PHILIPS

# P855/P860

# Interface Manual

First draft


data systems

PHILIPS

# P855/P860

## Interface Manual

First draft


data systems

This handbook is written for users wishing to integrate the P855 or P860 into a data processing system of their own design or, who want to connect non-standard peripheral equipment to the P855 or P860 central processor.
Interface means as much as "common boundary" between cpu and any equipment attached to it.

P855 and P860 operate into standard TTL - logic components. Devices using different logic levels can be connected without difficulty, provided that a level adaption card interfaces the control unit to the I/O bus.
Both processors are described in one manual since they are of similar design, the main differences being memory capacity (P855: 4k - 16k 16-bit words; P860: 4k - 32k 16-bit words) and read/write cycle time (P855: 1.2 usec; P860: 0.84 usec).

Great care has been taken to ensure that the information contained in this handbook is accurate and complete. However, should any errors or omissions be discovered, or should any user wish to make a suggestion for improving this handbook, he is invited to write his comments on the sheet provided at the end of the book and send it to:

MANUAL WRITING SMALL COMPUTERS

at the address on the opposite page.

Other manuals for the P855/P860 computers are:

P855/P860 Reference Manual (publ. nr. 5122 991 1835x)
A preliminary publication, to be replaced by the following manuals:

P850/P855/P860 Reference Data (publ. nr. 5122 991 1690x)
A pocket-size reference guide for programmer's, operators and service engineers.

Basic Executive Monitor (publ. nr. 5122 991 1135x)
The programmer's guide for Basic Executive Monitor.

Disc Monitor (publ. nr. 5122 991 1136x)
The programmer's guide for the Disc Operating System.

Real Time Monitors (publ. nr. 5122 991 11580)
The programmer's guide for basic and disc Real Time Monitors.

Processor Manual (publ. nr. 5122 991 1155x)
Containing the programmer's and operator's information for the assemblers, Update and text editor, linkage editor and debugging package.

Instruction set (publ. nr. 5122 991 1160x)
A syntactical description of the complete instruction set.

Basic FORTRAN (publ. nr. 5122 991 1143x)
Reference manual for FORTRAN programmer's (BASIC)

Full FORTRAN (publ. nr. 5122 991 1140x)
Reference manual for FORTRAN programmers (FULL).

Interface manual (publ. nr. 5122 991 1147x)
See the rest of this manual.

Installation manual (publ. nr. 5122 991 1148x)
Information on cpu, peripherals and site necessary for installation of the equipment.

CPU service manual (publ. nr. 5122 991 1231x)
The manual for cpu service engineers.

Control units (slow) service manual (publ. nr. 5122 991 1230x)

The manual for control unit service engineers.

Control units (fast) service manual (publ. nr. 5122 991 1691x)

The manual for control unit service engineers.

Table of contents

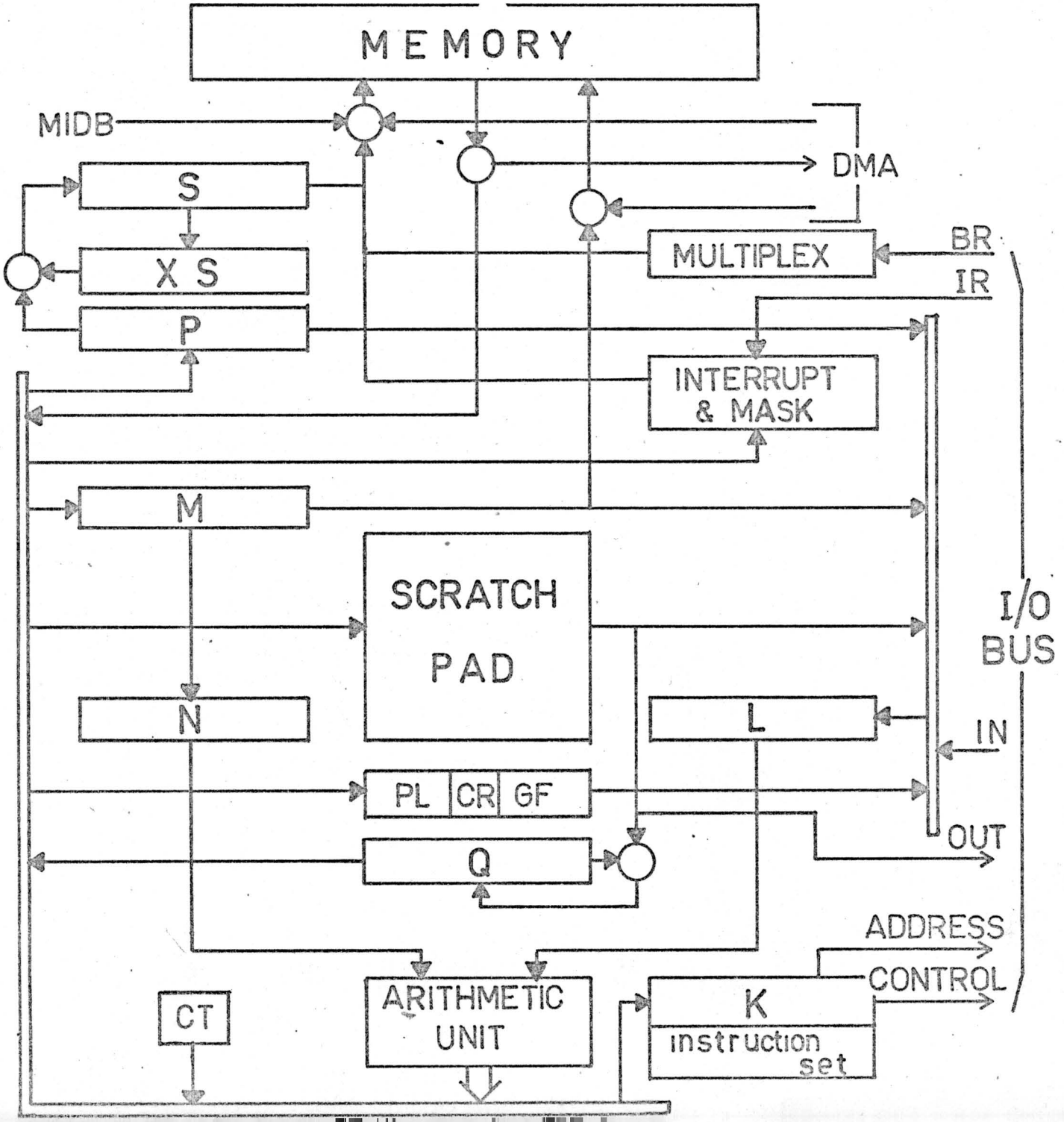
## List of illustrations



## List of tables

The ease with which a computer can be integrated into a user's system is a prime consideration of the system engineer, the programmer, and of course, the purchaser. Complicated device control unit busing, unwieldy device addressing, and restricted interfacing, which make competitive computers difficult to integrate into a system, have been avoided in the design of the Philips P855 and P860 computers. The clean, straightforward design of the central processor is easy to understand - an advantage to the system engineer and the service engineer. The simple I/O bus, used for programmed data transfer and for multiplex data transfer, is not subject to any critical timing or addressing restrictions - an advantage to the system engineer whether he uses the many available peripheral control units or designs control units to fit his own system requirements. Memory increment data break, with direct access to the memory address lines, is available for quantitive analysis applications. And a direct memory access channel is available for easy connection to high-speed peripheral devices.

The relationship of the I/O bus to the main hardware components of the central processor and the data paths through the processor are shown in figure 1. Data input from the I/O bus is shown taken through a data bus to the L-register and then, via the arithmetic unit and the main data distribution bus, to the scratch pad registers. Data output to the I/O bus is taken directly from the scratch pad registers. The device address and control bits of the program instruction are taken from the K-register to the address and control lines of the I/O bus. An interrupt request line (IR) from each device control unit on the I/O bus goes to the interrupt and mask logic and a break request line (BR) from each multiplex connected device control unit goes to the multiplex logic. The hardware components shown in figure 1 are:


MEMORY
MIDB
DMA
S
X S
P
MULTIPLEX
BR
IR
INTERRUPT
& MASK
M
SCRATCH
PAD
I/O
BUS
N
L
IN
PL
CR
GF
OUT
Q
ADDRESS
CONTROL
CT
ARITHMETIC
UNIT
K
instruction
set

MEMORY UNIT: A 4k to 16k word core memory is used in the P855 and a 4k to 32k core memory is used in the P860. The memory can be addressed by the memory address register (S-register), by the Memory Increment Data Break option, or by the Direct Memory Access Channel. Also, certain fixed locations are addressed by the interrupt and multiplex logic. The memory can accept data from either the CPU Memory Register (M-Register) or from the DMA channel. Data from the memory is taken via the main data distribution bus to a register selected by the instruction in the K-register and the instruction set.

SCRATCH PAD: A block of 16, 16-bit, registers addressed by the instruction word in the K-register. Register 0 is reserved for multiplex data transfers to the I/O bus and register 15 is reserved for use as a stack pointer for the interrupt system. The scratch pad registers can accept data from the memory of from any of the registers connected to the main data distribution bus. Data from the scratch pad can be loaded into the L register for processing in the arithmetic unit, or taken via the I/O bus to a peripheral device. In the latter case scratch pad addressing by programmed channel instructions is limited to registers 1 - 7 and register 0 is used for multiplex.

ARITHMETIC UNIT: Performs arithmetic and logic operations on data in the L and N registers. Data from the arithmetic unit can be transferred to any of the registers connected to the main distribution bus, or to the memory address register (S-register).

P-REGISTER: A 15-bit program instruction counter to hold the memory address of the next instruction to be executed.

The register is loaded from the main data distribution bus and is incremented each time an instruction is executed. Output from the register can be loaded into the L-register for processing in the arithmetic unit or into the S-register to address the memory.

S-REGISTER: A 16-bit register used to address the memory. The register can be loaded from the P-register, the scratch pad, or the XS-register, depending upon the type of instruction. The register contents can be loaded into the XS-register.

XS-REGISTER: A 16-bit register used for temporary storeage of the S-register contents.

M-REGISTER: A 16-bit memory buffer register to contain data transferred to and from the memory. Data from the register can be transferred to the N- or L-register for processing in the arithmetic unit.

N-REGISTER: One of two input registers to the arithmetic unit. It can be loaded from the M-register.

L-REGISTER: One of two input registers to the arithmetic unit. It can be loaded from the M-register, or from the P-register for address modification.

K-REGISTER: Instruction register loaded from the memory with the instruction to be executed. Output from the register is to the instruction set and, in the case of the I/O instructions, to the address and control lines (BAD and BOF) lines of the I/O bus.

Q-REGISTER: A 16-bit register used in double word shift operations.

PL/CR/GF-REGISTER: A 16-bit program status word register holding the priority level (PL) of the running program, the 2-bit condition register (CR), and 8 status bits.

CT-REGISTER: A 4-bit counter used in shift instructions and arithmetic operations. The counter is loaded from the K-register and the instruction set.

MULTIPLEX: A unit containing the logic to operate peripheral control units on the I/O bus in multiplex mode. In this mode of operation each character transfer is initiated by a break request (BR) from the peripheral control unit, which interrupts the CPU instruction cycle, and is completed without I/O programming in the CPU. The multiplex logic is also used to control memory increment data break (MIDB) operations.

INTERRUPT & MASK: A logic unit to initiate character transfers on the I/O bus under programmed channel (software) control. A mask register allows interrupt priority levels to be programmed and stack handling is used for sequential processing of interrupts.

MIDB: Memory increment data break provides direct access to the memory address lines and increments the value stored at the addressed memory location.

DMA: Allows direct access to the memory from an I/O channel (DMAC) using standard peripheral units on a second I/O bus.

Standard control units are available to connect many types of peripheral devices to the I/O bus or to the DMA channel. These include I/O typewriters, punched tape readers and punches, line printers, punched card readers and punches, magnetic disc units, digital plotters, and magnetic tape cassette units. For general purpose digital input and output of up to 16 bits per channel the DIOS system of control units is available with optional input buffering, signal detection, level adaptation, and galvanic isolation.

For analog or digital data acquisition there is the versatile MIOS system which can include output units for driving display equipment. For data communications a complete range of line control units is available. And for the engineer building his own control units for special applications there is a general purpose circuit card with power connections and a hole pattern to accept standard flat-pack logic and discrete component circuitry.

All peripheral units and other external devices for the input or output of data are connected to the I/O bus, or to the DMA-I/O bus, through a control unit. Each control unit has an address decoded from the address lines of the I/O bus and corresponding to the software instruction. Thus any control unit can be addressed by software I/O instructions. The function lines of the I/O bus are decoded in a similar manner to permit control of the peripheral devices from software instructions. The control unit will recognize its address, decode the function, and provide the necessary control signals or data synchronisation pulses to the peripheral device.

When a control unit, operating on programmed channel, has been switched into service by a software I/O instruction, it will give an interrupt request signal to the CPU when it is ready to receive or send data. The CPU will then service the interrupt request and interrupt the program according to the priority level assigned to the interrupt request line. Eight interrupt levels are included in the minimum configuration and these may be increased to a maximum of 48.
During the execution of each program instruction the interrupt levels can be sampled and compared with the priority level of the running program. If there is an interrupt request with a higher priority than the running program a new program initiated by the interrupt is run. Return to the original program is made possible by storing the memory address at which it must restart.

The interrupt system makes use of automatic stack handling to service both internal and external interrupts. This means that, for each interrupt, the contents of the P-register (this is the address at which the program must restart) and other vital information are stored in a part of the memory allocated for interrupt stacking. Access to the stack is made from scratch pad register 15 which always points to the next available location in the stack. This stack pointer is automatically decremented each time program information is loaded into the stack and is incremented when information is removed from the stack to run a program. The stack itself is able to generate an interrupt to warn the programmer that the stack is full and unable to handle further interrupts.
One of the eight interrupt levels in the basic configuration connects 16 interrupt request lines via a mask register - all other interrupt levels connect one interrupt request line each.

The 16 interrupt request lines connected to a common level can be inhibited or enabled by the 16-bit mask register. This register is loaded by a software instruction to allow the program to select which of the 16 interrupt requests is to be serviced. The 16 lines can also be sensed, by program instruction, to determine which lines have an active interrupt request.

There are, in addition to the interrupts generated by peripheral control units, as many as five interrupts generated by the CPU and one interrupt generated from a button on the control panel.

Of the five CPU interrupts two are used in all configurations and these are: the stack overflow interrupt already mentioned, and an operation interrupt produced when a Link to Monitor instruction is executed (this is explained in the Reference Manual) or when the CPU reads an instruction that is not valid for the particular hardware configuration. Interrupts are also generated by the power failure, real time clock, and memory protection hardware. Internal CPU interrupts are handled in exactly the same manner as external interrupts except that, to be effective, the power failure interrupt must be assigned the highest priority level, above all other internal and external interrupts.

FEATURES OF THE P855/P860 COMPUTERS

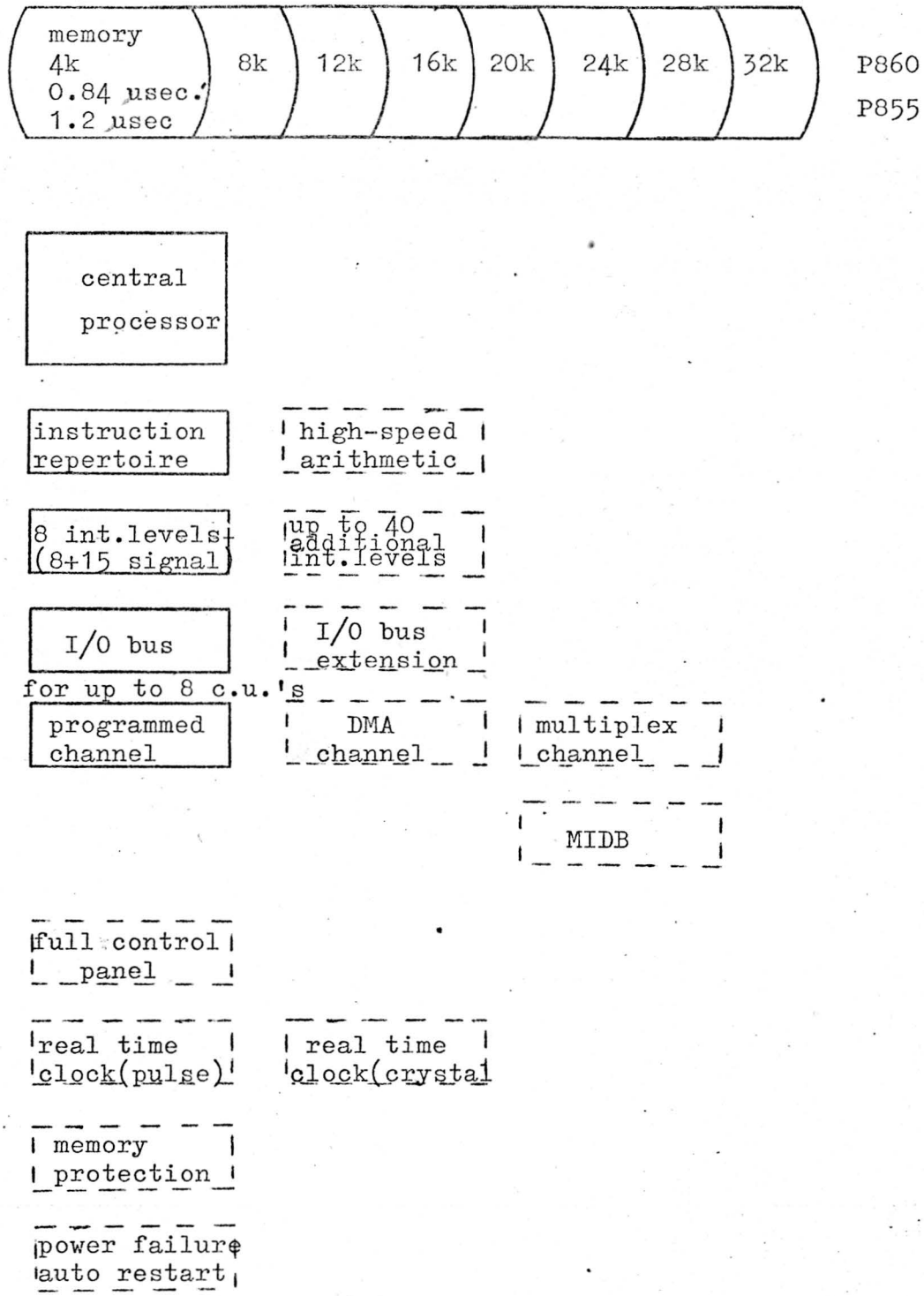



Figure 2. Hardware configuration chart showing the features of the P855/P860 computer.

MEMORY UNIT:

A 4k 16-bit core memory with a read/write cycle time of 1.2 microseconds is used in the P855 computer. This can be extended up to 32k in 4k modules. A core memory of 4k 16-bit words with a read/write cycle time of 0.84 microseconds is used in the P860 computer. This can also be extended to 32k in 4k increments.

CENTRAL PROCESSOR:

Contains the arithmetic unit, registers, etc., already described. Data paths are 16 bits wide and all data transfers are made in parallel format.

INSTRUCTION REPERTOIRE:

A powerful and versatile instruction set includes load, store, branch, logical, shift, and I/O instructions. Multiple load/store instructions allow for a very efficient usage of the registers.

Instructions for double length arithmetic, multiply, and divide may be included to extend the range of the instruction set.

Both single and double word instructions are available.

INTERRUPT LEVELS:

Eight interrupt levels including one level with 16 masked inputs are part of the minimum standard configuration. The interrupt system can be extended to give up to 40 additional levels.

I/O BUS:

The basic I/O is capable of driving 8 control units housed in the main cabinet. This can be extended to drive 21 control units in the main cabinet or to drive remote control unit installations.

PROGRAMMED CHANNEL:

Logic for programmed channel data transfer, i.e., single word or character transfer on the I/O bus under control of an I/O program, is included in the basic configuration.

Transfer rate P855: 14Kc-28Kc/s

P860: 20Kc-40Kc/s

MULTIPLEX: A unit containing the logic to operate peripheral control units on the I/O bus in multiplex mode. In this mode of operation blocks of data in memory are transferred via the I/O bus, each character transfer being initiated by a break request from the control unit. The break request is checked for priority (15 break request lines are used) over other break requests and, when accepted, breaks into the running program either between or during an instruction of more than five cycles. When the character transfer is completed the program is restarted.

Transfer rate P855: 93Kc-160Kc/s
P860: 150Kc-240Kc/s

DMA CHANNEL: An I/O channel to drive two control units on an I/O bus similar to the standard I/O bus. Data is transferred directly between the control units and the memory without using the standard CPU data paths. DMAC takes priority over the CPU for access to the memory and stops the CPU between memory read/write cycles.

Transfer rate P855: 415Kc-800Kc/s
P860: 600Kc-1.2Mc/s

MIDB: Memory increment data break is used in quantitive analysis applications. The same logic hardware is used as for multiplex described above. The user's equipment addresses the memory and gives a break signal to the multiplex logic. This break takes priority over all other multiplex break requests and causes the addressed memory location to be incremented.

POWER FAILURE/ AUTOMATIC RESTART: This feature provides the means to detect a power failure in time and automatically restart a program, without loss of information.

If the a.c. power fails completely or drops below the minimum level for error free operation, for longer than 20 milliseconds, an interrupt is generated 5 milliseconds before the d.c. operating voltages fall below normal. During this time all information relevant to the current instruction is stored in memory. Provided that the control panel key is in the "lock" position, the program is restarted automatically when the power is restored, i.e. all the hardware is reset, the stored information is retrieved and the interrupt instruction is processed again.

CONTROL PANEL:

The control panel contains switches and indicators which allow the operator or programmer to manually load or display the contents of the registers or memory locations. It is possible to interrupt the running program, to make any necessary alterations to it, and to restart it again at a selected program instruction without reloading the program.

A keylock is fitted which in one position permits all the switches on the panel to function, and in the other position only the INT button.

REAL TIME CLOCK:

This feature provides an interrupt each time a timing signal is received. The timing signal can be from either the 20 millisecond internal timer (which is tied to the $50^{c}/s$ mains supply) or from external timer. Also available is a crystal timer with an interval time of 1, 2, 5 or 10 milliseconds. The clock can be started or stopped by software and is programmable in the same way as a peripheral device.

Four input and three output channels are available on the P855/P860 central processors. The choice of which channel to use depends on the type of device to be connected and the application for which it is to be used. Progammed channel, where the transfer of each word or character must be programmed, is used only for low speed applications such as the operator's typewriter or punched tape or card devices. However, programmed channel is also used to address and control the devices on multiplex and direct memory access channels. Multiplex channel, where only blocks of data to be transferred are programmed, is used for both high and low speed devices. As many as 15 devices can be operated at the same time on the multiplex channel with word or character transfer taking place on a priority basis. In this way 15 memory data blocks for 15 devices can be programmed for simultaneous transfer in either direction. Direct memory access channel, operating with two device control units on its own I/O bus, is used with high speed peripheral devices. As with multiplex, only blocks of data are programmed.
Memory increment data break is the one channel used only for input. It has no data transfer lines but addresses the memory directly and the addressed location is incremented by the multiplex channel logic each time a break signal is given by the equipment connected to the memory increment channel. This channel is used for quantitive analysis applications where the sampling, scanning, and anolog to digital conversion normally required, is provided by the user.

## PROGRAMMED CHANNEL

Programmed channel provides what is effectively a direct connection from the CPU instruction register to the peripheral device control units via the control (BOF) and address (BAD) lines of the universal I/O bus.

