, (TOS)←(PC),(PC)←addr		-	-	-	-	-
RETURN GROUP									
C9	ret	RET	(PC)←(TOS)		-	-	-	-	-
C0		RNZ	If Z=0, (PC)←(TOS)		-	-	-	-	-
C8		RZ	If Z=1, (PC)←(TOS)		-	-	-	-	-
D0	z.c jp	RNC	If CY=0,(PC)←(TOS)		-	-	-	-	-
D8		RC	If CY=1,(PC)←(TOS)		-	-	-	-	-
E0		RPO	If P=0, (PC)←(TOS)		-	-	-	-	-
E8		RPE	If P=1, (PC)←(TOS)		-	-	-	-	-
F0		RP	If S=0, (PC)←(TOS)		-	-	-	-	-
F8		RM	If S=1, (PC)←(TOS)		-	-	-	-	-
RESTART GROUP									
C7		RST 0	(TOS)←(PC),(PC)←0_{16}		-	-	-	-	-
CF		RST 1	(TOS)←(PC),(PC)←8_{16}		-	-	-	-	-
D7		RST 2	(TOS)←(PC),(PC)←10_{16}		-	-	-	-	-
DF		RST 3	(TOS)←(PC),(PC)←18_{16}		-	-	-	-	-
E7		RST 4	(TOS)←(PC),(PC)←20_{16}		-	-	-	-	-
EF		RST 5	(TOS)←(PC),(PC)←28_{16}		-	-	-	-	-
F7		RST 6	(TOS)←(PC),(PC)←30_{16}		-	-	-	-	-
FF		RST 7	(TOS)←(PC),(PC)←38_{16}		-	-	-	-	-
ROTATE/CONTROL/ SPECIAL GROUP									
07		RLC	$(A_{n+1})\leftarrow(A_n),(A_0)\leftarrow(A_7),(CY)\leftarrow(A_7)$		-	-	-	-	+
0F	rrca	RRC	$(A_n)\leftarrow(A_{n+1}),(A_7)\leftarrow(A_0),(CY)\leftarrow(A_0)$		-	-	-	-	+
17		RAL	$(A_{n+1})\leftarrow(A_n),(A_0)\leftarrow(CY),(CY)\leftarrow(A_7)$		-	-	-	-	+
1F		RAR	$(A_n)\leftarrow(A_{n+1}),(A_7)\leftarrow(CY),(CY)\leftarrow(A_0)$		-	-	-	-	+
00		NOP	No operation		-	-	-	-	-
76		HLT	Processor stopped until interrupt or reset.		-	-	-	-	-
F3		DI	Interrupts disabled		-	-	-	-	-
FB		EI	Interrupts enabled after next instruction		-	-	-	-	-
E3	ex hl,(sp)	XTHL	(L)↔((SP)),(H)↔((SP)+1)		-	-	-	-	-
F9	ex hl,(sp)	SPHL	(SP)←(H) (L)		-	-	-	-	-
EB	ex hl,de	XCHG	(H)↔(D),(L)↔(E)		-	-	-	-	-
27		DAA	Decimal adjust accumulator		+	+	+	+	+
2F		CMA	$(A)\leftarrow(\bar{A})$		-	-	-	-	-
37		STC	(CY)←1		-	-	-	-	1
3F		CMC	$(CY)\leftarrow(\overline{CY})$		-	-	-	-	+

MACRo's

HEX CODES	ASSEMBLER FORM	FUNCTION	FLAGS				
			S	Z	AC	P	CY
TICC GROUP							
32 06 98	STXMT	(Transmit buffer)←(A)	-	-	-	-	-
3A 00 98	LDRCV	(A)←(Receive buffer)	-	-	-	-	-
32 07 98	STOUT	(output port)←(A)	-	-	-	-	-
3A 01 98	LDIN	(A)←(input port)	-	-	-	-	-
32 09 98	STTIM 1	(Timer 1)←(A)	-	-	-	-	-
32 0A 98	STTIM 2	(Timer 2)←(A)	-	-	-	-	-
32 0B 98	STTIM 3	(Timer 3)←(A)	-	-	-	-	-
32 0C 98	STTIM 4	(Timer 4)←(A)	-	-	-	-	-
32 0D 98	STTIM 5	(Timer 5)←(A)	-	-	-	-	-
3A 03 98	LDSTA	(A)←(TICC status)	-	-	-	-	-
32 04 98	STTCM	(TICC Command)←(A)	-	-	-	-	-
32 05 98	STCRR	(Rate register)←(A)	-	-	-	-	-
3A 02 98	LDIPR	(A)←(Interrupt pending)	-	-	-	-	-
32 08 98	STIMR	(Intr. Mask reg.)←(A)	-	-	-	-	-
GIC GROUP							
3E cd 32 03 1C	GICC am,bm	(GICC Cmd reg)←cd;cd=80+8 * am+bm	-	-	-	-	-
3E cc 32 03 1C	BCLR n	(P2Bn)←0;cc=2*n	-	-	-	-	-
3E cs 32 03 1C	BSET n	(P2Bn)←1;cs=2*n+1	-	-	-	-	-
3A 0m 1C	LDGI m	(A)←(Port m)	-	-	-	-	-
32 0m 1C	STGI m	(Port m)←(A)	-	-	-	-	-
3A 0m 5C	LDGIS m	(A)←(Port m)	-	-	-	-	-
32 0m 5C	STGIS m	(Port m)←(A)	-	-	-	-	-

5.6 INITIALIZATION OF THE DCE

Do IT!

The DCE contains an automatic power-on reset feature. When power is applied to the DCE, the following conditions are established:

- 8080 CPU Program Counter is set to zero.
- 8080 CPU Interrupts are disabled.
- All GIC ports are in input mode (high impedance state).
- All TICC registers contain random values.
- All RAM memory locations contain random values.
- 8080 CPU Registers, Accumulator, Flags, and Stack Pointer contain random values.

After Reset, the program will start executing, starting at memory address 0000 (F000 for DCE-LSA systems). The first few program steps must load the control registers of the GIC and TICC with appropriate values in order to set up the desired DCE configuration.

The TICC device does not have a hardware Reset signal. It must be Reset by software, via Bit 0 of its Command Register. After Power-on reset, this reset bit in the TICC Command Register may not necessarily be zero. It must be cleared, and then set to ensure correct TICC Reset.

The following steps must be included in the initialization sequence of the users program (unless program is entered via DCE Utility Program):

1. Clear TICC Interrupt Mask Register to zero.
2. Clear TICC Command Register to zero.
3. Delay 1 second (nominal) while the Reset line settles for Real-World interface cards connected to the DCE-BUS. This may be omitted if no Real-World cards are present.

4. Initialize the TICC by loading the TICC Command Register with a value having Bit 0 equal to 1 and Bits 4 to 7 equal to zero. Select Bits 1,2,3 for conditions desired, (see Section 3. 4. 2).

5. Load the TICC Communications Rate Register to set the desired Baud rate and the number of stop bits (see Section 3. 4. 3).

6. Load the TICC Interrupt Mask Register to enable desired interrupts (see Figure 6-2). Logical one in a bit position enables the corresponding interrupt.

7. Configure the GIC for the desired I/O mode (see Section 2. 7. 2).

8. Initialize the Stack Pointer. Load a value one greater than the highest RAM location (1200H for DCE-1 and 1800H for DCE-2). The first byte pushed on to the Stack will then be saved in the highest RAM location, since the Stack Pointer is decremented before data or an address is pushed on to the Stack.

9. Enable 8080 CPU interrupts, if desired.

5. 6. 1 A Typical DCE Start-Up Sequence

The following is a typical initialization procedure starting at address 0000.

```
        JMP     INIT
        .
        .
        .
INIT:   XRA     A
        STIMR              zero interrupt mask register.
        STTCM              zero TICC command register.
                           delay about 1 second
        LXI     H,0FFFFH
DELAY:  LXI     D,0FFFFH   D,E = -1
        DAD     D
        JC      DELAY

        MVI     A,0DH      reset TICC, select IN7 and interrupt
                           ack. enable
        STTCM
        MVI     A,1        select 110 Baud, 2 stop bits
        STCRR
        MVI     A,0C0H     enable IN7 and timer 4 interrupts
        STIMR
        GICC    1,0        configure GIC ports
        LXI     SP,1800H   initialize Stack Pointer (DCE-2)
        EI                 enable CPU interrupts
        .
        .
        .
```

After execution of this program segment, the TICC will be configured for operation with interrupts. IN7 is selected instead of Timer 5 as the interrupt 7 source. The serial I/O is programmed for 110 Baud with 2 stop bits. The interrupt mask register is programmed to pass interrupt no. 6 and 7 (see Figure 6-2). GIC Ports 0,1,2 (B0-B3) are configured as output, Port 2 (B4-B7) as input.

The first instruction at address 0000 in the previous example program is a Jump. It is necessary because address space from 0000 to 0008 (in PROM) is the interrupt 0 vector area. If none of the interrupts are used, this Jump instruction is not necessary, and the program could start directly at address 0000.

If Timer 1 is to be used, the program segment starting at address 0000 must first determine whether a power-on reset or a time-out has occurred. It can then jump to the initialization routine or Timer 1 service routine accordingly. One possible technique for realizing such a scheme is explained at the end of Section 3.2.2.

Note: All DCE Utility programs perform the DCE start-up procedure during initialization. If, subsequently, the user program performs a software reset of TICC via the reset bit (bit 0) of the TICC Command Register, then all other TICC registers must be re-initialized to desired values.

5.7 PROGRAMMING DCE FOR REAL-TIME MESSAGE TRANSFER

5.7.1 INTRODUCTION

One of the most important features of the DCE is the interrupt controlled serial communications I/O circuits. These circuits allow the DCE to input and output variable length messages on a character-by-character basis simultaneously, while executing programs that realize the work for which it is directed. In other words, the DCE can communicate in real time with other computers, remote printers, data acquisition devices and with other DCE cards.

In a typical application the DCE must control an isolated process, collect data, and transmit messages to a central computer system for further data processing. In turn, it must receive the control parameters from the central system to adjust its control function accordingly. In such an application both the serial input and output must simultaneously co-exist. Also, the length of the input and output messages can be variable and under the control of both the programmer and the input source. Furthermore, the DCE must continue its control function to which it is dedicated.

5.7.2 ARCHITECTURE

To realize simultaneous message transfer and control program execution, the TICC must be programmed to interrupt the CPU each time a character has been sent and each time a character has been received. This can be realized by setting up the TICC mask register to pass interrupts 4 and 5 (load logical ONE into bit positions 4 and 5) and starting the interrupt service routines in the corresponding PROM vector area (section 3. 2. 1 on page 3-10).

The routines needed to realize this application are: Message Initialization routine, Input Driver Routine and Output Driver Routine. The Message Initialization Routine is contained within the mainline control program. The Input and Output Driver Routines are interrupt driven.

5.7.2.1 Message Initialization Routine

The purpose of the message initialization routine is to initialize the control parameters needed by the Input and Output Driver Routines each time the main-line program desires to transmit a message or receive a message on the serial ports of the TICC. In our example the control parameters are: message length, first-character address of outgoing message, first-character storage address of incoming message.

An additional task of the Message Initialization Routine is to call the Output Driver Routine once each time a message unit is to be transmitted. This action causes the Driver to transmit the first character. The Driver will be activated by the TICC transmit-buffer-empty interrupt to transmit the rest of the characters.

5.7.2.2 Output Driver Routine

The task of the Output Driver Routine is to respond to the transmit-buffer-empty interrupt generated by the TICC, to send out the next character in the message string, to check if there are more characters to be transmitted, and to turn off the output message active flag that was set by the Message Initialization Routine.

5.7.2.3 Input Driver Routine

The task of the Input Driver Routine is to respond to the receiver-buffer-loaded interrupt generated by the TICC, to store the received character into the next sequential message storage location, to realize that the end of message has occurred,and to turn off the input message active flag that was set by the Message Initialization Routine.

5.7.3 DESIGN

5.7.3.1 Message Initialization Routine

The Message Initialization Routine is initiated by the main-line program under execution in the DCE at a moment when it wishes to send a message or receive a message on the serial ports of the DCE. Illustrated in Fig.5-1, page 5-15 is a flow diagram of the Message Initialization Routine.

When the main program is ready to send or receive a message string via the TICC serial port it must first check if the appropriate routine is free, or if it is still engaged in transmitting or receiving a previous message. The main program does this by examining the appropriate message active flag. If it is not set, the respective driver routines are free. In this case it places the address of the first string character and the message length into a predetermined location in the RAM memory. It then sets the appropriate message active flag and initiates execution of the first character.

5.7.3.2 Output Driver Routine

The Output Driver Routine is initiated by recognition of the interrupt by the TICC after it has completed the output of a character. Illustrated in Fig.5-2, page 5-16 is a flow diagram of the Output Driver Routine.

The output routine sends the character from the address specified in the length register to the TICC transmitter register, it then checks if there are more characters to be sent and returns control to the interrupted program. While the TICC is shifting out the character, the DCE continues with the main program.

When the transmitter register is empty the TICC interrupts the CPU and returns control to the output driver routine for transmitting the next character. When all the characters have been transmitted the output routine resets the OUT MSG flag as an indication to the main program that it is possible to accept the next desired message. Note that when the TICC interrupts the CPU after the last character, the output routine, receiving the command, sees the OUT MSG flag reset and returns control to the main program immediately.

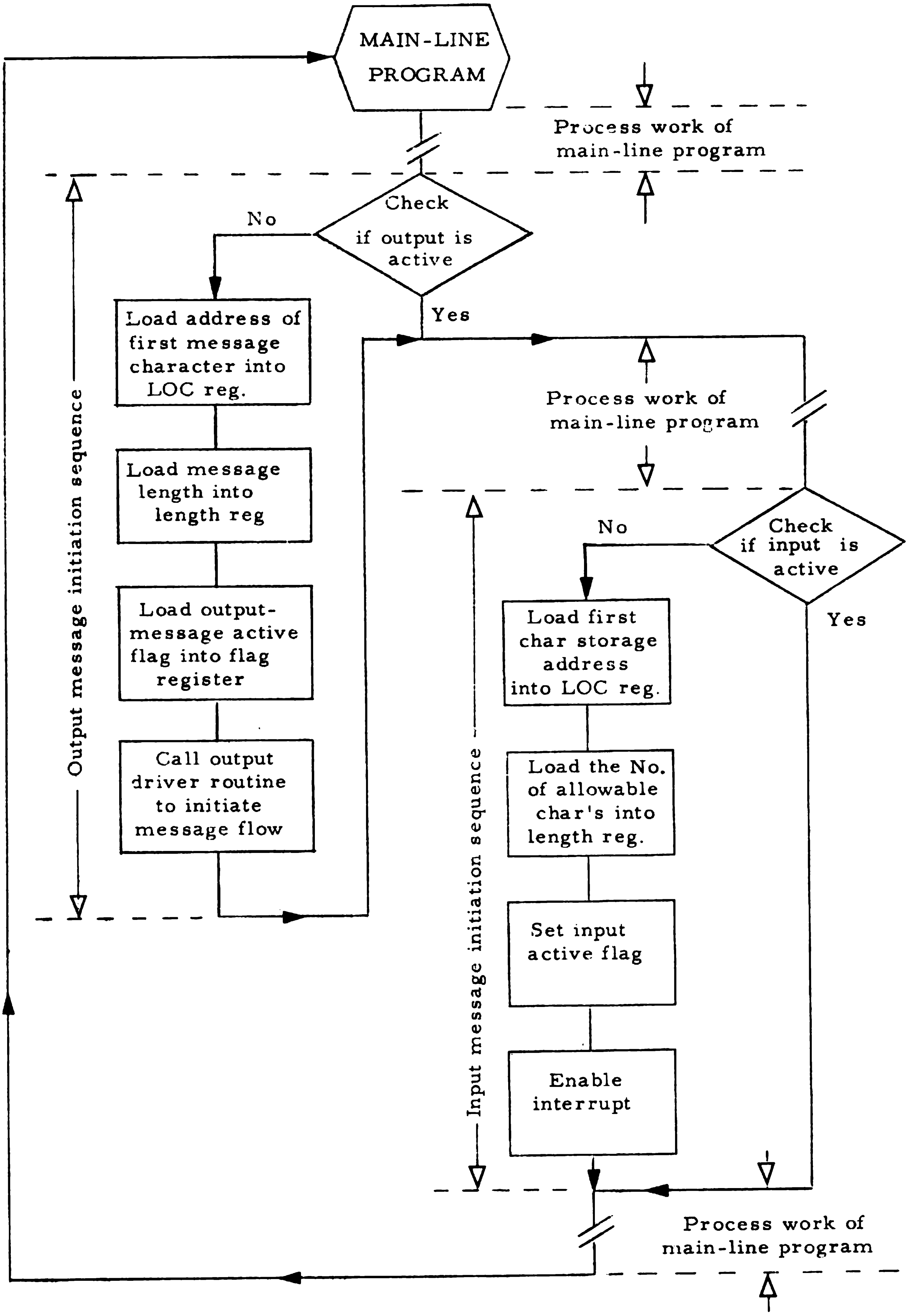
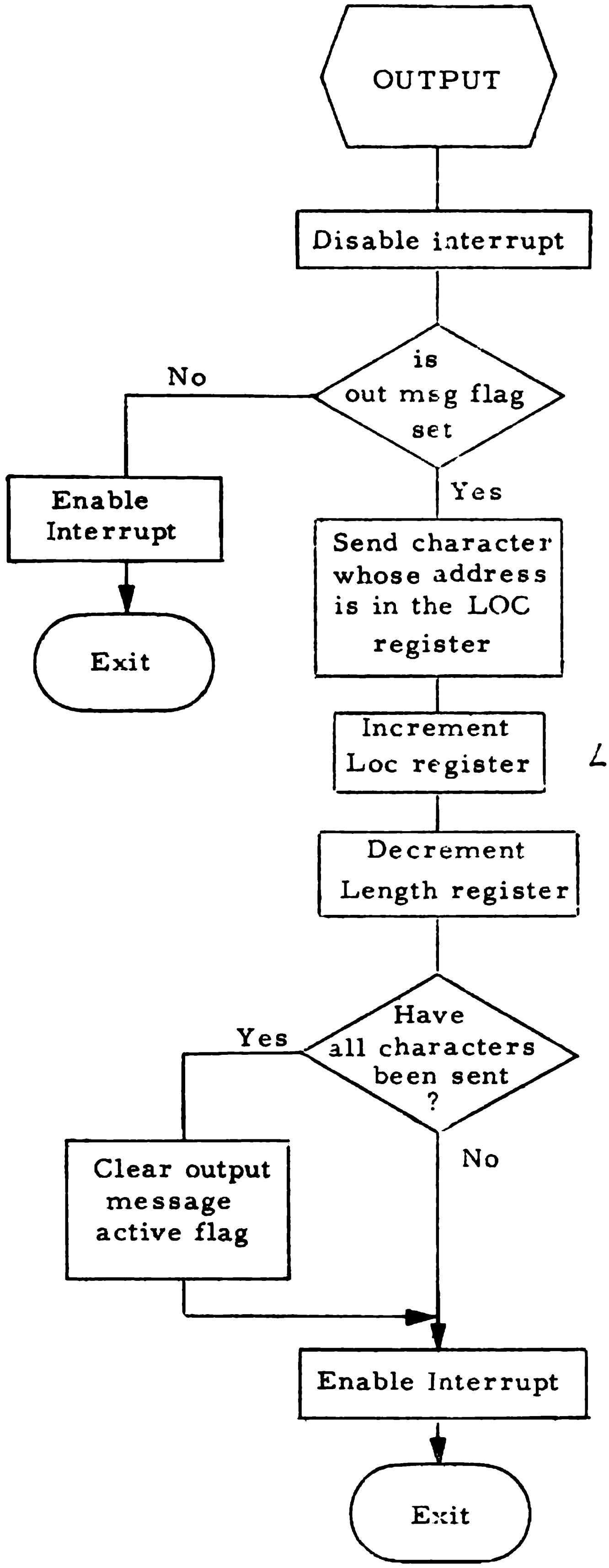

MAIN-LINE PROGRAM
Process work of main-line program
Check if output is active
No
Yes
Load address of first message character into LOC reg.
Load message length into length reg
Load output-message active flag into flag register
Call output driver routine to initiate message flow
Output message initiation sequence
Process work of main-line program
Check if input is active
No
Yes
Load first char storage address into LOC reg.
Load the No. of allowable char's into length reg.
Set input active flag
Enable interrupt
Input message initiation sequence
Process work of main-line program

Figure 5-1 Message Initialization Routine Flow Chart



Local

Figure 5-2 Output Driver Routine Flow Chart

5.7.3.3 Input Driver Routine

The Input Driver Routine is initiated by the recognition of an interrupt by the TICC after it has received a character from the serial input port. Illustrated in Fig.5-3, page 5-18 is a flow diagram of the Input Driver Routine.

The input routine is called automatically by the TICC when it has received a character from the serial input port. It then checks to see if the characte is a special end-of-message character. (The end of message character ma be any character desired by the programmer, although it is not necessary to have it at all). If the character is the special end of message character, the input routine clears the Input -message-active flag and returns control to the main-line program. When the end of message has not been specified,the input routine stores the received character into the memory location indicated by the contents of the ILOC register and increments the ILOC register to set up the address for the next coming character of the message. The input routine then decrements the length register and checks to determine if all allowable characters have been received. In the event that this condition is true, the input routine clears the Input-message-active flag and transfers control to the main-line program. In the event that this is false it simply transfers control back to the main-line program.

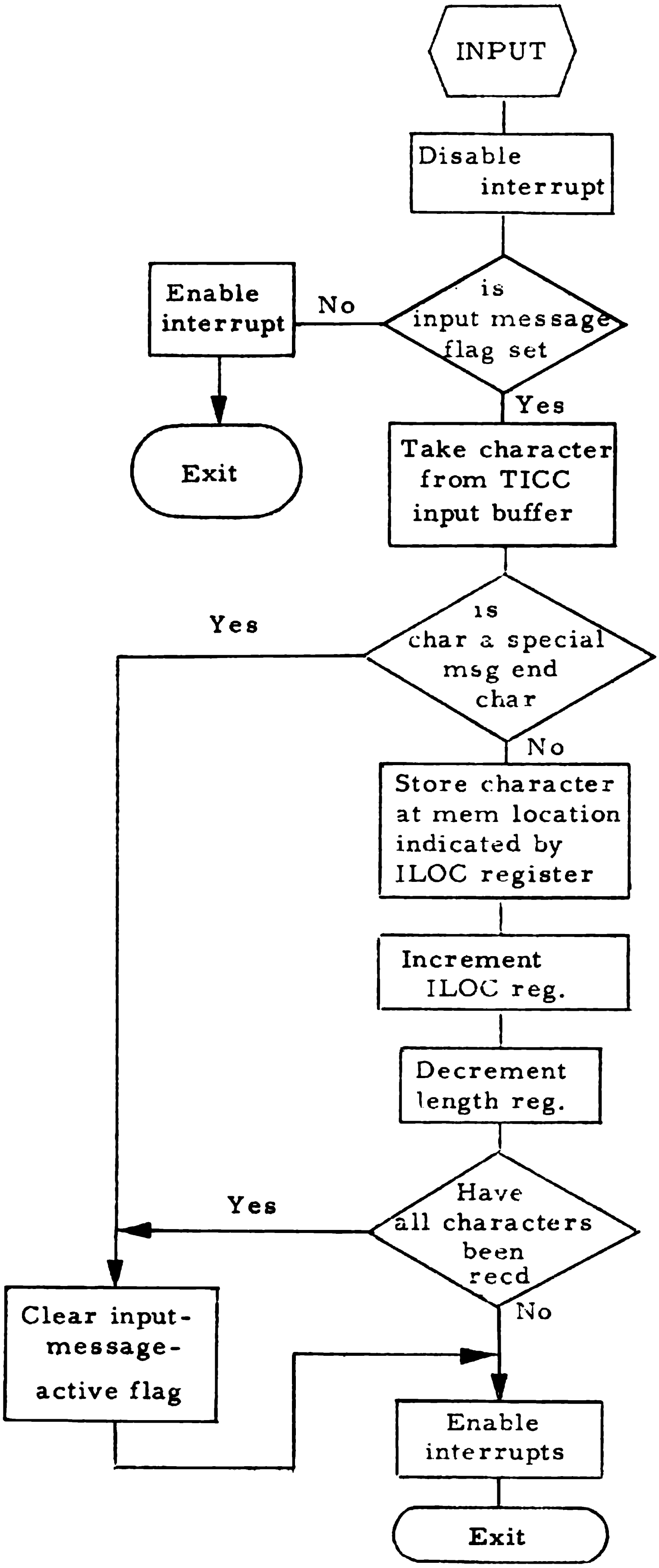
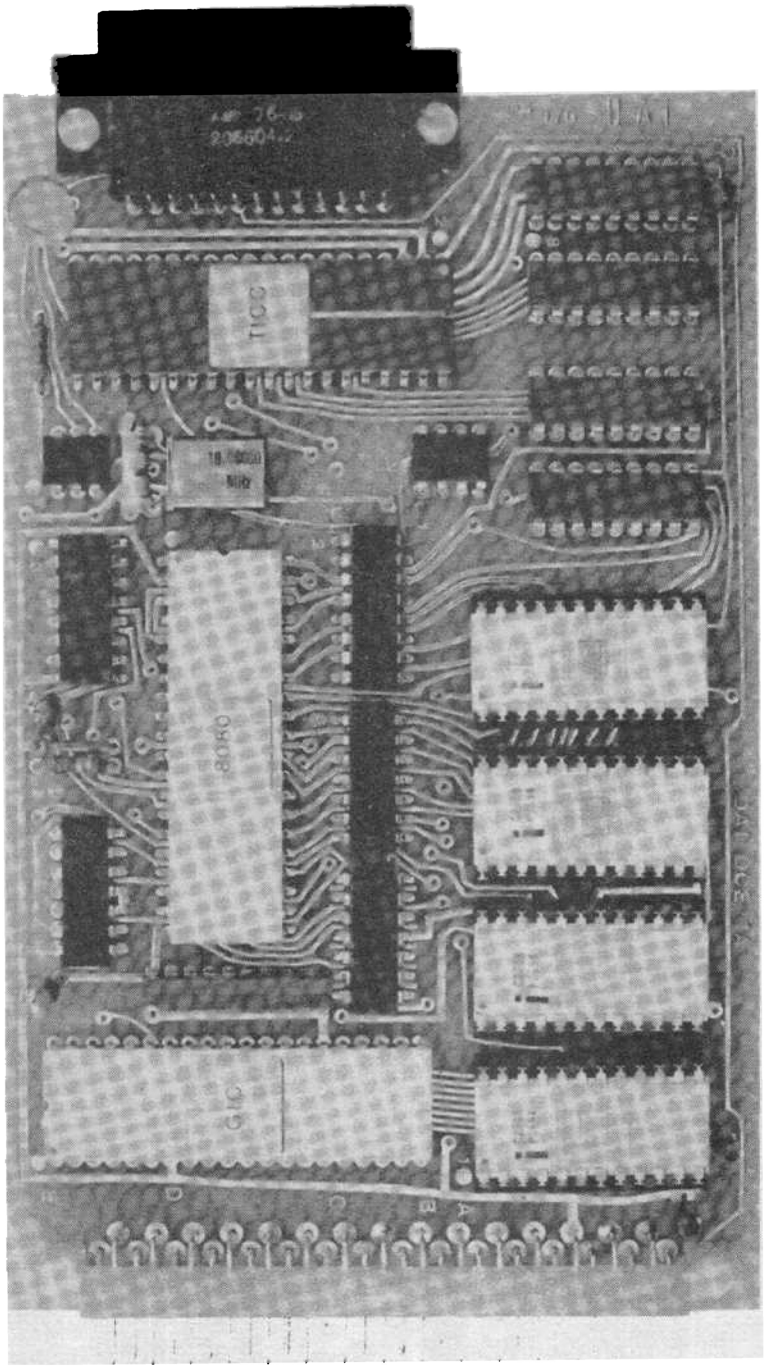

INPUT
Disable interrupt
is input message flag set
No
Enable interrupt
Exit
Yes
Take character from TICC input buffer
is char a special msg end char
Yes
No
Store character at mem location indicated by ILOC register
Increment ILOC reg.
Decrement length reg.
Have all characters been recd
Yes
No
Clear input-message-active flag
Enable interrupts
Exit

Figure 5-3 Input Driver Routine Flow Chart



DCE-1, Actual Size

6. DCE PROCESSOR MODULES

6.1 GENERAL SPECIFICATIONS

The DCE single eurocard (100 x 160 mm) microcomputer family provides the digital computing power of the popular 8080 microprocessor enhanced by fully implemented serial and parallel I/O capability, five independent interval timers and fully vectored interrupts. The opto-isolated serial communication interface has programmable Baud rates, and may be interrupt driven. The interval timers function independently of the CPU, and provide 64 microsecond resolution with crystal accuracy. There are 40 parallel I/O lines, 24 of which can be programmed as input, output, bi-directional or handshaking ports, with automatic handshake control signals.

