# 801 I/O SUBSYSTEM DEFINITION.


## Abstract

801 I/O Structure, Program Architecture and Functional Characteristics are defined. This definition is the result of the first complete look at 801 I/O.



Frank Carrubba x-2465
Albert Chang x-2281
Alain Couder x-2267
Richard Oehler x-1024


January 16, 1976

## Previous Versions

November 25, 1975
December 19, 1975

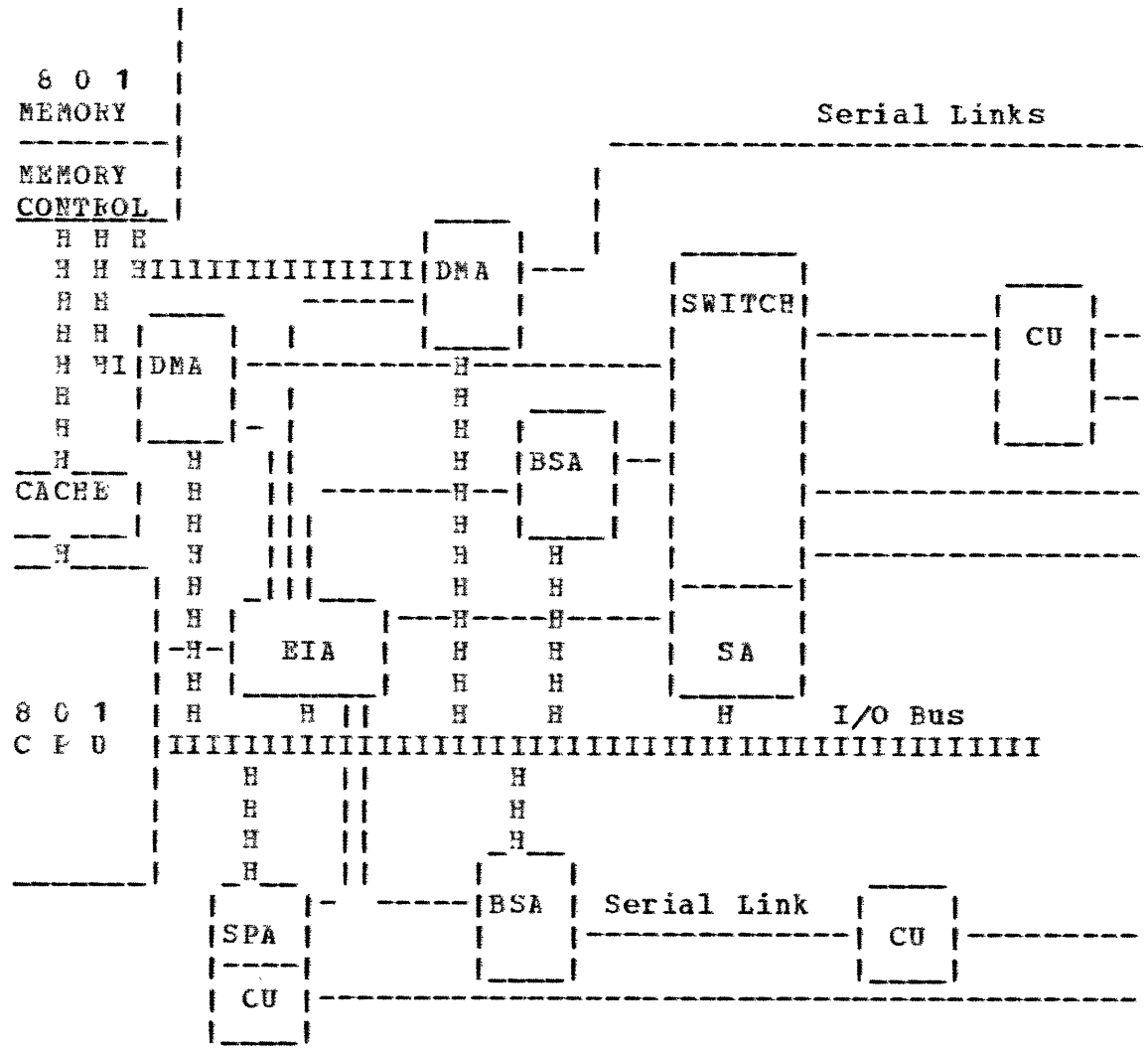
# CONTENTS

INTRODUCTION.