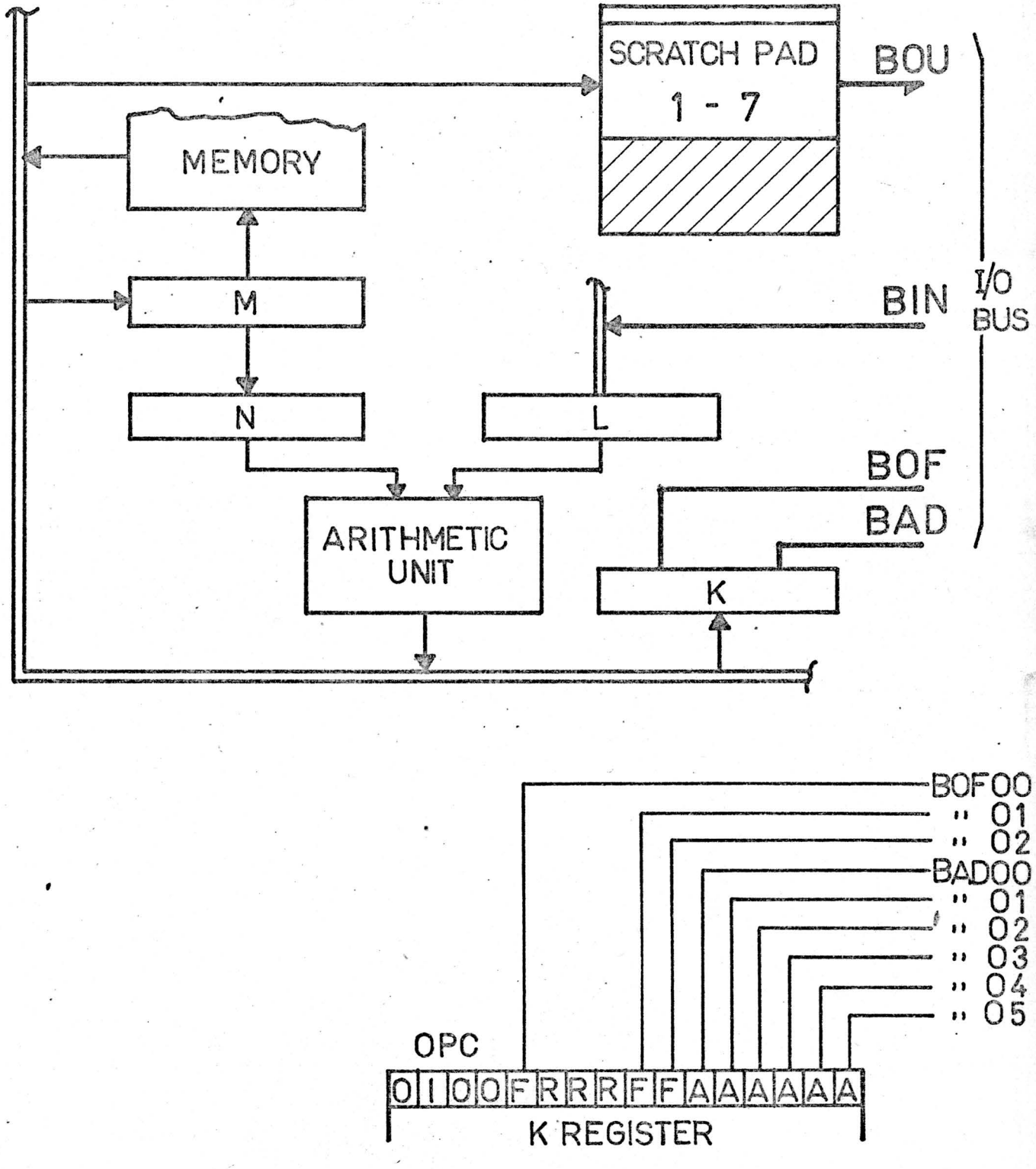



Figure 3. I/O bus connections to CPU

Figure 3 shows the I/O bus lines to the CPU. The BOU lines, 16-bit data output to the peripheral control units, are shown from the scratch pad and all data sent from the CPU on the I/O bus must come from one of the scratch pad registers. The BIN lines are taken through a data distribution bus, the L register and arithmetic unit, to the scratch pad and all data sent from the peripheral control units is loaded into one of the scratch pad registers. The peripheral device control unit function or control lines (BOF) and address lines (BAD) are taken from the K-register. An I/O instruction in the K-register always includes operation code (OPC) 0100 to signify that it is an I/O instruction. Also included in the instruction, and placed on the BOF lines, are three function bits (F in figure 3) which are used by both the control units and the CPU instruction set. K-register bit 4, for example, determines whether a control unit and the CPU are to operate in input mode or in output mode. Bits 5, 6, and 7 are used only by the CPU to determine which of the scratch pad registers is to be used for the data transfer and, since register 0 is not used for programmed channel, the addressing is limited to registers 1 through 7. Six device address bits are included in the K-register instruction word and these are placed on the I/O bus address lines to address the required control unit. Address 000000 cannot be used as a device address as it is used by the CPU to indicate that the instruction is internal to the CPU (to load the interrupt mask, for example) and does not address a peripheral device control unit.

Five I/O instructiond are used to program I/O transfers on programmed channel. Control Input/Output (CIO) stops or starts an I/O operation, depending on the state of function bit 4. Input to Register (INR) transfers a data word or character from the addressed control unit to the scratch pad register specified in the instruction. The number of data bits transferred depends on the type of peripheral device involved. Output from Register (OTR) transfers a data word or character from the scratch pad register specified in the instruction to the addressed control unit. Again, the number of bits transferred depends on the type of peripheral device involved. Send Status (SST) transfers the status word or character from the addressed control unit to the scratch pad register specified in the instruction.

The status word contains information as to the status of the peripheral device control unit.
Test Status (TST) is used to transfer a status word from the addressed control unit to the scratch pad register specified in the instruction but, unlike the SST instruction, it can be accepted by a control unit that has not been placed in the "ready state" by a CIO Start instruction. In this instruction only one bit of the status word is significant and the purpose of the instruction is to test whether or not the control unit is ready for a data transfer. Full details of these I/O instructions are given in chapter 3.

Programmed channel is used, in addition to data transfer of single words or characters, to start or stop a data transfer on the multiplex or direct memory access channels. In these applications the INR and OTR instructions are not used as it is only necessary to call the desired peripheral control unit to request the status and to place the control unit and the control logic in the "ready state" with a CIO instruction. The control unit, wired for multiplex or direct memory access operation, will then operate from the block transfer logic.

Input/Output instructions used in programmed channel provide, in addition to the address and function codes to the control units, coding to the CPU instruction logic to define the scratch pad register to be used in a data exchange and to select the CPU cycles required to complete the instruction. Full details of the operation of the CPU during the I/O, and other, instruction cycles can be found in the P855/P860 CPU Service Manual.
Programming for data transfer on programmed channel must include, a routine to call up the required control unit and place it in the ready state with a CIO Start instruction, a data transfer routine, and a routine to shut down the control unit when it is no longer required. The data transfer routine is initiated by an interrupt request from the control unit and will include subroutines for checking or assigning interrupt and program priority, stacking control words to allow the interrupted program to be restarted, checking the status of the control unit to ensure that the data transfer is completed (this also is initiated by an interrupt after an SST instruction),

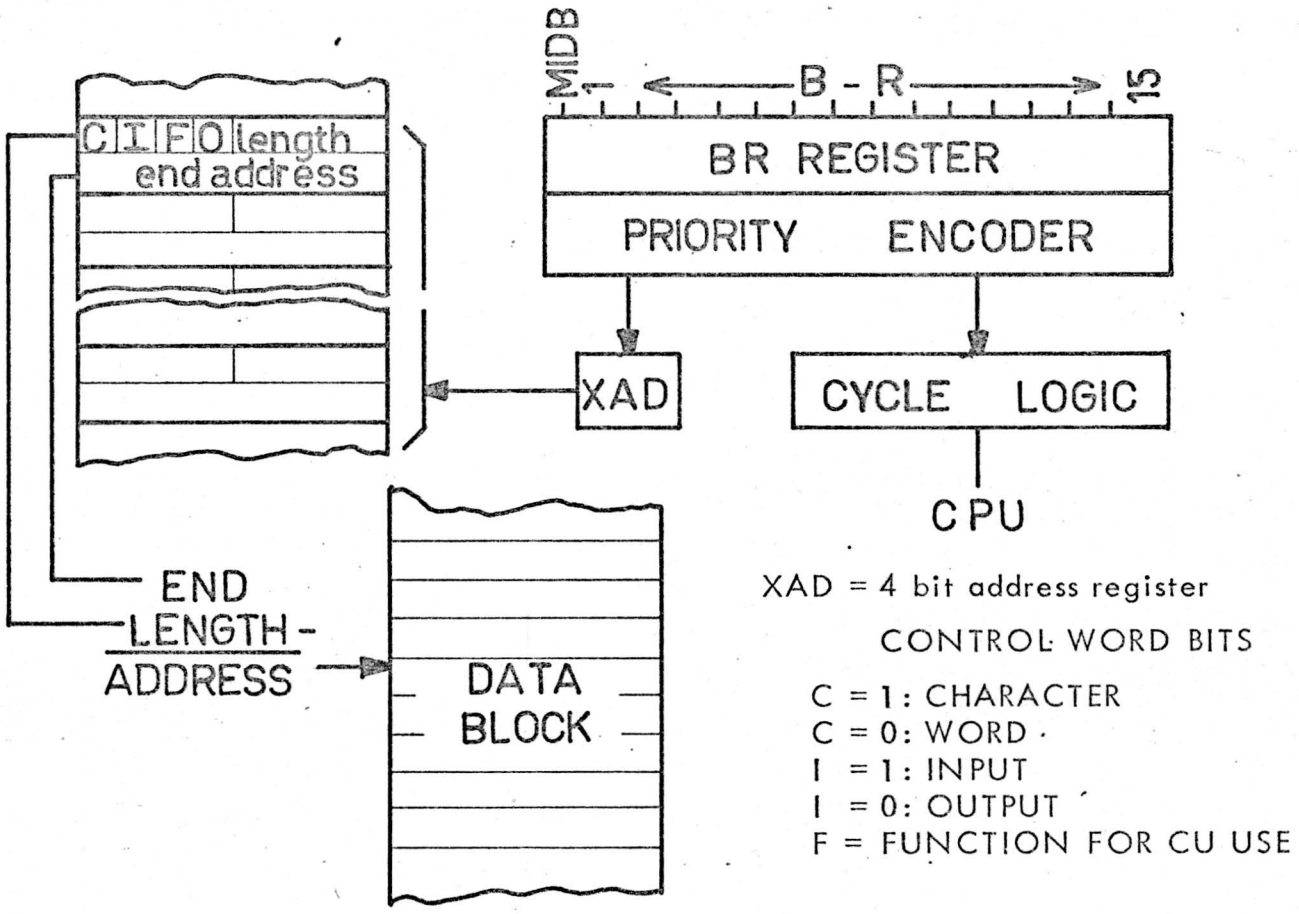



Figure 4. Multiplex System

and loading the data word into the scratch pad (in the case of an output transfer) or taking the data word from the scratch pad (in case of an input transfer). The Basic Executive Monitor contains a complete I/O routine and its use is described in the manual "Basic Monitor Details are given in chapter 4 of the I/O bus connections and timing for device control units used on programmed channel.

## MULTIPLEX CHANNEL

Multiplex channel operates in much the same way as programmed channel except that the data transfer routine is produced by hardware. The programmed channel and program interrupt system are used to address and control the peripheral control units on the multiplex channel but data transfer is initiated by a break request from the peripheral control unit. A break request signal is given by a control unit only when it is in the execute state, i.e., when it is ready to send or receive data, and only if the control unit is wired for multiplex operation. Wiring for multiplex operation is a simple matter of changing jumper connections in the control unit.
There is provision in the multiplex channel for 16 break request lines (see figure 4) arranged in priority order. The highest priority line is used only for memory increment data break and is not able to take a break request from a peripheral control unit. This leaves 15 break request inputs for normal multiplex operation. The break request lines are sampled by the CPU at least once during each intruction cycle and not more than three memory cycles can occur between samplings. When there are active break requests these are loaded into the break request register and the active break request with the highest priority is translated into a binary number in the priority and address encoder. At the end of a CPU instruction this number is loaded into the 4-bit multiplex address register and, with a low-order and two high-order bits added, becomes a 7-bit number to address even numbered memory locations from 66 through 94.

When the multiplex address register is loaded the multiplex data transfer routine takes control of the CPU to transfer one word or character and the running program is stopped. Five memory cycles are used for a multiplex transfer. In the first memory cycle the first of two multiplex control words (shown in figure 4) is read from memory at the address in the multiplex register. At this time the S-register contents (the next instruction address of the running program) are transferred from the S-register to the XS-register to allow the S-register to be used to address the data block. The data block length contained in the first control word in 2's complement form is incremented by one in the arithmetic unit and stored in the scratch pad. In the second multiplex cycle the second control word is read from memory - this is the ending address of the data block - and is added to the block length from the scratch pad to give the address of the data character in the data block. Another incrementing function may be included in this cycle as the required address depends upon whether a character or word is to be transferred - two halves of the same word are used for character transfer. The three remaining memory cycles perform the actual data transfer and end by restoring the running program address to the S-register, sampling the break requests and, if another data transfer is not requested by any other multiplex peripheral, the running program is resumed.

Maximum transfer rate requires that break requests are continuously available from peripherals. This would, of course, require that only high-speed devices such as magnetic tape or discs be connected. Also, under maximum transfer rate conditions, all other CPU programming would be locked out by the multiplex.

Programming for multiplex data transfer must include the programmed channel routine for calling up the required control unit with CIO and status instructions and for shutting down after a data block has been transferred. Routines must also be included to assign memory areas as data blocks and to load the control words with the applicable block information.

For output transfer the data blocks must be loaded and for input transfer the data must be assigned for processing. The Executive Monitor for multiplex contains complete routines for multiplex operation.

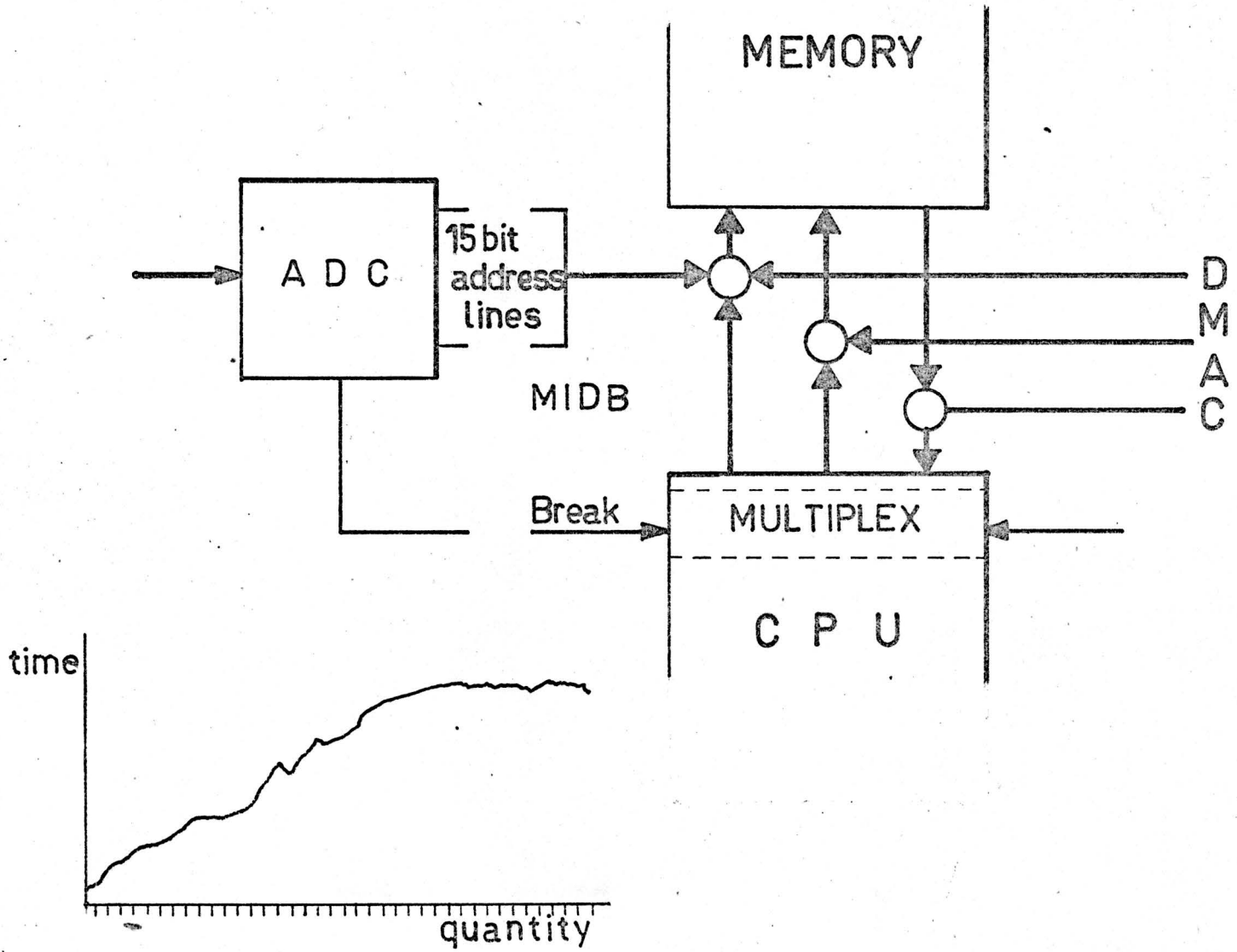



Figure 5 Memory Increment Data Break

Details are given in chapter 4 of the I/O bus and break request connections for device control units used on multiplex channel.

MEMORY INCREMENT DATA BREAK

Memory increment data break is used in quantitive analysis applications where it is convenient to increment the contents of memory locations when external sensors are scanned. A typical application is shown in figure 5 where analog values from sensing and scanning equipment are fed into an analog to digital converter (ADC) and the result is applied to the computer memory address lines. At fixed intervals a break signal is given by the ADC unit and on each break the memory location addressed by the ADC is incremented by one. The graph in figure 5 shows a quantity represented by memory locations plotted against the time that each value is maintained. Sensing, scanning, and analog to digital equipment used with memory increment data break is not included in the P855/P860 computer configuration. However, the Philips MIOS system is available for use with the P855/P860 in quantitive analysis applications.

Input to the computer for memory increment data break is taken to the memory address lines and, since there is no data transfer involved, no connection is made to the memory input or output lines. The multiplex channel logic is used for memory increment data break but only two of the five multiplex cycles are used; effectively the first and last multiplex cycles. The break request line is connected to the highest priority break signal input of the multiplex break request register and if the I/O bus is connected into the ADC unit, the programmed channel can be used for control purposes. Details are given in chapter 4 of the interface connections used for memory increment data break.

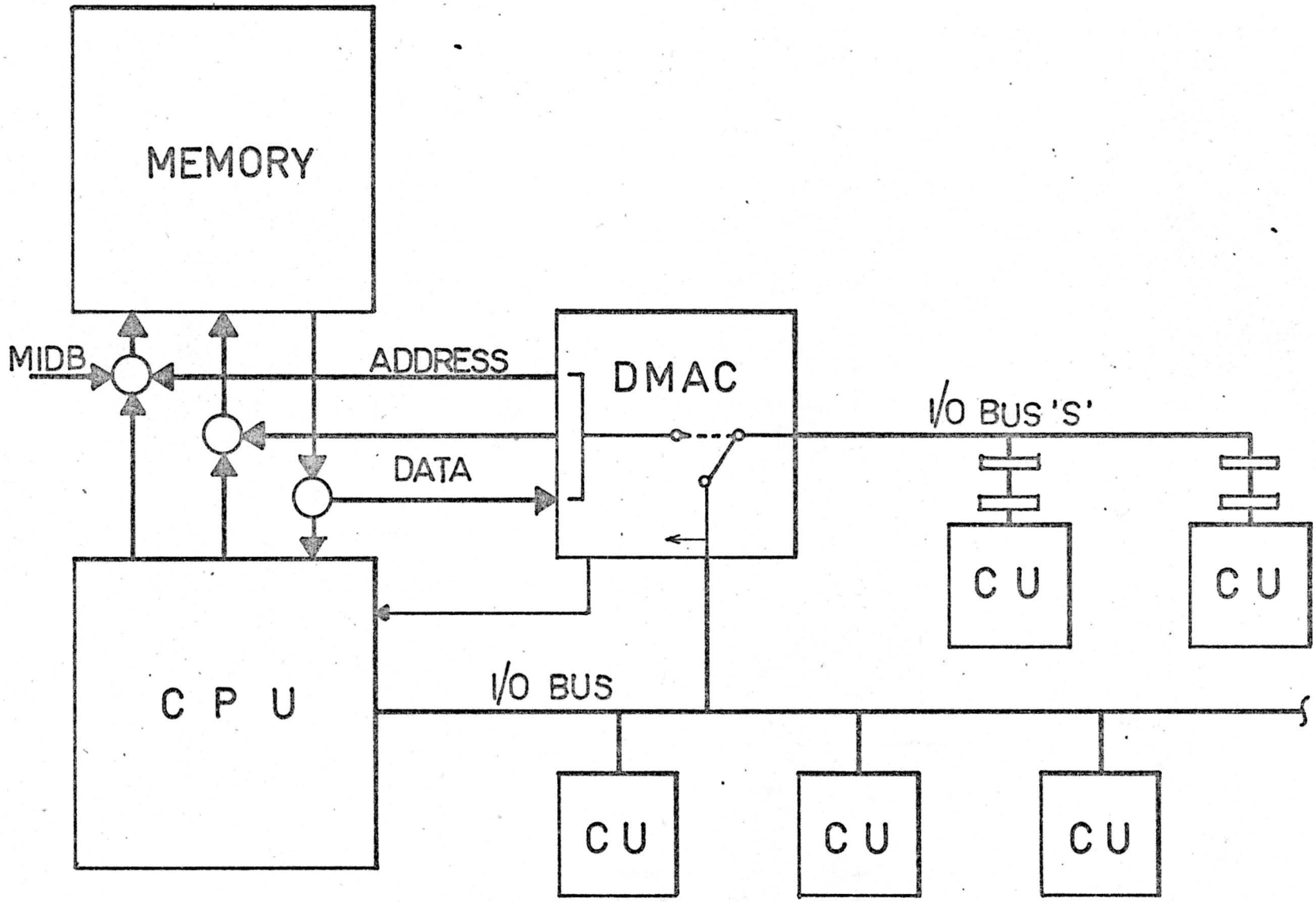



Figure 6. Direct Memory Access Channel

## DIRECT MEMORY ACCESS CHANNEL

This channel provides a direct link between the computer memory and two peripheral device control units only one of which may be operated at a time. The CPU memory addressing and data logic are not used for data transfer via direct memory access as the channel logic includes the necessary address and data registers and also has registers to allow data block transfer.

Figure 6 shows, in simplified form, how the direct memory access channel is connected into the system. The data and address lines of the unit are gated into the CPU memory data and address lines and there is provision in the CPU to stop the CPU instruction sequence at the end of any memory cycle and switch the memory access to the DMAC. Control of the DMAC unit and the peripherals connected to it is via the universal I/O bus. Control of the DMAC unit being a matter of activating the unit and assigning memory locations (starting address and block length) for blocks of data to be exchanged.

Once the DMAC unit has been activated the I/O bus is switched to the control units on the I/O Bus 'S' for CIO and status instructions. Control units on the DMAC are shown in figure 6 connected via bus extender line drivers and receivers. The control units normally used on DMAC operate high speed peripherals such as magnetic tape or disc drives and are housed in the peripheral device cabinets remote from the CPU. The length of cable between the CPU and peripheral device cabinets requires bus extenders.

The DMAC unit recognizes five I/O instructions from the CPU program. These are: CIO activation and deactivation (start and stop), OTR write block length, OTR write address, and INR read length. The DMAC unit is first activated by a CIO activation instruction and this is followed by OTR write length and write address instructions to define the area of memory to be used in the data block transfer. The data exchange is then started by a CIO instruction to the peripheral device control unit.

In the input mode the control unit gives a break request to the DMAC unit when it has a data character or word from the peripheral device ready for transfer to memory. The DMAC unit then loads this data into its data buffer and gives a request for memory access to the CPU.