All DCE processor modules can be interchanged without system modification. Source programs written for one DCE processor need only be re-assembled for any other DCE processor, and no back-panel or connector re-wiring is necessary.

DCE processor modules provide a standard hardware and software interface to the family of "Real-World" interface modules via the DCE-BUS. They plug into the parallel-wired DCE eurocard racks and are package compatible with all the support modules. Each DCE processor is tested and fully burned-in before delivery.

The exchange of data and control information between the DCE processor modules and the interface modules is achieved via the DCE-BUS. The DCE-BUS is realized by configuring the GIC ports on the DCE processor module in a specific manner. Complete specifications of the DCE-BUS are given in Section 4.

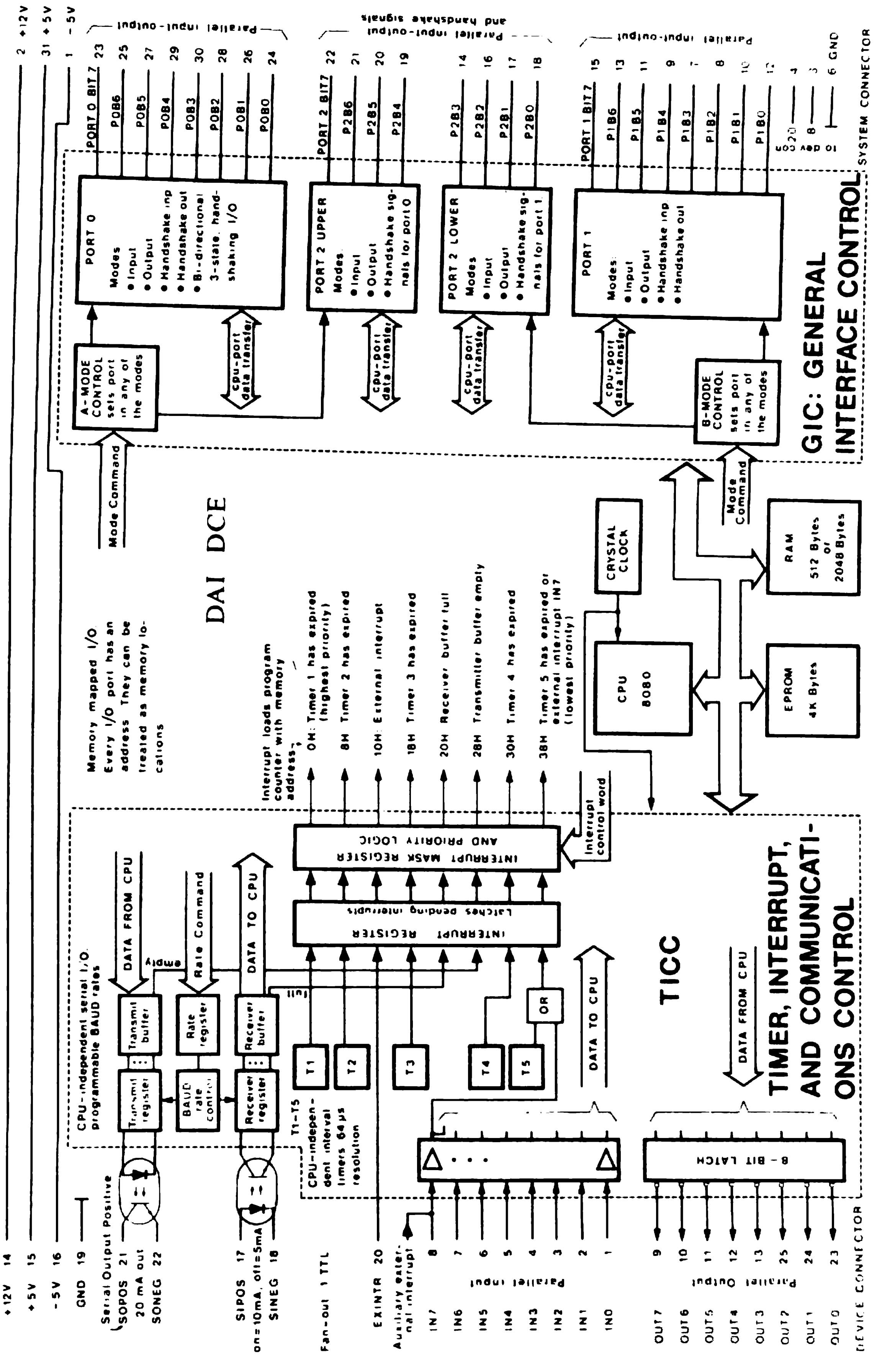
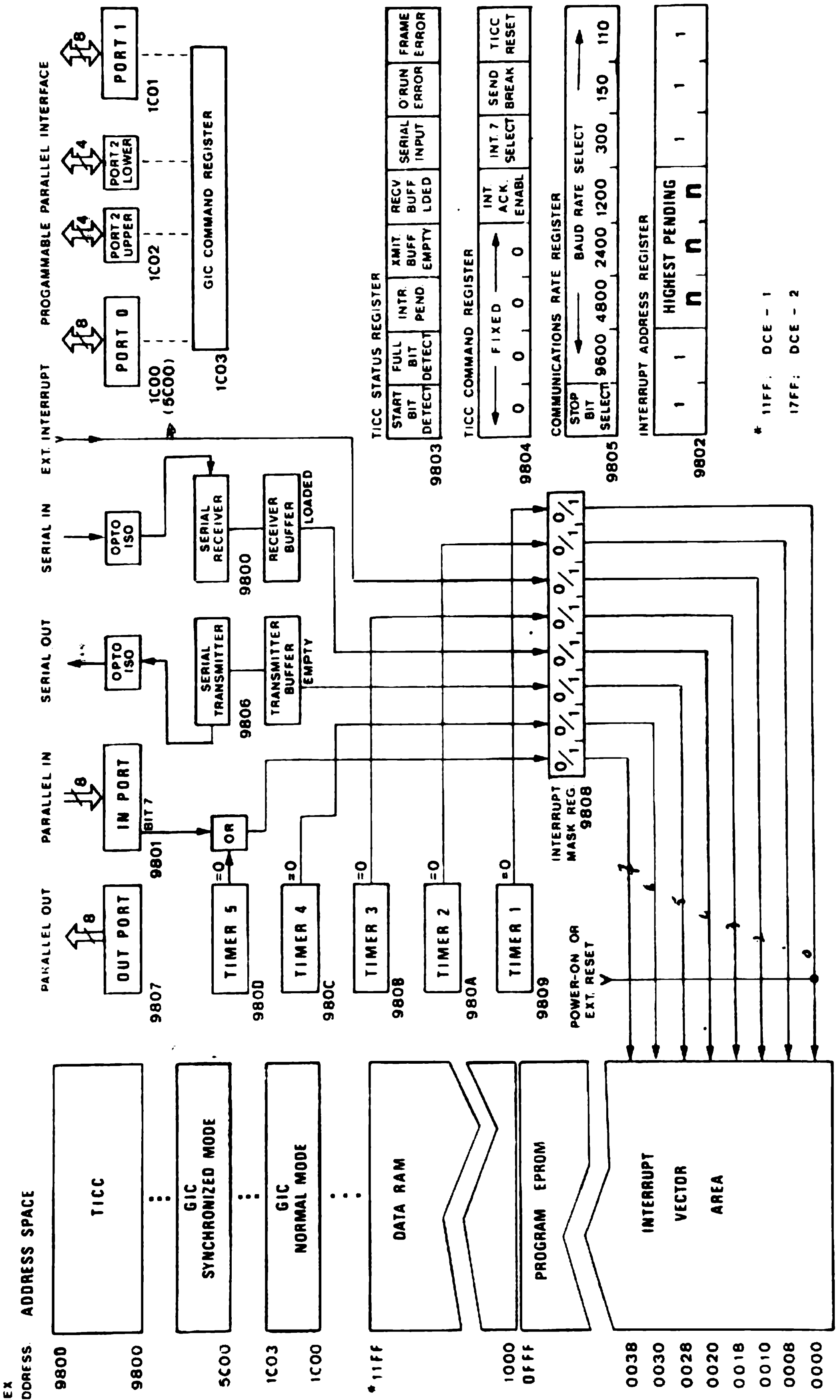


Figure 6-1. DCE Detailed Block Diagram



TICC STATUS REGISTER — 9803

START BIT DETECT	FULL BIT DETECT	INTR. PEND	XMIT. BUFF EMPTY	RECV BUFF LDED	SERIAL INPUT	O'RUN ERROR	FRAME ERROR

TICC COMMAND REGISTER — 9804

← FIXED →				INT. ACK. ENABL	INT. 7 SELECT	SEND BREAK	TICC RESET
0	0	0	0				

COMMUNICATIONS RATE REGISTER — 9805

STOP BIT SELECT	← BAUD RATE SELECT →						
	9600	4800	2400	1200	300	150	110

INTERRUPT ADDRESS REGISTER — 9802

		HIGHEST PENDING					
1	1	n	n	n	1	1	1

* 11FF: DCE - 1
17FF: DCE - 2

Figure 6-2. Memory and Register Diagram

REGISTER/PORT	ADDRESS (HEX)	
	DCE-1 DCE-2	**DCE-1A DCE-2A DCE-X**
GIC Command Register	1C03	**FF03**
GIC Port 0	1C00	FF00
GIC Port 1	1C01	FF01
GIC Port 2	1C02	FF02
GIC Port 0 (Sync. Mode)	5C00	FF08
TICC Serial Receiver	9800	**FF10**
TICC Parallel Input Port	9801	**FF11**
TICC Interrupt Address Register	9802	FF12
TICC Status Register	9803	FF13
TICC Command Register	9804	**FF14**
TICC Communications Rate Register	9805	FF15
TICC Serial Transmitter	9806	FF16
TICC Parallel Output Port	9807	**FF17**
TICC Interrupt Mask Register	9808	FF18
TICC Timer 1	9809	FF19
TICC Timer 2	980A	FF1A
TICC Timer 3	980B	FF1B
TICC Timer 4	980C	FF1C
TICC Timer 5	980D	FF1D

Table 6-1 : GIC and TICC Register Addressing for Different DCE Processor Modules

6.1.1 Detailed DCE Ί lock Diagram

Figure 6-1 shows the detailed block diagram for 8080 based DCE modules. The PROM and RAM memory capacity indicated is applicable to DCE-1 , 1A, 2, 2A modules only. The memory expandable DCE-X has no memory provided on the module. A flat-cable X-BUS connector is provided on the DCE-X for connecting memory modules to provide upto 64K bytes of memory.

6.1.2 DCE Memory and Register Diagram

Figure 6-2 shows the general memory and register diagram for DCE-1 and DCE-2 modules. The TICC and GIC register addresses, PROM memory addresses and the RAM memory start address are the same for DCE-1 and DCE-2. The TICC and GIC register addresses for DCE-X, 1A, 2A are different from those in Figure 6-2. Table 6-1 gives the addressing for the different processor modules. Register address differences between the different DCE processor modules are automatically resolved by the resident assembler (UAE) in the DCE-DM development system, when translating TICC and GIC macros. For this reason, users are advised to use the TICC and GIC macros specified in Section 5.2, when accessing the TICC and GIC registers.

6.1.3 DCE Program Memory

The DCE **modules** that accept program memory (DCE-1, DCE-2 etc.) have sockets provided for erasable and electrically reprogrammable read-only memory (EPROM). The sockets accommodate the 2708 1K byte, the 2716 2K byte, or pin compatible devices. These are packaged in 24 pin dual-in-line case with a transparent lid, which allows the user to expose the memory chip to ultraviolet light to erase the bit-pattern. The EPROMs can then be re-programmed. The erase-rewrite cycle can be repeated as many times as required.

For EPROM programming use DAI's low cost microcomputer development system, the DCE-DM. The DCE-DM system design includes facilities for testing, reviewing, and altering the program before actually programming the PROM. For complete information on these develoment systems refer to the DCE-DM User's Manual.

To erase the bit-pattern, expose the chip through the transparent quartz lid, to high intensity short-wave ultra-violet light at a wavelength of 2537Å. Among recommended ultra-violet light sources are Model UVS and Model S-52 lamps manufactured by Ultra-Violet Products Inc.
The lamps should be used without short-wave filters, and the EPROM to be erased should be placed about one inch away from the lamp tubes for about 20 to 30 minutes.

Figure 6-3 serves as reference for inserting the program memory into the sockets on the DCE card. The diagram shows the card as viewed from the component side. PROM position zero is the low order position referenced by start address 0000. Pin 1 of the PROM is shown as reference for insertion.

Program memory on the DCE-X processor resides on the memory modules connected via the flat-cable X-BUS. These memory modules have switch selectable address ranges, and therefore the program memory can be made to reside in any desired memory range.

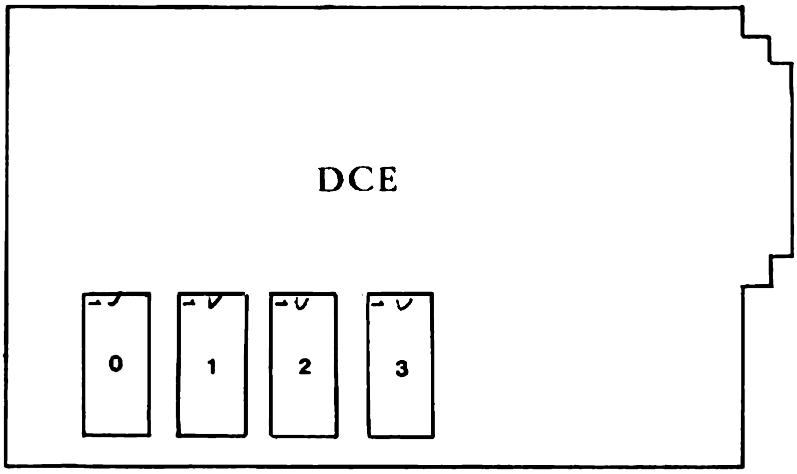


Figure 6-3 : Definition of Program Memory Position and Pin Reference for Insertion of EPROMs

6.1.4 System Connector Pin Definitions

Each DCE processor module has a 31-pin connector at one end. This is the System Connector which plugs into the DCE eurocard racks with the parallel-wired DCE-BUS.

The following pin definitions are valid for all DCE processor modules and DCE-BUS compatible interface modules.

SIGNAL NAME	DESCRIPTION		PIN	PC K1
P0B0	General Interface PORT 0	Bit 0	24	1
P0B1		Bit 1	26	2
P0B2		Bit 2	28	3
P0B3		Bit 3	30	4
P0B4		Bit 4	29	5
P0B5		Bit 5	27	6
P0B6		Bit 6	25	7
P0B7		Bit 7	23	8
P1B0	General Interface PORT 1	Bit 0	12	9
P1B1		Bit 1	10	10
P1B2		Bit 2	8	11
P1B3		Bit 3	7	12
P1B4		Bit 4	9	13
P1B5		Bit 5	11	14
P1B6		Bit 6	13	15
P1B7		Bit 7	15	16
P2B0	General Interface PORT 2	Bit 0	18	17/27
P2B1		Bit 1	17	28
P2B2		Bit 2	16	29
P2B3		Bit 3	14	30
P2B4		Bit 4	19	31
P2B5		Bit 5	20	32
P2B6		Bit 6	21	33
P2B7		Bit 7	22	34
EXINTR*	External Interrupt		4	
IN7 *	Parallel Input Bit 7 (aux. interrupt)		3	
EXRESET	External Reset (Ground for Reset)		5	
+12 *	+12V DC		2	–
+ 5 *	+ 5V DC		31	21/22/23/24
- 5 *	- 5V DC		1	–
GND *	Digital and Power Ground		6	18/19/20

*Signal present on both connectors in DCE processors.

6.1.5 Device Connector Pin Definitions

Each DCE processor module has a 25 pin D-type male connector at one end. This is the Device Connector carrying the serial I/O lines and the TICC parallel port lines.

The following pin definitions are valid for all 8080 based DCE processors.

SIGNAL NAME	DESCRIPTION		PIN
SOPOS	Serial Output (Positive)		21
SONEG	Serial Output (Negative)		22
SIPOS	Serial Input (Positive)		17
SINEG	Serial Input (Negative)		18
OUT0	TICC Parallel Output Port	Bit 0	23
OUT1		Bit 1	24
OUT2		Bit 2	25
OUT3		Bit 3	13
OUT4		Bit 4	12
OUT5		Bit 5	11
OUT6		Bit 6	10
OUT7		Bit 7	9
IN0	TICC Parallel Input Port	Bit 0	1
IN1		Bit 1	2
IN2		Bit 2	3
IN3		Bit 3	4
IN4		Bit 4	5
IN5		Bit 5	6
IN6		Bit 6	7
IN7 *	Auxiliary Interrupt, or,	Bit 7	8
EXINTR*	External Interrupt		20
+12 *	+12 from System Connector		14
+ 5 *	+ 5 from System Connector		15
- 5 *	- 5 from System Connector		16
GND *	Signal Ground		19

* Signal present on both connectors.

6.1.6 System Bus Electrical Specifications

SIGNAL	MIN.	TYP.	MAX.	CONDITIONS
GIC Ports 0-2:				
Input Mode			0.8V	Low input
	2.0V			High input
Output Mode			0.4V	-1.6mA
	2.4V			+50µA

6.1.7 Device Connector Electrical Specifications

SIGNAL	MIN.	TYP.	MAX.	CONDITIONS
Parallel Output			0.45V	-1.7mA
	3.7V			400µA
Parallel Input			0.8V	Low input
	3.3V			High input
Input Leakage			±10µA	0-5V input
Serial Input:				
ON state		10mA	15mA	Forward Current
		1.2V	1.3V	Forward Voltage
OFF state	2mA	5mA		Forward Current
	1.0V	1.1V		Forward Voltage
			20mA	Absolute max. Current
Serial Output:				
Forward Voltage	0.7V		1.4V	10mA (ON state)
			30V	Absolute max. (OFF state)
Reverse Voltage			7V	Absolute Maximum
Forward Current			100mA	Absolute Maximum

6.2 DCE-1 PROCESSOR MODULES

6.2.1 FEATURES

- complete 8080 microcomputer system on a single 100 x 160 mm eurocard.
- 512 byte RAM and sockets for 4K or 8K byte EPROM
- opto-isolated serial I/O with programmable Baud rates from 110 to 9600.
- 40 parallel I/O lines; 24 programmable as simple handshaking or bi-directional with automatic handshake control signals.
- 5 independent interval timers providing 64 microsecond resolution with crystal accuracy.
- 8 independently vectored interrupts.

6.2.2 FUNCTIONAL BLOCK DIAGRAM

Figure 1-2 on page 1-6 shows the general block diagram. Figure 6-1 shows a detailed functional block diagram. RAM memory occupies address space 1000H to 11FFH for DCE-1, and D000H to D1FFH for DCE-1A. EPROM memory starts from address 0000.

6.2.3 SYSTEM DESIGN PARAMETERS

6.2.3.1 Hardware Configuration

The DCE-1 processor module features two powerful LSI subsystems:

- Timer, Interrupt and Communication Control (TICC)
- General Interface Control (GIC)

Complete specifications of the GIC and TICC are given in Sections 2 and 3 of this manual.

The 24 programmable parallel I/O lines are provided by the GIC, and are available at the System Connector. The TICC also provides 8 parallel input lines and 8 parallel output lines. These are available at the Device Connector.

The DCE-1 has a memory-mapped I/O architecture and therefore all the TICC and GIC registers, and the I/O ports can be accessed simply as memory locations. This enables the usage of the powerful memory instruction set of the 8080 CPU, when accessing device registers and I/O ports.

6.2.3.2 Programming Specifications

See Section 5 for full DCE programming specifications.

If the DCE-1 processor module is connected to interface modules via the DCE-BUS, the GIC must be configured in a specific manner. See Section 4 for full details.

The GIC and TICC register addresses are as shown in Table 6-1.

6.2.3.3 Special Considerations

The 8080 CPU requires on Wait state when accessing the slower RAM memory on the earlier DCE-1s. This is automatically taken care of by the logic on the module. RAM memory is usually used for temporary data storage and Stack inmplementation, and therefore these Wait states do not cause any problem at all. However, when attempting to develop and run programs in RAM which utilise one or more TICC interrupts on these earlier versions of DCE-1 there is a possiblity of occasional loss of some interrupts. This is because the TICC interrupt control logic is not ideally suited to taking into account the possible presence of a CPU Wait state when transfering the interrupt RST instruction to the CPU. These earlier versions of DCE-1 can be recognised by the presence of a 74LS02 instead of a 74LS00 on the card.

6.2.3.4 Operational Requirements

Power Requirements

The DCE-1 requires three power supplies: +5V ± 5%, -5V ± 5%, +12V ± 5%. They are usually provided by the DCE-PWR module via the DCE-BUS. Typical power requirements in the quiescent state are given below. Active state values are typically 20% higher.

Number of EPROMs (2708)	+5V	-5V	+12V	unit
0	355	0.1	60	mA
1	355	30	110	mA
2	355	60	160	mA
3	355	90	210	mA
4	355	120	260	mA

Environmental Requirements

Operating temperature	0°C to 55°C
Storage temperature	-25°C to +85°C
Relative humidity	: 95% noncondensing

6.2.4 ORDERING INFORMATION

DCE-1 :	Standard Version with 4K EPROM space (4 x 2708)
DCE-1/8S :	with socket-mounted 8080
DCE-1A :	with 8K EPROM space (4 x 2716)
DCE-1A/8S :	with socket-mounted 8080

Serial Interface cables for standard terminals, Utility software package, Assembler/Editor, FORTRAN and BASIC must be ordered separately.

6.3 DCE-2 PROCESSOR MODULES

6.3.1 FEATURES

- complete 8080 microcomputer system on a single 100 x 160 mm eurocard.
- 2K byte RAM and sockets for 4K or 8K byte EPROM.
- opto-isolated serial I/O with programmable Baud rates from 110 to 9600.
- 40 parallel I/O lines; 24 programmable as simple, handshaking or bi-directional with automatic handshake control signals.
- 5 independent interval timers providing 64 microsecond resolution with crystal accuracy.
- 8 independently vectored interrupts.

6.3.2 FUNCTIONAL BLOCK DIAGRAM

Figure 1-2 on page 1-6 shows the general block diagram. Figure 6-1 shows a detailed functional block diagram. RAM memory occupies address space 1000H to 17FFH for DCE-2, and D000H to D7FFH for DCE-2A. EPROM memory starts from ADDRESS 0000.

6.3.3 SYSTEM DESIGN PARAMETERS

6.3.3.1 Hardware Configuration

The DCE-2 processor module features two powerful LSI subsystems:

- Timer, Interrupt and Communication Control (TICC)
- General Interface Control (GIC)

Complete specifications of the GIC and TICC are given in Sections 2 and 3 of this manual.

The 24 programmable parallel I/O lines are provided by the GIC, and are available at the System Connector. The TICC also provides 8 parallel input lines and 8 parallel output lines. These are available at the Device Connector.

The DCE-2 has a memory-mapped I/O architecture and therefore all the TICC and GIC registers, and the I/O ports can be accessed as simple memory locations. This enables the usage of the powerful memory instructions set of the 8080 CPU, when accessing device registers and I/O ports.

The 2K byte RAM memory on the DCE-2 is fast enough for 8080 CPU access without requiring any Wait states.

6.3 3.2 Programming Specifications

See Section 5 for full DCE programming specifications.

If the DCE-2 processor module is connected to interface modules via the DCE-BUS, the GIC must be configured in a specific manner. See Section 4 for full details.

The GIC and TICC register addresses are as shown in Table 6-1.

6.3.3.3 Operational Requirements

Power Requirements

The DCE-2 requires three power supplies: +5V ±5%, -5V ±5%, +12V ±5%. They are usually provided by the DCE-PWR module via the DCE-BUS. Typical power requirements in the quiescent state are given below. Active state values are typically 20% higher.

Number of EPROMs (2708)	+5V	-5V	+12V	unit
0	460	0.1	70	mA
1	460	30	120	mA
2	460	60	170	mA
3	460	90	220	mA
4	460	120	270	mA

Environmental Requirements

Operating temperature	:	0°C to 55°C
Storage temperature	:	-25°C to +85°C
Relative humidity	:	95% noncondensing

6.3.4 ORDERING INFORMATION

DCE-2	:	Standard Version with 4K EPROM space (4 x 2708)
DCE-2/8S	:	with socket-mounted 8080
DCE-2A	:	with 8K EPROM space (4 x 2716)
DCE-2A/8S	:	with socket-mounted 8080

Serial interface cables for standard terminals, Utility software package, Assembler/Editor, FORTRAN and BASIC must be ordered separately.

6.4 DCE-X PROCESSOR MODULE

6.4.1 FEATURES

- memory expandable 8080 microcomputer system on a single 100 x 160 mm eurocard.
- hardware, software and pin compatible with DCE-lA and DCE-2A microcomputers.
- memory expandable up to 64K bytes with MX PROM and/or RAM memory modules.
- flat-cable X-BUS connection to support 1 to 8 memory modules.
- opto-isolated serial I/O with programmable Baud rates from 110 to 9600.
- 40 parallel I/O lines; 24 programmable as simple, handshaking, or bidirectional, with automatic handshake control signals.
- 5 independent internal timers providing 64 microsecond resolution with crystal accuracy.
- 8 independently vectored interrupts.

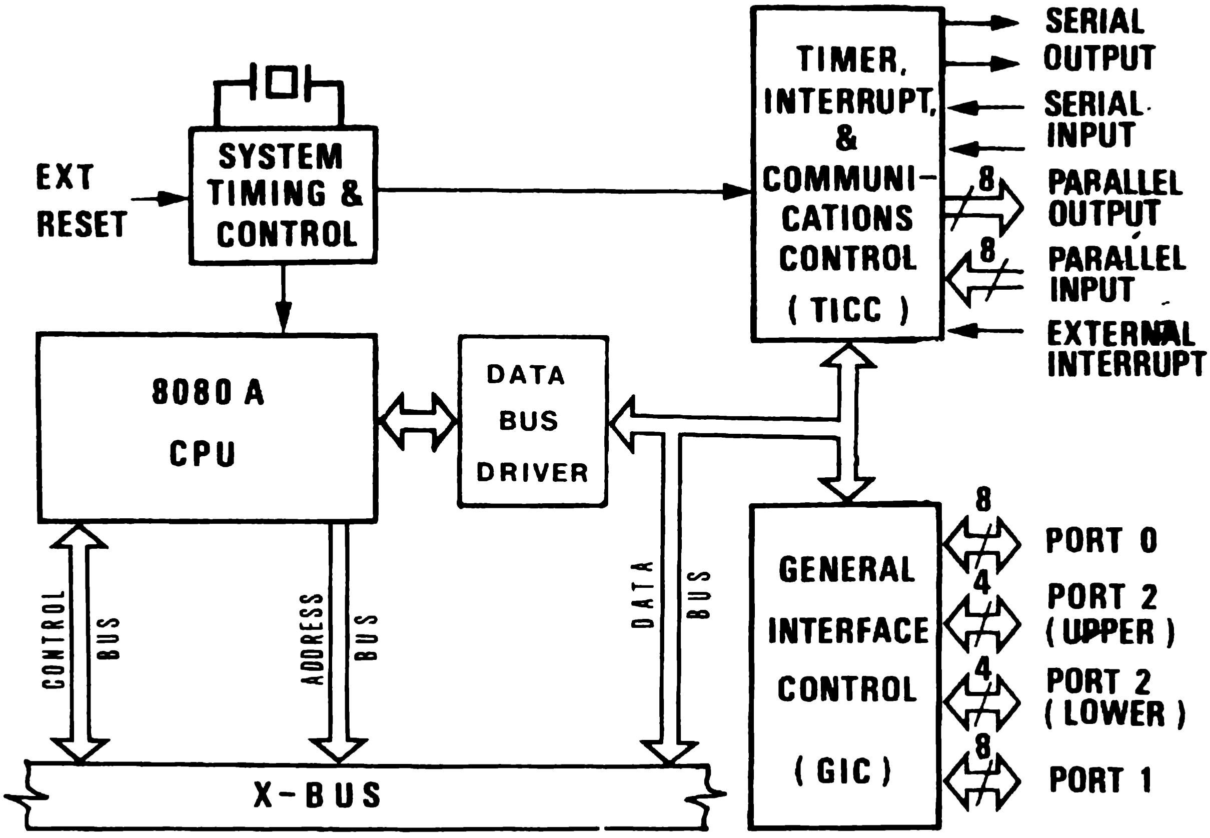


Figure 6-3 : Functional Block Diagram

6.4.2 FUNCTIONAL BLOCK DIAGRAM

Figure 6-3 shows the functional block diagram. The detailed block diagram of Figure 6-1 applies also to the DCE-X, except for the absence of memory on the DCE-X module.

6.4.3 SYSTEM DESIGN PARAMETERS

6.4.3.1 Hardware Configuration

The DCE-X is functionally equivalent to and plug compatible with DCE-1A and DCE-2A processor modules. All DCE-1A, DCE-2A documentation and specifications apply to the DCE-X with the exception of the resident memory on the module.

The DCE-X forms the base of the memory-expandable DCE series, with up to 64K bytes of PROM and RAM memory in any combination. A flat-cable X-BUS connector allows the DCE-X to be linked with up to eight MX memory modules in any combination, to provide up to a maximum of 64K bytes of memory. No memory is provided on the DCE-X. Standard X-BUS cables are available for connecting memory modules. All X-BUS signals are buffered. The cables are kept short with critical signals screened, to eliminate noise problems.

6.4.3.2 Programming Specifications

The TICC and GIC register addresses for the DCE-X are different from those given in Figure 6-2. These register address differences between the different DCE processor modules are automatically resolved by the resident Assembler in the DCE-DM development system, when translating TICC and GIC macros. Table 6-1 shows the TICC and GIC register addressing for the DCE-X.

The DCE-X address space is organised to allow optimum usage of external memory. A summary of DCE-X address space is given below:

ADDRESS (HEX)	ALLOCATION
0000 - FEFF	External Memory
FF00 - FF03	GIC Register and Ports
FF08	GIC Port 0 with synchronised I/O
FF10 - FF1D	TICC Registers, Ports and Timers

For GIC and TICC addressing, address bits 5,6,7 are don't care states.

6.4.3.3 X-BUS Signal Allocation

The DCE user need not normally be concerned with the X-BUS characteristics. This information is provided only for those users who intend designing their own X-BUS compatible modules.

PIN NUMBER	MNEMONIC	DESCRIPTION
1	GROUND	screening line
2	D0	data bit 0
3	GROUND	screening line
4	D1	data bit 1
5	GROUND	screening line
6	D2	data bit 2
7	GROUND	screening line
8	D3	data bit 3
9	GROUND	screening line
10	D4	data bit 4
11	GROUND	screening line
12	D5	data bit 5
13	GROUND	screening line
14	D6	data bit 6
15	GROUND	screening line
16	D7	data bit 7
17	GROUND	screening line
18	—	no connection
19	GROUND	screening line
20	MEMW	memory Write strobe
21	GROUND	screening line
22	DBIN	CPU input strobe
23	A10	address line 10
24	$\overline{\text{MEMR}}$	memory Read strobe
25	A14	address line 14
26	A11	address line 11
27	A12	address line 12
28	A13	address line 13
29	A9	address line 9
30	A15	address line 15
31	A7	address line 7
32	A8	address line 8
33	A5	address line 5
34	A6	address line 6
35	A3	address line 3
36	A4	address line 4
37	A1	address line 1
38	A2	address line 2
39	A0	address line 0
40	INTA	interrupt acknowledge
41	GROUND	screening line
42	$\overline{\text{WRQ}}$	Wait cycle request
43	WAIT	CPU Wait acknowledge
44	INTE	CPU interrupt enable
45	GROUND	screening line
46	$\overline{\text{HOLDA}}$	CPU Hold acknowledge
47	$\overline{\text{INT}}$	interrupt request
→ 48	CK2(TTL)	TTL level clock (2 MHz)
49	$\overline{\text{HOLD}}$	Hold request
50	SYNC	CPU SYNC signal

6.4.3.4 X-BUS Electrical Specifications

SIGNAL	MIN.	TYP.	MAX.	CONDITIONS
ADDRESS, INTE			0,45V	1.9 mA
	3.7V			-150µA
DATA OUTPUT		0.3V	0.45V	15mA
	3.65V	4.0V		-1mA
DATA INPUT			0.95V	
	2.0V			
DATA IN leakage				60µA
MEMW, $\overline{\text{MEMR}}$, INTA		0.35V	0.5V	8mA
	2.7V	3.4V		-400µA
$\overline{\text{HOLD}}$			0.4V	-1.2mA
	2.7V			0.25mA
$\overline{\text{HOLDA}}$		0.26V	0.4V	48mA
	2.4V	3.4V		-1.2mA
WAIT			0.4V	-0.4mA
	3.3V			
DBIN			0.45V	1.1mA
	3.7V			-190µA
CK2			0.45V	15mA
	2.4V			-1mA
SYNC			0.45V	1.65mA
	3.7V			-150µA
$\overline{\text{WRQ}}$			0.4V	-0.8mA
	2.7V			0.25mA
$\overline{\text{INT}}$			0.4V	-1.6mA
	2.4V			40µA

6.4.3.5 X-BUS Timing (relative to the 8080 CPU)

SIGNAL	TYP.	MAX.	CONDITIONS
DATA OUT	15ns	25ns	after 8080 output
DATA IN	20ns	30ns	before 8080 input
$\overline{\text{MEMR}}$, MEMW	10ns	20ns	relative to DBIN
$\overline{\text{HOLD}}$, $\overline{\text{HOLDA}}$	14ns	22ns	relative to 8080

6.4.3.6 Operational Requirements

Power requirements

The DCE-X requires three power supplies. These are usually provided by the DCE-PWR module via the DCE-BUS. Typical power requirements in the quiescent state are given below. Active state values are typically 20% higher.

+12V ± 5%	:	70mA
+5V ± 5%	:	375mA
-5V ± 5%	:	0.1mA

Environmental Requirements

Operating temperature	:	0°C to 55°C
Storage temperature	:	-25°C to +85°C
Relative humidity	:	95% noncondensing

6.4.4 ORDERING INFORMATION

DCE-X : Standard Version

DCE-X/8S : with socket mounted 8080

X-BUS(n) flat-cable bus connection between DCE-X and 'n' (n = 1 to 8) memory modules.
- must be ordered separately.

Interface cables for standard terminals, Utility software package, Assembler/Editor, FORTRAN and BASIC must be ordered separately.

6.5 MX-84 : COMBINATION MEMORY EXPANSION MODULE

6.5.1 FUNCTIONAL DESCRIPTION

The MX-84 module provides 8K bytes of PROM space and 4K bytes of static RAM as expansion memory for DCE-X processor based systems.

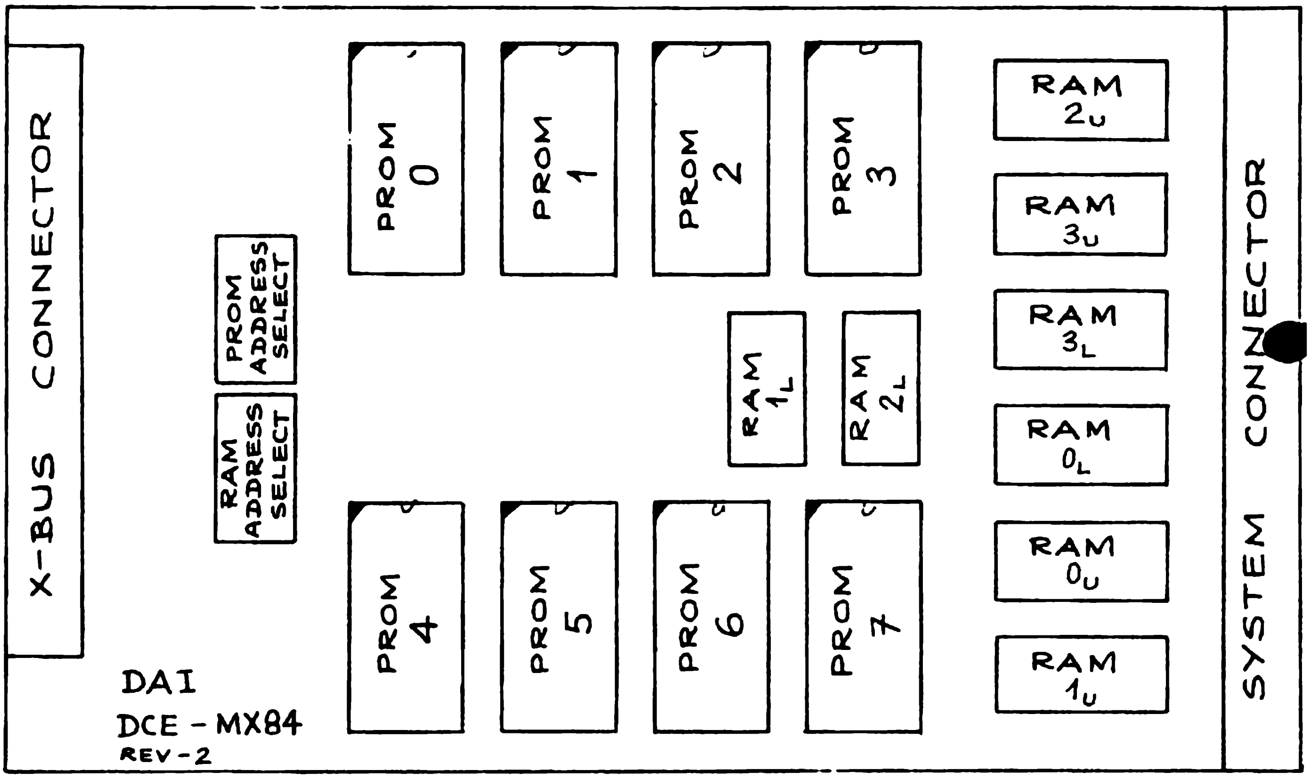
Up to eight 2708 or equivalent PROM/ROM devices can be inserted into the sockets provided. The PROM/ROM and RAM access times allow the 8080 CPU on the DCE-X to run at full speed. Two address select switches on the module allow the PROM and RAM memories to occupy any desired ranges in DCE-X memory address space.

One or more MX-84 modules, along with other MX memory modules, can be easily connected to a DCE-X processor via the flat-cable X-BUS connector.

6.5.2 FEATURES

- 8K byte PROM space and 4K byte fast RAM
- runs at full 8080 CPU speed (? 2MHz)
- switch selectable PROM memory base address
- switch selectable RAM memory base address
- X-BUS connection to DCE-X processor module
- single 100 x 160 mm eurocard format.

6.5.3 MODULE LAYOUT



6.5.4 ADDRESS SELECTION

The PROM and RAM base addresses can be separately selected using the two hexadecimal rotary switches located near the X-BUS connector on the module.

The upper switch, when the module is held with its name in the bottom left corner, controls the PROM addresses in steps of 8K bytes. The lower switch controls the RAM addresses in steps of 4K bytes.

PROM Base Address Selection

Each PROM socket on the MX-84 module is associated with a unique 1K address block relative to the selected PROM base address. The physical positions of the PROM sockets are as shown in the module layout diagram. PROM 0 occupies the lowest address range. All PROM sockets indicate the correct insertion orientation by a rounded internal corner at pin 1.

The PROM base address may be set to any 8K byte boundary as shown below:

Upper Switch Setting	PROM Start Address (Hex)	PROM End Address (Hex)
0	0000	1FFF
1	2000	3FFF
2	4000	5FFF
3	6000	7FFF
4	8000	9FFF
5	A000	BFFF
6	C000	DFFF
7*	E000	FFFF

0 → 4

7

* When using this range do not insert PROM 7, since it will conflict with addressing for the GIC and TICC registers on the DCE-X module.

RAM Base Address Selection

The MX-84 module has 4K bytes of fast static RAM, occupying a 4K address block relative to the RAM base address.

The RAM base address may be set to any 4K byte boundary as shown below:

Lower Switch Setting	RAM Start Address (Hex)	RAM End Address (Hex)
0	0000	0FFF
1	1000	1FFF
2	2000	2FFF
3	3000	3FFF
4	4000	4FFF
5	5000	5FFF
6	6000	6FFF
7	7000	7FFF
8	8000	8FFF
9	9000	9FFF
A	A000	AFFF
B	B000	BFFF
C	C000	CFFF
D	D000	DFFF
E	E000	EFFF
F*	F000	FFFF

PSC MAI B

C

* This setting should not be used due to the conflict with DCE-X GIC and TICC register addressing.

Boot - CFFF → RAM

6.5.5 OPERATIONAL REQUIREMENTS

Power Requirements

The MX-84 module requires three power supplies. These are obtained from the DCE-BUS via the System Connector. The module is not connected to any other signal on the DCE-BUS.

Typical power requirements in the quiescent state, without any PROMs, are given below. Active state values are typically 20% higher.

+5V	:	600mA
-5V	:	0
+12V	:	0

Environmental Requirements

Operating temperature	:	0°C to 55°C
Storage temperature	:	-25°C to +85°C
Relative humidity	:	up to 95% noncondensing

X-BUS Loading

SIGNAL	LEAKAGE	DRIVE (RAM)	LOAD (RAM)
Address Lines Data Lines Read, Write	one CMOS 80μA max.	2.1mA @ 0.4V -1mA @ 2.4V	one CMOS 80μA max. -360μA @ 0.4V

6.5.6 ORDERING INFORMATION

MX-84 : Standard Version, supplied without PROMs

The X-BUS(n) cable for connection between DCE-X and 'n' (1 to 8) memory modules must be ordered separately.

6.6 MXP-12 : 12K PROM MEMORY EXPANSION MODULE

6.6.1 FUNCTIONAL DESCRIPTION

The MXP-12 module provides 12K bytes of PROM space as expansion memory for DCE-X processor based systems. Up to twelve 2708 or equivalent PROM/RAM devices can be inserted into the sockets provided, and they allow the 8080 CPU on the DCE-X to run at full speed.

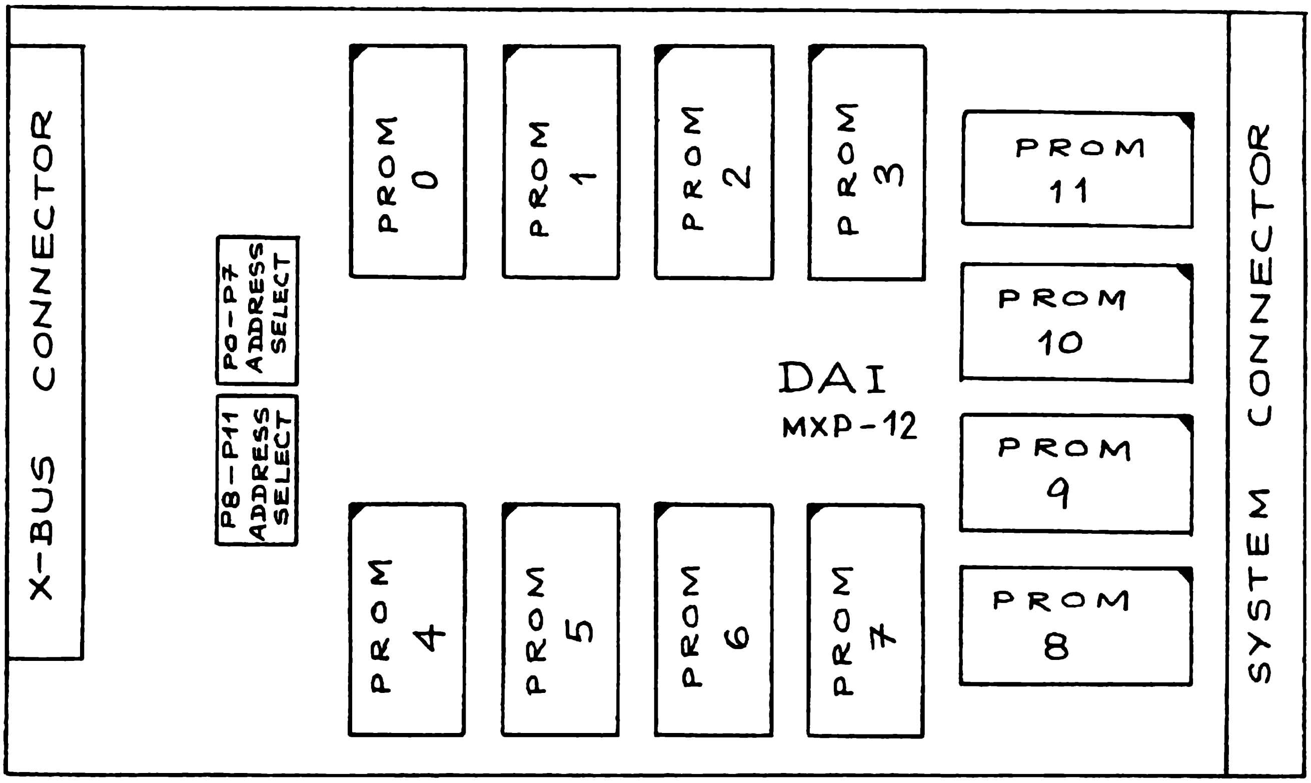
The memory space on the MXP-12 is organised into two banks of 4K and 8K bytes. Each bank can be assigned a separate base address by means of two rotary switches.

One or more MXP-12 modules, along with other MX memory modules, can be easily connected to a DCE-X processor via the flat-cable X-BUS connector.

6.6.2 FEATURES

- 12K PROM space, organised into two banks of 4K and 8K bytes
- separate switch selectable base addresses for the two memory banks
- runs at full 8080 CPU speed
- X-BUS connection to DCE-X processor module
- single 100 x 160 mm eurocard format.

6.6.3 MODULE LAYOUT



6.6.4 ADDRESS SELECTION

Each PROM socket on the MXP-12 module is associated with a unique 1K address block. PROMs 0 to 7 constitute the 8K bank, and PROMs 8 to 11 constitute the 4K bank. The physical positions of the PROM sockets are shown in the module layout diagram. PROMs 0 and 8 occupy the lowest address ranges in the two banks. All PROM sockets indicate the correct insertion orientation by a rounded internal corner at pin 1.

The base address for the 4K and 8K banks can be separately selected using the two hexadecimal rotary switches located near the X-BUS connector on the module.

The switch at the top, when the module is held so that the name can be correctly read, controls the base address of the 8K bank (PROM 0 to 7) in steps of 8K. The lower switch controls the base address of the 4K bank (PROM 8 to 11) in steps of 4K.

8K Memory Base Address Selection

The base address may be set to any 8K byte boundary as shown below:

Upper Switch Setting	8K Bank Start Address (Hex)	8K Bank End Address (Hex)
0	0000	1FFF
1	2000	3FFF
2	4000	5FFF
3	6000	7FFF
4	8000	9FFF
5	A000	BFFF
6	C000	DFFF
7*	E000	FFFF

* When using this range do not insert PROM 7, since it will conflict with addressing for the GIC and TICC registers on the DCE-X module.

4K Memory Base Address Selection

The base address may be set to any 4K byte boundary as shown below:

Lower Switch Setting	4K Bank Start Address (Hex)	4K Bank End Address (Hex)
0	0000	0FFF
1	1000	1FFF
2	2000	2FFF
3	3000	3FFF
4	4000	4FFF
5	5000	5FFF
6	6000	6FFF
7	7000	7FFF
8	8000	8FFF
9	9000	9FFF
A	A000	AFFF
B	B000	BFFF
C	C000	CFFF
D	D000	DFFF
E	E000	EFFF
F*	F000	FFFF

* When using this range do not insert PROM 11, since it will conflict with addressing for the GIC and TICC registers on the DCE-X module.

6.6.5 OPERATIONAL REQUIREMENTS

Power Requirements

The MXP-12 module requires three power supplies. These are obtained from the DCE-BUS via the System Connector. No other DCE-BUS signals are connected to this module.

Typical power requirements in the quiescent state, without any PROMs, are given below. Active state values are typically 20% higher.

+5V	100mA
-5V	0
+12V	0

Environmental Requirements

Operating temperature	0°C to 55°C
Storage temperature	-25°C to +85°C
Relative humidity	up to 95% noncondensing

X-BUS Loading

This depends on the PROMs used.

6.6.6 ORDERING INFORMATION

MXP-12 : Standard Version, supplied without PROMs.

The X-BUS(n) cable for connection between DCE-X and 'n' (1 to 8) memory modules must be ordered separately.

6.7 MXR-8 : 8K RAM MEMORY EXPANSION MODULE

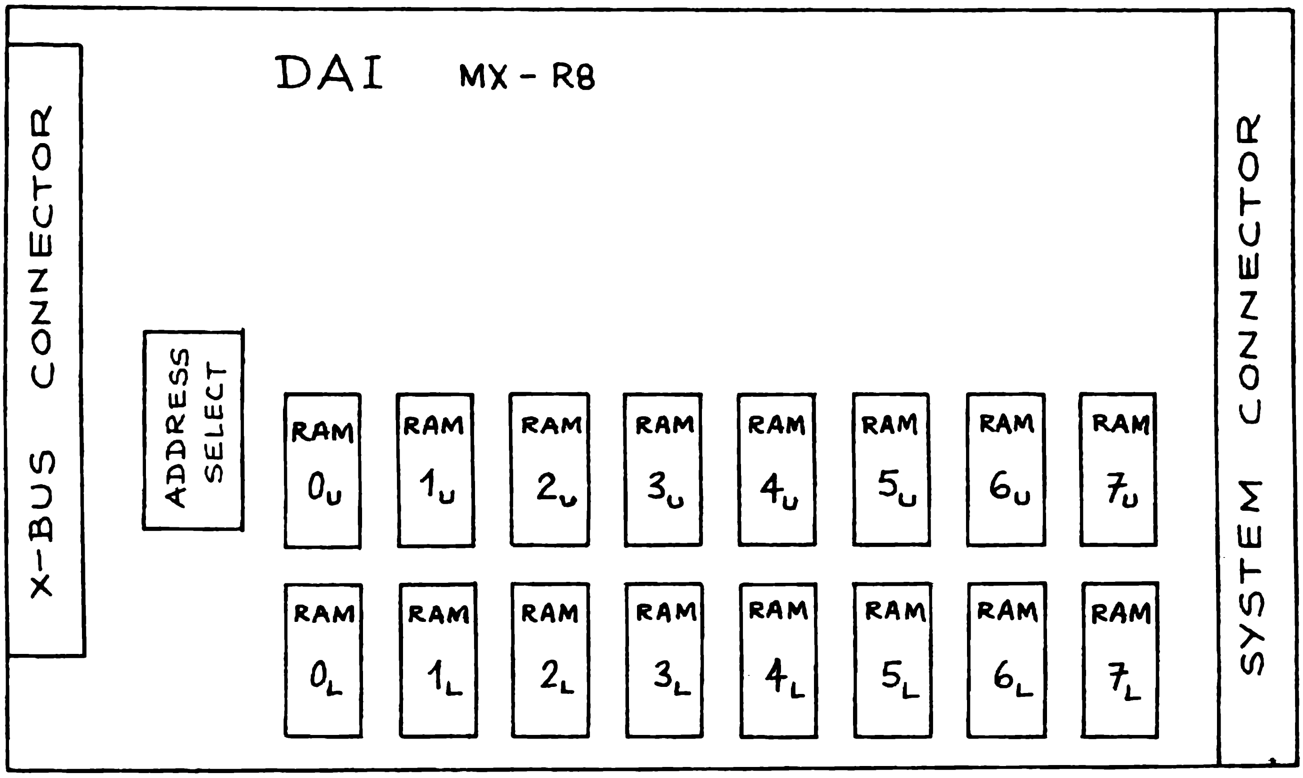
6.7.1 FUNCTIONAL DESCRIPTION

The MXR-8 module provides 8K bytes of static RAM as expansion memory for DCE-X processor based systems. The RAM on the module is fast enough to allow the 8080 CPU on the DCE-X to run at full speed. An address select switch allows the RAM to occupy any desired range in DCE-X memory address space. One or more MXR-8 modules, along with other MX memory modules, can be easily connected to a DCE-X processor via the flat-cable X-BUS connector.

6.7.2 FEATURES

- 8K byte RAM memory
- runs at full 8080 CPU speed
- switch selectable RAM base address
- X-BUS connection to DCE-X processor module
- single 100 x 160 mm eurocard format.

6.7.3 MODULE LAYOUT



6.7.4 ADDRESS SELECTION

The RAM memory base address can be selected using the hexadecimal rotary switch located near the X-BUS connector on the module.

The RAM base address may be set to any 8K byte boundary as shown below:

Switch Setting	RAM Start Address (Hex)	RAM End Address (Hex)
0	0000	1FFF
1	2000	3FFF
2	4000	5FFF
3	6000	7FFF
4	8000	9FFF
5	A000	BFFF
6	C000	DFFF
7*	E000	FFFF

* This setting should not be used due to the conflict with DCE-X GIC and TICC register addressing.

6.7.5 OPERATIONAL REQUIREMENTS

Power Requirements

The MXR-8 module requires a single +5V power supply. This is obtained from the DCE-BUS via the System Connector. No other DCE-BUS signals are connected to the module.

A typical power requirement in the quiescent state is given below. Active state value is typically 20% higher.

+5V : 1.2A

Environmental Requirements

Operating temperature	:	0°C to 55°C
Storage temperature	:	-25°C to +85°C
Relative humidity	:	up to 95% noncondensing

X-BUS Loading

SIGNAL	LEAKAGE	DRIVE	LOAD
Address Lines Data Lines	one CMOS 160µA max.	2.1mA @ 0.4V -1mA @ 2.4V	one CMOS 160µA max.
Read, Write			-360µA @ 0.4V

6.7.6 ORDERING INFORMATION

MXR-8 : Standard Version

The X-BUS(n) cable for connection between DCE-X and 'n' (1 to 8) memory modules must be ordered separately.

6.8 MXR-4 : 4K RAM MEMORY EXPANSION MODULE

6.8.1 FUNCTIONAL DESCRIPTION

The MXR-4 module economically provides 4K bytes of static RAM as expansion data memory for DCE-X processor based systems. The 8080 CPU on the DCE-X requires one Wait state when accessing the slower RAM memory on the MXR-4. This is automatically taken care of by the logic on the module.

An address select switch allows the RAM to occupy any desired range in DCE-X memory address space. One or more MXR-4 modules, along with other MX memory modules, can be easily connected to a DCE-X processor via the flat-cable X-BUS connector.

6.8.2 FEATURES

- 4K byte slow RAM memory
- switch selectable RAM base address
- X-BUS connection to DCE-X processor module
- single 100 x 160 mm eurocard format.

6.8.3 ADDRESS SELECTION

The RAM memory base address can be selected using the hexadecimal rotary switch located near the X-BUS connector on the module.

The RAM base address may be set to any 4K byte boundary as shown below:

Address Switch Setting	RAM Start Address (Hex)	RAM End Address (Hex)
0	0000	0FFF
1	1000	1FFF
2	2000	2FFF
3	3000	3FFF
4	4000	4FFF
5	5000	5FFF
6	6000	6FFF
7	7000	7FFF
8	8000	8FFF
9	9000	9FFF
A	A000	AFFF
B	B000	BFFF
C	C000	CFFF
D	D000	DFFF
E	E000	EFFF
F*	F000	FFFF

* This setting should not be used due to the conflict with DCE-X GIC and TICC register addressing.

6.8.4 SPECIAL CONSIDERATIONS

The 8080 CPU on the DCE-X requires one Wait state when accessing the slower RAM memory provided on the earlier MX-4s. This is automatically taken care of by the logic on the MXR-4 module. When this RAM memory is used for temporary data storage and Stack implementation, the Wait state does not cause any problems at all.

When attempting to run programs in RAM which utilize one or more TICC interrupts, there is a possibility of occasional loss or misinterpretation of interrupts. This is because the TICC interrupt control logic is not ideally suited to taking into account the possible presence of a CPU Wait state, when transferring the interrupt RST instruction to the CPU on the DCE-X.

6.8.5 OPERATIONAL REQUIREMENTS

Power Requirements

The MXR-4 module requires a single +5V power supply. This is obtained from the DCE-BUS via the System Connector. No other DCE-BUS signals are connected to the module.

A typical power requirement in the quiescent state is given below. Active state value is typically 20% higher.

+5V : 1.2A

Environmental Requirements

Operating temperature	:	0°C to 55°C
Storage temperature	:	-25°C to +85°C
Relative humidity	:	up to 95% noncondensing

X-BUS Loading

SIGNAL	LEAKAGE	DRIVE	LOAD
Address Lines	one CMOS		one CMOS
Data Lines	320µA	2ma @ 0.45V -200µA @ 2.4V	320µA
Read, Write			-360µA @ 0.4V

6.8.6 ORDERING INFORMATION

MXR-4 Standard Version

The X-BUS(n) cable for connection between DCE-X and 'n' (1 to 8) memory modules must be ordered separately.