At the end of the memory cycle the CPU instruction sequence stops and the data in the buffer is written into memory. The CPU sequence is then restarted and runs while the DMAC updates its memory block length and address and the control unit accepts another data word or character from the peripheral device. Another break request from the control unit restarts the DMAC cycle which is then repeated for each data word or character transferred. When the memory block length reaches zero an end-of-range signal (EOR) from the DMAC unit to the control unit stops the data transfer and the control unit then gives a programmed channel interrupt request. A CPU sub-program then checks the status of the peripheral device and gives a deactivation instruction to the DMAC.
In the output mode as soon as the OTR write address instruction has been given to the DMAC, it responds with a request to the CPU for memory access. At the end of the memory cycle the CPU instruction sequence stops and the data word or character required by the DMAC is taken from memory and loaded into a buffer. The CPU sequence is restarted and the DMAC updates its memory block length and address. After receiving the CIO start instruction the control unit gives a break request to the DMAC unit and the data word or character previously taken from memory, and now stored in the buffer, is transferred to the control unit. The DMAC unit again stops the CPU sequence, takes another word or character from memory, and updates the block length and address while the control unit to peripheral device data transfer takes place. This cycle is repeated until the block length becomes zero and the end-of-range signal is sent to the control unit. The control unit then gives a programmed channel interrupt request to pick up the sub-program required for status analysis and DMAC deactivation.

If the peripheral device end-of-record condition does not correspond to the DMAC end-of-range condition a throughput error will be given in the status analysis and the sub-program can include an INR read length instruction to find the remaining length of the block.
Programming for DMAC data transfer must include the programmed channel subroutines for calling up the device control unit and DMAC unit, as status analysis, and channel deactivation, as described above Routines must also be included to assign and load memory areas.

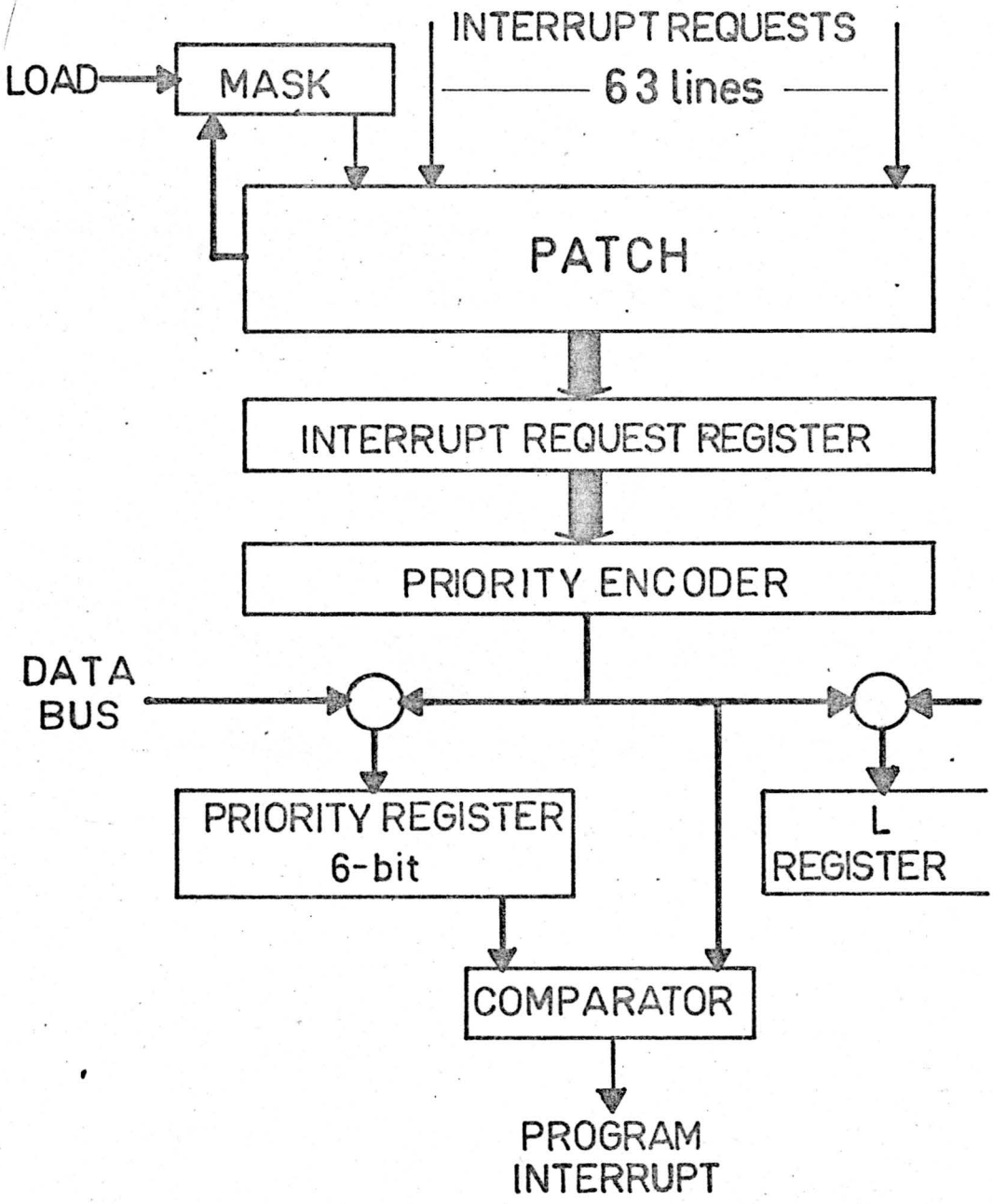



Figure 7. Interrupt request logic

The Disc Monitors contain complete routines for operation of the direct memory access channel.

Details of the I/O bus and I/O bus 'S' connections for device control units used on direct memory access channel are given in chapter 4.

## INTERRUPT AND STACKING SYSTEM

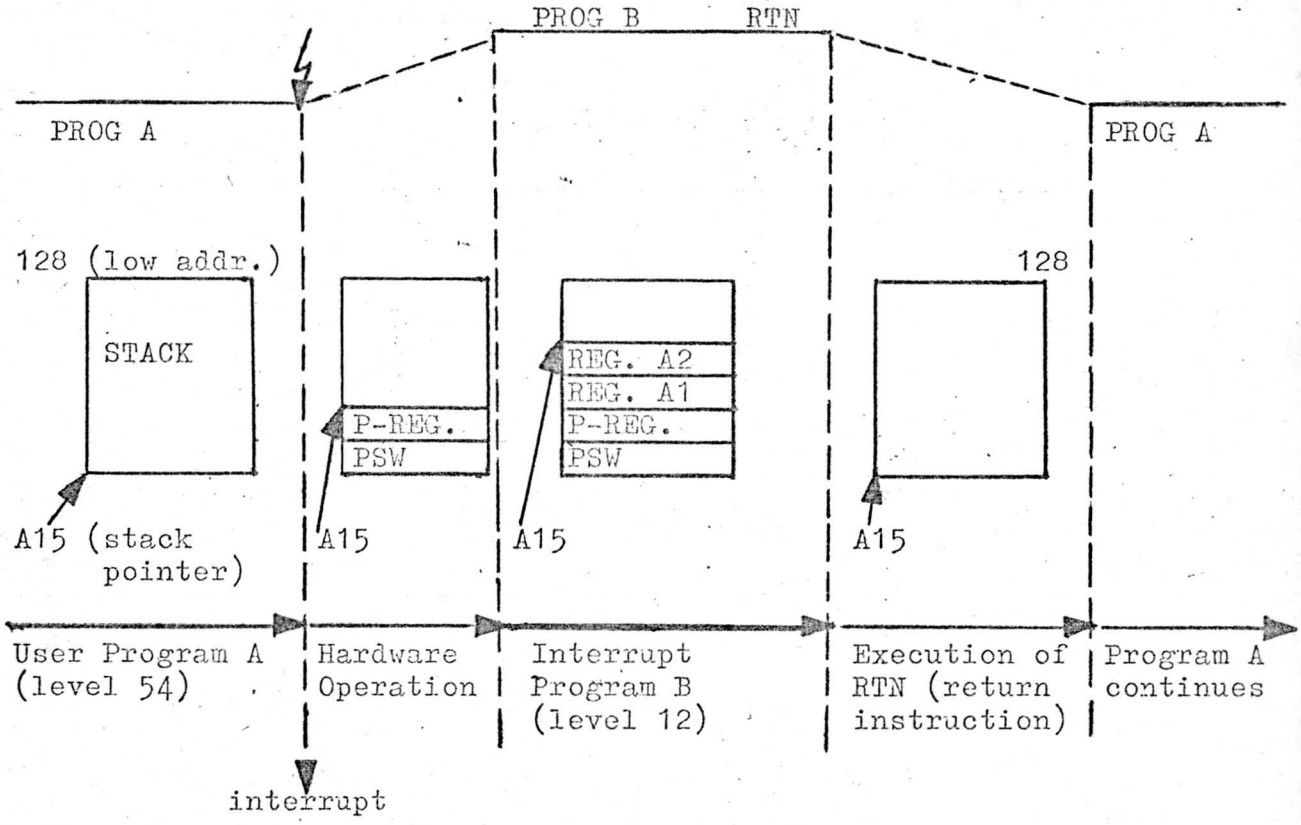
The interrupt system is used for all programmed channel peripheral operations and for handling internally generated interrupts. The system will handle up to 63 interrupt request lines including 16 masked lines to a single priority level. Interrupts are handled according to their priority, which is established by pre-wiring; the highest priority interrupt request is accepted and compared with the priority level of the running program. If the priority level of the interrupt is higher than that of the running program the program is interrupted if an enable instruction has been given and the P-register contents (the address of the next instruction in that program) and the program status word (containing the priority level and other information) are stored in a memory stack. A new program is then started by the interrupt and this program runs until stopped by a higher priority interrupt or until it is completed.

Figure 7 shows the interrupt system in simplified form. The 63 interrupt request lines can be patched to any of the 48 latches of the interrupt request register or to any of the 16 masked inputs. The masked lines are all patched to any one of the 48 latches. The mask is loaded by program instruction and only the selected (unmasked) lines are active. The interrupt requests are sampled during each program instruction and any active lines are gated into the interrupt request register at that time.

In the priority encoder the highest priority interrupt is selected and converted to a 16-bit binary number which is compared with the priority register contents. The comparator gives a program interrupt only if the priority level of the interrupt is higher than the priority level of the running program.

An eleven cycle hardware routine is started by the program interrupt signal from the comparator after the current program instruction is completed. During this routine the P-register contents and the program status word are stacked and the stack pointer (scratch pad register 15) is decremented to point to the next location in the stack. The 6-bit number from the priority encoder, which is the priority level of the new program started by the interrupt, is loaded into the priority register. This number is also loaded into the L-register, with a "1" added in the second most significant position, to address a location in memory which in turn contains the starting address of the new program.
The new program started by the interrupt will normally contain routines to save the contents of registers for the old program and will also include an instruction to enable the interrupt system to accept new interrupt requests. These routines are included in the executive monitors provided with the P855/P860 systems.

The illustration below shows, in simplified form, the operation of the interrupt and stacking system:



Return to the interrupted program is initiated by a return instruction (RTN) which picks up the hardware routine to take the program status word and program address from the stack. The priority level of the program, contained in the status word, is loaded into the priority register and the program address is loaded into the P-register.
The program then resumes and continues until completed or until interrupted by a new interrupt signal.
Instructions for programming the interrupt and stack system are given in Chapter 3 and details of the interrupt request line configuration are given in Chapter 4.

Masking is a convenient method of selecting the priority of interrupt requests by software. Sixteen interrupt request lines are controlled by the mask register; a "1" in the mask register inhibits the associated interrupt request line.
The mask register is loaded from one of the scratch pad registers by a program instruction so that the acceptable interrupts can be changed at any time. Normally, the mask is reloaded at the start of each interrupt software routine to designate the next acceptable interrupt requests. A second program instruction (RIL) reads the state of the 16 interrupt request lines into a scratch pad register for analysis before selecting subroutines.

## DIOS - DIGITAL INPUT/OUTPUT SYSTEM

The DIOS system provides the interface and control logic required to transfer digital information between user equipment and the P855/P860 computers. The DIOS system connects directly to the universal I/O bus and operates on programmed channel under software control. There are three basic DIOS units:

DIC - Digital Input Controller - has four 16-bit input channels.

DOC - Digital Output Controller - has four 16-bit buffered output channels.

DIOC - Digital Input/Output Controller - has two 16-bit input channels and two 16-bit buffered output channels.

Starting with these basic units DIOS can be extended to include buffering of incoming data, change-of-state detection - to clock new data into the input buffer and signal the computer that new data is available, level adaptation - to match the logic levels with those of the user equipment, and galvanic isolation - to completely isolate, electrically, the user equipment from the system.

Figure 8 shows the operation of DIOC unit for input and output of data. DIC and DOC units operate in the same way as the input and output parts of DIOC and are not shown separately. Although the unit is connected to the I/O bus and operated by programmed channel it does not operate in the same way as peripheral device control units in that CIO and Status instructions are not used. Only three instructions are used in DIOS programming: INR Word Address, INR Data Word, and OTR Data Word. Where data transfer is initiated by the user device, a call signal is sent to the DIOS unit to request each 16-bit data transfer. The call signal is stored in the call buffer and an interrupt request is sent to the CPU. The computer must respond to the interrupt with an INR Word Address instruction addressed to the unit. This instruction is decoded in the address decode and function select logic to cancel the interrupt signal and gate the contents of the call buffer onto the I/O BIN lines where they are picked up by the CPU and loaded into a register specified in the instruction.

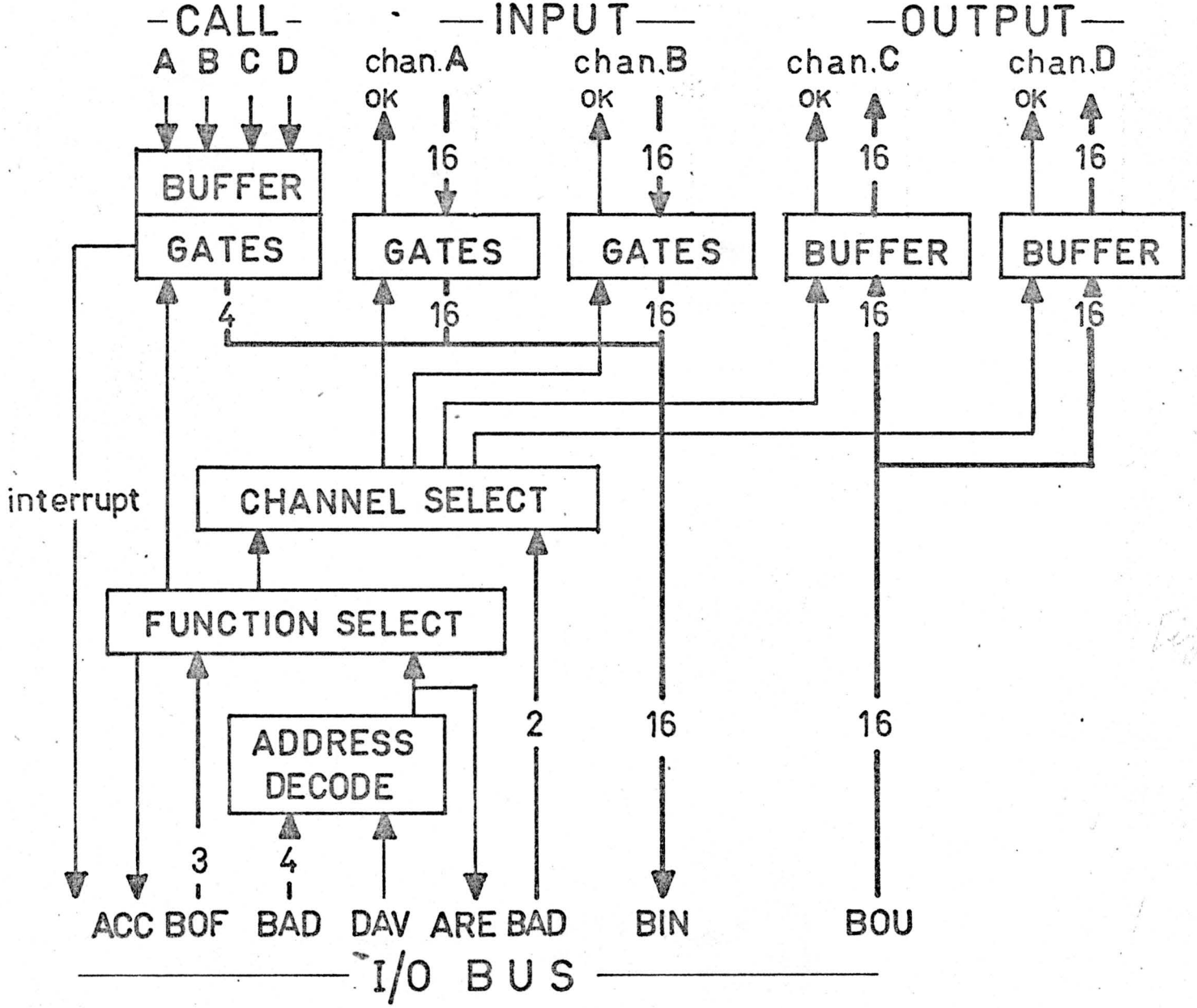



Figure 8. DIOS

The call information is examined in the computer to determine the channel requesting transfer, whether an input or output transfer is required, and to program the data handling within the CPU. A second instruction is then given to the DIOS unit - INR Data Word, for an input, or OTR Data Word, for an output. This instruction is decoded in the address decode, function select, and channel select logic to either gate a 16-bit input channel onto the BIN lines or strobe the 16-bit word from the BOU lines into an output buffer.

Data may also be transferred under software control without a call signal from the device and an interrupt from the DIOS unit. In this mode of operation INR Data Word instructions are used to sample the input lines at intervals controlled by the software program and an OK signal is sent to the device each time the lines are sampled. Output data is loaded into the DIOS buffers by OTR Data Word instructions and an OK signal sent to the device. Data remains in the buffer until overwritten by new data from the computer.

The Input Receiver and Buffer unit shown in figure 9 may be added to the basic DIOS input channels. This unit allows the input levels to be matched to the DIOS levels by means of jumpers on a printed circuit card. The 16-bit data input is then fed into a buffer connected to the data input lines of a basic DIOS unit. Data can be clocked into the buffer by a call signal from the device, a change -of-state of the input lines, or an OK signal from the DIOS channel. Or, the buffer can be placed in an open condition to accept data without a clock signal. The call signal is connected to the DIOS channel from the user device through a receiver circuit, identical to the data receivers, which allows signal level matching. The OK signal is connected to the device through a line driver to give the same provision for line length as the input receivers.

Galvanic isolation of the DIOS system from the user equipment is available and can be used either as a direct connection to the basic DIOS units or to the input buffer units described above. Complete electrical isolation is achieved by using optically coupled solid state isolators.

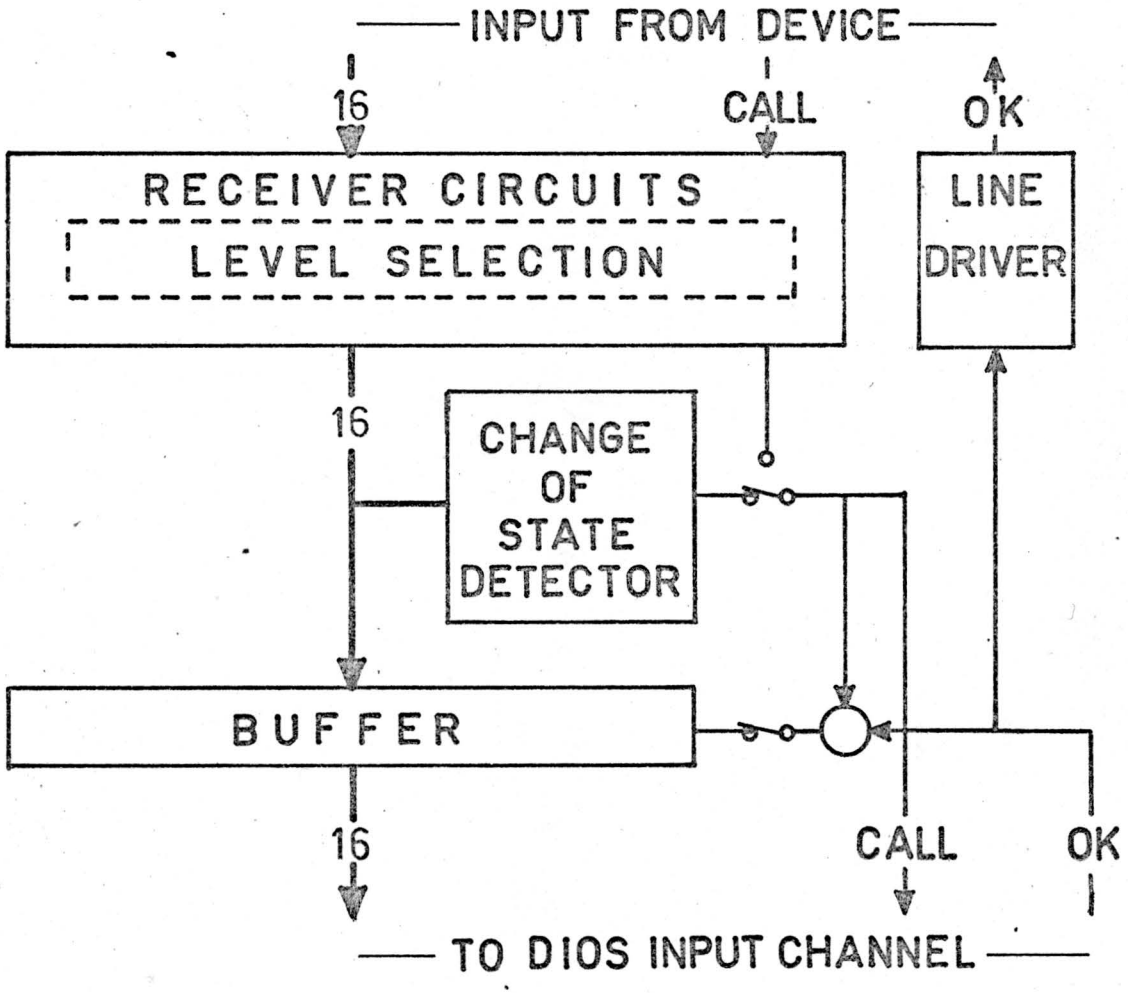



Figure 9. Input receiver and buffer

The input to the isolator drives a light emitting gallium arsenide diode. Light from the diode is detected by a photo sensitive transistor and amplified to match the logic levels of the circuit. Power for the input and output sides of the isolators is supplied separately by the device and the DIOS which simplifies the logic level matching.
Programming rules for DIOS are given in chapter 3 and full details of the hardware and connections are given in chapter 4.

## MIOS - MODULAR INPUT/OUTPUT SYSTEM

MIOS is a versatile input/output system which operates through a P855/P860 control unit for:

- Signal acquisition and conditioning (conditioning includes filtering and rationalisation of input levels).
- Sequential interrogation of inputs.
- Analog to digital conversion of input signals.
- Provision of output for display, control, or data acquisition.

Four types of MIOS systems are available for different applications.
MIOS-4S is for process control and similar applications where large numbers of analog inputs of various signal levels are involved.
MIOS-4D is for relatively simple applications involving small amounts of digital input and analog and digital output.
MIOS-4A is an extension of MIOS-4D and is able to handle both analog and digital input and output.
MIOS-4C is a unit with one analog input and one analog output, used in conjunction with MIOS-4A or as an independent channel.
MIOS is a self contained system built up of printed circuit modules housed in standard frame assemblies. The frames are designed for mounting in 19" racks.

Data transfer between input/output devices and the central processor are controlled by device control units each of which may have one or several devices attached to it, depending on the type of device. Each control unit is attached to the central processor by an interrupt line, address lines and other signal lines which are used by the computer to determine whether a data transfer can be performed. Data are transfered between the devices and the central processor via Programmed Channel - where each word or character transfer must be programmed, Multiplex Channel - where only blocks of data to be transferred are programmed, and the DMA Channel - also for block transfer, but at a very high speed.

## PERIPHERAL DEVICES

The standard software of the P855/P860 handles the following devices:

- I/O typewriter
- Punched tape equipment
- Punched card reader
- Magnetic tape equipment (also cassette tape)
- Disc equipment
- Line printing equipment
- Plotting equipment
- Display equipment

## BASIC INSTRUCTION SEQUENCE

The various commands are used as follows:

- TST (Test STatus): can be used before starting a data transfer (i.e. before CIO) to check whether the device control unit is in the ready state and, in the case of magnetic tape cassette, which side of the cassette is upwards;
- CIO (Control I/O): used to start or stop a device and to perform tape-handling operations, read and write on magnetic tape cassettes;
- INR (INput to Register): used to transfer a character or word from a device control unit to a specified register of the central processor;
- OTR (OuTput from Register): used to transfer a character or word from a specified register to the buffer of a device control unit;
- SST (Send STatus): used to get a status word from the device control unit.

It is advisable to use a different register for the sending of a status word than the register specified in the I/O command preceding SST. There are circumstances where an INR or OTR would be refused e.g. if an attempt was made to transfer an erroneous character or word and the ensuing SST command also refused because there was no device fault; in such a case, if the same register were specified in both commands the erroneous character or word would be erased by the status word.

A branch instruction should be used after each I/O command to check whether it has been accepted. The following conditions are indicated by the condition register:

CR = 0: command accepted (control unit ready);
1: command refused (control unit busy or command not compatible with state of control unit);
3: device address unknown.

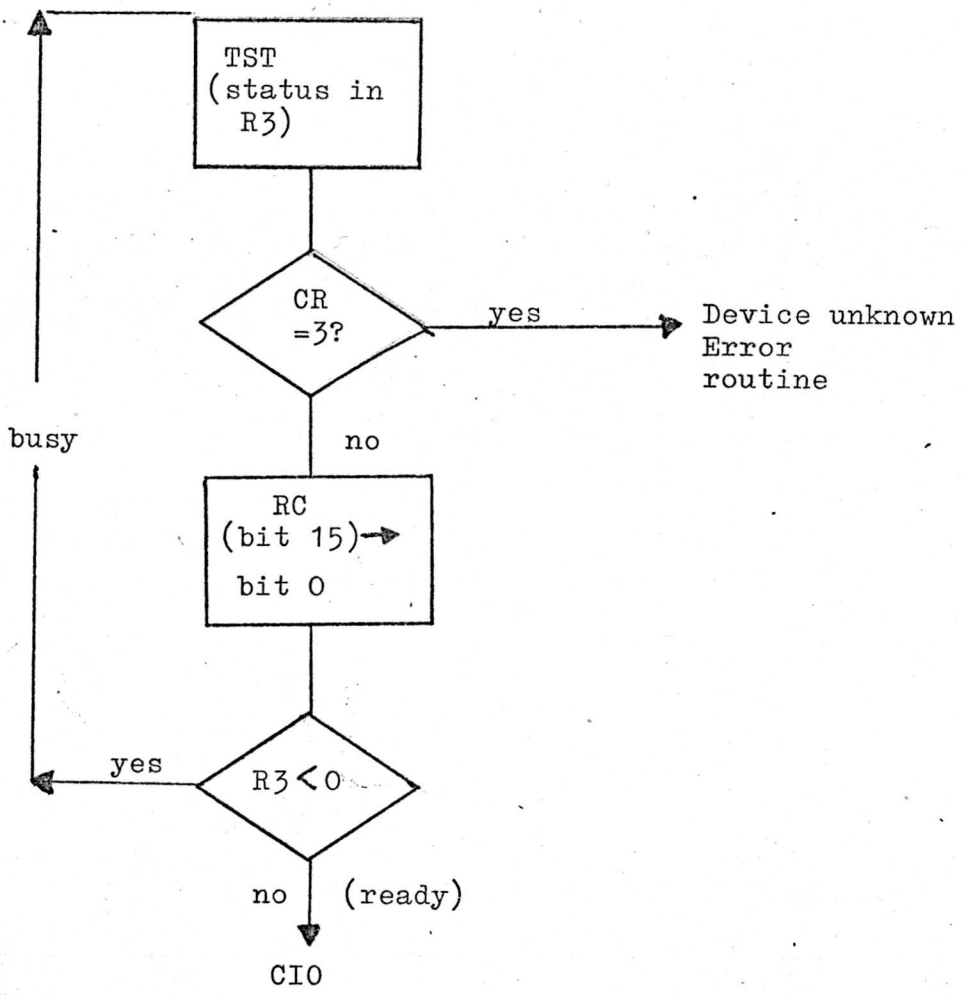
USE OF INTERRUPT PROGRAMS

Input/Output routines can be written either as a part of the main program, in which case processing speed is limited by the speed of the device (wait mode) or in seperate I/O routines which will, on request, interrupt the main program. In the latter case the main program, having activated the device by CIO start, can continue with other processing while mechanical actions are being performed by the device, and only when a character or word is ready for transfer will there be an interrupt from the device control unit.

INITIATION OF A DATA EXCHANGE

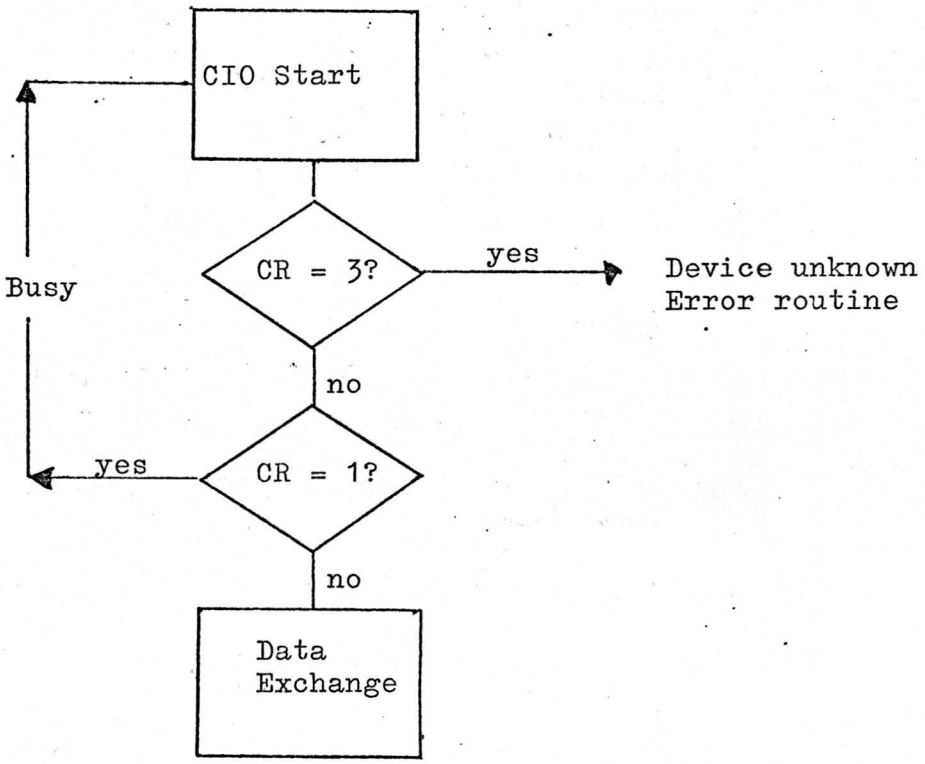
The Test Status instruction (TST) may be used before starting an I/O-operation, to test whether the device control unit is in the inactive state. This instruction must not, in any circumstances, be given after a Control I/O (CIO) instruction in the same routine. The TST instruction causes a status word to be transferred to a specified register; if bit 15 of this status word is 0 the control unit is in ready state; if 1, it is busy.

Example of use:



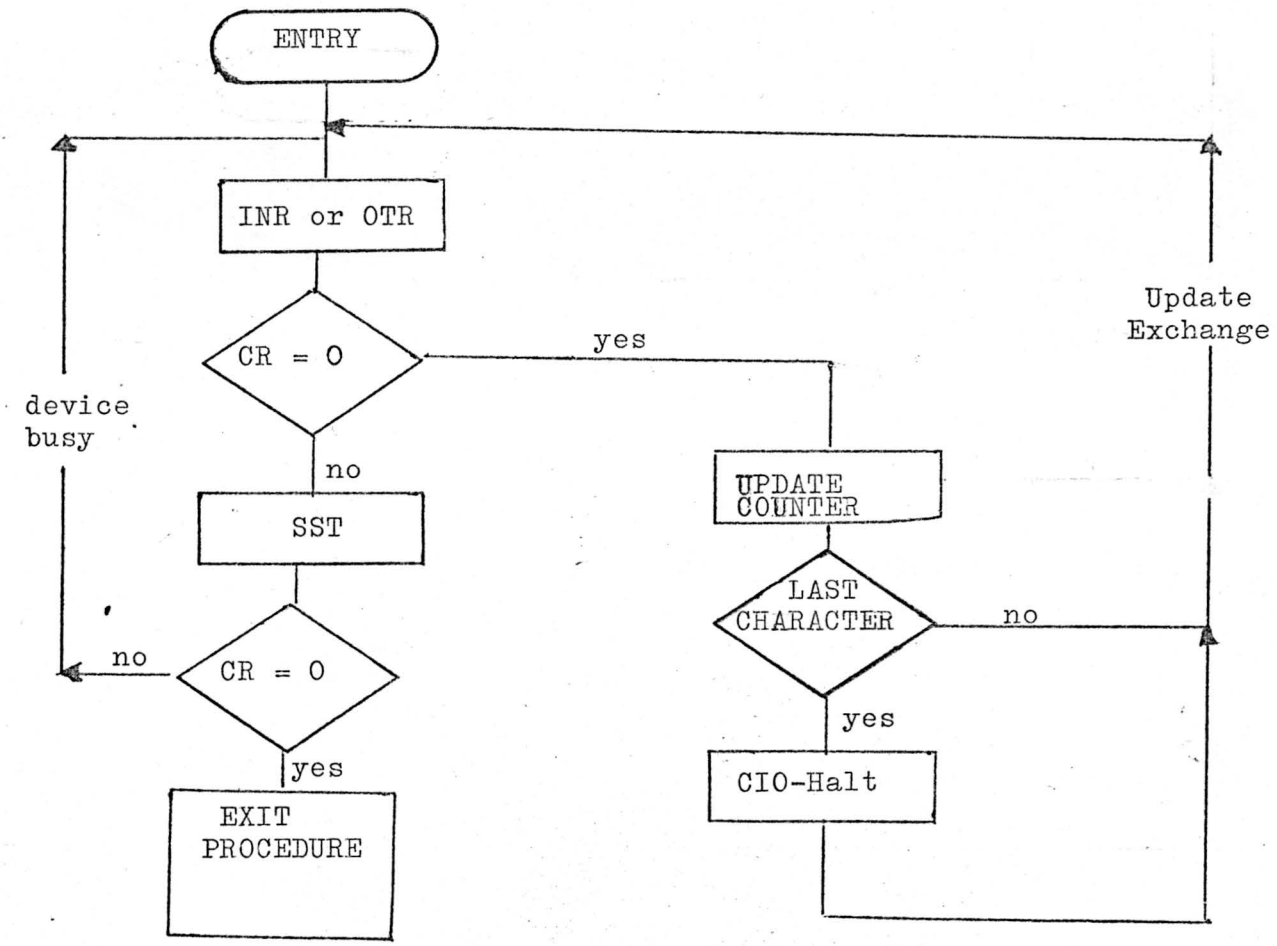
It is also possible to make a direct attempt to start, without using TST. In this case the readiness of the device control unit is tested by using a conditional branch after the CIO-instruction: the following states of the condition register are relevant:

- CR = 0: control unit ready: command accepted.
- CR = 1: control unit busy: command not accepted.
- CR = 3: device unknown.



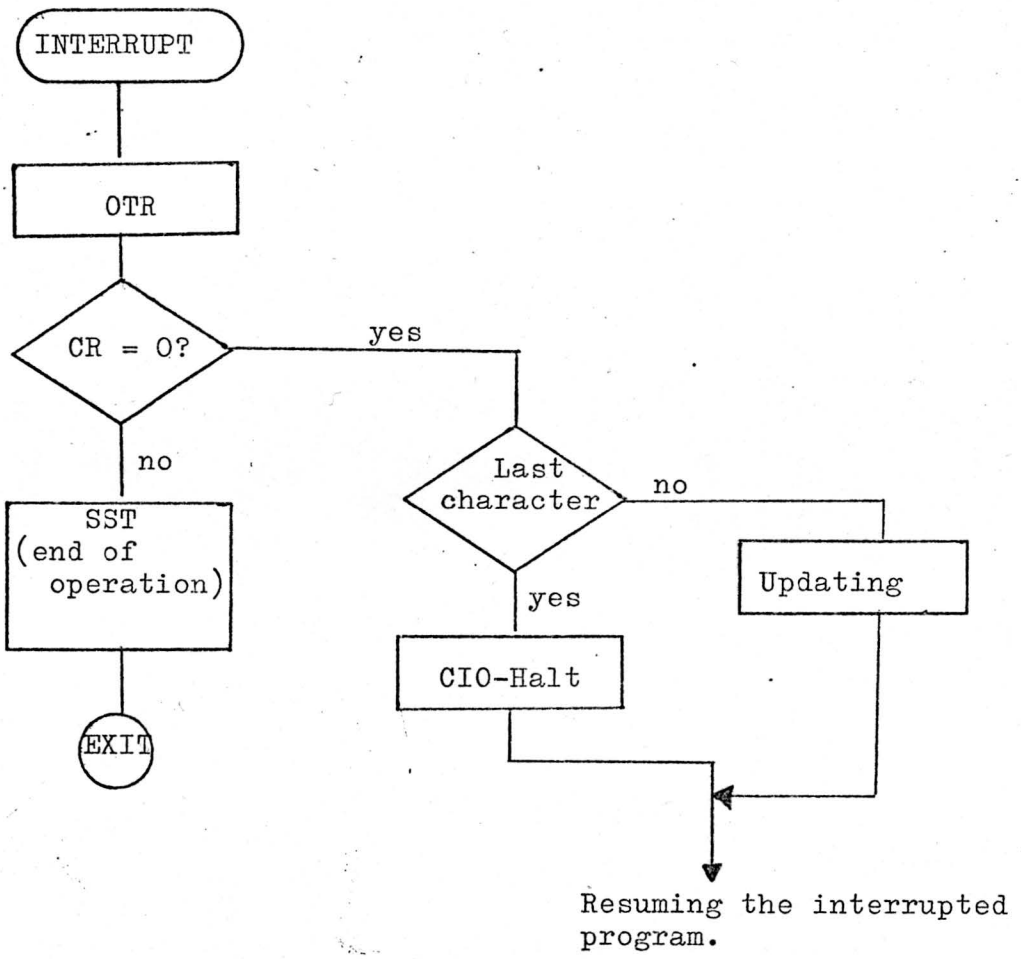
DATA TRANSFER IN WAIT MODE

Once it has been ascertained that the device is ready, the data transfer can be started. A test must be performed after each character or word transfer, to see whether the exchange is complete. If the device is still in ready state, the number of characters or words transferred is counted and, depending on the result of the count, either the transfer is continued or a CIO-instruction in halt mode is issued. If the device is not in ready state, either the exchange is complete (after CIO-Halt) or the device is still transferring a character or word. In either case ans SST-instruction is given followed by an exit routine or a loop through the exchange procedure depending whether SST is accepted or not.

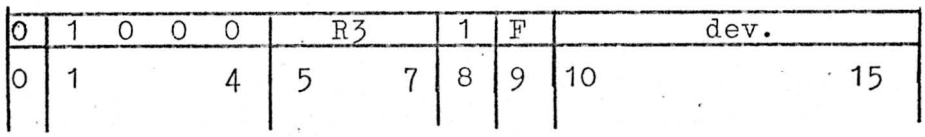


If the I/O routine is written as an interrupt program, it can be initialized in either of the two ways described earlier in this section - i.e. with or without TST intruction.

After this procedure the interrupted program can be processed: it will be interrupted, each time a data item has to be transferred, by the I/O exchange procedure, which is of higher priority.



## CIO: CONTROL INPUT/OUTPUT




Syntax

⌈<ident>⌉ ␣ CIO ␣ <r3>, ⌈0|1⌉, <dev>

This I/O instruction has the following functions:

- Start an I/O operation on a peripheral device (F = 1)
- Stop a data transfer or reset the state of a peripheral device control unit.

During execution <dev> field is sent, via the I/O bus, to the peripheral device control unit. The contents of the 16-bit register specified by R3 are also sent on the I/O bus to provide further information for some sophisticated control units. A start I/O command is not accepted if the device address is unknown or if the corresponding device is busy. Halt I/O is always accepted. The control unit will switch to the "wait" state after termination of data transfer with the peripheral device.

| Type | Execution Time (μsec) | |
|---|---|---|
| | P855 | P860 |
| T8 | 3.6 | 2.52 |

Condition register

CR = 0 command accepted
1 command not accepted
3 device address unknown

Remark

R3 ≠ 0

OTR: OUTPUT FROM REGISTER

| 0 | 1 | 0 0 0 | R3 | 0 | F | dev. |
|---|---|---|---|---|---|---|
| 0 | 1 | 4 | 5 7 | 8 | 9 | 10 15 |

Syntax

⌈<ident>⌉␣ OTR ␣ <r3> , ⌈0|1⌉, <dev>

If OTR is accepted a data character or word from the register specified by R3 is transferred to the device. According to the type of device the lower bits (right placed) or R3 or the whole 16 bits of R3 are taken into account by the control unit.

During execution, 'F' and <dev> field are sent to the device via the I/O bus. F may be used to specify a particular output function (e.g. binary or ASCII).

OTR is not accepted if the device control unit is not in the 'Exchange' state.

| Type | Execution Time (μsec) | |
|---|---|---|
| | P855 | P860 |
| T8 | 3.6 | 2.52 |

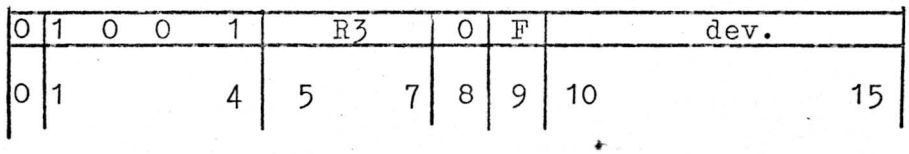
Condition register

CR = 0 command accepted
1 command not accepted
3 device address unknown

Remark

R3 ≠ 0

INR: INPUT TO REGISTER




Syntax

⌈<ident>⌉␣ INR␣<r3>,⌈0|1⌉,<dev>

If the I/O instruction is accepted a data word or character is transferred from the device to the register specified by R3. According to the type of the device the lower bits (right placed) or the 16 bits are significant. In the first case the higher bits of R3 are reset to zero. During execution 'F' and <dev> fields are sent to the device via the I/O bus. F bit may be used to specify a particular input function. This I/O instruction is not accepted if the device control unit is not in exchange state. After a not accepted INR instruction the contents of the R3-register are not significant.

| Type | Execution Time (μsec) | |
|---|---|---|
| | P855 | P860 |
| T8 | 3.6 | 2.52 |

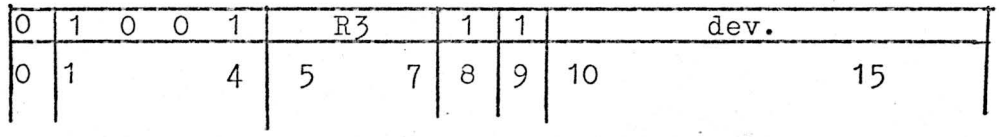
Condition register

CR = 0 command accepted
1 command not accepted
3 device address unknown

Remark

R3 ≠ 0

SST: SEND STATUS




Syntax

[<ident>]␣ SST␣ <r3>, <dev>

If SST is accepted a status character or word is transferred from the device to the register specified by R3. According to the type of the device only the right part of the register is used, in this case the left placed bits of R3 are reset to zero. The status word may have variable length. The following bits, if significant for the device concerned, have fixed positions:

15: not operable
14: throughput error
13: data fault
12: incorrect length

A device does not accept an SST instruction if it is not in "wait" status.
"Wait" status means that CT halt is executed by the computer and that data exchange with the peripheral device is terminated.
After a not accepted SST instruction the contents of the R3-register are not significant.

| Type | Execution Time (µsec) | |
|---|---|---|
| | P855 | P860 |
| T8 | 3.6 | 2.52 |

Condition register

CR = 0 command accepted
1 command not accepted
3 device address unknown

Remarks

<dev> must be ≠ 0.
R3 ≠ 0.

## RIL: READ INTERRUPT LINES

| 0 | 1 0 0 1 | R3 | 0 | 0 | 0 0 0 0 0 0 |
|---|---|---|---|---|---|
| 0 | 1 4 | 5 7 | 8 | 9 | 10 15 |

Syntax

⌈<ident>⌉ ␣RIL␣<r3>

This instruction is used to load in the register specified by R3 the state of the 16 interrupt lines, numbered from 0 to 15.

The active and non-masked lines are indicated by 1 in the corresponding bit position.

The inactive or masked or non-existent lines are indicated by 0.

| Type | Function | Execition Time (μsec) | |
|---|---|---|---|
| | | P855 | P860 |
| T8 | Interrupt lines→R3 | 2.6 | 2.28 |

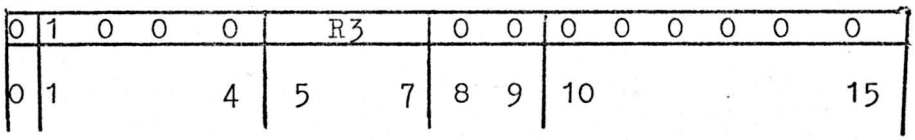
Condition register

Unchanged

Remark

R3 ≠ 0

## WIM: WRITE INTERRUPT MASK




Syntax

⌈<ident>⌉ ␣ WIM ␣ <r3>

The contents of the 16-bit register specified by R3 replace the contents of the mask register (16 bits).

The bits in the mask register are numbered from 0 to 15 and correspond to the interrupt lines 0 to 15.

A one-bit in the mask register indicates the corresponding line is inhibited; a zero-bit means the line is not inhibited.

| Type | Function | Execution Time (μsec) | |
|---|---|---|---|
| | | P855 | P860 |
| T8 | (R3) → mask register | 2.6 | 2.28 |

Condition register

Unchanged.

Remark

R3 ≠ 0.

RIT: RESET INTERNAL INTERRUPT

| 0 | 0 1 0 0 | 0 0 0 | 1 1 | dev. | 1 |
|---|---|---|---|---|---|
| 0 | 1 4 | 5 7 | 8 9 | 10 14 | 15 |

Syntax

⌈<ident>⌉uRIT

This I/O instruction is used to clear one of the internal interrupt bits which is indicated by a zero-bit in the <dev> field (other bits being one). Internal interrupts are:

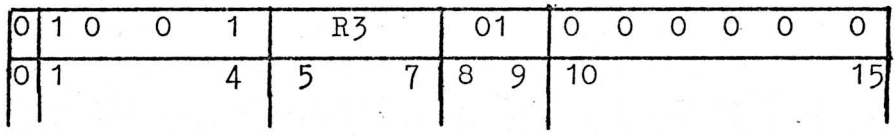
bit number 10 = control panel
11 = power failure/auotmatic restart
12 = real time clock
13 = program error
14 = memory protect error

| Type | Execution Time (μsec) | |
|---|---|---|
| | P855 | P860 |
| T8 | 1.9 | 1.56 |

Condition register

Unchanged

RCA: READ CHANNEL ADDRESS




Syntax

⌈<ident>⌉ ␣ RCA ␣ <r3>

This instruction is used to load into the register specified in R3 the channel number which has caused a MIDB overflow interrupt request. It deactivates the BUL line and resets the MIDB interrupt flip-flop.

| Type | Function | Execution Time (μsec) | |
|---|---|---|---|
| | | P855 | P860 |
| T8 | CNL → R3 | 3.6 | 2.52 |

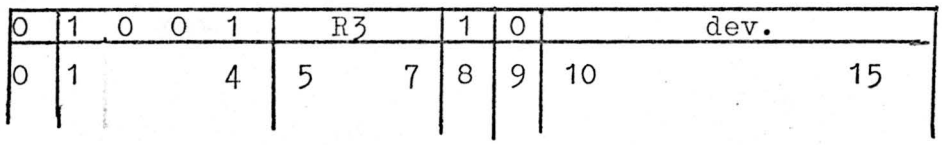
Condition register

Unchanged.