6.9 DCE-LSA : LARGE SYSTEM ADAPTER MODULE

Preliminary Specification

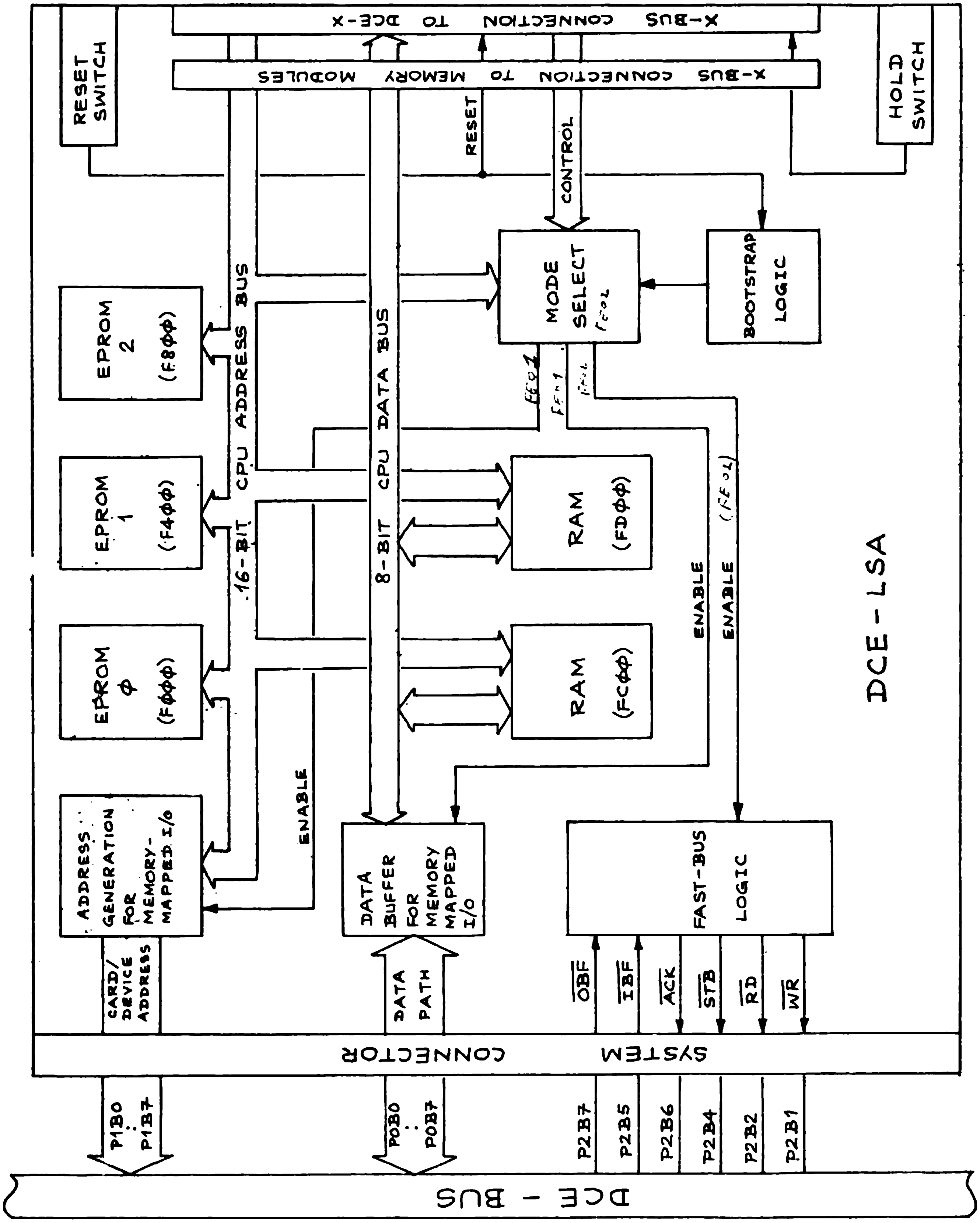
6.9.1 FUNCTIONAL DESCRIPTION

The DCE-LSA module enables the realisation of fast high-performance systems using the DCE-X microcomputer module. It is ideally suited for realising large systems based on the DCE-X microcomputer for optimum speed characteristics. It can provide a memory-mapped I/O architecture for DCE-BUS compatible modules to reduce the access time to two machine cycles of the DCE-X CPU. 512 bytes of RAM and sockets for 3K bytes EPROM in high address space are provided on the DCE-LSA module for housing the Utility or equivalent, and for work areas. This leaves the full capacity on added DCE-X memory expansion modules free for the user. A Reset button and an automatic bootstrap feature transfer control to the Utility program or equivalent residing on the card, in high address space, automatically on reset. This leaves lower address space starting from 0000H free for the user. A Hold switch enables the DCE-X CPU to be stopped and started at any time without loss of status. The DCE-LSA module can also be software configured for implementing the DCE Fast-Bus (see Section 4. 1. 4), or the normal DCE-BUS.

6.9.2 FEATURES

- reset button and bootstrap feature for automatic transfer of control to resident software in high address space on reset
- frees lower address space starting from 0000H for the user
- provides memory-mapped architecture, DCE Fast-Bus or normal DCE-BUS for accessing DCE-BUS compatible modules
- implements the DCE Fast-Bus by generating the necessary control signals automatically
- 512 bytes of RAM and sockets for 3K bytes EPROM in high address space, for the Utility program or equivalent and work areas
- full capacity on added memory expansion modules free for the user
- hold switch for stopping and starting the DCE-X CPU at will

6.9.3 MODULE LAYOUT AND BLOCK DIAGRAM



6.9.4 SYSTEM DESIGN PARAMETERS

6.9.4.1 Hardware Configuration

The DCE-LSA module has 2 flat-cable X-BUS connectors, for use in DCE-X based systems. The connector located at the edge of the module is for connection to the DCE-X processor module via an X-BUS (1) cable. The other is used for connecting DCE-X memory expansion modules via the X-BUS (n) cable (n = 1 to 8). The DCE-LSA thus acts as an intermediate stage between the DCE-X, and the memory and DCE-BUS compatible interface modules.

The DCE-LSA can be used to implement a memory-mapped I/O architecture, DCE Fast-Bus (see Section 4. 1. 4), or normal DCE-BUS for accessing DCE-BUS compatible modules. One of these three modes is selected when the DCE-X CPU performs a Write operation (data is irrelevent) to address locations 0FE01H, 0FE02H or 0FE03H respectively.

In the memory-mapped I/O mode, the DCE-LSA drives the DCE-BUS directly from the DCE-X CPU signals, and therefore the GIC ports on the DCE-X must be configured in the input mode to avoid conflicts. In this mode the registers and data devices on the DCE-BUS compatible modules appear as memory locations to the CPU on the DCE-X. The DCE-LSA generates the DCE-BUS $\overline{RD}$ and $\overline{WR}$ signals from the CPU Read and Write control signals, when the CPU address lines contain 0FE00H to 0FEFFH. When the 8 high-order address lines contain 0FEH, the 8 low-order address lines are gated onto the card/device address lines of the DCE-BUS, and the bi-directional drivers on the DCE-LSA between the DCE-BUS data path and CPU data bus are driven in accordance with the $\overline{RD}$ and $\overline{WR}$ signal status.

In the Fast-Bus mode the DCE-LSA does not drive the DCE-BUS, and instead the DCE-X GIC is configured in bi-directional mode for driving the DCE-BUS (see Section 4. 1. 4).

In the normal DCE-BUS mode the DCE-X GIC drives the DCE-BUS just as in non DCE-LSA systems.

The DCE-LSA has 512 bytes of RAM and sockets for 3K bytes EPROM in high address space. The Reset button on the DCE-LSA is usedas a system reset and to transfer control to the start of the EPROM space on the module. The reset activates a logic unit on the DCE-LSA, which inverts CPU address lines A15, A14, A13, A12, thus producing an effective address of 0F000H instead of 0000H. This is the start address of EPROM space on the DCE-LSA. A suitable program (eg: the Utility) at this location can then take over control, and perform necessary functions. The address inversion logic remains active until address line A15 goes high. By placing the instruction 'JMP 0F003H' or similar at address 0F000H it is possible to make address line A15 go high and disable the 4 address inverters. They will remain disabled until the reset button on the DCE-LSA is pushed again. This scheme acts as an automatic bootstrap procedure, while leaving the lower address space (including address 0000H) completely free for the user.

6. 9. 4. 2 Programming Specifications

Initialisation

The first instruction located at the beginning of PROM space on the DCE-LSA module (at start address 0F000H) should be 'JMP 0F003H' in order to reset the bootstrap logic as explained earlier.

After power-on or system reset, the DCE-LSA must be initialised to select one of the three possible DCE-BUS operation modes. This is done simply by performing a DCE-X CPU write operation (data is irrelevent) to one of three special address locations :

0FE01H : memory-mapped I/O mode
(DCE-LSA drives DCE-BUS)

0FE02H : Fast-Bus mode
(DCE-X GIC drives DCE-BUS)

0FE03H : normal DCE-BUS mode
(DCE-X GIC drives DCE-BUS)

For example, the Fast-Bus mode can be selected simply by a STA 0FE02H' instruction (contents of Accumulator are irrelevant).

Before selecting the memory-mapped I/O mode, ensure that the DCE-X GIC is configured in the input mode to avoid conflicts and possible damage to the GIC. The GIC will automatically be in the input mode after a power-on or system reset.

DCE-LSA Memory Addressing

The 512 bytes of RAM and sockets for 3K EPROM occupy high address space :

0F000H - 0FBFFH : 3K byte EPROM memory on DCE-LSA

0FC00H - 0FDFFH : 512 byte RAM memory on DCE-LSA

Memory-Mapped I/O Mode Addressing

In memory-mapped I/O mode the following addressing scheme is used:

0FE10H - 0FEFFH : read and write operations to DCE-BUS compatible interface modules

When used with DAI 'Real-World' interface modules (RWC-), the low-order 8 bits of the above address are decoded as follows :

Bits 4-7 : module select (1 to 15, corresponding to module address select switch setting in the range 1 to F).

Bits 0-3 register select within the selected module.

For example, RIC Port 1 of a RWC module with address select switch set to A (Hex) can be read into the DCE-X CPU Accumulator by the single instruction :

```
      LDA         0FEA1H
```

Similarly, all the RIC ports of a RWC module with address select switch set to B (Hex) can be configured as output as follows :

```
      MVI         A,80H     ;set up control word
      STA         0FEB3H    ;write to RIC command register
```

It is also possible for example, to set up a counter directly at one of the RIC Ports configured as output on a RWC module.

DCE Utility Software

Three different versions of the Utility program are available corresponding to the three modes of DCE-BUS operation :

UPT	LS.nn		3K Utility for DCE-LSA, standard DCE-BUS
UPT	LF.nn	:	3K Utility for DCE-LSA, Fast-Bus
UPT	LM.nn		3K Utility for DCE-LSA, Memory-mapped I/O

(nn = baud rate code)

6.9.4.3 Module Connector Definitions

System Connector

See Section 6.1.4 for the pin definitions.

X-BUS Connectors

See Section 6.4.3.3 for the pin definitions of both these connectors.

6.9.4.4 Operational Requirements

Power Requirements

The DCE-LSA requires three power supplies from the DCE-BUS. The values given below are for the quiescent state with the three 2708 EPROMs plugged in. Active state values are typically 20% higher.

+12V ±5%	:	80 mA
+ 5V ±5%	:	380 mA
- 5V ±5%	:	52 mA

Environmental Requirements

Operating temperature	:	0°C to 55°C
Storage temperature	:	-25°C to +85°C
Relative humidity	:	95% noncondensing

6.9.5 ORDERING INFORMATION

DCE-LSA	:	Standard version
X-BUS(1)	:	Flat-cable for connecting to DCE-X - must be ordered separately
X-BUS(n)	:	Flat-cable connecting upto 'n' DCE-X memory expansion modules - must be ordered separately

6.9.6 DCE-LSA APPLICATION EXAMPLE: INTEL MDS EMULATOR

The DCE-LSA card with a suitable DCE-X based configuration can emulate the entire functions of the paper-tape oriented INTEL MDS Development System. A DCE configuration for this requires a paper-tape reader, paper-tape punch, a printer and a console device as peripherals. The DCE-LSA Utility program contains all of the normal DCE Utility functions, together with peripheral driver routines. The DCE based MDS emulator allows development software such as Assembler and Editor to reside permanently in EPROM. This eliminates the need to load the Assembler and Editor from paper-tape each time they are needed.

The minimum configuration is given below.

1	RACK-I
2	DCE-PWR/H heavy duty power supply
1	DCE-X
1	DCE-LSA
1	X-BUS(3)
1	MXR-8 (adr. switch = 0)
1	MXR-8 (adr. switch = 1)
1	MXP-12 (adr. switches : top = 6, bottom = B)
2	RWC-T24
1	3K Utility program for DCE-LSA

The 2 RWC-T24 modules provide the handshake interfaces for the printer, paper-tape reader and punch. Driver software is included in the DCE-LSA Utility.

Address Space

0000 -	all available RAM
B000 - E3FF	PROM copy of MDS Editor and Assembler
F000 - FBFF	3K byte DCE-LSA Utility, MDS transfer vectors + I/O drivers.
FC00 - FDFF	512 byte RAM on DCE-LSA
FE00 - FEFF	DCE-LSA address space for DCE-BUS compatible module interfacing.
FF00 - FFFF	DCE-X GIC and TICC addresses

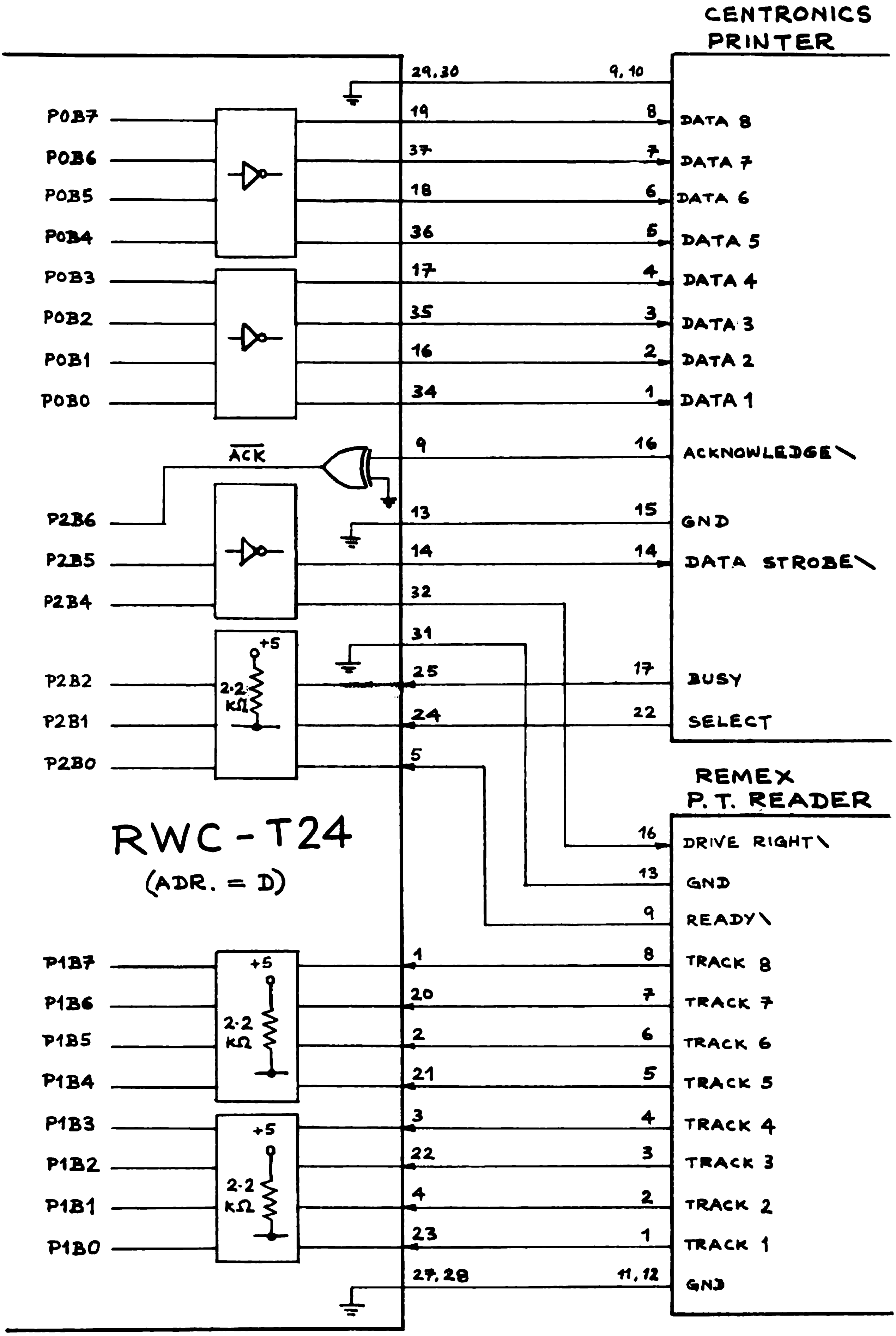


Figure 6-4 : RWC-T24 Interface to Printer and Paper-Tape Reader

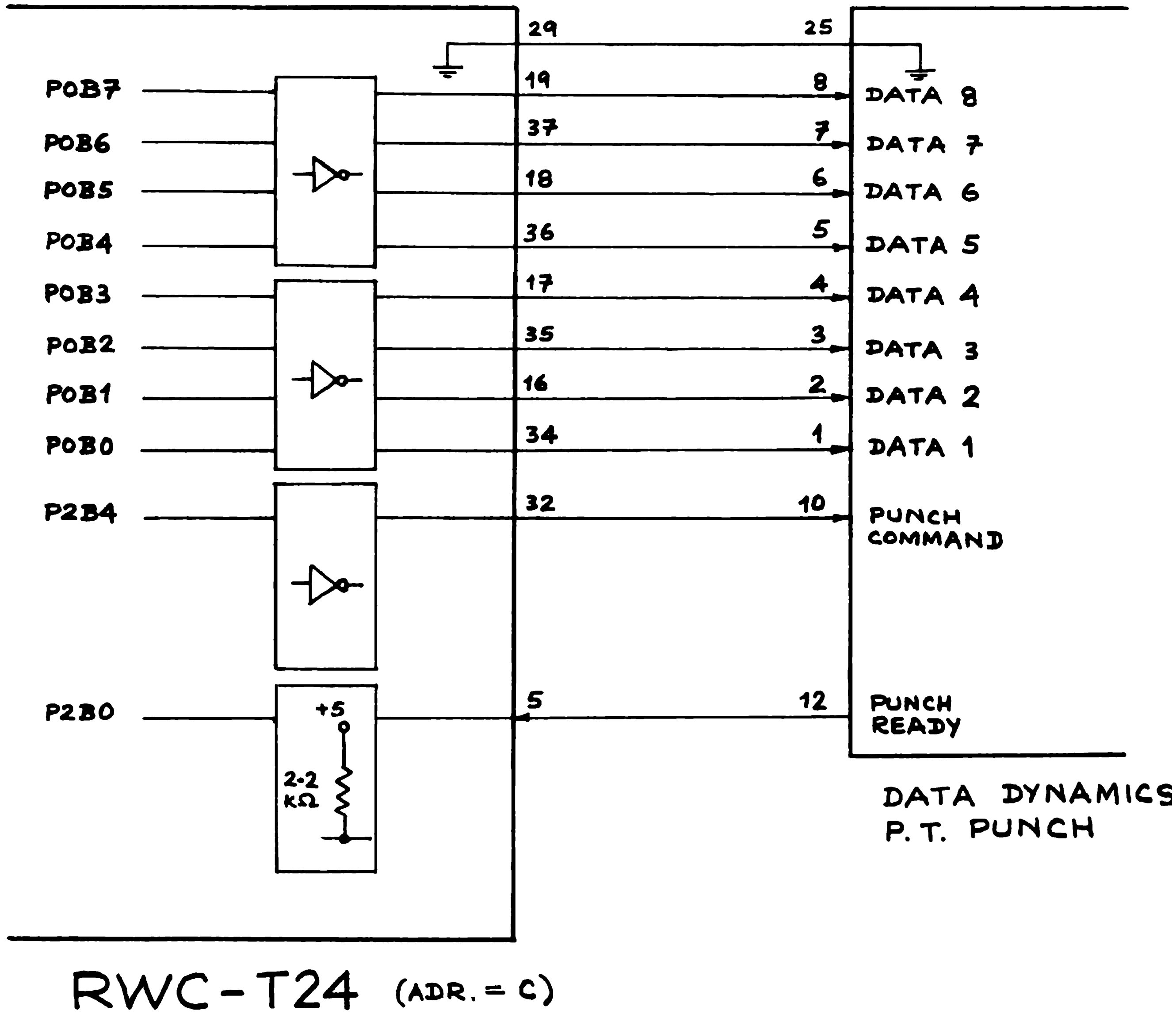


Figure 6-5 : RWC-T24 Interface to Paper-Tape Punch

6.10 MXR-4D : 4K DYNAMIC RAM MEMORY EXPANSION MODULE

6.10.1 FUNCTIONAL DESCRIPTION

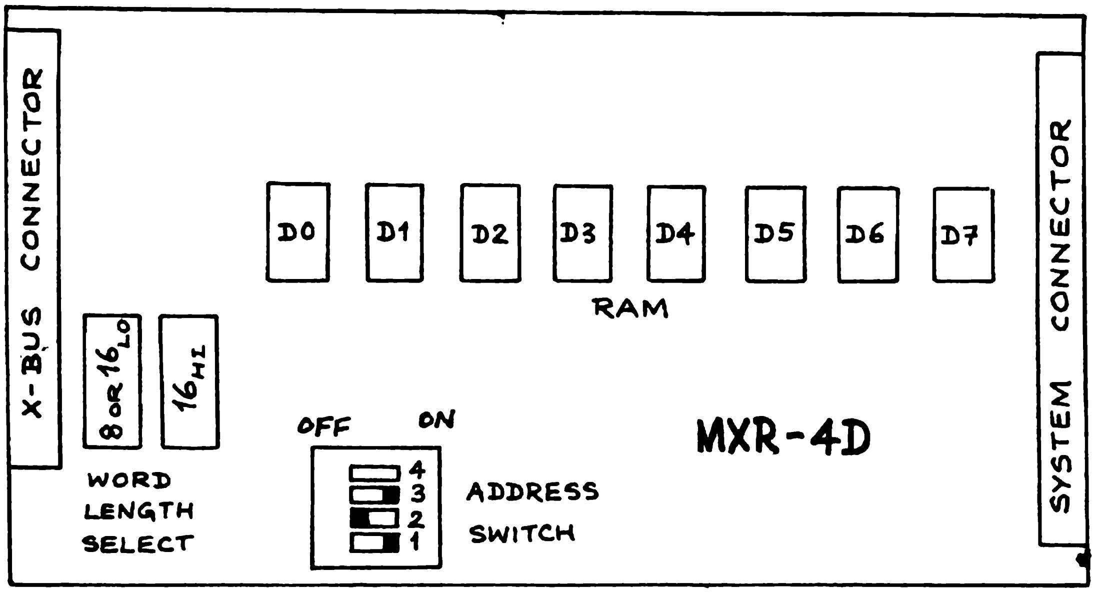
The MXR-4D module provides 4K bytes of dynamic RAM as expansion memory for DCE-X processor based systems. Automatic RAM refresh logic on the module allows the CPU on the DCE-X to run at full speed, and the memory appears just like static RAM to the user software.

An address select switch allows the RAM to occupy desired ranges in DCE-X memory address space. One or more MXR-4D modules, along with other MX memory modules, can be easily connected to a DCE-X processor via the flat-cable X-BUS connector.

6.10.2 FEATURES

- 4K byte dynamic RAM memory
- on-board refresh logic
- refresh maintained during system reset
- low power requirement
- runs at full 8080 CPU speed
- switch selectable RAM base address
- X-BUS connection to DCE-X processor module
- single 100 x 160 mm eurocard format

6. 10. 3 MODULE LAYOUT



6. 10. 4 WORD LENGTH SELECTION

The MXR-4D RAM module is designed for use with 8-bit as well as 16-bit DCE processor modules. When used with a 16-bit DCE processor, the RAM module must be assigned to either the low-order 8 bits or the high-order 8-bits of the 16-bit word. This is done by means of a component carrier which can be plugged into one of two sockets located near the X-BUS connector as shown above. When it is plugged into the left socket, the MXR-4D can be used either with an 8-bit DCE processor, or, as the low-order byte memory with a 16-bit DCE processor. When it is plugged into the right socket, the module can only be used as the high-order byte memory with a 16-bit DCE processor.

6. 10. 5 ADDRESS SELECTION

The RAM memory base address can be selected via a 4-bit switch located as shown in the above module layout diagram. Only the switches numbered 1 to 3 are used. Position of switch 4 is irrelevant.

The RAM base address may be set as shown on the next page.

Switch Setting 1 2 3	RAM Start Address (Hex)	RAM End Address (Hex)
On On On	0000	0FFF
Off On On	2000	2FFF
On Off On	3000	3FFF
Off Off On	4000	4FFF
On On Off	6000	6FFF
Off On Off	8000	8FFF
On Off Off	D000	DFFF
Off Off Off	A000	AFFF

Note 1 : The switch setting corresponding to start address D000 is for the DCE-X Utility system RAM (stack etc).

Note 2 : Referring to the physical module layout in Section 6. 10. 3 switch 1 is located towards the lower edge of the RAM card. To turn a switch on, move the switch rocker or slider to the right.

6. 10. 6 OPERATIONAL REQUIREMENTS

Power Requirements

The MXR-4D module requires 12V, ±5V power supplies. These are obtained from the DCE-BUS via the System Connector. No other DCE-BUS signals are connected to the module.

A typical power requirement in the quiescent state is given below. Active state values are typically 20% higher.

+5V	:	280 mA
-5V	:	720 µA
+12V	:	220 mA

Environmental Requirements

Operating temperature		0°C to 55°C
Storage temperature		-25°C to +85°C
Relative humidity	:	up to 95% noncondensing

X-BUS Drive and Load

SIGNAL	LEAKAGE	DRIVE (read data)	LOAD (write data)
Address Lines			+425µA
Data Lines	+10µA max.	+12.01mA@0.4V	+10µA max.
		-2.59mA@2.4V	
$\overline{\text{MEMR}}$			-250µA max.
MEMW			-360µA max

Note: + indicates current into card; - indicates current out of card.

6.10.7 ORDERING INFORMATION

MXR-4D : Standard Version

The X-BUS(n) cable for connection between DCE-X and 'n' (1 to 8) memory modules must be ordered separately.

6.11 MXR-8D : 8K DYNAMIC RAM MEMORY EXPANSION MODULE

6.11.1 FUNCTIONAL DESCRIPTION

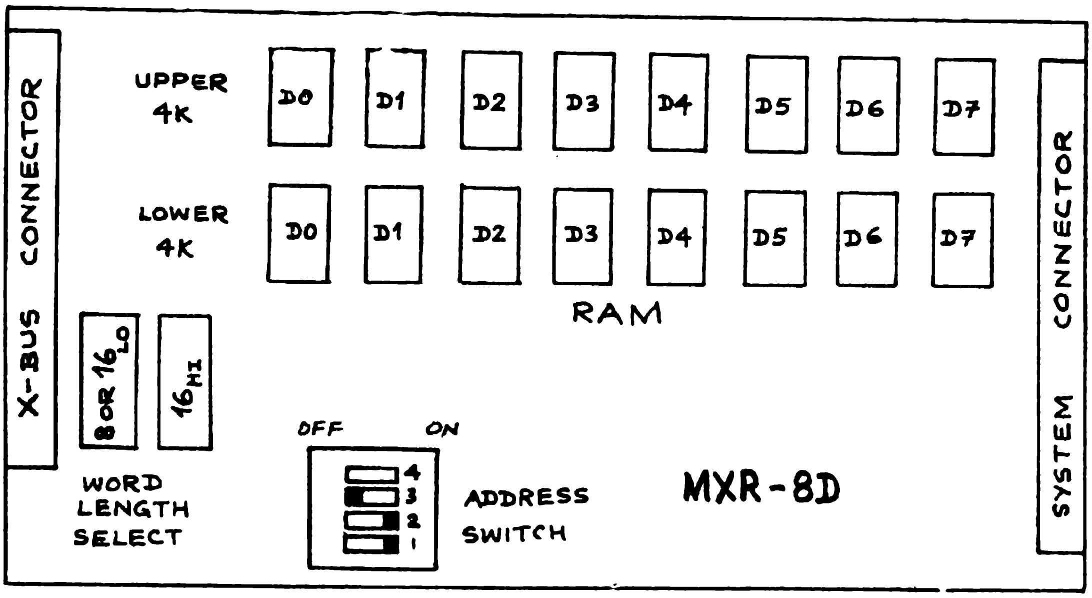
The MXR-8D module provides 8K bytes of dynamic RAM as expansion memory for DCE-X processor based systems. Automatic RAM refresh logic on the module allows the CPU on the DCE-X to run at full speed, and the memory appears just like static RAM to the user software.

An address select switch allows the RAM to occupy desired ranges in DCE-X memory address space. One or more MXR-8D modules, along with other MX memory modules, can be easily connected to a DCE-X processor via the flat-cable X-BUS connector.

6.11.2 FEATURES

- 8K byte dynamic RAM memory
- on-board refresh logic
- refresh maintained during system reset
- low power requirement
- runs at full 8080 CPU speed
- switch selectable RAM base address
- X-BUS connection to DCE-X processor module
- single 100 x 160 mm eurocard format

6.11.3 MODULE LAYOUT



6.11.4 WORD LENGTH SELECTION

The MXR-8D RAM module is designed for use with 8-bit as well as 16-bit DCE processor modules. When used with a 16-bit DCE processor, the RAM module must be assigned to either the low-order 8-bits or the high-order 8-bits of the 16-bit word. This is done by means of a component carrier which can be plugged into one of two sockets located near the X-BUS connector as shown above. When it is plugged into the left socket, the MXR-8D can be used either with an 8-bit DCE processor, or, as the low-order byte memory with a 16-bit DCE processor. When it is plugged into the right socket, the module can only be used as the high-order byte memory with a 16-bit DCE processor.