Remark

R3 ≠ 0.

## TST: TEST STATUS




### Syntax

[<ident>] ␣ TST ␣ <r3>, <dev>

This instruction may be used before starting any I/O operation to test if the device control unit is in the ready state. It is always accepted by the control unit.[1]
During the execution of the TST instruction a status word is sent from the control unit to the register specified by R3.
A 1-bit in position 15 of R3 indicates the control unit is not operable. Other bits are not significant.

| Type | Execution Time (µsec) | |
|---|---|---|
| | P855 | P860 |
| T8 | 3.6 | 2.52 |

### Condition register

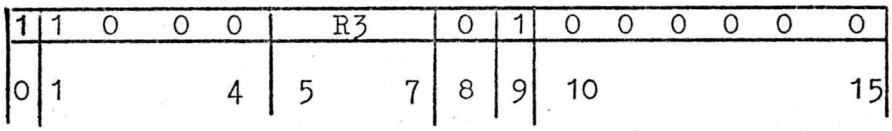
CR = 0 Accepted
3 device address unknown

### Remark

R3 ≠ 0.

1) "Ready" means that the device control unit is in the "inactive" state and can be used by other programs for I/O exchange (CT-start will be accepted).

## WMP: WRITE MASK PROTECTION




### Syntax

⌈<ident>⌉ ␣ WMP ␣ <r3>

The contents of the register specified in R3 replaces the contents of the memory protection key register.
Flip-flops in this register are numbered from 0 to 15 and correspond to the memory sectors 0 to 15.
A 1-bit in this key register indicates that the corresponding memory sector (= 1024 16-bit words) is protected; a 0-bit means that the memory sector is not protected.

| Type | Function | Execution Time (µsec) | |
|---|---|---|---|
| | | P855 | P860 |
| T8 | (R3)→M.P. key register | 2.6 | 2.28 |

### Condition register

Unchanged.

### Remark

R3 ≠ 0

## WM2: WRITE MASK PROTECTION NO.2

| 0 | 1 0 0 0 | R3 | 1 1 | 0 0 0 0 0 0 |
|---|---|---|---|---|
| 0 | 1 4 | 5 7 | 8 9 | 10 15 |

Syntax

[<ident>] ␣WM2␣<r3>

The contents of the register specified in R3 replace the contents of the memory protection key register no. 2.
Flip-flops in the key register are numbered from 0 to 15 and correspond to the memory sectors 16 to 31.
A 1-bit in the key-register indicates that the corresponding memory sector is protected; a 0-bit that it is not protected.

| Type | Function | Execution Time (µsec) P860 |
|---|---|---|
| T8 | (R3)→ M.P. key register | 2.28 |

Condition register

Unchanged

Remark

R3 ≠ 0.

## Memory Protection

If a certain bit is set, the corresponding memory area is protected.

MK1

| bit | memory area |
|---|---|
| 0 | 0000 - 07FE |
| 1 | 0800 - 0FFE |
| 2 | 1000 - 17FE |
| 3 | 1800 - 1FFE |
| 4 | 2000 - 27FE |
| 5 | 2800 - 2FFE |
| 6 | 3000 - 37FE |
| 7 | 3800 - 3FFE |
| 8 | 4000 - 47FE |
| 9 | 4800 - 4FFE |
| 10 | 5000 - 57FE |
| 11 | 5800 - 5FFE |
| 12 | 6000 - 67FE |
| 13 | 6800 - 6FFE |
| 14 | 7000 - 77FE |
| 15 | 7800 - 7FFE |

MK2

| bit | memory area |
|---|---|
| 0 | 8000 - 87FE |
| 1 | 8800 - 8FFE |
| 2 | 9000 - 97FE |
| 3 | 9800 - 9FFE |
| 4 | A000 - A7FE |
| 5 | A800 - AFFE |
| 6 | B000 - B7FE |
| 7 | B800 - BFFE |
| 8 | C000 - C7FE |
| 9 | C800 - CFFE |
| 10 | D000 - D7FE |
| 11 | D800 - DFFE |
| 12 | E000 - E7FE |
| 13 | E800 - EFFE |
| 14 | F000 - F7FE |
| 15 | F800 - FFFE |

## SPECIAL DEVICE HANDLING

The instruction sequence outlined in the foregoing varies slightly according to the device.

### I/O TYPEWRITER

The typewriter must be enabled in input or output mode. For this purpose the CIO-Start instruction must specify one of the registers 1 to 7, bit 15 of which must contain:

- 0: for output mode:
- 1: for input mode.

For the punched tape reader of the I/O typewriter the data stream must finish with X-off.

For the tape punch of the I/O typewriter, the data stream must start with 'tape-on' and end with 'tape-off'.

### HIGH-SPEED PUNCHED TAPE READER

If it is desired to process each character as it is input, the high-speed tape reader must be stopped between characters; i.e. a CIO-Start-INR-CIO Stop-SST loop must be performed for each character.

### HIGH SPEED TAPE PUNCH

No special restrictions.

MAGNETIC TAPE CASSETTE

The TST command indicates, not only whether the control unit is ready, but also which side of the cassette is uppermost:

- bit 7 of R3 register = 1: A-side up.
- bit 7 of R3 register = 0: B-side up.

The contents of the register specified in the CIO-command indicates the type of operation:

| Setting of R3-register bits 12-15 | Operation required |
|---|---|
| 0001 | Erase forward |
| 0010 | Backwards space block |
| 0011 | Forward space block |
| 0101 | Write a block forward |
| 0110 | Read a block reverse |
| 0111 | Read a block forward |
| 1000 | Rewind at fast speed |

The CIO-extensions are used to read and write records when the cassette unit is connected via the Multiplex Channel. If the Programmed Channel is used characters are transferred singly by INR and OTR commands.

MOVING HEAD DISC UNIT

The contents of the register specified in the CIO-command indicates the type of operation.

| Setting of R3-register bits | Operation required |
|---|---|
| bits 0-8: difference between cylinder numbers (absolute value)<br>bit 9: direction of motion<br>1 = towards inner track<br>0 = towards outer track<br>bit 14 = 1, bit 15 = 0 | seek |
| Bits 14-15: 11 | seek to zero |
| bits 3-7: sector address from 0-31.<br>bits 13-15: 000 | write a sector |
| bits 3-7: Sector address from 0-31.<br>bit 14 = 0, bit 15 = 1. | read a sector |

CONTROL UNIT/DEVICE ADDRESSING

All peripheral units and other external devices for the input or output of data are connected to the I/O bus, or to the DMA - I/O bus, through a control unit.
Each control unit must have a unique address decoded from the address lines of the I/O bus and corresponding to the I/O command.

If more than one device can be connected to a control unit, which is the case for magnetic tape and disc, separate bits are used to address the device. As an example, the following table shows addresses for control units and devices.

Table Example of C.U./device addresses

| I/O command | device address | | | | | | PROG. CH. | MULTIPLEX | DMAC | device |
|---|---|---|---|---|---|---|---|---|---|---|
| 9 | 10 | | | | | 15 | | | | |
| | 0 | 1 | 0 | 0 | 0 | 0 | x | | | I/O typewriter |
| | 1 | 0 | 0 | 0 | 0 | 0 | x | | | punched tape reader |
| | 1 | 1 | 0 | 0 | 0 | 0 | x | | | tape punch |
| | 0 | 0 | 0 | 0 | 0 | 1 | | x | x | F.H. disc no.1 |
| | 0 | 1 | 0 | 0 | 0 | 1 | | x | x | F.H. disc no.2 |
| | 1 | 0 | 0 | 0 | 0 | 1 | | x | x | F.H. disc no.3 |
| | 1 | 1 | 0 | 0 | 0 | 1 | | x | x | F.H. disc no.4 |
| | 0 | 0 | 0 | 0 | 1 | 0 | | x | x | M.H. disc no.1 |
| | 0 | 1 | 0 | 0 | 1 | 0 | | x | x | M.H. disc no.2 |
| | 1 | 0 | 0 | 0 | 1 | 0 | x | | | any control unit |
| | 1 | 1 | 0 | 0 | 1 | 0 | x | | | any control unit |
| | 0 | 0 | 0 | 0 | 1 | 1 | x | x | | magnetic tape no.1 |
| | 0 | 1 | 0 | 0 | 1 | 1 | x | x | | magnetic tape no.2 |
| | 1 | 0 | 0 | 0 | 1 | 1 | x | x | | magnetic tape no.3 |
| | 1 | 1 | 0 | 0 | 1 | 1 | x | x | | magnetic tape no.4 |
| | 0 | 0 | 0 | 1 | 0 | 0 | x | x | | cassette tape no.1 |
| | 0 | 1 | 0 | 1 | 0 | 0 | x | x | | cassette tape no.2 |
| | 1 | 0 | 0 | 1 | 0 | 0 | x | x | | cassette tape no.3 |
| | 1 | 1 | 0 | 1 | 0 | 0 | x | x | | cassette tape no.4 |
| | 0 | 0 | 0 | 1 | 0 | 1 | | x | | card reader |
| | 0 | 1 | 0 | 1 | 0 | 1 | x | | | any control unit |
| | 1 | 0 | 0 | 1 | 0 | 1 | x | | | any control unit |
| | 1 | 1 | 0 | 1 | 0 | 1 | x | | | any control unit |
| | 0 | 0 | 0 | 1 | 1 | 0 | | x | | line printer |
| | 0 | 1 | 0 | 1 | 1 | 0 | x | | | any control unit |
| | 1 | 0 | 0 | 1 | 1 | 0 | x | | | any control unit |
| | 1 | 1 | 0 | 1 | 1 | 0 | x | | | any control unit |

| I/O command 9 | device address 10 15 | PROG. CH. | MULTIPLEX | DMAC | device |
|---|---|---|---|---|---|
| | 0 0 0 1 1 1 | | | | |
| | 0 1 0 1 1 1 | | | | |
| | 1 0 0 1 1 1 | | | | |
| | 1 1 0 1 1 1 | | | | |
| | 0 0 1 0 0 0 | | | | DIOS no.1, 1st word |
| | 0 1 1 0 0 0 | | | | DIOS no.1, 2nd word |
| | 1 0 1 0 0 0 | | | | DIOS no.1, 3rd word |
| | 1 1 1 0 0 0 | | | | DIOS no.1, 4th word |
| | 0 0 1 0 0 1 | | | | DIOS no.2, 1st word |
| | 0 1 1 0 0 1 | | | | DIOS no.2, 2nd word |
| | 1 0 1 0 0 1 | | | | DIOS no.2, 3rd word |
| | 1 1 1 0 0 1 | | | | DIOS no.2, 4th word |
| | 0 0 1 0 1 0 | | | | MIOS 4C |
| | 0 1 1 0 1 0 | | | | MIOS 4C |
| | 1 0 1 0 1 0 | | | | MIOS 4S |
| | 1 1 1 0 1 0 | | | | MIOS 4D |
| | 0 0 1 1 0 0 | | | | |
| | 0 1 1 1 0 0 | | | | |
| | 1 0 1 1 0 0 | | | | |
| | 1 1 1 1 0 0 | | | | |
| | 0 0 1 1 0 1 | | | | adaptor |
| | 0 1 1 1 0 1 | | | | |
| | 1 0 1 1 0 1 | | | | |
| | 1 1 1 1 0 1 | | | | |
| | 0 0 1 1 1 1 | | | | DMAC |
| | 0 1 1 1 1 1 | | | | DMAC |
| | 1 0 1 1 1 1 | | | | MIDB |
| | 1 1 1 1 1 1 | | | | RTC |

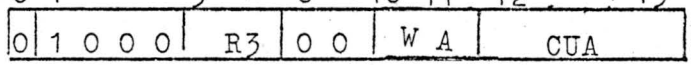
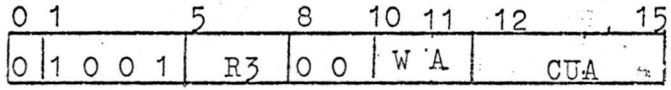
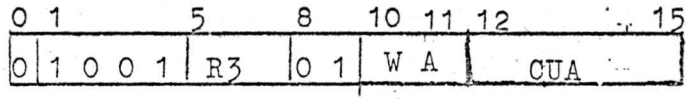
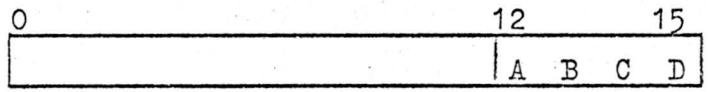
Control unit Status word configuration

| Bit | Description | control unit | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | ASR | CR | DU | LP | PTP | PTR | CASS Tape | PLOT |
| 0 | - | | | | | | | | |
| 1 | has become ready | | | x | | | | x | |
| 2 | - | | | | | | | | |
| 3 | tape mark has been read | | | | | | | x | |
| 4 | - | | | | | | | | |
| 5 | on cylinder | | | x | | | | | |
| 6 | seek error | | | x | | | | | |
| | write unable | | | | | | | x | |
| 7 | A or B side | | | | | | | x | |
| 8 | Device Address | | | | | | | x | |
| 9 | " | | | x | | | | x | |
| 10 | EOT | | | | | | x | x | |
| | tape low | | | | | x | | | |
| 11 | Program error | | | x | | | | x | x |
| 12 | incorrect length | | x | x | | | | x | |
| | Y limit overpass | | | | | | | | x |
| | | | | | | | | | |
| 13 | data fault | | | x | | | | | |
| 14 | throughput error | x | x | x | | | x | x | |
| 15 | not operable (only significant bit for TST) | x | x | x | x | x | x | x | x |

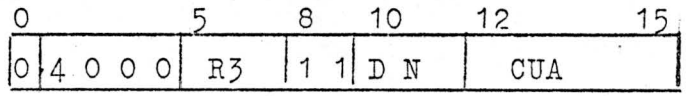
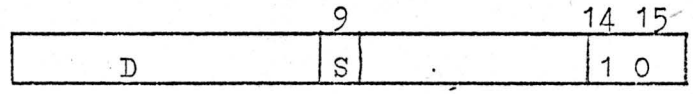
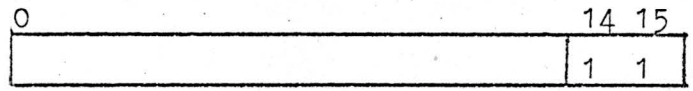
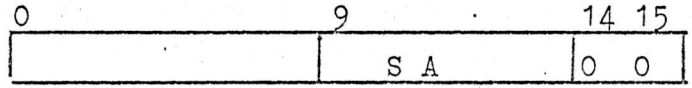
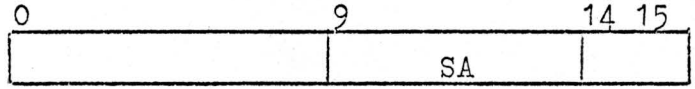
## DEVICES AND RECOGNIZED COMMANDS

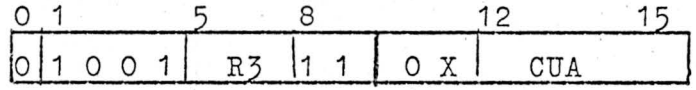
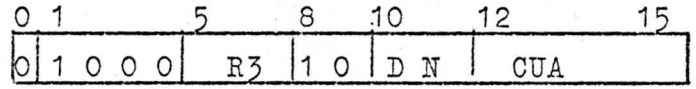
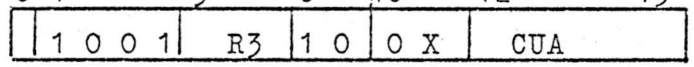
The following tables indicate per device, which commands are recognized, and also the bit-configuration and meaning of each command.

| Device | | | | | recognised command | Format | Meaning |
|---|---|---|---|---|---|---|---|
| PTR | CR | PLOT PTP | ASR | LP | | | |
| x | | | x | x | TST | 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15<br>0 1 0 0 1 \| R3 \| 1 0 \| D A | Test status<br>DA = Device Address<br>R3 = reg into which status is loaded; see status CU for contents R3 reg. |
| x | x | x | x | x | CIO start (read a card for CR) | 0 1 0 0 0 \| R3 \| 1 1 \| D A<br>15<br>Reg. indicated in R3 \| | R3 only significant for ASR<br>Start input<br>bit 15 = 1 start input<br>bit 15 = 0 start output |
| x | x | x | x | x | CIO stop | 0 1 0 0 0 \| R3 \| 1 0 \| D A | Stop input<br>R3 not significant |
| x | x | | x | | INR | 0 1 0 0 1 \| R3 \| 0 X \| D A | Input to register indicated in R3 field.<br>X: no significance |
| x | x | x | x | x | SST | 0 1 0 0 1 \| R3 \| 1 1 \| D A | Send status<br>The status word of the controlunit is sent to the register indicated in R3 field |
| | | X | X | X | OTR | 0 1 0 0 0 \| R3 \| 0 X \| D A<br>format control character<br>8 9 10 11 12 13 14 15<br>1 1 X 0 \| Chan.Nr | Output from the register indicated in R3 field to CU<br>Advance till given channel |
| | | | | x | | 8 9 10 11 12 13 14 15<br>1 1 X 1 \| Nr.of lines | Skip the given number of lines |

| Recognised Command | Format | Meaning |
|---|---|---|
| OTR |  | Output data word<br>R3 indicator register from data exchange<br>CUA = CU address<br>WA = Word address<br>00 = word A<br>01 = " B<br>10 = " C<br>11 = " D |
| INR |  | Input data word<br>R3 indicates register into which data is exchanged.<br>CUA = CU address<br>WA = Word address (see above). |
| INR |  <br>Format of R3<br>  | Input word address<br>R3 indicates the register ... into which the address is loaded<br>CUA = CU address<br>WA = not significant<br><br>A one-bit in field 12-15 indicates that the corresponding word is active. |

# Moving head disc CU

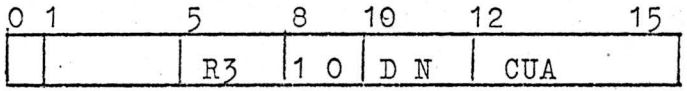
| recognised commands | Format 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 | Meaning |
|---|---|---|
| CIO start general |  | Start an Operation<br>DN = Device number 00 or 01<br>CUA = Control Unit address<br>R3 = specifies the command |
| | R3 Formats | |
| seek command |  | move the head to another cylinder<br>D = difference between cylinder numbe<br>S = direction of motion<br>1 = forward<br>0 = reverse |
| Seek to zero command |  | move heads to cylinder 00 |
| Write a sector |  | Start a write command on the sector address on the BOU lines on the cylinder of the moment.<br>SA = Sector Address 0-31 |
| Read a sector |  | Start a read operation on the sector which is addressed on the BOU lines on the cylinder of the moment<br>SA = Sector address 0-31 |

| Recognised commands | Format<br>0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 | Meaning |
|---|---|---|
| SST |  | Exchange Status between CPU and CU<br>CUA = CU address<br>X = not significant<br>R3 = specifies register into which status is sent (see status word) |
| Stop command |  | Stop data transfer<br>R3 = not significant<br>DN = Device number<br>CUA = CU address |
| TST |  | Test status of CU<br>CUA = CU Address<br>R3 indicates register into which status is exchanged. |

## Cassette tape CU

### TST

Format

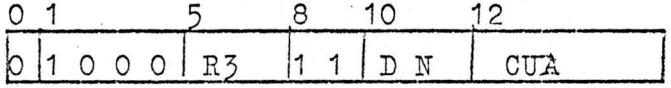



Meaning

Test status
CUA = CU address
DN = Device number (00-11)
R3 = indicating register into which status is exchanged
see status.

### CIO Start

Format

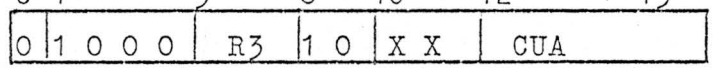
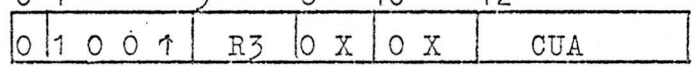
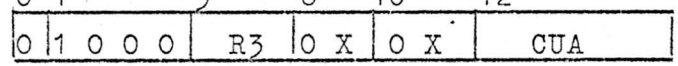
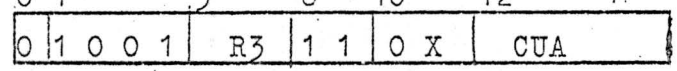



Meaning

Start command
CUA = CU address
DN = Device number
R3 = indicates register specifying the command.

R3 contents

| bit number 12 13 14 15 | Command |
|---|---|
| 0 0 0 0 | Lock/unlock |
| 0 0 0 1 | Erase forward |
| 0 0 1 0 | Backwards space block |
| 0 0 1 1 | Forward " " |
| 0 1 0 1 | Write a block forward |
| 0 1 1 1 | Read " " " |
| 1 0 0 0 | Rewind at fast speed |
| 1 0 1 0 | Search tape mark backwards |
| 1 0 1 1 | " " " forward |

| Recognised commands | Format<br>0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 | Meaning |
|---|---|---|
| CIO Stop |  | Stop transfer<br>CUA = CU address<br>DN = not significant<br>Contents of register indicated in R3 not significant.<br>X = |
| INR |  | Input to register<br>DN = not significant<br>R3 indicates register into which data is exchanged.<br>X not significant |
| OTR |  | Output from register<br>DN = not significant<br>X = not significant<br>R3 indicates register from which the data is exchanged. |
| SST |  | Send Status<br>see status meaning<br>for explanation of field see above |

## EXAMPLE OF AN I/O PROGRAM IN INTERRUPT MODE

First, words are reserved to store parameters.
The 'PHILIPS-ELECTROLOGICA' 'APELDOORN' is written into an area labeled BUFFER, 34 characters long. (CR-LF are given after 'PHIL.-EL.' and after 'APELDOORN').
Then, the main program:

Stack pointer is set in A15.
A mask (with bit 8=0) is written into A7 and loaded into the mask-register so that only the typewriter is enabled (this is not realistic, of course, only for the purpose of this example).
The address (INTY) of the interrupt routine is loaded into A1 and stored in interrupt location 14A.
The buffer length is loaded into A1 (34).
The buffer address is loaded into A2 (BUFFER).
Zeroes are loaded into A3 and A4.
These 4 parameters are stored respectively in the reserved words BULEN, BUAD, ACT and READY.
The AIR is started for output.