6.11.5 ADDRESS SELECTION

The RAM memory base address can be selected via a 4-bit switch located as shown in the above module layout diagram. Only the switches numbered 1 to 3 are used. Position of switch 4 is irrelevant.

The RAM address may be set as shown on the next page.

Switch Setting 1 2 3	RAM Start Address (Hex)	RAM End Address (Hex)	Possible Usage
On On On	0000	1FFF	RAM at 0 for DCE-LSA
Off On On	2000	3FFF	Above 8K PROM
On Off On	3000	4FFF	Above 12K PROM
Off Off On	4000	5FFF	Above 16K RAM (MXR-16D)
On On Off	6000	7FFF	Above 24K PROM (2 x MXP-12
Off On Off	8000	9FFF	Above 32K RAM (MXR-32D)
On Off Off	0000 D000	0FFF DFFF	For running DCE-X, 1A or 2A software on a DCE-LSA development system
Off Off Off	C000	DFFF	Above 16K + 32K RAM

Note 1: The switch setting corresponding to start addresses 0000 and D000 is for the DCE-X Utility system RAM (stack etc).

Note 2 : Referring to the physical module layout in Section 6. 11. 3 switch 1 is located towards the lower edge of the RAM card. To turn a switch on, move the switch rocker or slider to the right.

6. 11. 6 OPERATIONAL REQUIREMENTS

Power Requirements

The MXR-8D module requires 12V, ±5V power supplies. These are obtained from the DCE-BUS via the System Connector. No other DCE-BUS signals are connected to the module.

A typical power requirement in the quiescent state is given below. Active state values are typically 20% higher.

+5V :	280 mA
-5V :	1440 μA
+12V :	225 mA

Environmental Requirements

Operating temperature	0°C to 55°C
Storage temperature	-25°C to +85°C
Relative humidity	up to 95% noncondensing

X-BUS Drive and Load

SIGNAL	LEAKAGE	DRIVE (read data)	LOAD (write data)
Address Lines			+425μA
Data Lines	+20μA max.	+12.02mA@0.4V	+20μA max.
		-2.58mA@2.4V	
$\overline{\text{MEMR}}$			-250μA max.
MEMW			-360μA max.

Note: + indicates current into card; - indicates current out of card.

6.11.7 ORDERING INFORMATION

MXR-8D Standard version

The X-BUS(n) cable for connection between DCE-X and 'n' (1 to 8) memory modules must be ordered separately.

6.12 MXR-16D : 16K DYNAMIC RAM MEMORY EXPANSION MODULE

6.12.1 FUNCTIONAL DESCRIPTION

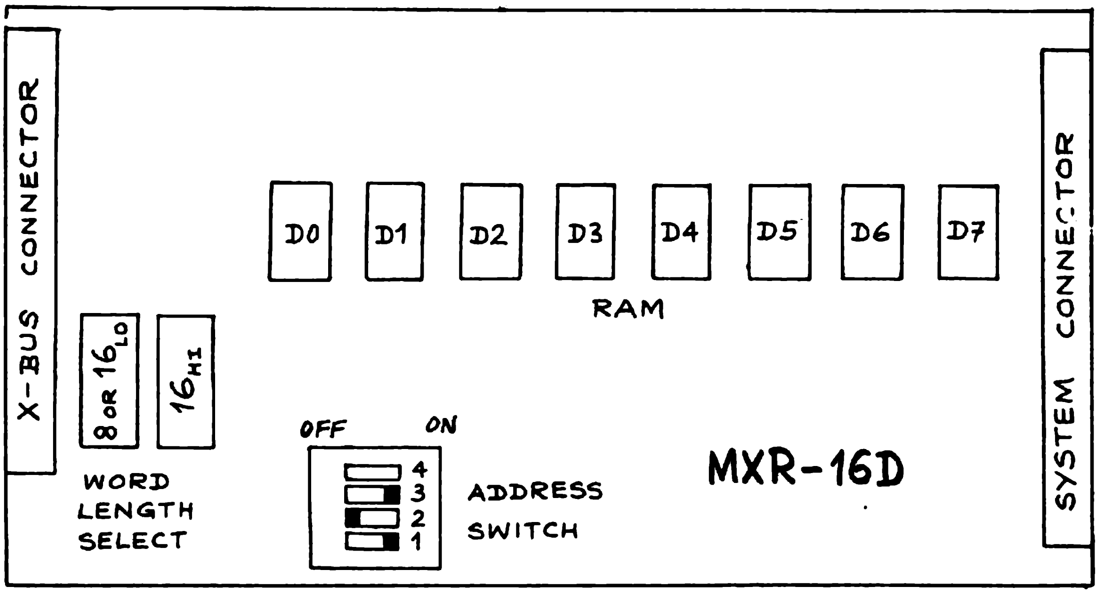
The MXR-16D module provides 16K bytes of dynamic RAM as expansion memory for DCE-X processor based systems. Automatic RAM refresh logic on the module allows the CPU on the DCE-X to run at full speed, and the memory appears just like static RAM to the user software.

An address select switch allows the RAM to occupy desired ranges in DCE-X memory address space. One or more MXR-16D modules, along with other MX memory modules, can be easily connected to a DCE-X processor via the flat-cable X-Bus connector.

6.12.2 FEATURES

- 16K byte dynamic RAM memory
- on-board refresh logic
- refresh maintained during system reset
- low power requirement
- runs at full 8080 CPU speed
- switch selectable RAM base address
- X-BUS connection to DCE-X processor module
- single 100 x 160 mm eurocard format

6.12.3 MODULE LAYOUT



6.12.4 WORD LENGTH SELECTION

The MXR-16D RAM module is designed for use with 8-bit as well as 16-bit DCE processor modules. When used with a 16-bit DCE processor, the RAM module must be assigned to either the low-order 8-bits or the high-order 8-bits of the 16-bit word. This is done by means of a component carrier which can be plugged into one of two sockets located near the X-BUS connector as shown above. When it is plugged into the left socket, the MXR-16D can be used either with an 8-bit DCE processor, or, as the low-order byte memory with a 16-bit DCE processor. When it is plugged into the right socket, the module can only be used as the high-order byte memory with a 16-bit DCE processor.

6.12.5 ADDRESS SELECTION

The RAM memory base address can be selected via a 4-bit switch located as shown in the above module layout diagram. Only the switches numbered 1 to 3 are used. Position of switch 4 is irrelevant.

The RAM address may be set as shown on the next page.

Switch Setting 1 2 3	RAM Start Address (Hex)	RAM End Address (Hex)	Possible Usage
On On On	0000	3FFF	RAM at 0 for DCE-LSA
Off On On	2000	5FFF	Above 8K PROM
On Off On	3000	6FFF	Above 12K PROM
Off Off On	4000	7FFF	With 2nd MXR-16D, or 16K PROM
On On Off	6000	9FFF	Above 24K PROM
Off On Off	8000	BFFF	With MXR-32D 32K RAM module
On Off Off	0000 D000	2FFF DFFF	For running DCE-X, 1A or 2A software on a DCE-LSA development system
Off Off Off	B000	EFFF	Highest position

Note 1 : The switch setting corresponding to start address D000 is for the DCE-X Utility system RAM (stack etc).

Note 2 : Referring to the physical module layout in Section 6. 12. 3, switch 1 is located towards the lower edge of the RAM card. To turn a switch on, move the switch rocker or slider to the right.

6.12.6 OPERATIONAL REQUIREMENTS

Power Requirements

The MXR-16D module requires 12V, ±5V power supplies. These are obtained from the DCE-BUS via the System Connector. No other DCE-BUS signals are connected to the module.

A typical power requirement in the quiescent state is given below. Active state values are typically 20% higher.

+5V :	280 mA
-5V :	55 mA
+12V :	220 mA

Environmental Requirements

Operating temperature	0°C to 55°C
Storage temperature	-25°C to +85°C
Relative humidity	up to 95% noncondensing

X-BUS Drive and Load

SIGNAL	LEAKAGE	DRIVE (read data)	LOAD (write data)
Address Lines			+425μA
Data Lines	-10μA to +10μA	12. 01mA max.@0. 4V	-10μA to +10μA
		-2. 61mA max.@2. 4V	
$\overline{\text{MEMR}}$			-250μA max.
MEMW			-360μA max.

Note: + indicates current into card; - indicates current out of card.

6.12.7 ORDERING INFORMATION

MXR-16D : Standard Version

The X-BUS(n) cable for connection between DCE-X and 'n' (1 to 8) memory modules must be ordered separately.

6.13 MXR-32D : 32K DYNAMIC RAM MEMORY EXPANSION MODULE

6.13.1 FUNCTIONAL DESCRIPTION

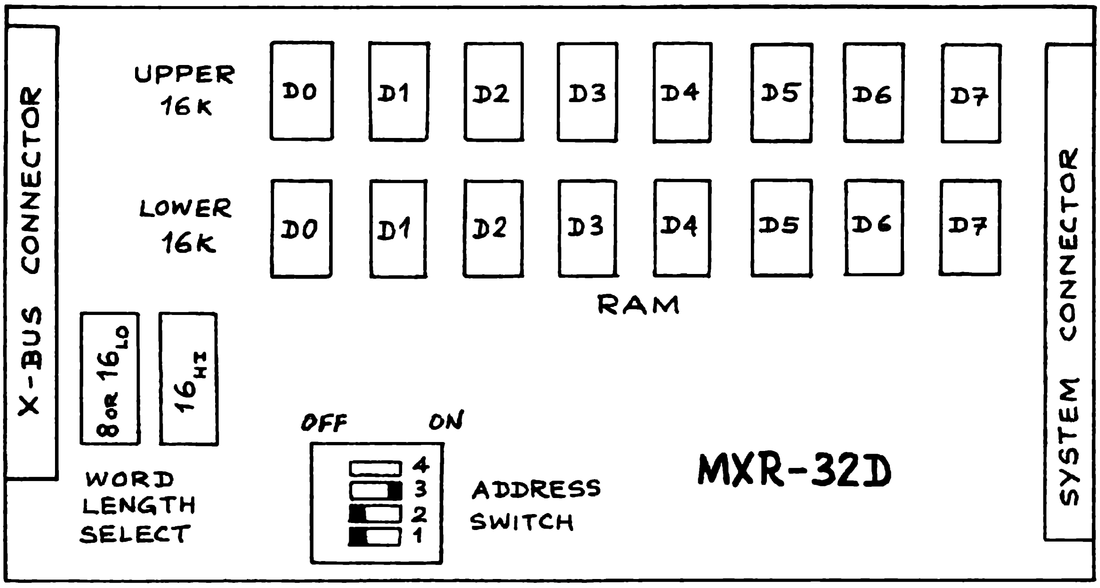
The MXR-32D module provides 32K bytes of dynamic RAM as expansion memory for DCE-X processor based systems. Automatic RAM refresh logic on the module allows the CPU on the DCE-X to run at full speed, and the memory appears just like static RAM to the user software.

An address select switch allows the RAM to occupy desired ranges in DCE-X memory address space. Upto two MXR-32D modules along with other MX memory modules, can be easily connected to a DCE-X processor via the flat-cable X-BUS connector.

6.13.2 FEATURES

- 32K byte dynamic RAM memory
- on-board refresh logic
- refresh maintained during system reset
- low power requirement
- runs at full 8080 CPU speed
- switch selectable RAM base address
- X-BUS connection to DCE-X processor module
- single 100 x 160 mm eurocard format

6.13.3 MODULE LAYOUT



6.13.4 WORD LENGTH SELECTION

The MXR-32D RAM module is designed for use with 8-bit as well as 16-bit DCE processor modules. When used with a 16-bit DCE processor, the RAM module must be assigned to either the low-order 8-bits or the high-order 8-bits of the 16-bit word. This is done by means of a component carrier which can be plugged into one of two sockets located near the X-BÜS connector as shown above. When it is plugged into the left socket, the MXR-32D can be used either with an 8-bit DCE processor, or, as the low-order byte memory with a 16-bit DCE processor. When it is plugged into the right socket, the module can only be used as the high-order byte memory with a 16-bit DCE processor.

6.13.5 ADDRESS SELECTION

The RAM memory base address can be selected via a 4-bit switch located as shown in the above module layout diagram. Only the switches numbered 1 to 3 are used. Position of switch 4 is irrelevant.

The RAM address may be set as shown on the next page.

Switch Setting 1 2 3	RAM Start Address (Hex)	RAM End Address (Hex)	Possible Usage
On On On	0000	7FFF	RAM at 0 for DCE-LSA
Off On On	2000	9FFF	Above 8K PROM
On Off On	3000	AFFF	Above 12K PROM
Off Off On	4000	BFFF	Above 16K RAM
On On Off	6000	DFFF	Above 24K PROM
Off On Off	1000	8FFF	Above 4K PROM
On Off Off	0000 D000	6FFF DFFF	For running DCE-X, 1A or 2A software on a DCE-LSA development system
Off Off Off	8000	EFFF	With 2nd MXR-32D and DCE-LSA

Note 1 : The switch setting corresponding to start address D000 is for the DCE-X Utility system RAM (stack etc).

Note 2 : Referring to the physical module layout in Section 6. 13. 3 switch 1 is located towards the lower edge of the RAM card. To turn a switch on, move the switch rocker or slider to the right.

6.13.6 OPERATIONAL REQUIREMENTS

Power Requirements

The MXR-32D module requires +12V, ±5V power supplies. They are obtained from the DCE-BUS via the System Connector. No other DCE-BUS signals are connected to the module.

A typical power requirement in the quiescent state is given below. Active state values are typically 20% higher.

+5V :	280 mA
-5V :	110 mA
+12V :	230 mA

Environmental Requirements

Operating temperature	:	0°C to 55°C
Storage temperature		-25°C to +85°C
Relative humidity	:	up to 95% noncondensing

X-BUS Drive and Load

SIGNAL	LEAKAGE	DRIVE (read data)	LOAD (write data)
Address Lines			+425µA
Data Lines	-20µA to +20µA	12.02mA max.@0.4V -2.62mA max.@2.4V	-20µA to+20µA max.
$\overline{\text{MEMR}}$			-250µA max.
MEMW			-360µA max.

Note: + indicates current into card; - indicates current out of card.

6.13.7 ORDERING INFORMATION

MXR-32D	:	Standard Version

The X-BUS(n) cable for connection between DCE-X and 'n' (1 to 8) memory modules must be ordered separately.

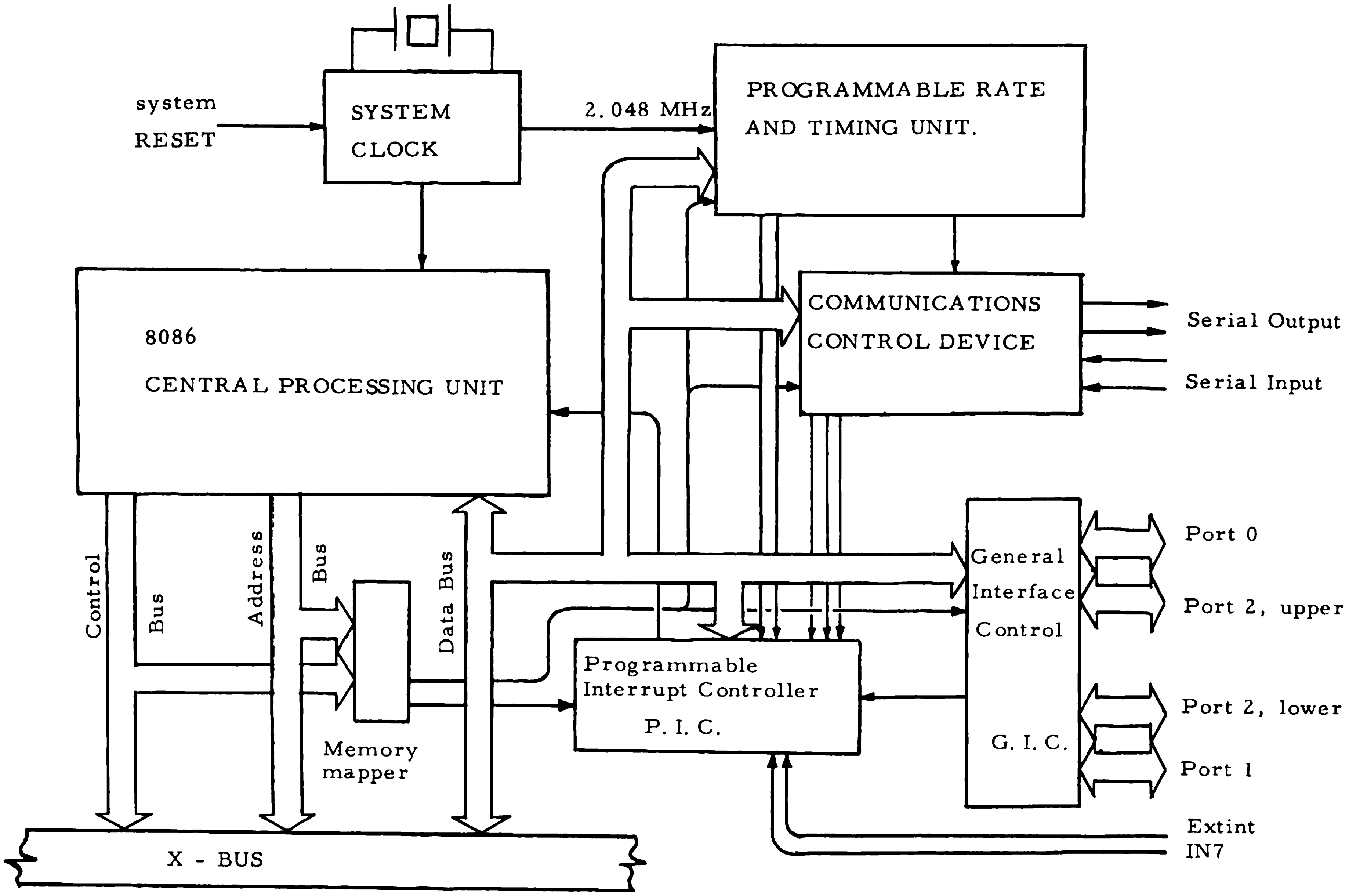
6.14 DCE-X86 16-BIT PROCESSOR MODULE

6.14.1 FEATURES

- ° Complete 8086, 16 bit microcomputer system on a single 100 x 160 mm eurocard.
- ° 2K byte RAM and sockets for 4K or 8K byte EPROM.
- ° Memory expandable with upto 60K x 16-bit via flat-cable X-Bus.
- ° Opto-isolated serial I/O with programmable baud rates.
- ° Synchronous, asynchronous, and isosynchronous modes.
- ° 24 parallel I/O lines, programmable as simple, handshaking or bi-directional with automatic handshake control signals.
- ° Hardware support for DAI's "fast-bus" transfer mode.
- ° 2 independent interval timers providing 500 nano-second resolution, with crystal accuracy.
- ° Advanced interrupt controller, provides both simple and rotating priority modes.
- ° Instructions to manipulate 8 and 16-bit words, including hardware multiply and divide.
- ° 3 external interrupt lines.

The DCE-X86 is a single eurocard format microcomputer module incorporating the 8086 highspeed 16-bit microprocessor. It provides the user the possibility of combining the software power of this microprocessor together with the complete interface capabilities provided by the family of DCE-BUS compatible modules.

The DCE-X86 module can be used either as a single card microcomputer, or, by the addition of compatible memory expansion modules, it becomes a central processing and system controller module. The memory space provided can be translated into a high address field, thus leaving all low address space (from zero) free for the memory expansion modules.



DCE-X86: Functional Block Diagram

(Memory not shown)

In addition to the microprocessor and memory facilities, the DCE-X86 module also provides a total system control architecture. This includes a centralised interrupt controller, programmable serial input and output interfaces, and a timer-counter group. The following sections will describe them in more detail, including the necessary software aspects related to each in turn.

6.14.2 Memory

The DCE-X86 module provides the user both RAM and E PROM memory, thus enabling its application as a stand-alone controller. However, this can be augmented by a collection of memory expansion cards. The on-card memory will then be used as stack/ workspace (RAM) and Utility/Bootstrap (EPROM). All Addresses from 00000H upto 1DFFFH are then left free for the memory expansion modules (120K bytes).

In order to release the memory at low address space, commencing with address zero (necessary for the operation of the 8086 device), an optional memory configuration is provided. This must be ordered specifically at the time of purchase and is only required if the module is to have memory expansion attached.
All of the peripheral devices found on the eurocard are included into the 8086 memory space. The addresses relative to each device, and also for the memory, are defined in the following table:

Memory Segment	Address Bits																				Description
Number	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	B	32K + 32K byte system memory
1A	0	0	0	1	0	A	A	A	A	A	A	A	A	A	A	A	A	A	A	B	16K + 16K byte system memory
1B	0	0	0	1	1	0	A	A	A	A	A	A	A	A	A	A	A	A	A	B	8K + 8K byte system memory
1C	0	0	0	1	1	1	0	A	A	A	A	A	A	A	A	A	A	A	A	B	4K + 4K byte system memory
1D	0	0	0	1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	DO NOT USE
3	0	0	1	1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	A	0	Communications control
5	0	1	0	1	1	1	1	X	X	X	X	X	X	X	X	X	X	A	A	0	Timing and Rate Controller
7	0	1	1	1	1	1	1	X	X	X	X	X	X	X	X	X	X	A	A	0	General Interface controller
9	1	0	0	1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	A	0	Programmable interrupt controller
B	1	0	1	1	1	1	1	X	X	A	A	A	A	A	A	A	A	A	A	B	RAM with optional memory mapper
F	1	1	1	1	1	1	1	C	A	A	A	A	A	A	A	A	A	A	A	B	2716/2732 EPROM memory
0A	0	0	X	X	X	X	X	X	X	A	A	A	A	A	A	A	A	A	A	B	RAM with memory map supplied.

Notes:

A = Memory address line
B = Bank select address controls both A_0/8086 and $\overline{\text{BHE}}$.
C = Ø for 2716 devices or Address line for 2732's.
X = Don't care.

DON'T use EVEN segments (with the exception of segment zero)!

Memory arrangement for DCE-X86 microcomputer module.

6.14.3 Memory Organisation

The 8086 microprocessor manipulates data transfers to and from memory over 2 independant 8-bit data busses. These relate to a high and low byte of a 16 bit data-word. It is therefore possible, so long as an even address is selected, to read or write a full 16 bits with only one memory cycle. The two data busses share a common memory cycle. 16 bit memory access can occur with anODD address, but they need 2 cycles for execution bits. Available at the X-Bus connector, the X-Bus addresses now form 8086 address A1 through A16. AØ is used to select the low byte memory bank, whilst a special signal, $\overline{\text{BHE}}$, is used to select the high bank of memory. In 16 bit transfers these two signals can simultaneously become active. Also organised on the X-Bus cable, and supported by DAI's family of dynamic RAM memory expansion modules, are 8 extra databus lines. The memory modules can be connected either to the high or low databus lines as required. The same memory module cannot, however, be split between both data lines.

The on-card EPROM memory is organised as 1 chip per bus. All peripheral devices reside on the low byte bus.

6.14.4 Programmable Interrupt Controller

The device employed on the DCE-X86 module enables the user to select the system responses to the available interrupts.

Eight system interrupt lines are supported. These are known as INØ to IN7. Normally, INØ will have the highest priority, and IN7 the lowest. This priority order can be influenced by commands from the user. The assignment of functions to each of these interrupt lines is as follows:

INØ	EXTINT	System bus interrupts
IN1	INT7	
IN2	GIC	
IN3	TIMER 2	
IN4	TIMER 1	
IN5	Receive Buffer full	Serial Interface Interrupts.
IN6	Transmit Buffer Empty	
IN7	Break or Sync. Detect.	

Internal to the interrupt controller, there are 3 important registers. They operate closely together and have the following functions.

a) Interrupt Request Register (IRR)

This register provides one bit for each of the interrupt lines. A bit is set whenever the related interrupt exhibits the condition required to request an interrupt. This condition may be either level or edge sensitive. The selection of which is made during the software initialisation.

b) Interrupt Mask Register (IMR)

Each interrupt request input can be individually masked by the Interrupt Mask Register, (IMR), programmed by the user. Each bit in the IMR masks one interrupt channel if it is set to 1. Bit Ø masks IRØ, bit 1 masks IR1 and so forth. Edge-triggered interrupts during the time that an interrupt is masked will cause a service request when the mask is removed. Thus, if the interrupt mask is used to selectively ignore interrupts, care must be taken to trap these supurious requests.

c) Interrupt Service Register (ISR)

These register bits become set when the 8086 microprocessor acknowledges an interrupt. This is normally performed by hardware, and is thus transparant to the user. In response to the interrupt acknowledgement, the bit corresponding to the highest request priority will be set in the ISR, and the corresponding bit in the IRR will be reset. Only requests which are unmasked will be considered for service. The service request register normally masks off requests from interrupts of lower priority. This feature can be disabled by supplying a suitable operations command to the controller. This is described in section 6. 14. 4. 5 under the title "special mask mode".

The appropriate bit in the ISR will remain set until a suitable End of Interrupt command is given to the controller. Unmasked interrupts of higher priority can still cause an interrupt of the service routine. Thus, more than one bit may be set in the ISR allowing for several interrupt service routines to be nested.

6. 14. 4. 1 Initialisation procedure

The user must always initialise the interrupt controller before any interrupts can be processed. If required, initialisation can occur more than once. This is useful when the application demands a change of the fundamental operation of the controller.

Three important parameters are transferred to the controller during the initialisation sequence. There are defined by the code values of three successive bytes which are to be sent to the controller device. The device employed is the 8259A interrupt controller. Manufacturers' documentation relating to this device mentions a large collection of operating modes. These each have a mnemonic associated with every command word issued to the device. Those commands chosen are listed, together with their mnemonic codes for reference.

Mnemonic	Address	Data	Description
ICW1 I	9FFF0	X X X 1 0 X 1 1	Select edge trigger
ICW1 J	9FFF0	X X X 1 1 X 1 1	Select level sense
ICW2	9FFF2	V V V V V X X X	Issue vector
ICW4 B	9FFF2	0 0 0 0 0 0 0 1	Programmed EOI
ICW4 D	9FFF2	0 0 0 0 0 0 1 1	Automatic EOI

Notes:

X	Don't care
0,1	Literal 0 or 1
VVVVV	5 bit vector

Table 1. Initialisation command words

The normal mode of operation is for the interrupts to be edge triggered. Level sense mode will cause a continuous interrupt request if the corresponding interrupt is not removed before an end of interrupt (EOI) is issued.

Never use Level sense mode if either interrupts IN2,5,6, or 7 are enabled.

ICW1 must be issued, and always followed by ICW2 and ICW4 before any other communication with the controller device can commence.

ICW2 defines the address block used by the 8086 to carry the entry points of all of the interrupt service routines. This 5 bit value is multiplied by 32 to define the entrypoint for INØ. All of the other interrupt entry points follow successively at 4 byte intervals. The 4 bytes provided for each interrupt allows for a set Interrupt Instruction (STI , to enable interrupt servicing of higher priority interrupts), followed by a suitable JUMP or CALL instruction.

A call instruction is useful for IN5 (Receive buffer full). The consequence of this is to immediately follow with transmit character service routine. This echoes the received character back to the attached console device.

The first 4 vector fields (VVVVV = 0,1,2,or3) are reserved for future DAI Software. The user is advised to use vectors other than these. Vector field ∅ can also conflict with the internal trap vectors of the 8086 (e.g. trap if division by zero).

ICW4 specifies if the device should operate in the Automatic End of Interrupt mode (AEOI). When this mode is selected, the ISR bit will be reset immediately the service routine is entered. Thus, the device can immediately register requests for interrupt from all of the unmasked lines. If AEOI mode is used, the interrupts of the 8086 should not be reenabled until the 8086 instruction, IRET (Interrupt Return), restores the Interrupt Enable Flag into the flag registers.

6.14.4.2 End of Interrupt Processing and Priority Control

In applications which use several of the available System Interrupts, it is often advantageous to use the programmed end of interrupt mode of operation. This command is available in several different forms. These are given in the following table:

Mnemonic	Address	Data	Description
0CW2 E	9FFF0	0 0 1 0 0 0 0 0	Reset bit of highest priority set in ISR.
0CW2 SE	9FFF0	0 1 1 0 0 n n n	Reset bit in ISR corresponding to IN'n'
0CW2 RE	9FFF0	1 0 1 0 0 0 0 0	Reset bit of highest priority in ISR and assign this interrupt to the lowest priority level.
0CW2 RSE	9FFF0	1 1 1 0 0 n n n	Reset bit in ISR corresponding to IN'n' and assign interrupt to the lowest priority level.

OCW2 R	9FFF0	0 1 0 0 0 0 0 0	Enable automatic assignment to lowest priority level if in AEOI mode.
OCW2 CR	9FFF0	0 0 0 0 0 0 0 0	Disable automatic assignment of lowest priority if in AEOI mode.
OCW2 RS	9FFF0	1 1 0 0 0 n n n	Assign IN'n' as lowest priority. Does not affect ISR.

Table 2. End of Interrup Command Group.

The specific end of interrupt commands (OCW2 SE and RSE) are normally used when the service routine sequence disturbs the normal nesting of the interrupt requests. Thus, a low priority interrupt service routine can be completed from within the service routine for an interrupt operating on a higher level.

The current priority status can be modified under program control. The function of the rotating priority commands in the EOI command Group (OCW2 SE through RS) is to define the interrupt request of lowest priority. This is usual to be the request that was last serviced. The relative priorities of all of the other interrupt requests are implicitly defined by the lowest priority assignment.

Interrupt assigned to lowest priority	Interrupt requests in ascending priority	Notes
IN0	0 - 7 - 6 - 5 - 4 - 3 - 2 - 1	
IN1	1 - 0 - 7 - 6 - 5 - 4 - 3 - 2	
IN2	2 - 1 - 0 - 7 - 6 - 5 - 4 - 3	
IN3	3 - 2 - 1 - 0 - 7 - 6 - 5 - 4	
IN4	4 - 3 - 2 - 1 - 0 - 7 - 6 - 5	
IN5	5 - 4 - 3 - 2 - 1 - 0 - 7 - 6	
IN6	6 - 5 - 4 - 3 - 2 - 1 - 0 - 7	
IN7	7 - 6 - 5 - 4 - 3 - 2 - 1 - 0	Assignment at RESET

The priority status can be modified at any time, without affecting the content of the ISR, by issuing the command OCW2 RS.

6.14.4.3 Communication with the Internal Registers of the Programmable Interrupt Controller.

After initialisation, it is necessary to write the required mask byte into the IMR (Interrupt Mask Register). This register is initially reset to unmask all interrupts. Thus, the interrupts not required for service must be selectively masked.

The Interrupt Mask Register can normally be written to at any time after initialisation by executing a memory write to address 9FFF2.

The content of the mask register can also be read by executing a memory read from address 9FFF2.

The other status registers, IRR and ISR, can be exclusively read from address 9FFF0. Unless otherwise initialised, a status read will provide the user the current content of the IRR (Interrupt Request Register). Selection and deselection of the ISR can be made by issuing a special command word to address 9FFF0. These are as follows:

Mnemonic	Address	Data	Description
0CW3 RIS	9FFF0	0 0 0 0 1 0 1 1	Select ISR
0CW3 RR	9FFF0	0 0 0 0 1 0 1 0	Deselect ISR

Table 3. Status Register Select Group

It is not necessary to repeat these command words before the execution of every status read. The interrupt controller will always return the content of the current selected register.

6.14.4.4 Polled Interrupts - Their philosophy and implementation

It is sometimes necessary to be able to operate an interrupt priority structure under control of the operational program. When this interrupt method is used, the microprocessor shall not respond to an interrupt request. The interrupt controller will then wait for a software interrupt acknowledgement sequence. The user must first issue a poll request command. This is as follows:

Mnemonic	Address	Data	Description
0CW3 P	9FFF0	0 0 0 0 1 1 0 0	Poll, request status.

All of the interrupt requests will now be frozen at the current requesting priority (if one exists). A subsequent status read from address 9FFF0 will now reveal one of two conditions:

a) 0 X X X X X X X	No interrupt active
b) 1 X X X X n n n	Interrupt "nnn" requesting service.

If an interrupt is indicated by the polled status, its corresponding bit in the IRR is cleared, and set in the ISR. The ISR responds as normal to the End of Interrupt Group of commands.

6.14.4.5 Special Mask Mode

Normally, and following initialisation, an interrupt level currently being serviced (ISR bit set) will automatically mask requests from all interrupts of lower priority. This feature can be disabled and reenabled by the following two commands:

Mnemonic	Address	Data	Description
OCW3 SM	9FFF0	0 1 1 0 1 0 0 0	Set special mask mode
OCW3 RSM	9FFF0	0 1 0 0 1 0 0 0	Reset special mask mode

Special Mask Mode - commands

6. 14. 4. 6 Conclusion

The Programmable Interrupt Controller provides the user an extremely powerful control structure. The above section describes all of the functional commands relating to its application on the DCE-X86 module. Examination of a manufacturers' documentation for the device will reveal an extended set of possible commands relating to other applications for this particular device. The user is thus recommended to confine himself to the description given above in order to avoid any confusion with this device.

6. 14. 5 Communications Control Device

The serial communications capability is provided by a USART (Universal Synchronous/Asynchronous Receiver and Transmitter) type n° 8251A. This device is functionally identical to that described in the chapter on the RWC-CCE interface module.

The USART registers appear as memory addresses to the 8086 microcomputer. These addresses, which may be either read from or written to, are as follows:

Address	Description
3F000	Transmit or Received data
3F002	Mode/Command or Status.

Communications Device Addressing.

The device is first initialised by writing the required mode and command bytes to address 3F002. The same address can then be read, to establish status, and data transferred either to or from the device by using address 3F000. Details of this initialization and execution sequence is given on page 7-126 of this manual, under the sections entitled "USART Devices for channel 1 and 2". The reader must be sure not to be confused with the different register addressing found in those chapters, and also to remember that the RWC-CCE documentation describes a module that includes 2 USART devices.

6. 14. 5. 1 Communications device connector.

The assignment of the signals on this connector is different from that found on the DCE-1, -2, -X family of modules. Several signal definitions have been changed in order to exploit the operational advantages of the communications control device.

The following pin definitions are valid for the DCE-X86 processor module:

SIGNAL NAME	DESCRIPTION	PIN
SOPOS	Serial Output (Positive)	21
SONEG	Serial Output (Negative)	22
SIPOS	Serial Input (Positive)	17
SINEG	Serial Input (Negative)	18
$\overline{\text{DSR}}$	Data set Ready	1
$\overline{\text{CTS}}$	Clear to send	4
TxC	Transmit clcck input	5
RxC	Receive clock input	6
CkOUT	Serial clock output	7
TxD	Transmit data - TTL	2
RxD	Received data - TTL	3
+12V	+12V from system connector	14
+5V	+5V from system connector	15
-5V	-5V from system connector	16
GND	Signal and supply ground	19

The connector fitted to the module is a 25 pin D-type male.

Under usual operating conditions, the clock connections (pins 5, 6, and 7) are connected together by user selectable jumpers provided on the module. These jumpers are fitted to all standard DCE-X86 modules. The individual connections are present to enable either synchronous or isosynchronous communication methods to be employed.

TTL level signals of the transmitted and received data are available. These enable communications to be made at rates above 9600 bauds. Opto-isolated communications can only be made at transmission rates at or below 9600 bauds. The maximum rate allowed with these TTL signal lines is 50k Baud. Modem handshake lines are provided for the control of character transmission. Received characters must ALWAYS be handled by the application program. No control for the received communications is provided.

6. 14. 5. 2 Serial Interface - Interrupts

The serial communications interface is supported by 3 of the system's interrupts. Two of these, "Transmit buffer empty" and "Receive buffer full" are also found on the other DCE cards which employ the TICC device. In addition, a special interrupt is provided for.
This interrupt, SYNC/BREAK detect, causes a request for system interrupt whenever a specific condition occurs. When programmed to use asynchronous communications, the communications device will cause an interrupt request whenever a break condition occurs. Thus, the break key on the attached terminal device can be used to cause a specific system interrupt. This feature is employed in DAI Support software and is used to cause a return to the system utility program.

When communications are being made with synchronous communication, this interrupt will occur whenever the receiver detects a valid synchronisation code. This interrupt will then be used to ready the receive character routines to receive a block of data. By using a suitable application program, high-level communications protocols including BSC can therefore be supported by the DCE-X86 module.

6. 14. 6 Programmable Rate and Timing device

This device is identical to that employed in DAI's RWC-CCE module. However, the assignment of functions to each of the three timer-counters is dedicated as follows:

Timer counter 0 : Communications rate generator. Connected to device connector pin 7.
Timer counter 1 : Interrupting system timer. Connected to interrupt controller.
Timer counter 2 : idem.

A full description of this device, the 8253A, is found under the sections relating to the RWC-CCE module. Exceptions to this however are as follows:

6.14.6.1 8253A Addressing

The 8253A device can be selected at the following hexadecimal addresses.

Address	Description
5F000	Counter 0, read or write
5F002	Counter 1, read or write
5F004	Counter 2, read or write
5F006	Write mode word.

Note: DO NOT read from location 5F006.

6.14.6.2 8253A Modes

Each of the three timer-counters of the 8253A can be selected to operate in any one of six possible pulse or waveform modes. Under the normal operating conditions applied to the DCE-X86 module, the following modes are used:

Channel	Function	Mode select byte	
		Binary	BCD
0	Baud rate generator	36	37
1	Timer	70	71
2	Timer	B0	B1

Timers 1 and 2 can also be programmed to perform a repeating timer interrupt function. This is selected by writing the following mode select bytes:

Channel	Function	Mode select byte	
		Binary	BCD
1	Repeating timer	74	75
2	Repeating timer	B4	B5

In all cases, a required count is written to the selected timer by the following sequence:

(i) Send "Mode select byte" to mode address.

(ii) Send least significant 8 bits followed by most significant 8 bits to selected counter address.

In all cases, the counters, each of 16 bit operation, can be selected to operate in either a binary or BCD (Binary Coded Decimal) count-down format.

The timers are also provided with an extra 16 bit register. This can be loaded with the current count of any of the 3 counters by issuing one of the following mode select bytes to address 5F006.

Counter	Mode Select byte to latch count
0	00
1	40
2	80

Following the issuance of one of these bytes, the next two read operations from the selected counter will reveal first the least and then the most significant bytes of the latched counter.

In order to reset the internal operation of the device, it is necessary to perform two dummy read operations from each counter (see RWC-CCE section, page 7-132).

6.14.6.3 Baudrate generation

In order to generate any of the standard Baud rates used for serial communications, a correct 16 bit count value must be loaded for the operation of counter-timer 0. In order to acheive rate errors of less than 5 %, the following table of hexadecimal count values has been prepared. The counter is to be operated in the binary mode.

Baud rate	Count values (Hexadecimal)		
required	x1	x16	x64
45	B1C7	0B1C	02C7
50	A000	0A00	0280
75	6AAB	06AB	01AB
110	48BA	048C	0123
134,5	3B7B	03B8	00EE
150	3555	0355	00D5
200	2800	0280	00A0
300	1AAB	01AB	006B
600	0D55	00D5	0035
1200	06AB	006B	001B
2400	0355	0035	000D
4500	01AB	001B	0007
9600	00D5	000D	-
19200	006B	0007	-

Note: Both high and low bytes MUST be written to the device, even when high byte is zero.

6.14.6.4 Timing calculation

The count frequency, common to all three timer-counters, is 2.048 MHz. Thus, the period for each count is approximately 500 ns.

6.14.7 16-bit, X-BUS DESCRIPTION

In order to support the extended data and address requirements of the DCE-X86 module, the following pin allocations have been made.

Pin number	Name	Description
1	D8	16 bit data-bus
2	D0	
3	D9	
4	D1	
5	D10	
6	D2	
7	D11	
8	D3	
9	D12	
10	D4	
11	D13	
12	D5	
13	D14	
14	D6	
15	D15	
16	D7	
17	$\overline{\text{BHE}}$	High bank enable D8-D15
18	AO-86	Low bank enable D0-D7

19	GROUND	
20	MEMW	Memory write strobe
21	GROUND	
22	N. C.	
23	A.11	Address line 11
24	$\overline{\text{MEMR}}$	Memory read strobe
25	A15	Address lines
26	A12	
27	A13	
28	A14	
29	A10	
30	A16	
31	A8	
32	A9	
33	A6	
34	A7	
35	A4	
36	A5	
37	A2	
38	A3	
39	A1	
40	INTA	Interrupt acknowledge
41	GROUND	
42	$\overline{\text{WRQ}}$	Wait cycle request
43	N. C.	
44	N. C.	
45	GROUND	
46	$\overline{\text{HOLDA}}$	Hold acknowledgement
47	N. C.	
48	CK2(TTL)	CPU clock, phase 2.
49	$\overline{\text{HOLD}}$	Hold request
50	ALE	Address latch enabled.

Notes:

1. The signal names are left tabulated if they are standard X-BUS assignments, and right-tabulated if unique to the DCE-X86 module.
2. N. C. No connection.

6.14.8 Operational Requirements

Power Requirements

The DCE-X86 module requires three power supplies. These are usually provided by the DCE system power supply module via the DCE-BUS. Typical power requirements in the quiescent state are given below. Active state values are typically 20 % higher.

+12V	± 5%
+5V	± 5%
-5V	± 5%

Environmental requirements

Operating temperature	: 0°C to 55°C
Storage temperature	: -25°C to +85°C
Relative humidity	: 95% non-condensing.

6.14.9 ORDERING INFORMATION

DCE-X86	: Standard DCE-X86 module with provision for 2716 device EPROMS.
DCE-X86/S	: Same as DCE-X86, but with 8086 socket mounted.
DCE-X86A	: DCE-X86 module with provision for 2732 device EPROMS.
DCE-X86A/S	: Same as DCE-X86A, but with 8086 socket mounted.

All of the above versions are fitted with the standard memory mapper (RAM at address 0) and software DCE-BUS mode operation.

Optionally, the following modules can be ordered:

X86-MAP	Optional memory mapper, places RAM at address B0000.
X86-FBUS	Optional FAST-BUS hardware handshake module.

Interface cables for standard terminals and memory expansion must be ordered separately.

8. SYSTEM ENHANCEMENT MODULES

8.1 DCE-PWR : PLUG-IN POWER SUPPLY MODULE

8.1.1 FUNCTIONAL DESCRIPTION

The DCE-PWR is a regulated power supply module, which simply plugs into any standard DAI eurorack or eurobox. It operates from 220V or 110V A. C. input, and supplies the DCE-BUS power rails with regulated +5V, -5V and +12V D. C. The current outputs of 2. 5A (+5V), 1A (-5 V) and 1A (+12V), are sufficient for DCE microcomputer systems with medium power requirements.

8.1.2 FEATURES

- Regulated plug-in power supply module.
- Plugs into any DAI eurobox or eurorack.
- Outputs of regulated +5V (2. 5A), -5V (1A) and +12V (1A).
- 220V or 110V A. C. input.
- On/off switch and power-on LED indicator on front panel.
- Ripple less than 100 mV with severe dynamic load conditions.
- 3 mV typical ripple.

8.1.3 HARDWARE CONFIGURATION

The DCE-PWR module has an on/off switch and a power-on LED indicator on the front panel. All input and output connections are brought to 31-pin and 13-pin male connectors at the back of the module. These plug into corresponding female connectors on the DCE-BUS motherboard at the back of all standard DAI euroboxes and euroracks.

The pin definitions of these two male connectors are as follows:

31-pin connector

pin	signal
1,2	-5V
3,4	+12V
5 → 28	GND
29 → 31	+5V

13-pin connector

pin	signal
2	0V
6	110V
10	220V
13	GND

Mains power is derived via a separate Mains Power Adaptor module (PSM-MPA or PSM-MPA/C), which can be mounted at the front or back of the corresponding eurorack or eurobox. This MPA unit provides a mains power cable and connector, a power line noise filter, a fuse, an on/off switch, and an optional cooling fan (MPA/C). The power line noise filter provides approximately 30 dB attenuation for mains interference in the frequency range 600 Hz to 100 MHz.

An optional bench-top adaptor (PSM-PWR/B) is available for enclosing the DCE-PWR module and converting it to a bench-top version with 220V A. C. input. This adaptor provides an enclosure for plugging in the DCE-PWR module and has rubber feet, a fuse, a D. C. power output cable with a 4-pin connector and a mains cable.

8. 1. 4 ORDERING INFORMATION

DCE-PWR : Standard version.

PSM-PWR/B : The optional bench-top adaptor must be ordered separately.

PSM-MPA/C, MPA : The mains power adaptor module, with or without cooling fan, must be ordered separately.

8.2 DCE-PWR/H : HEAVY-DUTY PLUG-IN POWER SUPPLY MODULE

8.2.1 FUNCTIONAL DESCRIPTION

The DCE-PWR/H is a heavy-duty regulated power supply module, which simply plugs into any standard DAI eurorack or eurobox. Because of its extra width it occupies more space than the DCE-PWR module when plugged in, and covers one extra card guide adjacent to it. It operates from 220V or 110V A. C. input, and supplies the DCE-BUS power rails with regulated +5V, -5V and +12V. The current outputs of 10A (+5V), 2A (-5V) and 2A (+12V) are sufficient for powering all DCE microcomputer configurations, including the diskette system.

The module contains an over-heating cut-out, which will cause all the power outputs to be switched off automatically if over-heating occurs for any reason. If any of the three output voltages rise above specified limits, an over-voltage protection feature will blow the fuse and switch off all outputs.

The module continuously monitors the A. C. mains supply, and the absence of a half-cycle produces a mains failing signal, which can be used to implement an automatic 24V battery back-up feature.

8.2.2 FEATURES

- Heavy-duty regulated plug-in power module.
- Plugs into any standard DAI eurobox or eurorack.
- Outputs of regulated +5V (10A), -5V (2A) and +12V (2A).
- Over-heating and over-voltage protection.
- 110V and 220V A. C. inputs.
- Ripple less then 200 mV under full load.
- Mains power failing signal, and 24V battery back-up capability.

8.2.3 HARDWARE CONFIGURATION

The DCE-PWR/H module is based on a switching type design giving a higher efficiency than the conventional types. It contains an over-heating cut-out, which will cause all the power outputs to be switched off automatically if over-heating occurs for any reason. The over-heating cut-out will switch off the power outputs without blowing the fuse. After such a cut-out, the power supply must be disconnected from the mains and left to cool before it can function again.

If any of the three output voltages rise above specified limits, an over-voltage protection feature will switch off all outputs by blowing the fuse. The over-voltage protection feature is triggered at the following approximate levels:

+5V	:	5.8V
-5V	:	1.4V
+12V	:	12.7V

A LED indicator on the front panel of the module provides a visual indication of active mains or battery power input.

The power supply module continuously monitors the A. C. mains supply, and the absence of a half-cycle causes a logic 1 to appear on the "Mains Failing" signal at pin 13 of the 31-pin connector. During normal operation this signal is at logic 0. This signal is not latched, and therefore a logic 1 appears on it only during the time that the half cycles of the mains supply are not being detected. This Mains Failing signal can be used to implement an automatic switch-over to an external 24V battery, to ensure system operation without interruption.

All input and output connections are brought to 21-pin and 31-pin male connectors at the back of the module. These plug into corresponding female connectors on the DCE-BUS motherboard at the back of all standard DAI euroboxes and euroracks.

The pin definitions of these two male connectors are as follows:

31-pin connector

pin	signal
1,2	-5V
3,4	+12V
5 → 8	GND
9 → 12	24V battery input
13	Mains Failing
28 → 31	+5V

21-pin connector

pin	signal
8	0V
12	110V
16	220V
19	GND

Mains power is derived via a separate Mains Power Adaptor module (PSM-MPA or PSM-MPA/C), which can be mounted at the front or back of the corresponding eurorack or eurobox. This MPA unit provides a mains power cable and connector, a power line noise filter, a fuse, an on/off switch, and an optional cooling fan (MPA/C). The power line noise filter provides approximately 30 dB attenuation for mains interference in the frequency range 600 Hz to 100 MHz. A separate PSM-MPA/B module, very similar to PSM-MPA/C, provides additional connections to a 24V battery and a trickle charging supply. Another separate module, PSM-BAT, provides a 24V 1.8 Ah battery with trickle-charge circuitry on a panel, for mounting on the back of a large eurorack or eurobox in place of the PSM-PCP/L cover panel.

8.2.4 ORDERING INFORMATION

DCE-PWR/H	: Standard version.
PSM-MPA, MPA/B, MPA/C	: The mains power adaptor module, with or without the battery back-up and cooling fan options, must be ordered separately.
PSM-BAT	: The optional back-up battery module must be ordered separately.

8.3 DCE-PRG : EPROM PROGRAMMER MODULES

8.3.1 FUNCTIONAL DESCRIPTION

The DCE-PRG/8 and DCE-PRG/16 modules provide all the necessary hardware functions for programming and accessing 2708 or 2716 EPROMs under DCE control. Each module has a zero-insertion-force socket for easy insertion of EPROMs. A programming disable switch on the module prevents accidental alteration of EPROM contents while in the socket. A LED lamp provides a visual indication of when a programming operation is in progress.

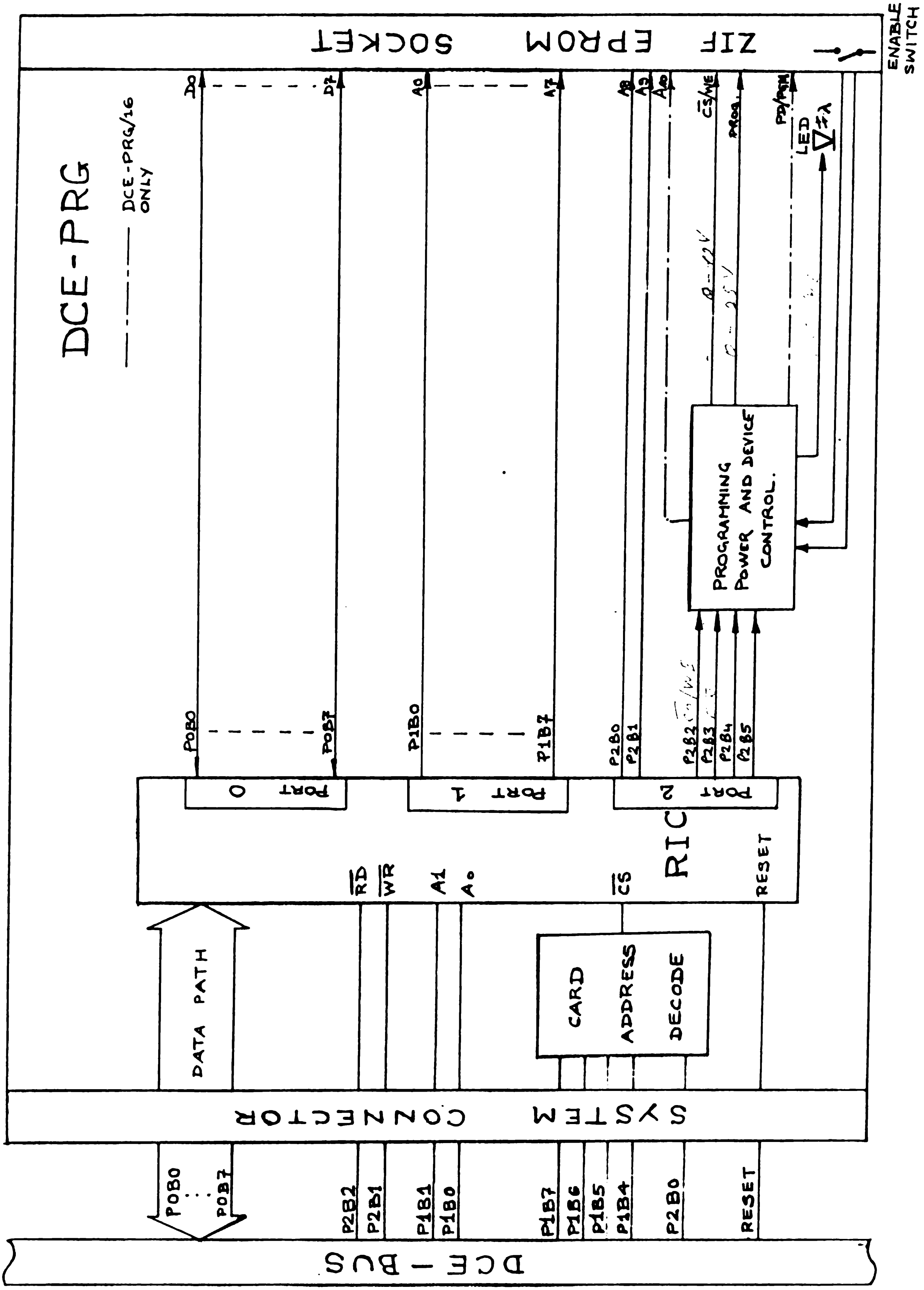
All DCE microcomputer Utility software packages contain routines for driving the corresponding versions of DCE-PRG modules. A Programming function enables the contents of any DCE system memory block to be transferred to one or more locations of an erased EPROM in the socket. After programming, the new contents of the EPROM locations are automatically compared with the corresponding memory contents, and any discrepancies reported. Utility commands can be used to compare contents of selected locations of an EPROM with memory, and to transfer selected locations to system RAM memory.

Each card has an identification address defined by a hexadecimal switch, and up to fifteen can be directly connected to the DCE-BUS. This feature enables several DCE-PRG modules to be configured together for programming several EPROMs simultaneously.

8.3.2 FEATURES:

- Programs 2708, or 2716 EPROMs under DCE microcomputer control.
- LED lamp for visual indication of programming operation.
- Programming disable switch prevents accidental alteration of EPROM contents.

8. 3. 3 FUNCTIONAL BLOCK DIAGRAM



- ° Includes zero-insertion-force EPROM socket.
- ° Uses standard DCE-BUS power supplies.
- ° Standard hardware and software interface to the DCE-BUS.
- ° Switch selectable module address.
- ° Single 100 x 160 mm eurocard format.

8. 3. 4 SYSTEM DESIGN PARAMETERS

8. 3. 4. 1 Hardware Configuration

The functional block diagram given in section 8. 3. 3 illustrates the hardware configuration of the DCE-PRG module. It has the standard RIC interface to the DCE-BUS.
PORT 1 of the RIC and two (or three) lines of PORT 2 are used to address the EPROM contents. PORT 0 is used to transfer data both to and from the EPROM. A Hexadecimal switch allows the module card address to be configured. Thus, multiple EPROM programming modules can be independently accessed.

8. 3. 4. 2 Programming Specifications

The DCE-PRG is addressed via the DCE-BUS via the standard DCE-BUS interface (see section 4. 1 of this manual).
DAI System software assumes the card addresses as follows:

2704/2708	DCE-PRG/8	ADDRESS = 'F'
2516	DCE-PRG/16	ADDRESS = 'B'

The timing requirements of the programming process are controlled by the system CPU. Suitable drive software is included in DAI's UTILITY and REAL-WORLD BASIC programs, versions of which exist for either 2708 or 2716 device programming.

Reading data from an EPROM inserted into the DCE-PRG module can be achieved by initialising the RIC with control word 90H. Thereafter, any address written to RIC port 1, and bits 0, 1 and 2 of port 2, can be read directly on RIC port 0. All other bits on port 2 shall be set to 0.

Programming is enabled by setting the program-switch up. The RIC shall be initialised with control-word 80H. Then, for each byte set on RIC port 0, the required address is set on port 1 and 2 (as above) with the port 2 data merged with either 0CH (for DCE-PRG/8) or 20H (for DCE-PRG/16). The programming pulse is then given by using the RIC bit-set and bit-clear facility. P2B2 is pulsed low on the DCE-PRG/8. For the DCE-PRG/16 module, P2B3 is pulsed high.

The user must never insert or remove the EPROM whilst the device is selected for programming. After the programming cycle is complete, initialisation of the module to read (control word 90H) will extinguish the indicator LED.

8.3.4.3 Operational Requirements

Signal characteristics.

When operated by the standard DAI programming software, the DCE-PRG modules provide an approved EPROM programming facility. All signal levels and timing are according to the specified programming conditions provided by the device manufacturer.

EPROM types

All versions of the 2708, 8708, 2704 family of EPROMs may be programmed with the DCE-PRG/8.

Note: Only single +5V supply EPROMs may be programmed with the DCE-PRG/16. Earlier 3 supply devices (e.g. TMS-2716) should not be used with this programmer module.

Power Requirements

The DCE-PRG card uses all 3 DCE-BUS supply lines. Typical power consumption from each, under programming conditions, are:

PRG/8:	+5V	120 mA	PRG/16:	+5V	175 mA
	+12V	75 mA		+12V	85 mA
	-5V	50 mA		-5V	50 mA

Environmental Requirements

Operating temperature	: 0°C to 55°C
Storage temperature	: -25°C to 85°C
Relative humidity	: 95 % non-condensing

Bus Loading

The RWC-PRG module presents 1 unit-load to the DCE-BUS (see Section 4.4).

8.3.5 ORDERING INFORMATION:

DCE-PRG/8	: Standard version for 1K byte 2708 EPROMs.
DCE-PRG/16	: Standard version for 2K byte 2716 EPROMs.