Now, the interrupt routine:
Contents of A1 to A4 are stored in stack (A15).
The parameters 34, BUFFER, 0 are loaded into A1, A2 and A3. One character of the buffer is loaded and output on the Teletype. If it is not accepted: SST to test the status of the typewriter. If accepted the ACTual length (A3) is incremented by 1 and compared with A1, until it is 34. Then all characters have been output. If not, return is made to main program. If output is ready: stop ASR.

```
M:LD
    IDENT   ASM8K   1880
:EOS  3A58
:EOF
M:ST
A:1023R

   00000                            IDENT    OUTINT
   00001                        *
   00002                        *  OUTPUT PROGRAM IN INTERRUPTMODE
   00003                        *  INTERRUPT ADDRESS IS /4A
   00004                        *  BIT 8 OF MASK REGISTER
   00005                        *
   00006                        TY        EQU      /10
   00007                                  RORG     *+/100
   00008  0100                  BULEN     RES      1            BUFFERLENGTH
   00009  0102                  BUAD      RES      1            BUFFERADDRESS
   00010  0104                  ACT       RES      1            ACTUAL LENGTH
   00011  0106                  READY     RES      1            READY AND ERRORBIT
   00012  0108  5048            BUFFER    DATA     'PHILPIPS-ELECTROLOGICA '
          010A  494C
          010C  5049
          010E  5053
          0110  2D45
          0112  4C45
          0114  4354
          0116  524F
          0118  4C4F
          011A  4749
          011C  4341
          011E  2020
   00013  0120  0D0A                  DATA     /0D0A
   00014  0122  4150                  DATA     'APELDOORN'
          0124  454C
          0126  444F
          0128  4F52
          012A  4E20
   00015  012C  0D0A                  DATA     /0D0A
   00016                        *
   00017                        *  MAIN PROGRAM
   00018                        *
   00019  012E  87A0            STOUT     LDKL     A15,BULEN-2  STACKPOINTER
          0130  00FE  R
   00020  0132  8720                  LDKL     A7,/FF7F
          0134  FF7F
   00021  0136  4700                  WIM      A7           ONLY TY INTERR.
   00022  0138  8120                  LDKL     A1,INTY
          013A  0000  F
   00023  013C  8141                  ST       A1,/4A       FILL IN INT. ADDR.
          013E  004A
   00024  0140  0122                  LDK      A1,34
   00025  0142  8220                  LDKL     A2,BUFFER
          0144  0108  R
   00026  0146  0300                  LDK      A3,0
   00027  0148  0400                  LDK      A4,0
   00028  014A  BA41                  MS       4,BULEN      FILL IN PARAMETERS
          014C  0100  R
   00029  014E  0500                  LDK      A5,0         OUTPUT MODE
   00030  0150  45D0                  CIO      A5,1,TY      START ASR
   00031  0152  8140            WAIT      LD       A1,READY     TEST READY BIT
          0154  0106  R
   00032  0156  5806                  RB(0)    WAIT
   00033  0158  207F                  HLT
```

```
00034                          *
00035                          *  TYPEWRITER INTERRUPT ROUTINE
00036                          *
00037  015A  BA3F       INTY     MSR      4,A15        A1,A2,A3, A4 TO STACK
00038  015C  B9C0                ML       3,BULEN      3 PARAMETERS
       015E  0100  R
00039  0160  920C                ADR      A2,A3        COMP. CHAR. ADDR.
00040  0162  E428                LCR      A4,A2        LOAD CHAR.
00041  0164  4410                OTR      A4,0,TY
00042  0166  5100  F             RF(1)    SST          NOT ACCEPTED
00043  0168  9041                IM       ACT          UPDATE ACTLEN
       016A  0104  R
00044  016C  E90C                CWR      A1,A3        TEST IF READY
00045  016E  5100  F             RF(1)    RETURN       NOT READY
00046  0170  4190                CIO      A1,0,TY      STOP ASR
00047  0172  5700  F             RF       RETURN
00048  0174  4AD0       SST      SST      A2,TY
00049  0176  5100  F             RF(1)    ERROR
00050  0178  8120                LDKL     A1,/8000
       017A  8000
00051                          *
00052                          *  SET CONTROL BITS
00053                          *  BIT 0 IS READY BIT
00054                          *  BIT 1 IS ERROR BIT
00055                          *
00056  017C  8141       SETBIT   ST       A1,READY     STORE CONTROL BITS
       017E  0106  R
00057  0180  BA3E       RETURN   MLR      4,A15
00058  0182  F03E                RTN      A15
00059  0184  8120       ERROR    LDKL     A1,/C000
       0186  C000
00060  0188  5F0E                RB       SETBIT
00061                             END      STOUT
        SYMBOL TABLE

  TY      0010  A   BULEN   0100  R   BUAD    0102  R   ACT     0104  R
  READY   0106  R   BUFFER  0108  R   STOUT   012E  R   INTY    015A  R
  WAIT    0152  R   SST     0174  R   RETURN  0180  R   ERROR   0184  R
  SETBIT  017C  R
        ASS.ERR.    00000
:EOF
A::EOF

EXIT
```

EXAMPLE OF A PROGRAM TO READ 20 CHARACTERS VIA MULTIPLEX

```
M:ST
A:1020R

   00000                          IDENT    MPLEX
   00001                    *
   00002                    *  PROGRAM TO READ 20 CHARACTERS
   00003                    *  VIA MULTIPLEX
   00004                    *  DEVICE ADDRESS IS 2
   00005                    *  THEN THE CONTROL WORDS ARE /88 AND /8A
   00006                    *  INTERRUPT LEVEL IS 9
   00007                    *
   00008  0000  0000        EOS      DATA     0
   00009  0002              READY    RES      1
   00010  0004              BUF      RES      10
   00011                    *
   00012                    *  MAIN PROGRAM
   00013                    *
   00014  0018  87A0        START    LDKL     A15,EOS      STACK POINTER
          001A  0000  R
   00015  001C  8120                 LDKL     A1,INT
          001E  0000  F
   00016  0020  8141                 ST       A1,/52       LEVEL 9
          0022  0052
   00017  0024  8120                 LDKL     A1,/CFEC     CHAR,INP,0,0,-20
          0026  CFEC
   00018  0028  8220                 LDKL     A2,BUF+/12   END OF BUF
          002A  0016  R
   00019  002C  B941                 MS       2,/88        FILL IN CW
          002E  0088
   00020  0030  0300                 LDK      A3,0
   00021  0032  8341                 ST       A3,READY     CLEAR READY
          0034  0002  R
   00022  0036  44C2                 CIO      A4,1,2       START DEVICE
   00023  0038  8140        WAIT     LD       A1,READY
          003A  0002  R
   00024  003C  5806                 RB(0)    WAIT         NOT READY
   00025  003E  207F                 HLT
   00026                    *
   00027                    *  INTERRUPT ROUTINE
   00028                    *
   00029  0040  813F        INT      STR      A1,A15       A1 TO STACK
   00030  0042  49C2                 SST      A1,2
   00031  0044  9041                 IM       READY        SET READY
          0046  0002  R
   00032  0048  813E                 LDR*     A1,A15
   00033  004A  F03E                 RTN      A15
   00034                             END      START
            SYMBOL TABLE

  EOS     0000  R   READY   0002  R   BUF     0004  R   START   0018  R
  INT     0040  R   WAIT    0038  R
           ASS.ERR.     00000
:EOF
A::EOF

EXIT
```

Interface with the P855/P860 computers may take any of five different forms. The simplest interface is between a standard control unit and the device for which the control unit has been designed. Interface with the DIOS digital input/output system is achieved by selecting a suitable DIOS module. For the user who designs his own control unit interface with the universal I/O bus and the direct memory access channel I/O bus 'S' it is possible - and quite simple with readily available TTL components. Interface for Direct Memory Access or Memory Increment Data Break is, again, quite simple with readily available TTL components.

Control units and DIOS units are built on printed circuit cards and housed in the CPU cabinet or an extension cabinet. Interface with these units is made by one or two cable connectors on the outer end of the circuit cards. Customer designed units to interface with the I/O bus or I/O bus 'S', when built on P855/P860 standard circuit cards, are also housed in the CPU cabinet or an extension cabinet. The card slots allocated to control units are wired onto the I/O bus and I/O bus lines are designed to connect a fan-out or fan-in of one TTL load in each card slot. Similarly, the cards used for Direct Memory Access or Memory Increment Data Break are connected to the Memory gates on the CPU 'ALU' card and each line is designed to connect one TTL load or output. Typical characteristics of TTL components used in the P855/P860 series are as follows:

Supply.....+5.0 volts
Logic '0' output .....+0.2 volts
Logic '1' output .....+3.0 volts.

UNIVERSAL I/O BUS AND I/O BUS'S'

The universal I/O bus comprises 28 output lines and 20 input lines hard wired between the CPU logic card slots and the card slots allocated for I/O control units. Three additional lines are provided for use only with the MIOS system control unit. Pin connections are standardized for all control unit card connectors and any control unit card can be used in any control unit slot without modification. The I/O bus 'S' is identical to the universal I/O bus but is wired, in the CPU cabinet, only from the DMAC logic card slots to two slots allocated for line driver cards. A cable from each of the line driver cards takes the bus to extension cabinets where the cables are terminated on a line receiver card and the bus is wired to the circuit card connectors in the extension cabinet. Fan-in and fan-out is one TTL load for each card slot. Figure 10 shows the distribution of the universal I/O bus and I/O bus'S' in the CPU cabinet.

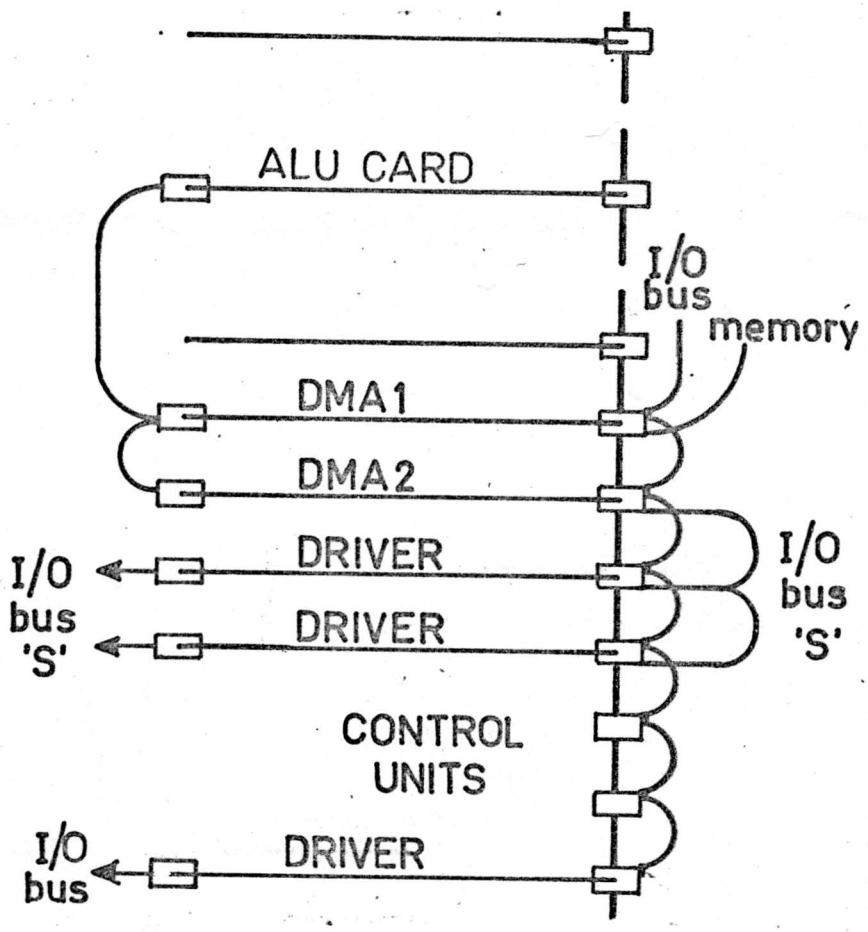



Figure 10 I/O bus and I/O bus 'S' distribution

Details of the universal I/O bus, the I/O bus 'S', and direct memory access interface are shown in Figure 11. Two programmed channel connected peripheral device control units are shown on the universal I/O bus and two peripheral device control units are shown on the I/O bus 'S' of the direct memory access channel. Signal lines contained in the universal I/O bus in the I/O bus 'S' are:

BIN lines:
Sixteen data input lines from the device control units to a CPU register, in the case of the universal I/O bus, and to the DMAC memory register in the case of the I/O bus 'S'.

BOU lines:
Sixteen data output lines from a CPU register, in the case of the universal I/O bus, to the device control units and from the DMAC memory register to the device control units in the case of the I/O bus 'S'.

BAD lines:
Six lines used to send the 6-bit address from the CPU to the device control units either directly or via the DMAC. Decoding logic in each device control unit allows it to respond to only one address code.

ARE line:
The addressed device control unit gives a response signal on this line to indicate that the address on the BAD lines has been recognized.

BOF lines:
Three lines used to send a 3-bit function code from the CPU to the device control unit, via the DMAC in the case of the I/O bus 'S'.

DAV line:
The signal sent on this line - by the CPU - validates the data on the BAD and BOF lines.

ACC line:
The ACC signal from the device control unit indicates to the CPU that the function on the BOF lines has been accepted.

EOR line:
Used only in multiplex and DMAC operation. The EOR (End-of-Range) signal indicates to the device control unit (addressed by the BAD lines) that the data block transfer is complete.

MCL line:
The master clear signal sent on this line by the CPU is used to clear the control unit registers and reset the sequence logic.

---

The above I/O bus lines are common to all device control units. The remaining lines are not common but are unique to each device control unit.

IR line:
This programmed channel interrupt request line, is activated by the device control unit when it is ready to effect a data transfer or send status information to the CPU under programmed channel control.

BR lines:
BR1, break request line, is activated by the device control unit when it is ready to effect a Multiplex or DMAC data transfer.
BR2, BR3, and BR4 are reserved for MIOS operation.

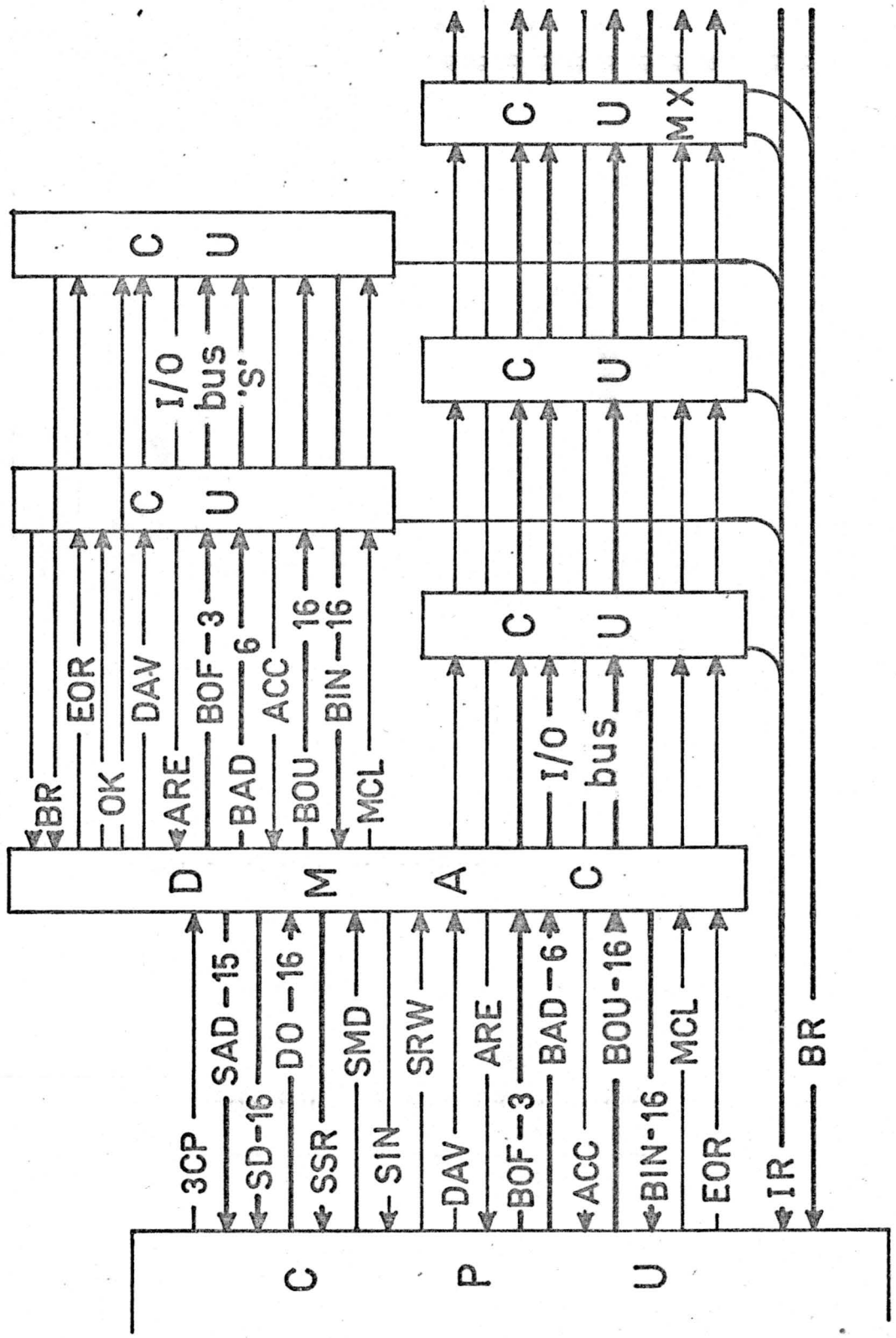
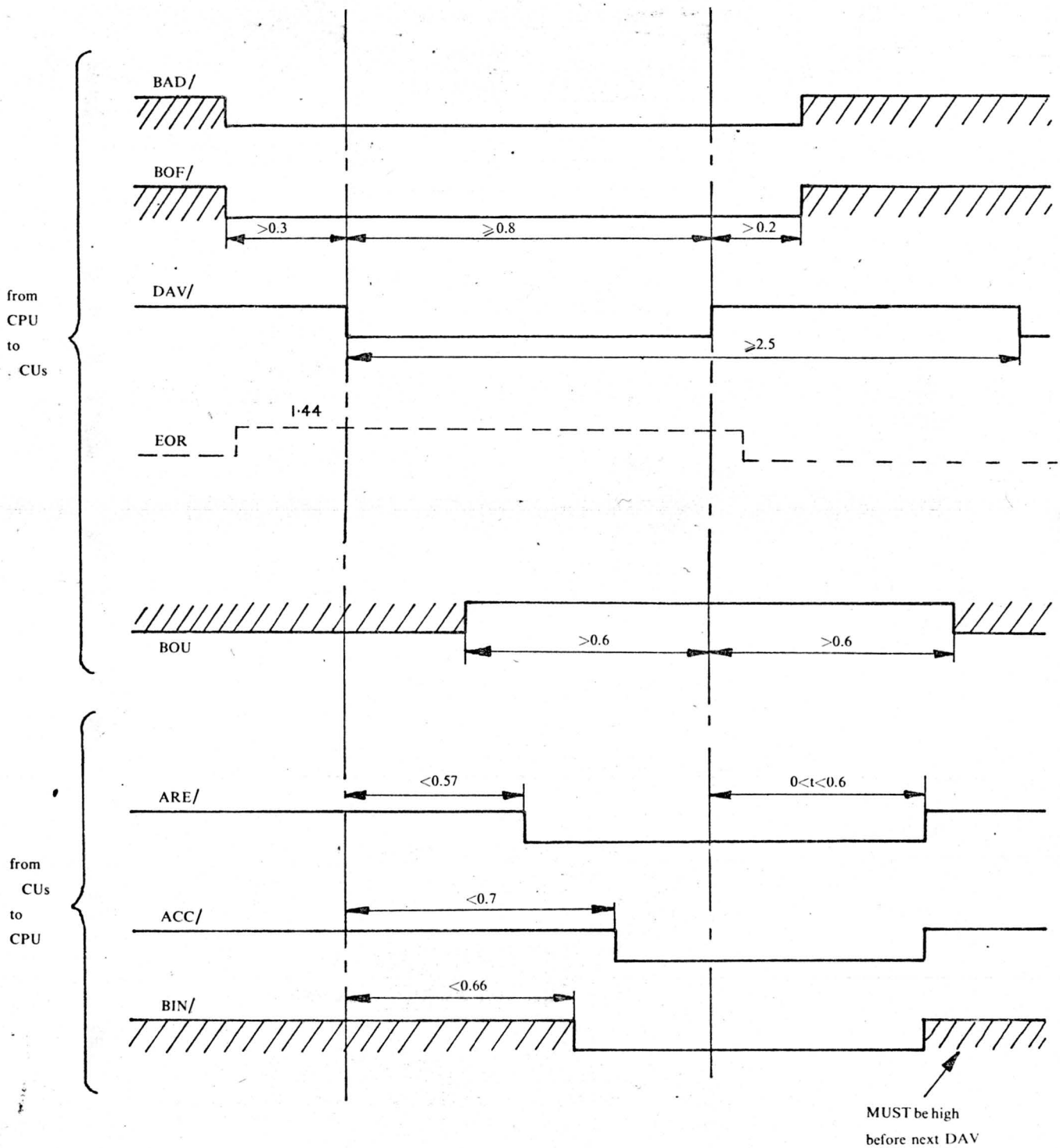



Figure 11 Universal I/O bus and I/O bus 'S'

## I/O BUS SIGNAL TIMING

The timing of the I/O bus signals is shown in Figure 12. IR, BR and MCL signals are not shown on the timing diagram because each is dependent upon factors such as the mechanical action of peripheral devices that are unrelated to the system clock.




Note:

1 All levels are active low except BOU which is active high.

2 This timing does not take into account any delays due to the addition of I/O bus extenders.

Figure 12 Bus signal timing

## CONTROL UNIT OPERATION ON THE I/O BUS

The following commands originate in I/O program instructions and are sent to the device control unit (addressed by the BAD lines) on the BOF lines of the I/O bus:

CIO - To start or stop a data transfer.
INR - To effect a transfer from a device to the CPU.
OTR - To effect a transfer from the CPU to a device.
TST - To test the status of a device (busy or not busy).
SST - To test the status of a device after the transfer has been completed.

The device control unit address is taken from the 6-bit address field of the I/O instruction and placed on the BAD lines. Each control unit has a BAD decoding logic to recognize its own address. Address selection in the decoding logic is made by jumper wires on the circuit card of all system control units. Where more than one device or channel is controlled by a single control unit the address logic is arranged to recognize all required addresses.

The commands on the BOF lines are decoded in the function code logic of the control unit as follows:

| | Instruction bit | 4 | 8 | 9 |
|---|---|---|---|---|
| Instruction | BOF line | 00 | 01 | 02 |
| CIO Start | | 0 | 1 | 1 |
| CIO Stop | | 0 | 1 | 0 |
| INR (Input transfer) | | 1 | 0 | X * |
| OTR (Output transfer) | | 0 | 0 | X * |
| TST (Test status) | | 1 | 1 | X |
| SST (Send status) | | 1 | 1 | 1 ** |

* May be decoded and used to specify variations in the type of transfer required.

** Significant only in control unit 'Wait' mode.

Each BOF line function code is accompanied by a BAD address code that must be recognized by the control unit before the function code is effective.

Acceptance of a function code depends on which mode the control unit is in at the time. P855/P860 control units are sequenced through four modes during data transfer:

INACTIVE MODE

The only commands accepted by a control unit in this mode are CIO start and TST. In this mode the control unit responds to a TST command by placing zeros on all BIN lines to indicate a 'not busy' condition. A CIO Start command switches the control unit from the Inactive to the Execute or Exchange mode depending upon whether the control unit is designed for input or output.

EXECUTE MODE

In this mode the control unit causes the peripheral device to make a single operation - read one character, for example in the case of a tape or card reader - after which it switches to the Exchange mode. A CIO command switches the unit to the Wait Status mode and a TST command places a 'busy' signal on the BIN lines without switching the mode. No other commands are accepted in the Execute mode.

EXCHANGE MODE

In this mode the control unit activates either the BR (Break Request) line - if the control unit is connected for multiplex or DMAC - or the IR (Interrupt Request) line - if the control unit is connected for programmed channel operation - to indicate to the CPU that it is ready to send or receive data. The INR, OTR and TST commands are acceptable in this mode and will effect a data transfer. After the transfer the control unit reverts to the Execute mode.

WAIT MODE

The control unit switches to this mode after receiving a CIO stop command whilst in the Execute mode. TST and SST commands are accepted in this mode and in both cases the control unit places the device status on the BIN lines. The control unit also responds to a SST command with an interrupt request on the IR line and then switches to the Inactive mode.

Indication of device status in response to an SST command will vary with the kind of device connected to the control unit. The most common status bits are:

BIN line 15: Not operable/manual intervention required.
BIN line 14: Throughput Error.
BIN line 13: Data fault.
BIN line 12: Incorrect length.

## DEVICE CONTROL UNIT HARDWARE

Control units used in the CPU cabinet are each built on a circuit card. The more elaborate control units used for high speed devices on DMAC require a number of circuit cards and are housed in the peripheral device cabinet. Each control unit in the CPU cabinet plugs into a card connector wired to the I/O bus and to the power supply lines. One or two connectors on the other end of the card connect via a cable (maximum length 15 metres) to the peripheral device. Card slots in the CPU cabinet are assigned to control units and these are shown in the card location diagram in Figure 13.