8.4 DCE-SBM : SYSTEM BUS MONITOR

8.4.1 FUNCTIONAL DESCRIPTION:

The DCE-SBM System Bus Monitor module provides 24 LED indicator lamps for monitoring the status of the address, data and control lines of the DCE-BUS. 24 switches, also connected to the DCE-BUS lines, enable manual generation of input signals to the bus. A system Reset switch enables the entire DCE microcomputer system including RWC Real-World interface modules to be reset. Two jumper contacts allow manual generation of the two interrupt request signals carried by the DCE-BUS.

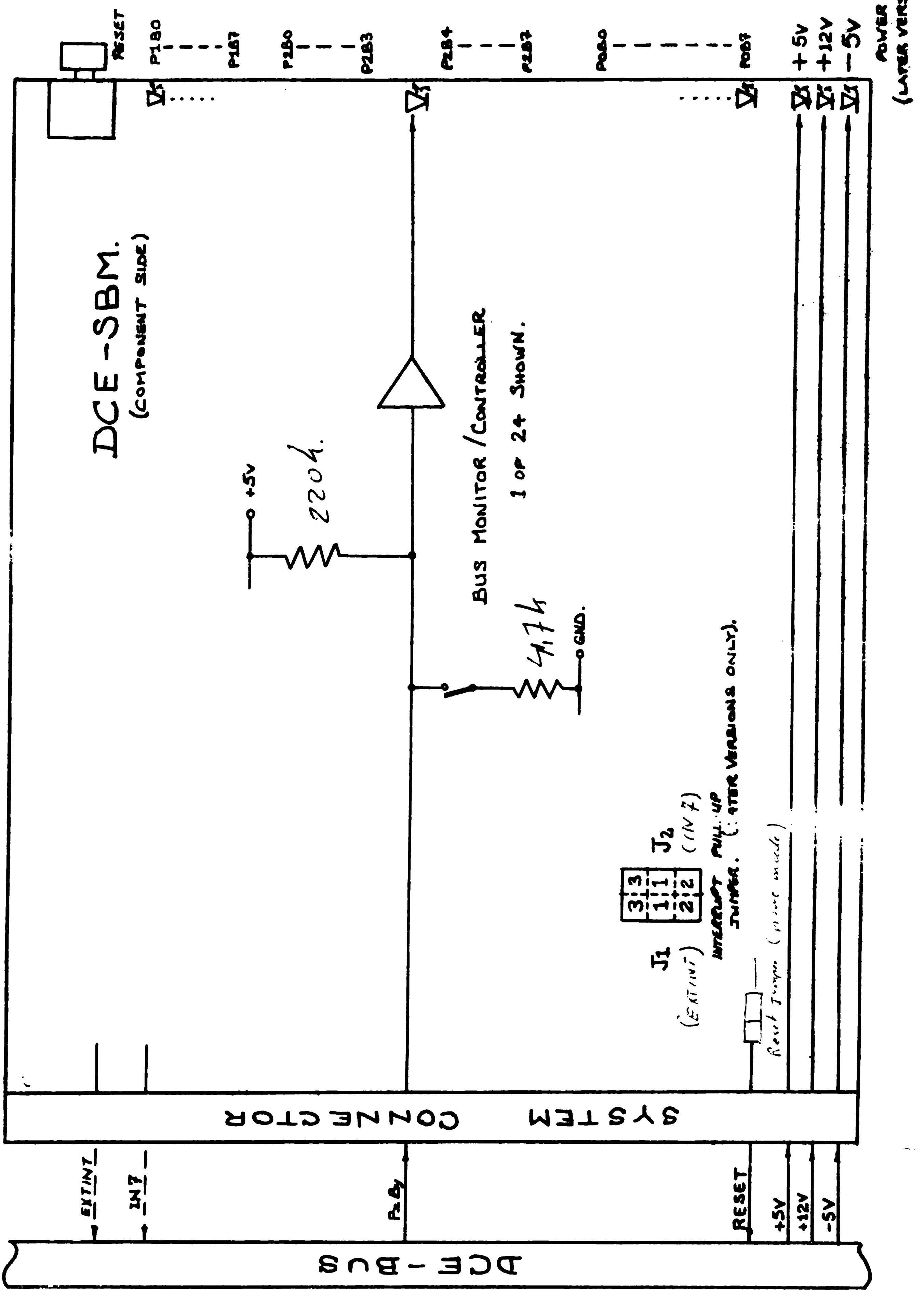
The 24 LED indicators are arranged as two groups of eight and two groups of four, to match the organisation of the three programmable ports of the DCE microcomputers. Similarly, the 24 switches are arranged as three groups of eight. When the DCE microcomputer is used as a stand-alone controller, the DCE-SBM can be used during the software development phase to simulate data inputs, and monitor signal outputs. It can also be used to drive RWC interface modules manually for testing purposes.

The Reset switch and the LED indicators are mounted at the front edge of the module, for ease of access when plugged into a system. In order to make the switches accessible, the module should be inserted into the eurobox or eurorack, via a DCE-EX extender card. All the circuitry of the DCE-SBM module will then be completely accessible.
Later versions of the DCE-SBM also include LED indicators for each of the three system power supply rails.

8.4.2 FEATURES:

- 24 LED indicator lamps displaying the status of DCE-BUS lines.
- 24 switches for manual input of signals to the DCE-BUS lines.
- System Reset switch.
- Jumper contacts for manual generation of the two DCE-BUS interrupt request signals.
- Single 100 x 160 mm eurocard format.

FUNCTIONAL BLOCK DIAGRAM



8.4.4 SYSTEM DESIGN PARAMETERS

8.4.4.1 Hardware Configuration

The DCE-SBM provides LED indicators of the DCE-BUS. The status of DCE-lines, normally driven by the three ports of the GIC, are given by a correspondance of: 0 = LED off; 1 = LED on.

The module also provides 24 switches, arranged as three banks of eight. The switches can be used to drive the 24 lines of the DCE-BUS for manually performing data transactions to and from any DCE-BUS compatible module.

A pushbutton switch is also provided on the DCE-SBM module. This is connected to the system reset line of the DCE-BUS. Activation of this push-button will cause a hardware reset of the DCE microcomputer, and also of any Real-World cards that are connected to the reset line. Thus, all RIC devices are configured as read ports, etc.

Later versions of the DCE-SBM module also provide the following additional features:
Firstly, all three DCE power supplies are monitored, and their presence is shown by three status LEDs provided for this purpose. Secondly, a pull-up or pull-down feature can be selected by positioning a small movable jumper. One is provided for each system interrupt line. J1 controls the EXTINT line, whilst J2 controls IN7. Configuration by each is as follows:

Pull-up:	Link 1 and 3
Pull-down:	Link 1 and 2

The module is supplied with the jumpers positioned to affect a pull-down function.

Use of the switches provided on the DCE-SBM module.

The switches can be used to manually set up addresses on the DCE-BUS. They can also be used to generate the other signals necessary to affect a communication of data over the DCE-BUS. Each switch corresponds directly with each of the 24 data LEDs provided on the module when the lines are in input mode.

Driving a DCE-BUS line by switch on DCE-SBM

There are 24 switches available for driving the 24 lines of the DCE-BUS. If a particular switch is required to drive any DCE-BUS line, any other device connected to the same line must be operating as an input. (If a device is currently driving the line, the operation of the switch will have no effect). In this way the status of a DCE-BUS line is determined by its respective switch as follows:

switch closed: line pulled to logic '0'
switch open: line pulled to logic '1'

The LEDs indicate the status of the line as usual.
If a particular switch is not used for determining the status of a DCE-BUS line, it must be placed in its 'OPEN' position.
Whenever the DCE-SBM module is only used for monitoring the DCE-BUS, all 24 switches must be in their OPEN positions.

8.4.4.2 Operational Requirements

DCE-BUS Loading

DCE-BUS	Switch Status	Current
0	OPEN	: 50 µA
1		: 10 µA
0	CLOSED	: 50 µA
1		: 1 mA

Power Requirements

+5V	: 280 mA, all LEDs on.
+12V	: 20 mA
-5V	: 20 mA

Environmental Requirements

Operating temperature	: 0°C to 55°C
Storage temperature	: -25°C to +85°C
Relative humidity	: 95 % non-condensing

Bus Loading

The DCE-SBM module presents 1 unit-load to the DCE-BUS (see section 4. 4). All switches are open.

8. 4. 5 ORDERING INFORMATION

DCE-SBM	: Standard version.
DCE-EX	: Extender card must be ordered separately.

8.5 PDM-KDU : HAND-HELD KEYBOARD/DISPLAY UNIT

8.5.1 FUNCTIONAL DESCRIPTION

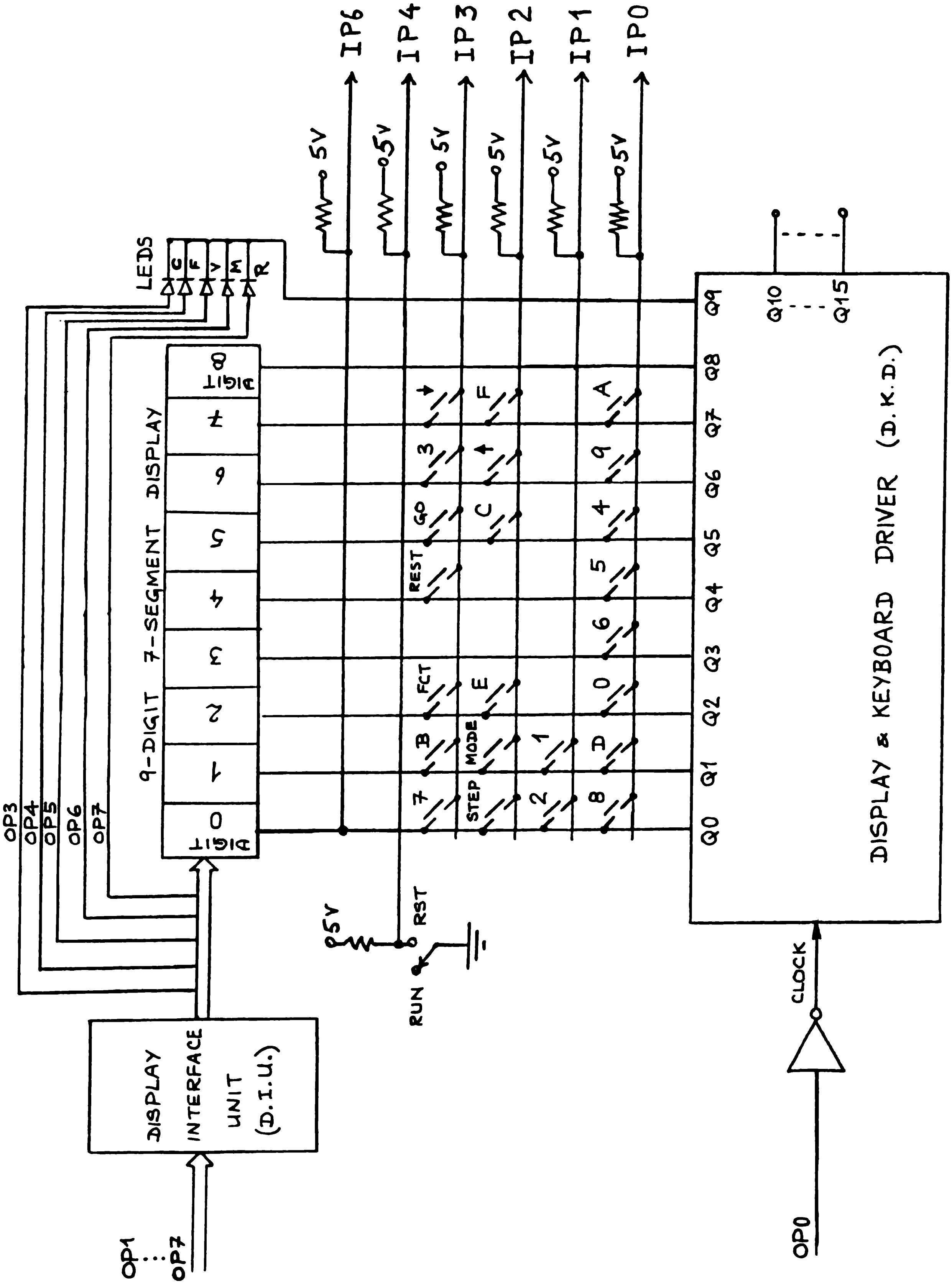
The PDM-KDU is a very low-cost alternative to a teletype or a terminal for interacting with DCE microcomputer systems. It is provided with a connecting cable, and can be plugged directly into the TICC parallel ports of the DCE microcomputer via the device connector.

The PDM-KDU driver Utility program UPK provides the following functions and facilities: system initialization, display/modify memory locations and CPU registers, move and fill memory blocks, set-up or change the RAM vector addresses of interrupt service routines, initiate program execution, single-step and trace program execution and program EPROMs including compare-with-memory and transfer-to-memory functions. The PDM-KDU provides all necessary functions for program modification in the field. Only an EPROM programmer card is necessary for transferring the modified program on to EPROMs.

8.5.2 FEATURES

- Hand-held keyboard/display terminal for DCE microcomputer systems.
- 23 push switches for hexadecimal data and function input.
- 7-segment LED display for 9 hexadecimal digits.
- 5 modes of operation: Register, Memory, Vector, Function, Compare.
- 5 LEDs for indicating the current mode of operation.
- EPROM programming functions: Program, Transfer, Compare.
- Utility functions: Zero and initialize, Move, Look, Fill.
- Program debug functions: enter program, single-step and program trace.
- Display or modify DCE memory and CPU registers.
- Direct interface to DCE microcomputers via TICC parallel ports.

8.5.3 FUNCTIONAL BLOCK DIAGRAM



8. 5. 4 SYSTEM DESIGN PARAMETERS

8. 5. 4. 1 Hardware Configuration

Section 8. 5. 3 shows a functional block diagram of the Keyboard Display Unit (KDU). The KDU houses twenty four switches. Twenty three of these switches are Key pad type and the twenty fourth is a two position switch. The keys in the block diagram are marked with the same symbols as on the front plate of the KDU. The KDU also houses a 9-digit 7-segment display, and five light emitting diodes indicated by the symbols C, F, V, M, R.

Communication with the KDU is done via 16 lines. Fourteen of these lines are signal lines, and the other two lines are used to supply +5V and 0V power to the KDU. Eight of the signal lines (OP0 to OP7) are used to send data to the KDU. The remaining six lines (IP0 to IP4 and IP6) are used to receive data from the KDU. This 16-line interface enables the nine-digit display to be activated, any or all of the five LEDs to be illuminated, and all the switches to be monitored.

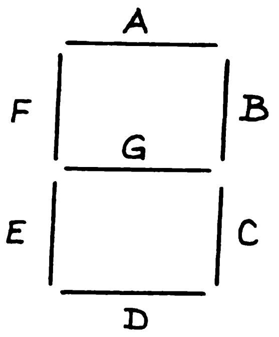
9-digit Display

The 9-digit display is used dynamically, and must be periodically refreshed. Each digit is made up of seven segments. The 'Display Interface Unit' (D. I. U.) directly drives all anodes of all the digits. The particular digit of the display, whose segments are to be illuminated by the data from the D. I. U. is selected by a signal from the ' Display and Keyboard Driver' (D. K. D). The D. K. D. allows current to pass through to the cathode of any one of the digits or LEDs thereby illuminating it.

Signal lines OPl to OP7 drive the D. I. U, whose input requirements are those of TTL. They drive the anodes of the five LEDs and of the segments in the selected digit. Table 8. 5. 1 indicates the relationship between these signal lines, the anodes of the LEDs, and the anodes in the segments of the digit selected by the D. K. D.

The D. K. D. is used to select the cathodes of the required digit or LED to be activated at any given time. The D. K. D. is a dynamic device and needs a clocking signal. Signal line OP0 is used for this. The requirements of OP0 are those of TTL. The D. K. D. essentially comprises a 16-stage counter with 16 output lines Q0 to Q15. Eight of these output lines are used to select the cathodes of the digit display. The required cathode to be activated is chosen by clocking the D. K. D. an appropriate number of times. A high going edge on OP0 is used to provide the clock. The D. K. D. has an automatic time-out of 7. 5 msecs. This means that if clock pulses are not provided every 7. 5 msecs, the 9-digit display and the LEDs are deactivated and none of the cathodes are allowed to sink current. In this condition the 16-bit counter within the D. K. D. is set to count 15. This condition remains until the arrival of the next clock pulse.

The counter within the D. K. D. unit counts from 0 to 15 and activates outputs Q0 to Q15 sequentially (active low). Each clock pulse increments its count by one and activates the next output line. When at count 15, a further clock puts it back to count 0. To activate the cathode of digit "i" requires the counter within the D. K. D. to be clocked to count "i" (0 to 8). The KDU allows the monitoring of the state of the counter within the D. K. D. The signal line IP6 is used for this. This signal is TTL compatible. A logic '0' on this line indicates that the D. K. D. counter is at value 0.



Segment in selected digit	Anode drive Signal	LED whose anode is driven
A	OP7	R
B	OP6	M
C	OP5	V
D	OP4	F
E	OP3	C
F	OP2	
G	OP1	

Note: A logic "1" on the anode drive signal line will activate the corresponding anode on LED and/or segment.

Table 8. 5. 1 : Anode Driver Signals

LED Display

The five LEDs are activated in the same way as the digits of the display, and the five lines OP1 to OP7 are used to activate their anodes. Table 8. 5. 1 indicates signal line allocation. The five LEDs have a common cathode driven from one of the D. K. D.

outputs (Q9). A value of 9 within the D.K.D. counter allows current to pass through to all the cathodes of the LEDs.

The clocking rate requirement for the D.K.D. (at least every 7.5 msecs) also applies for the LEDs. Failure to meet this requirement causes the LEDs to turn off due to the fact that the counter will be automatically set to 15.

As can be seen by the above details, the signal lines OP1 to OP7 control the display of either one digit, or, any or all of the LEDs at any one time. The display and LEDs therefore require regular refreshing to provide an apparent static display to the viewer.

Scanning the Switches

The 23 Keypad switches are arranged as a matrix for ease of monitoring. The matrix has 8 columns and 4 rows. The columns are activated by the D.K.D. via its counter. The signals from the D.K.D. connecting to the cathodes of digits 0 to 7 of the display are also used to drive the columns. The rows are read via signal lines IP0 to IP4. A particular keypad switch is read by activating the column containing that switch, and by reading the appropriate row from IP0 to IP3 to see if it is activated. A logic zero read from a row implies that the corresponding Key on the activated column has been pressed. For example, to sense if the key "6" at the intersection of column Q3 (which also connects to digit 3) and row 0 (which corresponds to signal line IP0) is being pressed, the following is done. The counter of the D.K.D. is clocked to a count of 3. The signal line IP0 is then read to see if it is at logic "0". If it is, then the key "6" is being pressed.

It should be noted that the output signals from the counter within the D.K.D. used to drive the keyboard matrix columns are also used to drive the cathodes of the digit display and LEDs with the specified refresh requirement.

The two-position switch on the KDU is not on the matrix connecting the Keypad switches. The position of this switch is monitored by simply reading signal line IP4. This line is TTL compatible.

Table 8. 5. 2 gives the correlation between the KDU switches, the required values in the D. K. D. counter, and the pin numbers on the device connector carrying the sensing signals (IP0 to IP3).
Note: Only one key should be pressed at any one time.

8. 5. 4. 2 Software Driver

The Utility program UPK provides all the necessary functions for using the PDM-KDU module, including the use of the DCE-PRG EPROM programmer modules. It provides sophisticated facilities for developing programs in machine language at minimum cost, and for program modification in the field. These functions include system initialization, display/modify memory locations and CPU registers, move and fill memory blocks, set-up or change the RAM vector addresses of interrupt service routines, initiate program execution, single-step and trace program execution by displaying all CPU registers after each instruction execution, and program EPROMs including compare-with-memory and transfer-to-memory functions.

The PDM-KDU can also be used as an I/O device for the input and output of hexadecimal data by using a suitable software driver routine. Section 8. 5. 6 gives an example of a complete software routine that enables the usage of the PDM-KDU as a normal input and output device.

KDU Switch	Count in D. K. D.	Device connector pin for sense signal (IP0 to IP3)
0	0	1
1	1	2
2	0	2
3	6	4
4	5	1
5	4	1
6	3	1
7	0	4
8	0	1
9	6	1
A	7	1
B	1	4
C	5	3
D	1	1
E	2	3
F	7	3
FCT	2	4
GO	5	4
REST	4	4
↓	7	4
STEP	0	3
MODE	1	3
↑	6	3
RUN/ RST	-	5 RST: Sense '0' RUN: Sense '1'
Q0 of D. K. D.	0	7

Table 8. 5. 2 : Correlation between Switches, Counter Values and Sensing Signals.

8.5.4.3 Module Connector Definitions

The PDM-KDU includes a connecting cable with a 25-pin D-type female connector. It can be plugged directly into the device connector of any 8080 CPU based DCE microcomputer and can interface directly to the TICC parallel ports.

The pin definition of the 25-pin D-type female connector:

Pin Number	Signal
1	IP0
2	IP1
3	IP2
4	IP3
5	IP4
7	IP6
23	OP0
24	OP1
25	OP2
13	OP3
12	OP4
11	OP5
10	OP6
9	OP7
15	+5V
19	0V

Table 8.5.3 : PDM-KDU Connector Pin Definition

8. 5. 4. 4 Operational Requirements

Power Requirements

The PDM-KDU requires a single +5V power supply. When used with 8080 CPU based DCE microcomputers, this can be obtained directly from its device connector.

Maximum power requirement is:

+5V 200 mA

Environmental Requirements

Operating temperature	: 0°C to 55°C
Storage temperature	: -25°C to +85°C
Relative humidity	: upto 95 % non-condensing.

8. 5. 5. ORDERING INFORMATION

PDM-KDU : Standard version including connecting cable.

UPK : The PDM-KDU oriented Utility program must be ordered separately (specify for which DCE microcomputer).