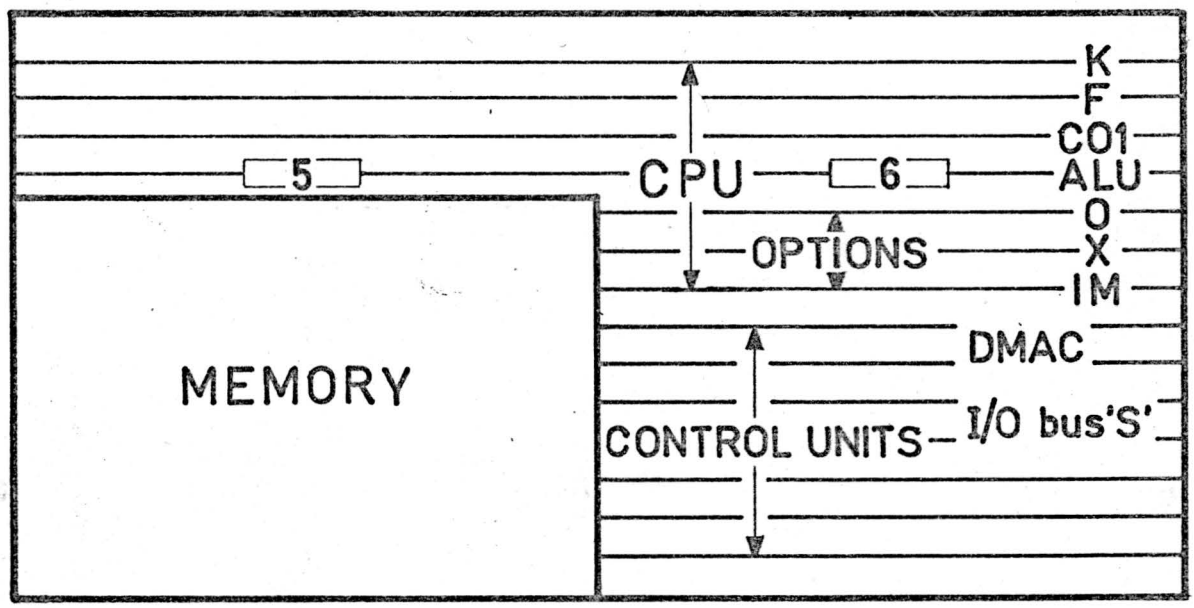



Figure 13 Circuit Card Locations in P860 Cabinet

USER DESIGNED CONTROL UNITS

Non-standard control units may be designed and built by the user on circuit cards to fit the card slots assigned to control units. Card dimensions and connector placement is shown in figure 14.

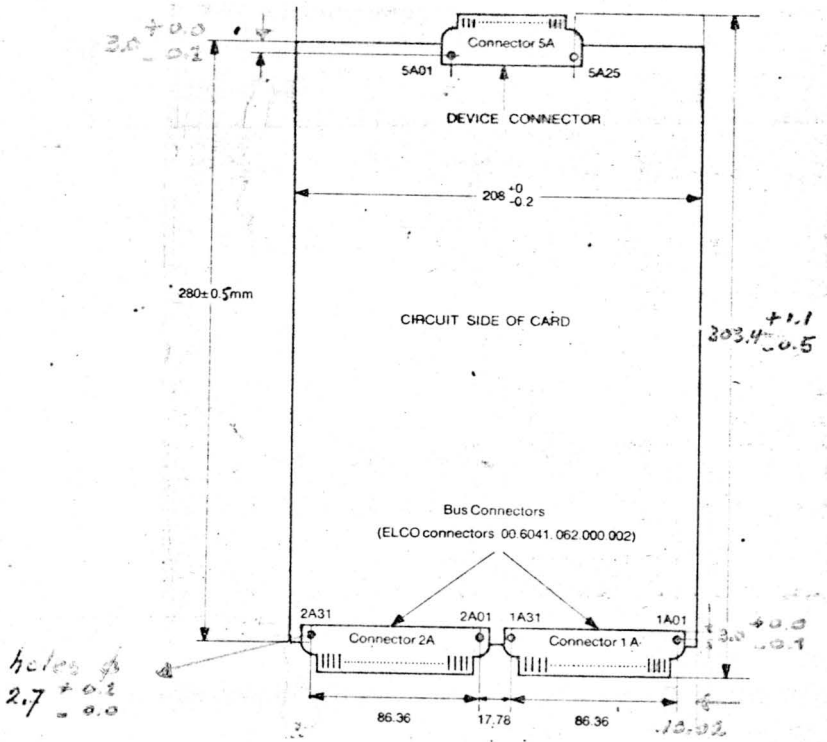
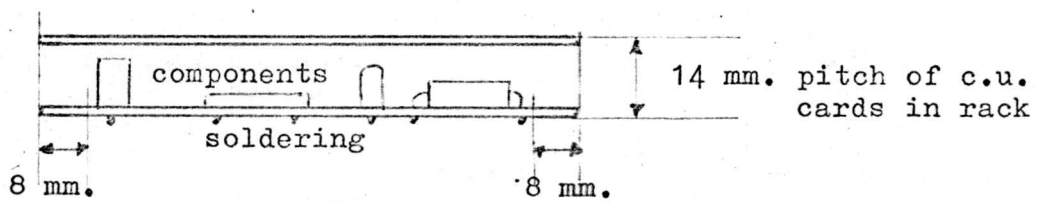



Figure 14 DCU Circuit Card (with dimensions in millimetres)

General purpose circuit card P849-022 shown in Figure 15 will accept both TTL components and discrete component circuitry for user designed control units. The circuit pattern provides power connections for TTL logic components and short leads from the logic connections for circuit jumper wires. Provision is made on the card for two connectors to make with the card slot connectors and two connectors for peripheral device cables.



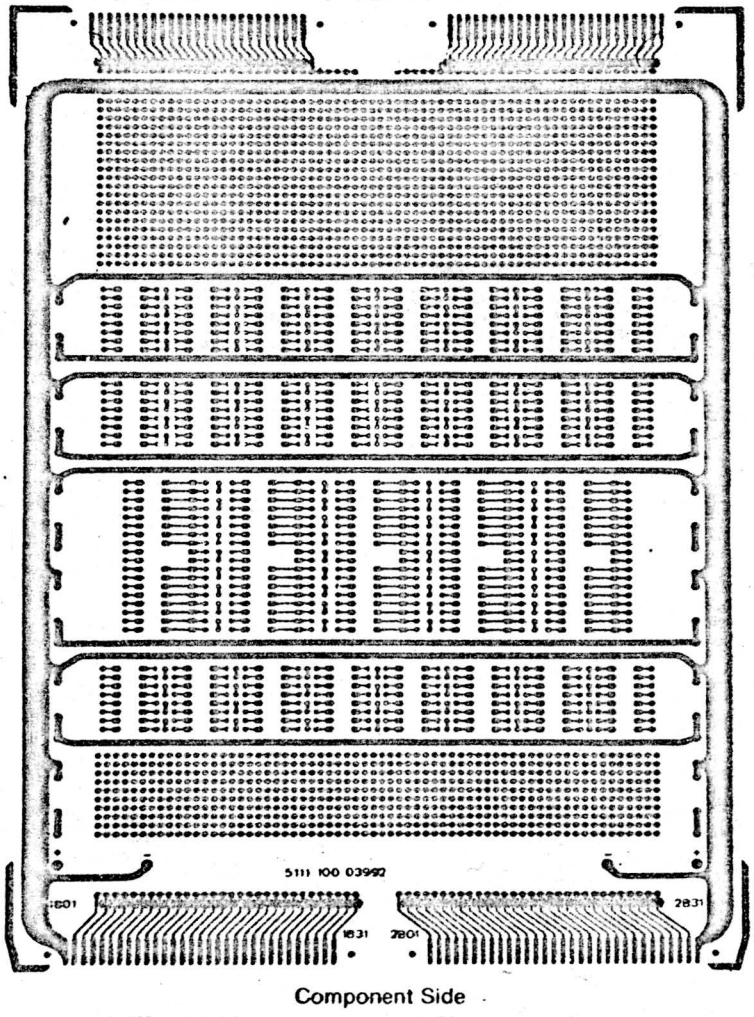



Figure 15 P849-002 General Purpose card

## TYPICAL DCU AND SUGGESTED CIRCUITS

Figure 16 shows a block diagram of a typical DCU. Operation of the DCU is controlled by I/O commands and is extremely simple due to the design of the Universal I/O bus. To assist the user who wishes to design his own DCU's, the following description of operation also includes typical logic circuits.

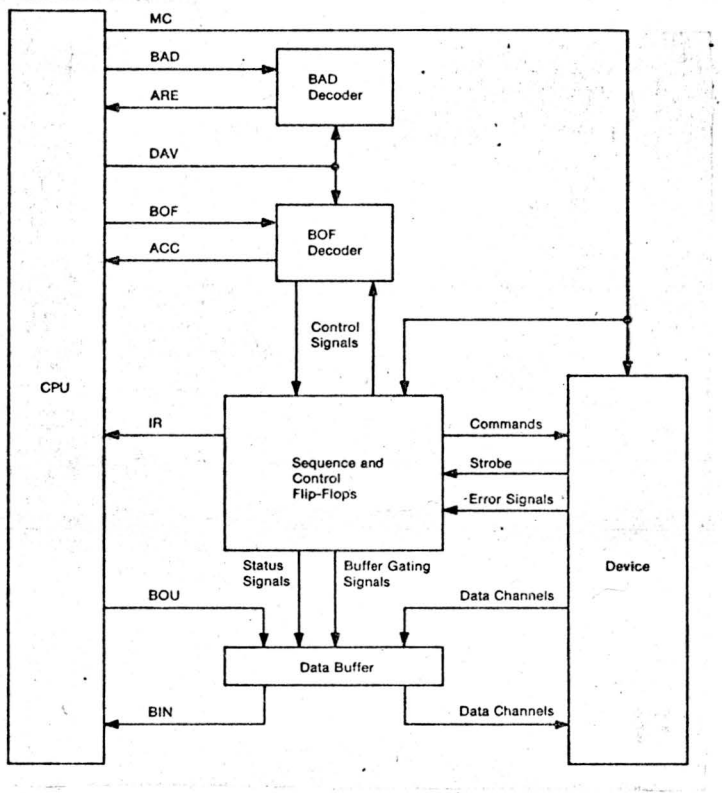



Figure 16 Block Diagram of Typical DCU

ADDRESSING

Each DCU must have a unique address that it can decode from the addresses that are sent on the BAD lines. The Philips designed DCU's have decode circuits similar to the one shown in figure 17.

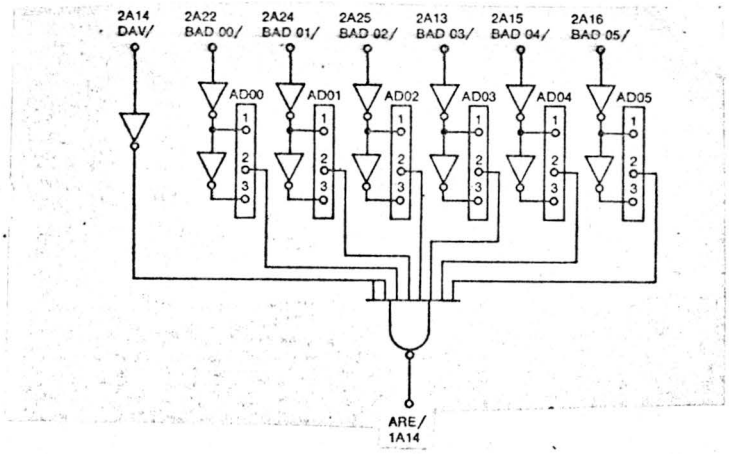



Figure 17 Typical Address Decode Logic

The inclusion of the link connectors - AD00 to AD05 - allows the user to change the address of the DCU by simply changing the position of the appropriate links.

Example:
For device address 5 (in binary) the links would be connected as follows:

| AD00 | AD01 | AD02 | AD03 | AD04 | AD05 |
|---|---|---|---|---|---|
| 1 to 2 | 2 to 3 | 1 to 2 | 2 to 3 | 2 to 3 | 2 to 3 |

Data on the BAD lines is validated by the DAV/ signal from the CPU. When the address is recognized, the ARE/signal is sent back to the CPU by the DCU.

FUNCTION DECODING

Once the address has been recognized, the function on the BOF lines is decoded and a typical circuit is shown in figure 18.

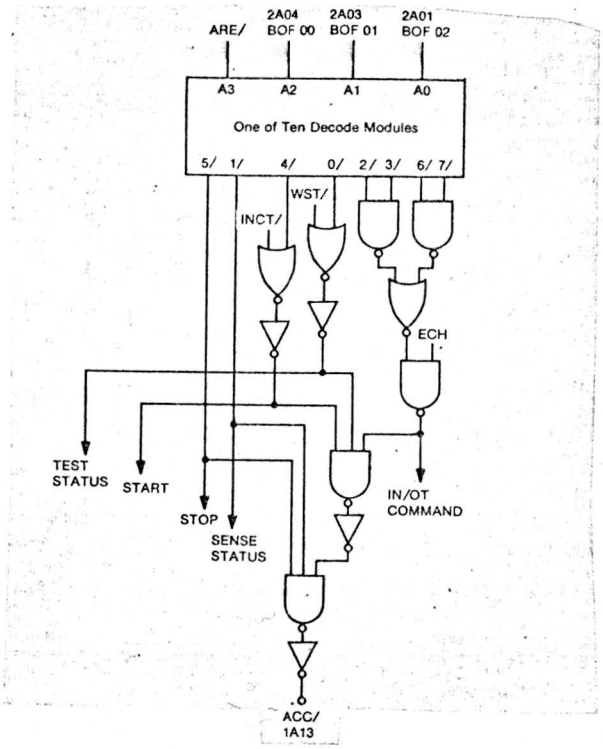



Figure 18 Function Decode Logic

The above logic will decode the function on the BOF lines, check the state of the DCU and the device and decide if the command can be accepted. The CIO Stop and the Test Status commands are always accepted by the DCU no matter what its state. For all other commands the DCU must be in the appropriate state. The change-of-state is controlled by the sequencer logic.

SEQUENCER LOGIC

This logic controls data transfers by switching the DCU into one of four states. Figure 19 shows a typical circuit.

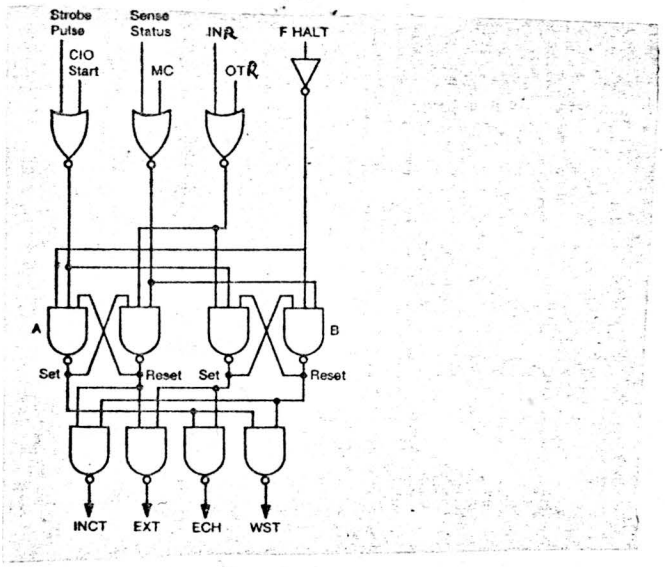



Figure 19 Sequence Logic

The four state of this logic interface the I/O bus timing to the mechanical actions of the device. These four states are:

INCT This is the inactive state where the DCU is waiting for a CIO Start command - from the CPU - to enable it to start a data exchange. It is switched into this state when both flip-flops A and B are reset by either a Master Clear (MC) or Send Status signal. Output from this gate enables the Function Decode logic.

EXT This is the execute state where a single mechanical action is performed by the peripheral device (i.e. a read or write operation). It is switched into this state when flip-flop A has been reset and flip-flop B has been set by the DCU accepting either an INR (input) or OTR (output) command from the CPU.

ECH This state is the exchange state when the data transfer - between the CPU and the DCU - is actually executed. It is switched into this state when both the A and B flip-flops have been set by either a CIO Start command (from the CPU) or a strobe pulse (from the device) depending on the type of exchange.

WST This is the Wait Status state. It is switched into this state when flip-flop A is set and flip-flop B is reset by signal FHALT. FHALT will be produced by either a CIO Stop command (from the CPU) or when an error has been detected by the DCU. The DCU will remain in this state until it receives a Send Status command or MC signal from the CPU. When either of these are received, the DCU will switch back to the INCT state to await the next CIO Start command.

CONTROL FLIP-FLOPS

These flip-flops are very simple and are used to remember that certain events have occurred; such as commands from the CPU, device states or errors. The number of flip-flops needed by a DCU will depend on the type of device, therefore only two are shown in figure 20.

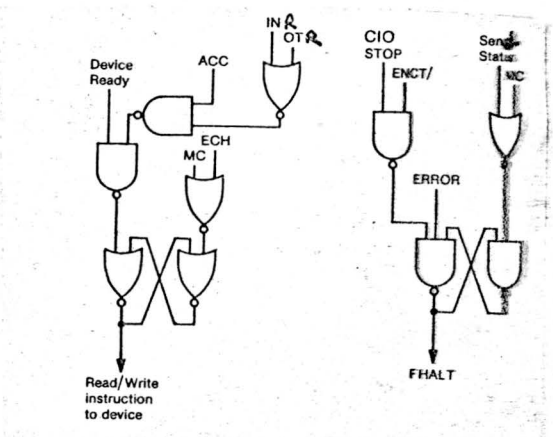



Figure 20 Typical Control Flip-Flops

ECH This state is the exchange state when the data transfer - between the CPU and the DCU - is actually executed. It is switched into this state when both the A and B flip-flops have been set by either a CIO Start command (from the CPU) or a strobe pulse (from the device) depending on the type of exchange.

WST This is the Wait Status state. It is switched into this state when flip-flop A is set and flip-flop B is reset by signal FHALT. FHALT will be produced by either a CIO Stop command (from the CPU) or when an error has been detected by the DCU. The DCU will remain in this state until it receives a Send Status command or MC signal from the CPU. When either of these are received, the DCU will switch back to the INCT state to await the next CIO Start command.

CONTROL FLIP-FLOPS

These flip-flops are very simple and are used to remember that certain events have occurred; such as commands from the CPU, device states or errors. The number of flip-flops needed by a DCU will depend on the type of device, therefore only two are shown in figure 20.

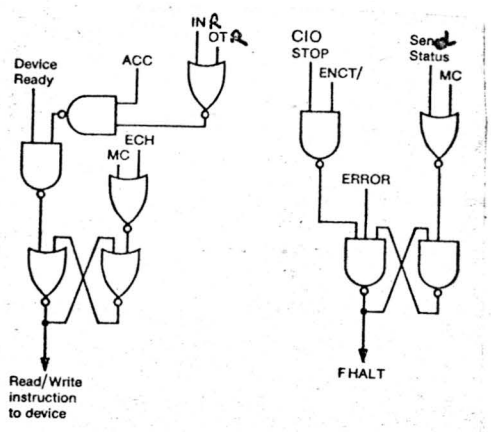



Figure 20 Typical Control Flip-Flops

## NOTES FOR DESIGNERS

All the circuits given in the previous section use logic modules producing an active low output when all inputs are active high. These circutis are based on the standard DCU cards produced by Philips and it should be stressed that they may not suit individual user devices without modification or additional logic. This will also apply to devices using different logic levels. A final aid to user designers is the list of I/O bus pin connections that follows these notes.

## BUS PIN CONNECTIONS

Two connectors - on each card - plug into the bus sockets. The pin configuration of these connectors has been standardized so that any card can be plugged into any bus socket. For the user designing his own control units, adherance to the following table will enable the cards to be plugged directly into the I/O bus sockets.

| | | | |
|---|---|---|---|
| 1A01 | +5 volts | 2A01 | BOF 02/ |
| 1A02 | reserved | 2A02 | EOR |
| 1A03 | BIN 00/ | 2A03 | BOF 01/ |
| 1A04 | BIN 01/ | 2A04 | BOF 00/ |
| 1A05 | BIN 02/ | 2A05 | BOU 14 |
| 1A06 | BIN 03/ | 2A06 | BOU 13 |
| 1A07 | BIN 04/ | 2A07 | BOU 12 |
| 1A08 | BIN 05/ | 2A08 | BOU 09 |
| 1A09 | BIN 06/ | 2A09 | BOU 10 |
| 1A10 | BIN 07/ | 2A10 | BOU 11 |
| 1A11 | ground | 2A11 | BOU 15 |
| 1A12 | reserved | 2A12 | MC/ |
| 1A13 | ACC/ | 2A13 | BAD 03/ |
| 1A14 | ARE/ | 2A14 | DAV/ |
| 1A15 | PIL/ | 2A15 | BAD 04/ |
| 1A16 | BRL1 | 2A16 | BAD 05/ |
| 1A17 | reserved | 2A17 | BOU 08 |
| 1A18 | reserved | 2A18 | BOU 07 |
| 1A19 | reserved | 2A19 | BOU 06 |
| 1A20 | reserved | 2A20 | BOU 05 |
| 1A21 | BIN 08/ | 2A21 | BOU 04 |
| 1A22 | BIN 14/ | 2A22 | BAD 00/ |
| 1A23 | +6 volts | 2A23 | BOU 03 |

| | | | |
|---|---|---|---|
| 1A24 | -6 volts | 2A24 | BAD 01/ |
| 1A25 | -12 volts | 2A25 | BAD 02/ |
| 1A26 | BIN 13/ | 2A26 | BOU 02 |
| 1A27 | BIN 15/ | 2A27 | BOU 01 |
| 1A28 | BIN 09/ | 2A28 | BOU 00 |
| 1A29 | BIN 10/ | 2A29 | +24 volts |
| 1A30 | BIN 11/ | 2A30 | -5 volts |
| 1A31 | BIN 12/ | 2A31 | +5 volts |
| | | | |
| 1B31 | ground | 2B31 | ground |

Note:

The + and -6 volts, the +24 volts, the -12 and 5 volts are not supplied by the CPU power supply. These supplies may be needed by special circuit cards, in which case a separate power supply will also be needed. The pins that are reserved for these supplies do not form part of the I/O bus and are in no way connected to the CPU cards. For example, 2A29 is designated +24 volts and all 2A29 pins of the bus sockets are wired in parallel. If that particular supply is needed by one or more cards, it will be connected to one 2A29 pin and will be available at all bus sockets. However, only the cards that need +24 volts will have their 2A29 pins connected to any circuitry.

DCU TO PERIPHERAL CONNECTIONS

The peripheral to DCU connections are made by twisted pairs of wires. One of the wires is connected to the signal level, the other to ground.

Below is the pin configuration for the DCU cards supplied by Philips for punched tape equipment and the I/O typewriter.

| | | | | |
|---|---|---|---|---|
| ASR card | 5A02 | signal line | | |
| | 5A05 } | ground | | |
| | 5A06 } | | | |
| PTR card | 5A07 | | OPER | (card reader operable) |
| | 5A08 | FWD POC | | (tape forward/pick one card) |

| | | | |
|---|---|---|---|
| | 5A09 | STOP | (stop tape) |
| | 5A10 | CP | (card present) |
| | 5A11 | LTOPER/ | (load tape level operable) |
| | 5A12 | PKT BZS | (sprocket strobe/column strobe) |
| | 5A13 | EOT | (end of tape) |
| | 5A14 | CH7 CH03 | |
| | 5A15 | CH8 CH02 | |
| | 5A16 | CH5 CH05 | |
| | 5A17 | CH6 CH04 | |
| | 5A18 | CH3 CH07 | |
| | 5A19 | CH4 CH06 | (data channels) |
| | 5A20 | CH1 CH09 | |
| | 5A21 | CH2 CH08 | |
| | 5A22 | CH11 | |
| | 5A23 | CH12 | |
| | 5A24 | CH01 | |
| | 5A25 | CH00 | |

5B01 to 5B25 ground

| | | | |
|---|---|---|---|
| PTP card | 5A01 | CH1 | (data channel) |
| | 5A02 | | Punch Instruction |
| | 5A03 | CH9 | (sprocket hole) |
| | 5A04 | CH4 | |
| | 5A05 | CH2 | |
| | 5A06 | CH3 | |
| | 5A07 | CH5 | (Data channels) |
| | 5A08 | CH8 | |
| | 5A09 | CH6 | |
| | 5A10 | CH7 | |
| | 5A11 | Sense Direction (always low to indicate forward) | |
| | 5A12 | Tape Low signal | |
| | 5A13 | Error signal | |
| | 5A21 | Punch Ready signal. | |

Connector pin configurations for all other Philips control units are given in the applicable logic drawings in the P850/55/60 Control Unit Service Manual (publication number 5122 991 1230x).

The DIOS system is a general purpose input/output system that allows any external digital equipment to interface with the P855/P860. The system is connected to the I/O bus in the same manner as a peripheral device control unit and operates on the programmed channel. Three basic DIOS units are available:

P839-150: DIC - four 16-bit input channels.
P839-250: DOC - four 16-bit output channels.
P839-050: DIOC - two 16-bit input channels.
two 16-bit output channels.

In these basic units output channels only are buffered and only DTL and TTL logic levels are provided for at the input and output interfaces. The basic DIOS can be extended with additional circuit card modules:

INIS - two 16-bit input buffers with signal detection and logic level adaptation.
ONIS - two 16-bit output level adaptation circuits.
IIS - two 16-bit input isolation circuits.
OIS - two 16-bit output isolation circuits.

Connections between the basic DIOS circuits - DIC, DOC, and DIOC - and the user devices are made using twisted pair cables to 62-way, male, ELCO connectors (type 6041) on the circuit cards. The circuits match standard TTL or DTL components and have the following typical characteristics:

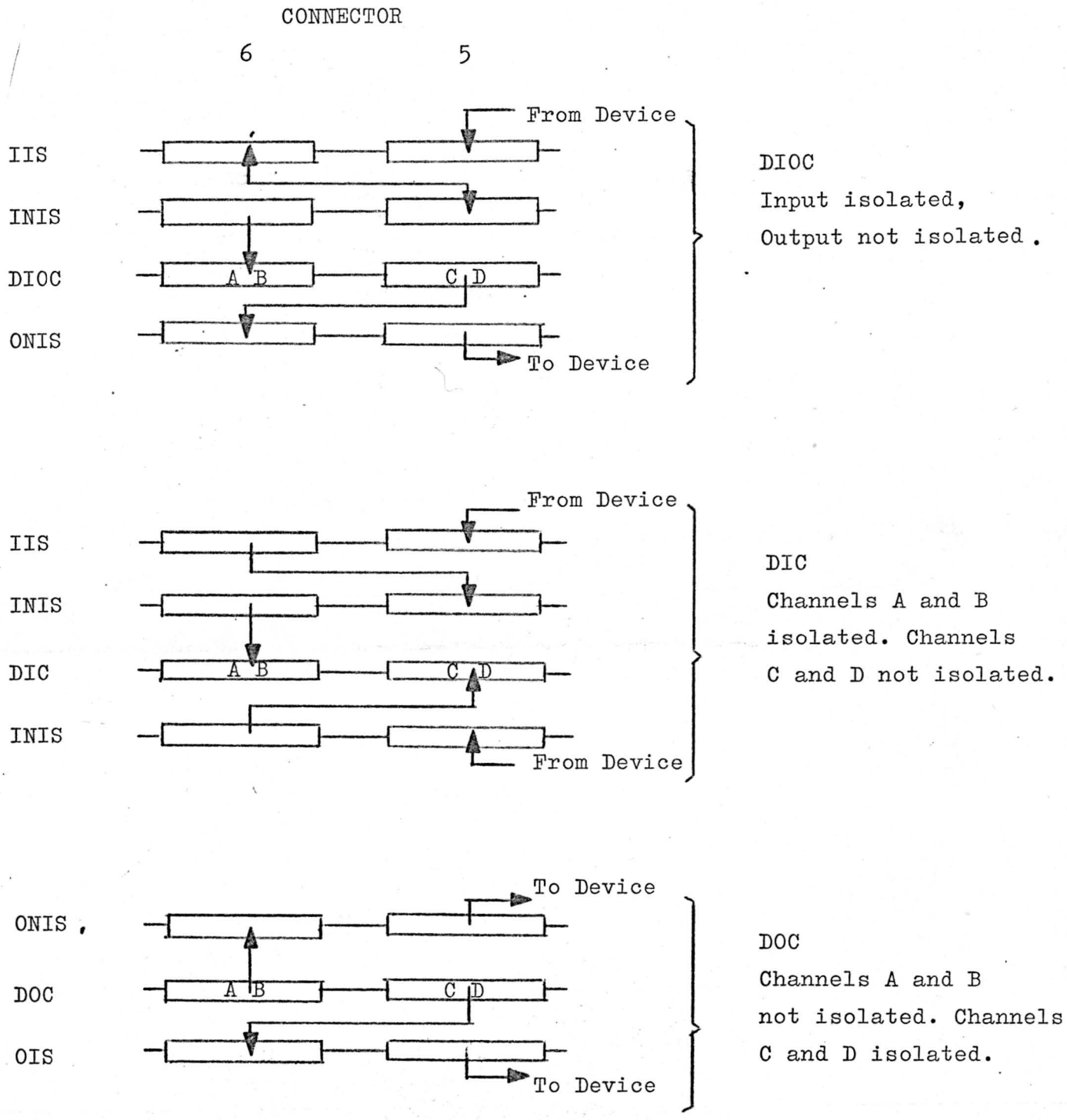
| | | |
|---|---|---|
| Input Low | 0.2V - Fan In | 2 |
| Input High | 3.4V - Fan In | 2 |
| Output Low | 0.2V - Fan Out | 10 |
| Output High | 3.2V - Fan Out | 10 |

Two ELCO connectors on the outer end of each circuit card, numbered 5 and 6 are used for data, call and response signals. The pin configuration for these connectors is given in Table 1 below and the interconnections required when the system is extended with input and output circuit card modules is given in Figure 22.

| Signal | | Connector & Pin Number | | | |
|---|---|---|---|---|---|
| | | 6 | | 5 | |
| | Channel A | A | B | C | D |
| DATA | 00 | 6A25 | 6B25 | 5A07 | 5B07 |
| | 01 | 6A24 | 6B24 | 5A08 | 5B08 |
| | 02 | 6A23 | 6B23 | 5A09 | 5B09 |
| | 03 | 6A22 | 6B22 | 5A10 | 5B10 |
| | 04 | 6A21 | 6B21 | 5A11 | 5B11 |
| | 05 | 6A20 | 6B20 | 5A12 | 5B12 |
| | 06 | 6A19 | 6B19 | 5A13 | 5B13 |
| | 07 | 6A18 | 6B18 | 5A14 | 5B14 |
| | 08 | 6A14 | 6B14 | 5A18 | 5B18 |
| | 09 | 6A13 | 6B13 | 5A19 | 5B19 |
| | 10 | 6A12 | 6B12 | 5A20 | 5B20 |
| | 11 | 6A11 | 6B11 | 5A21 | 5B21 |
| | 12 | 6A10 | 6B10 | 5A22 | 5B25 |
| | 13 | 6A09 | 6B09 | 5A23 | 5B24 |
| | 14 | 6A08 | 6B08 | 5A24 | 5B23 |
| | 15 | 6A07 | 6B07 | 5A25 | 5B22 |
| | CAL | 6A16 | 6A05 | 5A16 | 5A05 |
| | OK | 6B16 | 6B05 | 5B16 | 5B05 |
| | Ground | 6A01 | 6B01 | 5A01 | 5B01 |
| | | 6A17 | 6B17 | 5A17 | 5B17 |

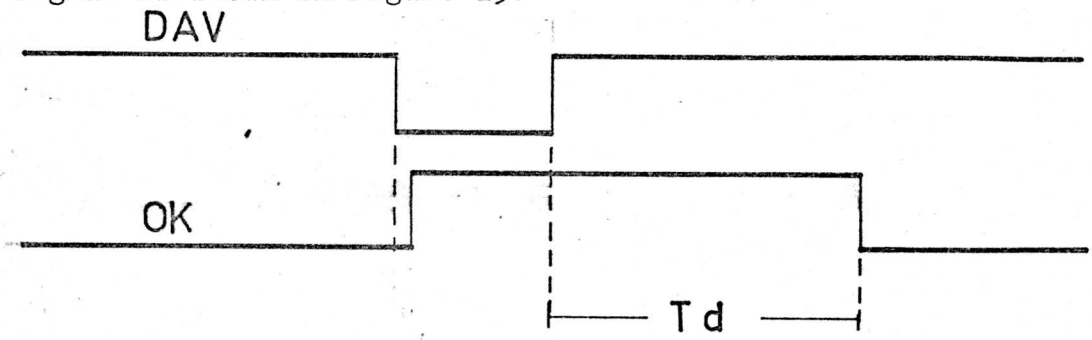
Table 1 Basic DIOS Connector Pin Configuration

Figure 22. DIOC SYSTEM INTERCONNECTION DIAGRAM




NOTE: The OIS card may be connected directly to the output channels of the DIOS. The IIS, however, must always be used in conjunction with the INIS.

A delay circuit is included in the response (OK) signal line for each channel and the duration of this delay, after the DAV signal, can be specified by the user. The delayed signal relative to the DAV signal is shown in Figure 23.




Td is selected in the range 1 to 20 μseconds.

Figure 23 Response signal delay timing

INIS - INPUT CIRCUIT CARD

An INIS card may be used with either a DIC or the input channels of the DIOC module ot buffer and provide change-of-state detection for two 16-bit input channels. The INIS card also includes logic level changing circuits to allow the incoming data and call signals to be matched to the DIOS logic levels and to allow the outgoing response signals to be matched to the user device. The INIS card may be specified in either of two forms: the INIS LT, to match symetrical input logic levels of those of the DIOS, and the INIS HT to match assymetrical input logic levels to those of the DIOS.

Receiving circuits

Characteristics of the receiver circuits in the data and call lines are given below and the receiver circuit is shown in Figure 24.

With the low threshold option signals can be in the range:

High = +2.0V to +48V

Low = -48V to +0.4V

With the high threshold option input signals can be in the range:

High = +7.5V to +48V

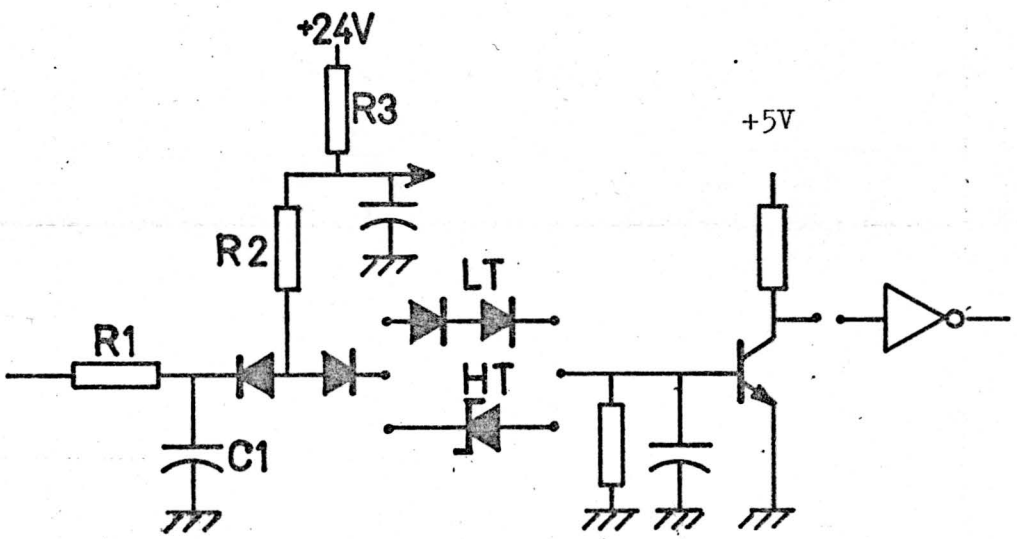
Low = -48V to +4.5V

In both cases Isink at low level $\leq$1.8 mA

Iload at high level $\leq$1.0 uA

Cabling recommendations are as follows:

| Distance | Type | Noise Margin | Input high | Input low |
|---|---|---|---|---|
| <15m | single wires | >3V | INISLT ≥ 5V<br>INISHT ≥ 10.5V | ≤-2.2V<br>≤1.5V |
| 15 - 50 | twisted pair | >4V | INISLT ≥ 6V<br>INISLT ≥ 11.5V | ≤-3.2V<br>≤0.5V |
| >50m | twisted pair | 6V | | |




R1 = 121 Ohms  Zener Diode 6.2V
R2 = 46.4K
R3 = 17.8K
C1 = 330 pF

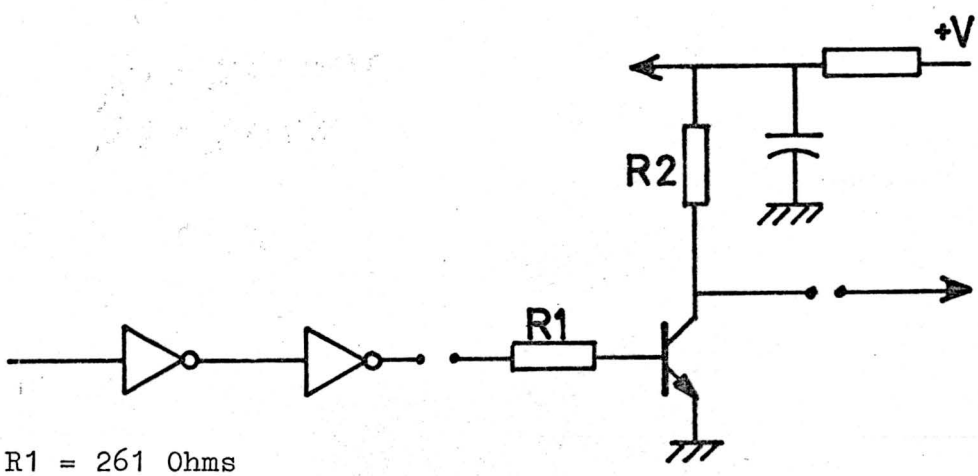
Figure 24. Receiver Circuit

The transmitter circuits used to transmit the response (OK) signal of each channel and the output data bits is shown in Figure 25. The power for the transmitting circuits is supplied by the user and can be adjusted to give output voltages within the range:

High = 2.4V to 48V at 0.1 mA
Low = <0.4V, Imax = 50 mA

The length of the cable connecting the card to the device sets a minimum value on these output voltages as follows:

| Distance | Type | Device Output high |
|---|---|---|
| <15m | single wires | >12V |
| 15 - 50m | twisted pair | >15V |
| >50m | twisted pair | <24V |




R1 = 261 Ohms
R2 = 21.5K

Figure 25. Transmitter Circuit

Connection from the user device is made to connector 5 on the INIS card and connection to the basic DIOS card is made from connector 6 on the INIS card. Connector pin configuration is given in Table 2 below.

| Signal | Connector & Pin Number | | | |
|---|---|---|---|---|
| | DIOS | | DEVICE | |
| | Channel | | Channel | |
| | A | B | A | B |
| DATA 00 | 6A25 | 6B25 | 5A07 | 5B07 |
| 01 | 6A24 | 6B24 | 5A08 | 5B08 |
| 02 | 6A23 | 6B23 | 5A09 | 5B09 |
| 03 | 6A22 | 6B22 | 5A10 | 5B10 |
| 04 | 6A21 | 6B21 | 5A11 | 5B11 |
| 05 | 6A20 | 6B20 | 5A12 | 5B12 |
| 06 | 6A19 | 6B19 | 5A13 | 5B13 |
| 07 | 6A18 | 6B18 | 5A14 | 5B14 |
| 08 | 6A14 | 6B14 | 5A18 | 5B18 |
| 09 | 6A13 | 6B13 | 5A19 | 5B19 |
| 10 | 6A12 | 6B12 | 5A20 | 5B20 |
| 11 | 6A11 | 6B11 | 5A21 | 5B21 |
| 12 | 6A10 | 6B10 | 5A22 | 5B25 |
| 13 | 6A09 | 6B09 | 5A23 | 5B24 |
| 14 | 6A08 | 6B08 | 5A24 | 5B23 |
| 15 | 6A07 | 6B07 | 5A25 | 5B22 |
| CAL | 6A16 | 6A05 | 5A16 | 5A05 |
| OK | 6B16 | 6B05 | 5B16 | 5B05 |
| Ground | 6A01 | 6B01 | 5A01 | 5B01 |
| | 6A17 | 6B17 | 5A17 | 5B17 |
| +24V | | | | 5B02 |
| +V device | | | 5A03 | 5B03 |

Table 2. Connector pin configuration for extender DIOS modules

An ONIS card may be used with either a DOC or the output channels of a DIOC module to allow the outgoing data and response signals to be matched to the user device and to allow the incoming call signals to be matched to the DIOS logical level. The ONIS card may be specified in either of two forms: the ONIS LT, to match symetrical input and call signals to those of the DIOS, and INIS HT to match assymetrical input and call signals to those of the DIOS. The transmitter and receiver circuits are the same as those used on the INIS card and are shown in Figures 25 and 26.

ONIS connector configuration

Connection from the user device is made to connector 5 on the ONIS card and connection to the basic DIOS card is made from connector 6 on the ONIS card. Connector pin configuration is the same as that given for the INIS card in Table 1.

IIS - INPUT ISOLATION CARD

The IIS card provides electrical isolation (± 1K. volt peak) for two 16-bit input channels. Optically coupled isolator circuits on the IIS card between the device interface and DIOS effectively separate the device from the computer system. Each isolator is a sealed unit containing a light emitting gallium arsenide diode and a photo sensitive transistor. The circuit input current drives the diode and light from the diode is detected by the transistor, giving a signal that is amplified in a simple transistor amplifier circuit.
Power for the DIOS side of the isolator circuits is taken from the DIOS supply while power for the device side is taken from the device supply, regulated, in the case of transmitting isolators, in a series regulator circuit.

## Receiver circuits

Figure 26 shows the circuit used for input signal isolators. A two-wire cable, either twisted pair or coaxial, must be used to connect the device. The length of cable is not limited but the noise level delivered at the card must be less than 3 Volts peak.
The device can handle input logic levels in the range:

| | |
|---|---|
| Low | +2V |
| High | between +12V and +48V |
| Iload | 50 mA max. |

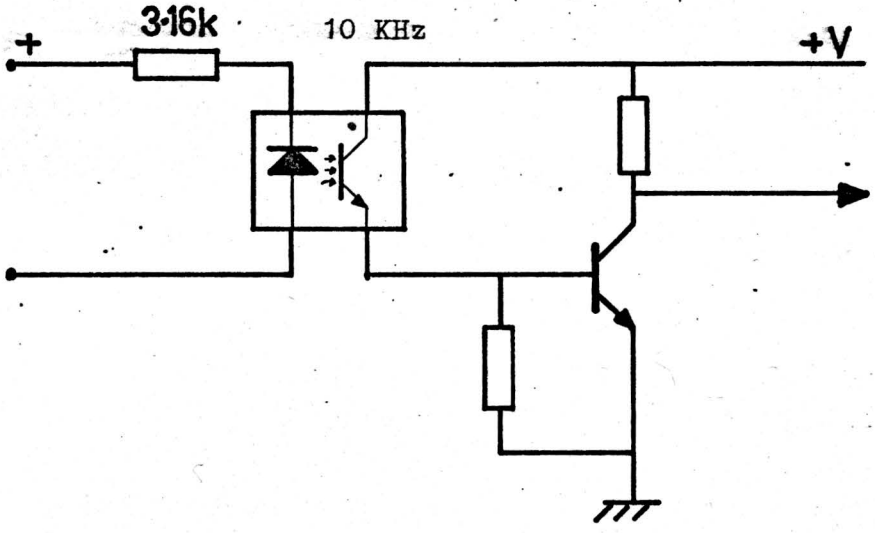



Figure 26 Isolating Receiver Circuit

Transmitter circuits

Figure 27 shows the circuit used for input signal isolators in the response (OK) lines. Power for the output (device) side of the isolator is taken from the device supply after regulation while power for the output driver transistor is taken directly from the device supply without regulation. As the high logic level output of the circuit is close to the supply voltage the high level can be adjusted (by means of supply voltage) by the user to suit his application. The supply voltage must be within the range of +12 to +48 Volts. Output voltages are in the range:

Low +0.4V (Isink 50 mA max.)
High between +12V and +48V (Iload 0.1 mA max.)

IIS Connector Configuration

Connection from the user device is made to connector 5 on the IIS card and connection to the basic DIOS card or to the INIS card is made from connector 6 on the IIS card. Connector pin configuration for the IIS card is shown in Table 2.

OIS - OUTPUT ISOLATOR CARD

The OIS card provides electrical isolation (± 1KV peak) for two 16-bit output channels. Optically coupled isolator circuits on the OIS card between the device interface and DIOS separate the device from the computer system in the same way as that described for the IIS card. The circuits used for transmitter and receivers are described in the IIS paragraph above.

OIS Connector Configuration

Connection from the user device is made to connector 5 on the OIS card and connection to the basic DIOS card or the ONIS card is made from connector 6 on the OIS card. Connector pin configuration for the OIS card is shown in Table 2.

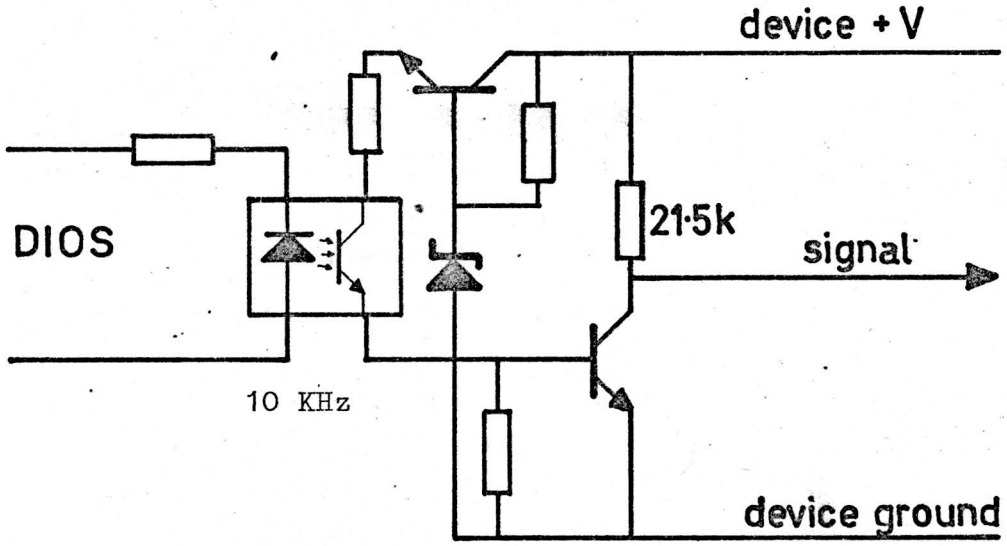



Figure 27 Isolating Transmitter Circuit

## MEMORY INCREMENT DATA BREAK INTERFACE

Memory address lines are available on connectors 5 and 6 (shown in Figure 13) on the outer end of the ALU circuit card. Connector pin configuration is as follows:

| Address line | Connector and Pin number |
|---|---|
| 00 | 5B12 |
| 01 | 5A11 |
| 02 | 5B17 |
| 03 | 5A14 |
| 04 | 5B20 |
| 05 | 5A18 |
| 06 | 5B22 |
| 07 | 5A21 |
| 08 | 6A11 |
| 09 | 6B09 |
| 10 | 6A14 |
| 11 | 6B12 |
| 12 | 6A18 |
| 13 | 6B15 |
| 14 | 6B21 |

P855/P860 Interface Manual

Name _____

Company _____

Department _____

Address _____

City _____ Country _____

Telephone number _____ Ext. _____

Comments or Suggestions