8.5.6 SOFTWARE DRIVER FOR PDM-KDU

```
;KDU INPUT/OUTPUT ROUTINE.
;D. A. I.
;

;THIS ROUTINE ALLOWS THE PROGRAMMER TO DISPLAY
;A SIMPLE TEXT ON THE KDU.
;IT ALSO PROVIDES A DATA ENTRY FACILITY.

;ONCE THE TEXT HAS BEEN TRANSFERED TO THE
;DISPLAY BUFFER, ITS OUTPUT IS CONTINUOUS & AUTOMATIC
;UNTIL THE INPUT ROUTINE IS TERMINATED BY THE OPERATOR
;OR BY THE TIME-OUT OPTION.

;OUTPUT.
;ON ENTRY, THE HL REGISTER PAIR IS SET TO POINT TO
;THE LOCATION OF THE TEXT TO BE DISPLAYED. IT IS
;IMPORTANT TO NOTE THAT THIS TEXT MUST BE STORED
;IN DISPLAY CODE (SEE TABLE "DIST" FOR EXAMPLES).

;INPUT.
;THE B REGISTER IS SET TO THE NUMBER OF DIGITS
;TO BE ENTERED (FIELD WIDTH), AND THE C REGISTER
;IS SET TO THE LEAST SIGNIFICANT DIGIT POSITION
;OF THE FIELD IN THE 9-DIGIT DISPLAY (POSITION
;0 IS AT THE LEFT, 8 AT THE RIGHT).

;DIGITS ENTERED ON THE KEYBOARD ARE SHIFTED LEFT
;THROUGH THE FIELD IN THE DISPLAY, AND THE HEX
;VALUE OF EACH DIGIT IS COPIED INTO A DATA BUFFER.

;THE ONLY KEYS VALID IN THIS ROUTINE ARE THE HEX DIGITS
;(0-9, A-F) AND THE "ROLL-UP" (^) KEY (TO END THE ENTRY).
;A NULL ENTRY (NULLE) FLAG IS SET IF THE ROUTINE IS
;TERMINATED BY THE ROLL-UP KEY BEFORE ANY DIGITS
;HAVE BEEN ENTERED.
```

```
;EXAMPLE:-

;LXI     H,CHMSG                       ;DISPLAY "CH-        ".
;THIS ASKS FOR A CHANNEL NUMBER.
;MVI     B,2              ;2 DIGITS IN FIELD.
;MVI     C,4              ;ENTRY POINT IN DISPLAY,
;I.E. 5TH POSITION FROM LEFT.
;CALL    KDUIO            ;CALL ROUTINE.
;MOV     A,M              ;RETRIEVE MS DIGIT.
;INX     H                ;POINT TO NEXT LS DIGIT.
;ADD     A
;ADD     A
;ADD     A
;ADD     A                ;MULTIPLY BY 16.
;ADD     B                ;ADD LS DIGIT.
;STA     CHANNEL                       ;SAVE BINARY VALUE OF CHANNEL.

;THE "KDU BUFFER AREA" ORIGIN CAN BE MOVED TO
;SUIT YOUR PROGRAM.

;SUBROUTINES :-

;DISP    DISPLAY DRIVER, KEYBOARD SCAN & DECODE.
;XFER    MOVE TEXT (H,L) TO DISPLAY BUFFER.
;DISPA   SETS UP TIMER INTERRUPT THEN JUMPS TO DISP.
;DISPI   INTERRUPT ROUTINE.
;TRAD    TRANSLATE HEX BYTE TO 2 DISPLAY CHARACTERS.
;TRSP    TRANSLATE HEX DIGIT TO DISPLAY CHARACTER.
```

```
;APART FROM ITS USE IN "KDUIO", THE "DISP"
;SUBROUTINE CAN BE USED ON ITS OWN (TO ALLOW ENTRY OF
;SINGLE DIGITS WITH NO COPY TO THE DISPLAY), OR
;WITH THE TIME-OUT TO END THE WAIT-FOR-KEY LOOP
;IN "DISP". IN THIS CASE, THE PROGRAM FOLLOWING
;THE CALL ON "DISPA" WOULD CHECK THE ZERO FLAG.
;IF IT IS SET, THEN A TIME-OUT HAD OCCURED. OTHERWISE
;THE "A" REGISTER CONTAINS THE HEX DIGIT ENTERED
;(0-9, A-F).

;EXAMPLE.
;THE PROGRAM READS A CHANNEL OF AN ANALOG INPUT CARD
;AT 100 MS INTERVALS. A NEW CHANNEL CAN BE SELECTED
;WITH THE KDU. NOTE THAT "COUNT" IS RELOADED BEFORE
;EVERY CALL OF DISPA. THIS ALLOWS VARIOUS
;TIME INTERVALS TO BE SET UP AT DIFFERENT POINTS
;IN THE PROGRAM.


;LOOP:  MVI    A,10             ;10*10 MS
;       STA    COUNT            ;SET TIME-OUT COUNTER.
;       CALL   DISPA            ;(MIN TIME = 10 MS).
;       JNZ    NEW              ;KEY PRESSED. SET NEW CH.
;       CALL   READ             ;READ ANALOG CARD.
;       JMP    LOOP
;NEW:   STA    CHAN             ;SET UP NEW CHANNEL.
;       JMP    LOOP
```

```
;KDUIO.

KDUIO:  PUSH    B                   ;SAVE DIGITS, POSITION.
        CALL    XFER                ;MOVE TEXT TO DISPLAY.
        MVI     A, 1
        STA     NULLE               ;PRESET NULL ENTRY FLAG.
        MVI     B, 10               ;CLEAR DATA BUFFER.
        LXI     H, DATAB
KDIOG:  MVI     M, 0
        INX     H                   ;STEP POINTER.
        DCR     B                   ;COUNT DOWN
        JNZ     KDIOG               ;LOOP FOR MORE.

KDIOF:  POP     B                   ;RETRIEVE DIGITS, POSITION.
KDIOB:  CALL    DISP                ;DISPLAY MESSAGE, WAIT FOR KEY.
        CPI     ROLU                ;CHECK IF ^.
        JZ      KDIOH               ;RETURN IF SO.
        MOV     D, A                ;SAVE DIGIT.
        ANI     0F0H                ;TEST FOR CONTROL DIGIT.
        JNZ     KDIOB               ;IGNORE IF SO.

        STA     NULLE               ;CLEAR NULL ENTRY FLAG.
        MOV     A, D                ;RETRIEVE DIGIT.
        PUSH    PSW                 ;SAVE ON STACK.

        LXI     H, DISB             ;PUT DIGIT IN DISPLAY BUFFER.
        MVI     D, 0                ;ADD DIGIT ENTRY POSITION.
        MOV     E, C                ;TO BUFFER BASE.
        DAD     D
        MOV     D, M                ;SAVE OLD DIGIT.
        PUSH    D

        CALL    TRSP                ;TRANSLATE & STORE DIGIT.
        POP     D                   ;RETRIEVE OLD DIGIT.
        DCX     H                   ;TO OFFSET INX H IN TRSP.
        PUSH    B                   ;SAVE DIGITS, POSITION.
```

```
KDIOC:   DCX     H                   ;NEXT MS BUFFER POSITION.
         DCR     B                   ;COUNT DOWN "DIGITS".
         JZ      KDIOD               ;IF END OF SHIFT.
         MOV     E,M                 ;SAVE OLD DIGIT.
         MOV     M,D                 ;REPLACE WITH NEW DIGIT.
         MOV     D,E                 ;SET FOR NEXT SHIFT.
         JMP     KDIOC

KDIOD:   POP     B                   ;RETRIEVE "DIGITS", "POSITION".
         POP     PSW                 ;AND LATEST DIGIT.
         LXI     H,DATAB                     ;POINT TO DATA BUFFER.
         MVI     D,0                 ;ADD "POSITION".
         MOV     E,C
         DAD     D
         MOV     D,M                 ;SAVE OLD DIGIT.
         MOV     M,A                 ;REPLACE WITH NEW ONE.
         PUSH    B                   ;SAVE B,C.

KDIOE:   DCX     H                   ;POINT TO NEXT MS POSITION.
         DCR     B                   ;COUNT DOWN "DIGITS".
         JZ      KDIOF               ;END OF SHIFT.
         MOV     E,M                 ;SAVE OLD DIGIT.
         MOV     M,D                 ;REPLACE WITH NEW ONE.
         MOV     D,E                 ;SAVE OLD ONE FOR NEXT SHIFT.
         JMP     KDIOE               ;LOOP FOR MORE.

KDIOH:   LXI     H,DATAB                     ;POINT TO DATA BUFFER.
         MOV     A,C                 ;ADD "POSITION".
         SUB     B                   ;SUB "DIGITS"-1.
         INR     A
         MOV     E,A
         MVI     D,0                 ;MOVE TO D,E.
         DAD     D
         RET                         ;RETURN.
                                     ;H,L POINTS TO MS DATA DIGIT.
```

```
;XFER. MOVE TEXT TO DISPLAY.

XFER:     LXI     D,DISB              ;POINT TO BUFFER.
          MVI     B,10                ;SET TO MOVE 10 CHARS.
XFERA:    MOV     A,M                 ;GET TEXT CHAR.
          STAX    D                   ;MOVE TO DISPLAY.
          INX     H
          INX     D                   ;STEP POINTERS.
          DCR     B                   ;COUNT DOWN.
          JNZ     XFERA               ;MORE.
          RET

          ;DISPA.
          ;SET TICC FOR TIMER 3 INTERRUPTS.
          ;SET TIME-OUT, AND JUMP TO "DISP".

          ;PUT "JMP DISPI" AT ORIGIN 18H

DISPA:    XRA     A
          STTCM
          MVI     A,9
          STTCM                       ;SET TICC FOR INTS.
          MVI     A,8
          STIMR                       ;ALLOW TIMER 3 INT.
          MVI     A,0A0H              ;SET TIMER 3 FOR 10 MS.
          STTIM   3
          EI
          JMP     DISP                ;GO TO DISPLAY ROUTINE.

DISPI:    PUSH    PSW                 ;TIMER 3 INT. ROUTINE.
          MVI     A,0A0H
          STTIM   3                   ;RELOAD TIMER 3.
          LDA     COUNT
          DCR     A
          STA     COUNT               ;COUNT DOWN "VALUE".
          JZ      TMOUT               ;TIME-OUT.
          POP     PSW
          EI
          RET                         ;NORMAL RETURN.
TMOUT:    POP     PSW                 ;STRAIGHTEN STACK.
          POP     PSW                 ;DISCARD INT. RETURN ADDR.
          XRA     A                   ;CLEAR A-REG.
          RET                         ;TO CALL OF "DISP".
```

```
        ;DISP STEPS THE DISPLAY AND CHECKS FOR KEYBOARD INPUT.

DISP:   LHLD    DISBP           ;DISPLAY BUFFER POINTER
        MVI     A,0             ;TO CLOCK KDU
        STOUT                   ;OUTPUT TO KDU
        LDIN                    ;TO CHECK ROW ONE
        ANI     040H            ;SEE IF ROW ONE
        JNZ     DROW1           ;IF SO RELOAD POINTER
        LXI     H,DISB          ;WITH BUFFER START
DROW1:
        MOV     A,M             ;COLLECT DISPLAY DATA
        ORI     061H            ;SET CLOCK BIT
        STOUT                   ;SEND TO KDU
        INX     H               ;INDEX TO NEXT DISPLAY
        SHLD    DISBP           ;PUT IN STORE
        LXI     D,KEYB-DISB-1   ;INCREMENT TO KEY INPUT
        DAD     D               ;FOR INPUT COMPARE
        LDIN                    ;READ KEY INPUT
        ANI     00FH            ;ISOLATE BITS
        XRI     00FH            ;AND COMPLEMENT
        CMP     M               ;COMPARE WITH LAST RESULT
        MOV     M,A             ;AND UPDATE STORED RESULT

        JZ      DISP            ;JUMP IF NO CHANGE
        LXI     D,-KEYB                 ;TO SUBTRACT BASE
        DAD     D               ;LEAVING INDEX
        DAD     H               ;MULTIPLY BY FOUR
        DAD     H
        ANI     0FH             ;REMOVE TOP BITS
        JZ      DISP            ;JUMP IF NO INPUT
DROW2:
        INR     L               ;CYCLE TO COUNT WHICH BIT IS SET
        RAR
        JNC     DROW2
        MVI     H,0             ;COMPLETE INDEX
        LXI     D,KC-1          ;KEYBOARD TABLE
        DAD     D               ;ADD INDEX
        MOV     A,M             ;FINAL CODE
        RAL                     ;CHECK CONTROL BIT
        MOV     A,M             ;FINAL CHARACTER AGAIN
        RNC                     ;EXIT IF NO CARRY
        ANI     07FH            ;CLEAR TOP BIT
        JZ      DISP            ;NO INPUT - LOOP
        MOV     A,M             ;FINAL CHARACTER
        RET
```

```
;       DISPLAY CODES

;       DIGIT     HEX.  CODE    SEGMENTS
;
DIST:
DSC0    EQU     0FCH                ;ABCDEF
        DB      DSC0
DSC1    EQU     060H                ; BC
        DB      DSC1
DSC2    EQU     0DAH                ;AB DE G
        DB      DSC2
DSC3    EQU     0F2H                ;ABCD  G
        DB      DSC3
DSC4    EQU     066H                ; BC  FG
        DB      DSC4
DSC5    EQU     0B6H                ;A CD FG
        DB      DSC5
DSC6    EQU     0BEH                ;A CDEFG
        DB      DSC6
DSC7    EQU     0E0H                ;ABC
        DB      DSC7
DSC8    EQU     0FEH                ;ABCDEFG
        DB      DSC8
DSC9    EQU     0E6H                ;ABCD FG
        DB      DSC9
DSCA    EQU     0EEH                ;ABC EFG
        DB      DSCA
DSCB    EQU     03EH                ;  CDEFG
        DB      DSCB
DSCC    EQU     09CH                ;A  DEF
        DB      DSCC
DSCD    EQU     07AH                ; BCDE G
        DB      DSCD
DSCE    EQU     09EH                ;A  DEFG
        DB      DSCE
DSCF    EQU     08EH                ;A   EFG
        DB      DSCF
DSCH    EQU     06EH                ;  C EFG
DSCI    EQU     020H                ;   D
DSCO    EQU     03AH                ;  CDE G
DSCS    EQU     0B6H                ;A CD FG
DSCP    EQU     0CEH                ;AB  EFG
DSCR    EQU     00AH                ;    E G
DSCU    EQU     07CH                ; BCDEF
DSCL    EQU     01CH                ;   DEF
ROLU    EQU     010H
```

```
;       KEYBOARD CODES
                                 ;SELECT         RETURN
KC:     DB      08H              ;0              1
        DB      02H              ;0              2
        DB      CSTEP            ;0              4
        DB      07H              ;0              8
        DB      0DH              ;1              1
        DB      01H              ;1              2
        DB      CMODE            ;1              4
        DB      0BH              ;1              8
        DB      00H,             ;2              1
        DB      080H             ;2              2
        DB      0EH              ;2              4
        DB      CFUN             ;2              8
        DB      06H              ;3              1
        DB      080H             ;3              2
        DB      080H             ;3              4
        DB      080H             ;3              8
        DB      05H              ;4              1
        DB      080H             ;4              2
        DB      080H             ;4              4
        DB      CREST            ;4              8
        DB      04H              ;5              1
        DB      080H             ;5              2
        DB      0CH              ;5              4
        DB      CGO              ;5              8
        DB      09H              ;6              1
        DB      080H             ;6              2
        DB      ROLU             ;6              4
        DB      03H              ;6              8
        DB      0AH              ;7              1
        DB      080H•            ;7              2
        DB      0FH              ;7              4
        DB      ROLD             ;7              8
        DB      080H             ;8              1
        DB      080H             ;8              2
        DB      080H             ;8              4
        DB      080H             ;8              8
;
CSTEP   EQU     088H
CMODE   EQU     086H
CFUN    EQU     082H
CREST   EQU     040H
ROLU    EQU     010H
ROLD    EQU     020H
CGO     EQU     084H

;       END KEYBOARD CODES
```

```
        ;TRSP.  TRANSLATE & STORE ONE DISPLAY CHARACTER.

TRSP:   ANI     0FH             ;LS4 ONLY
        LXI     D,DIST          ;TRANSLATE TABLE.
        ADD     E
        MOV     E,A             ;SET INDEX.
        JNC     TRNC
        INR     D
TRNC:   LDAX    D               ;GET TRANSLATION.
        MOV     M,A             ;PUT INTO BUFFER.
        INX     H
        XRA     A               ;CLEAR A-REG.
        RET

; KDU BUFFER AREA
;
        ORG     1000H

DISB:   DS      16              ;DISPLAY BUFFER
KEYB:   DS      16              ;KEY BUFFER.
DISBP:  DS      2               ;BUFFER POINTER
DATAB:  DS      9               ;DATA BUFFER
COUNT:  DB      0               ;COUNTER
NULLE:  DB      0               ;NULL ENTRY FLAG.
;
     END
```

Note on KEYBOARD CODES Table in Page 8-36:

The entry 080H in this table indicates an unused position within the keyboard matrix. The Function keys are indicated by 'CSTEP', 'CMODE' etc., and they are equated to their actual values. These Function codes correspond to the actual Function keys of the PDM-KDU as follows:

CSTEP	=	STEP
CMODE	=	MODE
CFUN	=	FCT
CREST	=	REST
ROLU	=	Upward arrow
ROLD	=	Downward arrow
CGO	=	GO

The RUN - RST switch is not used in this example.

8.6 PDM-KBD : HEAVY-DUTY KEYBOARD

8.6.1 FUNCTIONAL DESCRIPTION

The PDM-KBD heavy-duty keyboard module provides 24 momentary-contact type keys, arranged in a 6 x 4 matrix. Each key has a diode in series with it for column isolation during scanning. All the columns can be activated together to determine if any keys have been pressed, before scanning to identify them individually.

The module can be driven directly from the parallel input and output ports available at the device connector of all 8080 CPU based DCE microcomputers. It can also be driven via the RWC interface modules with parallel interfaces.

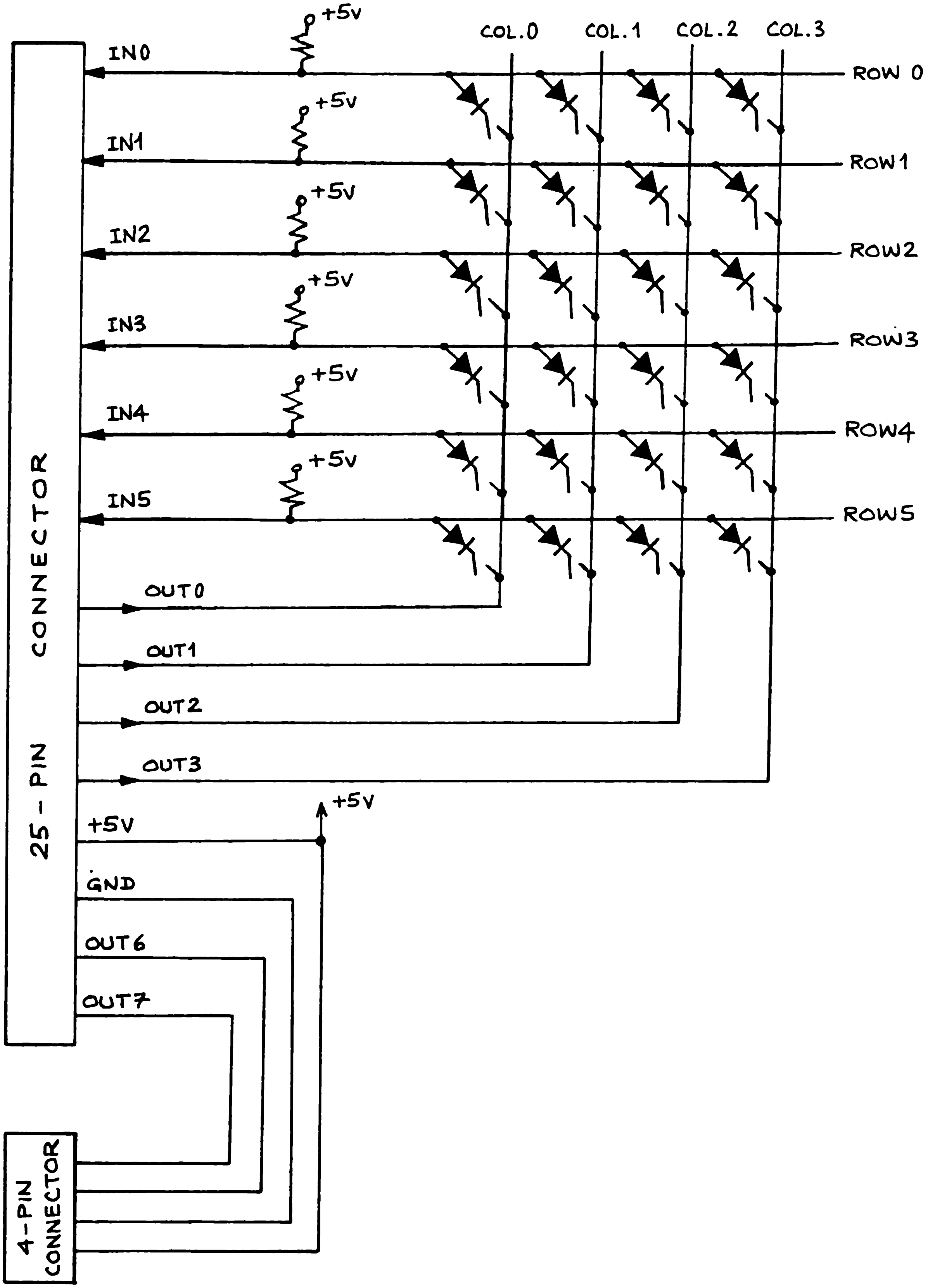
Two of the signal lines together with +5V and ground are available at a 4-pin male connector, which enables an external device such as the PDM-DSP display module to be directly connected to the PDM-KBD.

An optional kit enables the keyboard module to be directly mounted at the front or back of any DAI eurobox or eurorack.

8.6.2 FEATURES

- 24 keys arranged in a 6 x 4 matrix.
- Series diode with each key for column isolation during scanning.
- Parallel interface connection to DCE microcomputer, or suitable Real-World module.
- 4-pin connector for driving the PDM-DSP display module, or similar device.
- Removable transparent key tops, for user definition of key symbols.
- Keys assembled on a 100 x 160 mm metal mounting plate.
- Optional kit for direct mounting on any DAI eurobox or eurorack.

8.6.3 FUNCTIONAL BLOCK DIAGRAM



8. 6. 4 SYSTEM DESIGN PARAMETERS

8. 6. 4. 1 Hardware Configuration

Section 8. 6. 3 shows a functional block diagram of the Heavy-duty Keyboard module. The module contains 24 momentary-contact type keys arranged in a 6 x 4 matrix. Each key has a diode in series with it for column isolation during scanning. All the keyboard matrix columns can be activated together if sufficient current can be sunk by the column drivers. This can be used to check if any key has been pressed, before scanning the keyboard. If a key has been pressed, the keyboard can be scanned column by column to identify that key.

The 4 columns of the keyboard matrix must be driven via the signal lines OUT0 to OUT3. The closing of a key is monitored by driving its column with a logic 0. If upon reading its row a logic 0 is detected, the key at the junction of the selected row and column has been pressed. The 6 rows of the keyboard matrix are read via the signal lines IN0 to IN5. The loading on each of these input lines is one TTL load.

The 4 column driver signal lines and the 6 row scanning signal lines are brought to a 25-pin D-type female connector on the module. It also has a 4-pin male connector which enables an external device to be physically linked to it. These 4 lines are directly routed to the 25-pin connector, and two of them are used for +5V supply and ground.

The PDM-KBD module can be driven directly from the parallel TICC input and output ports available at the device connectors of 8080 CPU based DCE microcomputers.

In this case, two of the output lines together with +5V and ground will be available at the 4-pin connector. The large 9-digit display module PDM-DSP can be directly connected to this 4-pin connector, in which case the two output signal lines OUT6 and OUT7 must carry a clock signal and serial data to it. The keyboard module and any external device connected via the 4-pin connector can also be driven via any suitable Real-World interface module.

Figure 8. 6. 1 gives the dimensions of the metal supporting plate for the 24 keys. An optional packaging kit PSM-KBD/M enables the keyboard module to be directly mounted at the front or back of any DAI eurobox or eurorack, with four quarter-turn screws. It mounts the keyboard horizontally (4 rows of 6 keys) using a 203 mm wide panel. The keys have removable transparent keytops to enable user definition of key symbols.

8. 6. 4. 2 Module Connector Definitions

The PDM-KBD module includes a 25-pin D-type female connector. This may be connected directly to the device connector of any 8080 CPU based DCE microcomputer. Table 8. 6. 1 shows the connector pin definitions, together with the corresponding TICC signals when connected to a DCE microcomputer. This table also gives the pin definitions for the 4-pin male connector on the module. It is situated next to the 25-pin connector, and its pin adjacent to the 25-pin connector is designated pin 4.

Signal Name	Description	TICC Signal	Pin Number
IN0	Matrix Row 0	TICC Input Port Bit 0	1
IN1	1	Bit 1	2
IN2	2	Bit 2	3
IN3	3	Bit 3	4
IN4	4	Bit 4	5
IN5	5	Bit 5	6
OUT0	Matrix Column 0	TICC Output Port Bit 0	23
OUT1	1	Bit 1	24
OUT2	2	Bit 2	25
OUT3	3	Bit 3	13
OUT6	To 4-pin connector pin 1	Bit 6	10
OUT7	To 4-pin connector pin 3	Bit 7	9
+5V	+5V to PDM-KBD and to 4-pin connector pin 4		15
GND	To 4-pin connector pin 2		19

Note: The TICC Signals are applicable only when the PDM-KBD is directly connected to a suitable DCE microcomputer.

Table 8. 6. 1 : PDM-KBD Connector Pin Definitions

8.6.4.3 Operational Requirements

Power Requirements

The PDM-KBD requires a single +5V power supply. When used with 8080 CPU based DCE microcomputers, this can be obtained directly from its device connector.

When the columns on the key matrix are not driven to logic 0, the power supply requirements are negligible. When a column is driven to logic 0, for each key closed on that column, the power supply requirement is approximately 4.2 mA.

Column Driver Signal Requirements

The 4 columns in the key matrix must be driven via signals OUT0 to OUT3. If an output line is driven to logic 0, for each key closed on the column the buffer driving the output line must sink 5.6 mA. If an output line is driven to logic 1, the current to be supplied by the driving buffer is negligible.

Environmental Requirements

Operating Temperature	: 0°C to 55°C
Storage Temperature	: -25°C to +85°C
Relative Humidity	: upto 95 % non-condensing

8.6.5 ORDERING INFORMATION

PDM-KBD	: Standard version.
PSM-DCC/25	: The optional connecting cable to a DCE microcomputer must be ordered separately.

PSM-FCC/4 : The optional connecting cable for the 4-pin connector must be ordered separately. It is 40 cm long, and has two female connectors at the ends.

PSM-KBD/M : The optional packaging kit for·mounting the keyboard on any DAI eurorack or eurobox must be ordered separately.

8.7 PDM-DSP : LARGE DIGIT DISPLAY

8.7.1 FUNCTIONAL DESCRIPTION

The PDM-DSP display module provides nine large 7-segment LED displays with decimal points, and eight LED indicators. The LED digits are each 7,62 mm high and 4,78 mm wide. They can be used to display hexadecimal data, and a limited character set. The eight LED indicators are mounted in a line, and can be used to identify different modes of operation etc. The entire module is covered by a thick red filter mounted in front, for protection and enhanced display legibility.

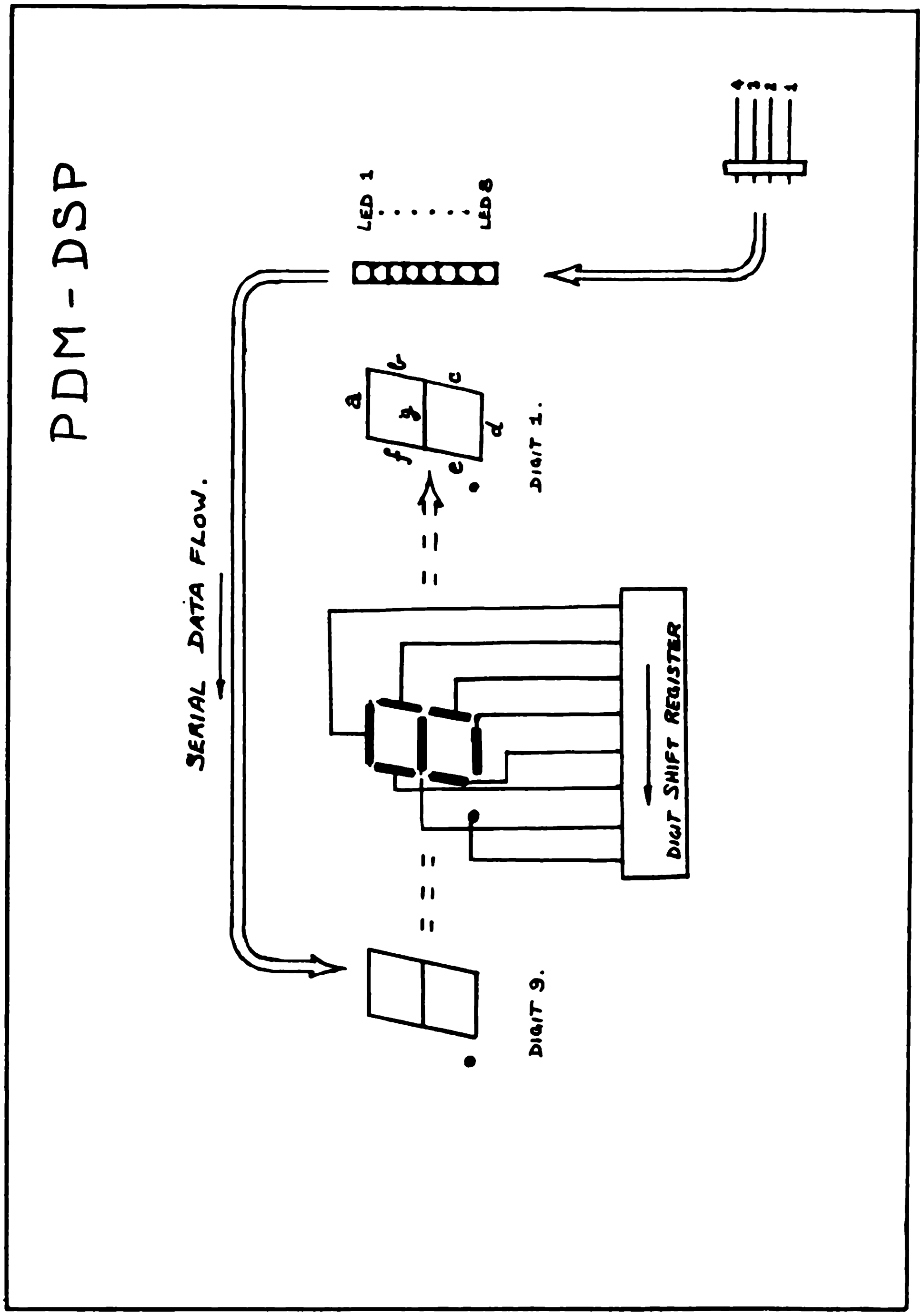
The module must be driven serially by data and clock signals, supplied via the 4-pin male connector. The remaining two pins are for a +5V supply and ground. The clock signal is used to transfer the serial data to an 80-bit shift-register on the module. The seven segments and the decimal point of each of the nine digits, and the eight indicators are driven in accordance with the data bits stored in this shift register. Once the shift-register is loaded with the necessary bit pattern, the corresponding display will remain stable until new data is clocked into the shift-register by driver software.

The display module can be directly connected to the PDM-KBD keyboard, or to any other suitable interface, via a separate 4-wire flat-cable connection. An optional kit enables the display module to be directly mounted at the front or back of any DAI eurobox or eurorack, with four quarter-turn screws. The width of the mounting panel is 137 mm, which is exactly one-third that of a 19-inch rack or box.

8.7.2 FEATURES

- 9 large 7-segment LED displays with decimal points.
- 8 LED indicators.
- 4-pin connector for serial interface and power.
- Static operation with no data refresh requirements.
- Directly connectable to the PDM-KBD keyboard module.
- Red cover filter for enhanced legibility and protection.
- Optional kit for direct mounting on any DAI eurobox or eurorack.

8.7.3 FUNCTIONAL BLOCK DIAGRAM



8. 7. 4 SYSTEM DESIGN PARAMETERS

8. 7. 4. 1 Hardware configuration

The functional block diagram in section 8. 7. 3 illustrates the hardware configuration.

Each individual segment, decimal-point, or indicator LED is driven by an output of the 80 bit, static, shift register memory. Thus, synchronous communications in bit-serial form can be acheived with only two latched TTL output lines from the DCE system. These outputs are driven by the system software to operate as data and data-clock lines to this shift register. Typically, two of the TICC output lines are dedicated for this purpose.

Since the displays are controlled in a bit-per-segment fashion, non-numeric characters can also be represented. This includes most alphabetical characters, and therefore system messages can usually be displayed. Since all the decimal digits, and the letters A to F can be represented; hexadecimal data presentation is thus an inherent application of the PDM-DSP module.

Data is shifted at every 0 to 1 transition of the data-clock input. Data is first transferred through the eight indicator LED's, and then through the digits 9 to 1 (see section 8. 7. 3). The individual segment sequence is given below:

1st	bit shifted in	display decimal point		or indicator	N° 3
2nd	bit shifted in	display segment	g	or indicator	N° 4
3rd	"	"	f	"	N° 6
4th	"	"	e	"	N° 5
5th	"	"	d	"	N° 8
6th	"	"	c	"	N° 1
7th	"	"	b	"	N° 2
8th	"	"	a	"	N° 7

Note: 1) The LED indicators are not loaded in sequence.
2) As each bit is shifted in, ALL 80 segments take the old status of their predecessors.

Table 1. : Bit/segment relationship

An all cases, a logic zero is needed to illuminate the selected segment.

Selection of required digit

Since data must pass through all previous stages in the shift register, before it arrives at its required destination, extra clock pulses must be given. These are generally those used to shift the remaining data into the module. However, for reference the following table of counts has been prepared.

Digit position required for bit-pattern	Number of clock pulses required after 8-bit data has been shifted in.
9	8
8	16
7	24
6	32
5	40
4	48
3	56
2	64
1	72
Indicator LED's	0

Table 2: Post-clock counts for display positions

8.7.4.2 Connector Details

Connector pin configuration

pin 1	clock, transfers data on low to high transition.
pin 2	ground
pin 3	serial data, 0 to illuminate the required segment.
pin 4	+5V supply.

A 4 wire flat cable with suitable connectors is available. This cable, type N° PSM-FCC/4, is suitable to connect the PDM-DSP module to the connector provided on the PDM-KBD module.

8.7.4.3 Operational Requirements

Signal characteristics

The signal levels conform to standard TTL logic levels. The inputs impose the following loading on their respective drivers:

Clock input:	Logic 0, sink 14.4mA max.
	Logic 1, source 360 µA max.
Data input:	Logic 0, sink 3.2 mA max.
	Logic 1, source 80 µA max.

Signal timing

Data bits can be shifted into the PDM-DSP module at upto 25 MHZ. Thus, maximum DCE system speed can be accomodated.

However, the data must be written at least 15 nS before the rising edge of the clock. Thus, the data bit should be written with the clock output set low, and then a second instruction should transfer the clock level from 0 to 1.

Power requirements

The PDM-DSP module typically consumes 60 mA from the single +5V supply.

Environmental Requirements:

Operating temperature	: 0°C to 55°C
Storage temperature	: -25° to 85°C
Relative humidity	: 95 % non-condensing.

8.7.5 ORDERING INFORMATION

PDM-DSP : Standard version, with red filter plate.

PSM-FCC/4 : The 4-wire flat-cable connection,must be ordered separately.

PSM-DSP/M : The optional packaging kit for mounting the display module on any DAI eurorack or eurobox must be ordered separately.