801 I/O Structure, Program Architecture and Functional Characteristics are defined. This definition is the result of the first complete look at 801 I/O. This document is distributed for review and comment. If you have any comments, criticisms or suggestions please contact the authors. The next release of this document is planned for the end of February, 1976.

I/O STRUCTURE OVERVIEW.

The following figure gives the general structure of the 801 I/O subsystem.




The I/O components are described as:

- I/O Bus is a parallel bus which moves control and data to and from the 801 under control of 801 instructions.
- A: Adapter is any unit attached to the I/O Bus and addressed by 801 I/O instructions.
- BSA: Bus Serial Adapter is an adapter which connects a Control Unit (CU) to the I/O Bus through a Serial Link.
- EIA: External Interrupt Adapter collects all the interrupt requests from the BSAs, Switches and other

external devices and sends an external interrupt signal to the 801 when appropriate.

- Serial Link is used to attach a Control Unit either to a BSA or to a DMA. The Serial Link allows the attachment of Control Units over a far greater distance than the I/O Bus.

- CU: Control Unit is any box which controls one or several devices. It may be connected by the Serial Link to a BSA or DMA (perhaps through a Switch) or connected locally to a Special Purpose Adapter (SPA).

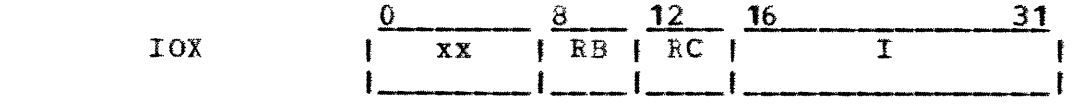
- Switch is a cross point switch of Serial Links. It can be inserted in any Serial Link. All connections are bidirectional. The Switch is transparent with respect to the Serial Link. An integral part of the Switch is the Switch Adapter (SA). The Switch is addressed and controlled on the I/O Bus thru the SA.

- DMA: Direct Memory Access is an adapter which transfers blocks of data between the CU and memory without using the I/O Bus or CPU. Before the first block transfer can take place, initial conditions must first be established in the CU and DMA using Programmed I/O. At the end of the transfer, an interrupt request is presented to the External Interrupt Adapter. The DMA contains full BSA function.

- SPA: Special Purpose Adapter is any adapter that is not a BSA, SA or EIA. It is designed to meet a specific need. An example of such an adapter is the UC Bus Adapter. This adapter connects the UC Bus to the 801's I/O Bus, thereby permitting indirect attachment of UC Bus attached devices to the 801.

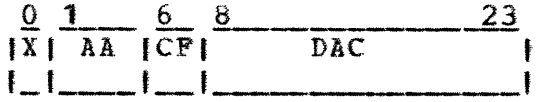
I/O INSTRUCTIONS.

The 801 I/O instructions are I/O Read (IOR) and I/O Write (IOW). Both instructions have the D instruction format.




Associated with both instructions is a 24 bit address/command field equal to the result obtained by replacing the high order bit of the 24 bit sum of I and 0/(RC) with 0 (IOR) or 1 (IOW).

The address/command field has the following format:




The first bit represents a Read or a Write.

The adapter address (AA) is a 5 bit field used to select an adapter on the I/O Bus.

The 2 bit control field (CF) and the 16 bit device address/command field (DAC) contain control information for the selected adapter. In general, the use of these fields is adapter, control unit and device dependent.

An I/O instruction is executed jointly by the CPU and an adapter. At the time the instruction is initiated by the CPU, the selected adapter may or may not be able to execute the I/O operation (within the time out period of the I/O Bus - see below). If it is not able to execute the instruction, no data transfer between register RB and the adapter takes place and the I/O Busy bit in the Condition Register is set to a 1. Otherwise, the I/O Busy bit is set to a zero and the lower order 2 bytes of RB are transferred to the the adapter (IOW) or 2 bytes from the adapter are placed in the lower order 2 bytes of RB and the higer order byte is set to zero (IOR).

The above description of the operation of I/O instructions assumed no detectable hardware or programming errors. Otherwise, an I/O Check interrupt occurs at the completion of the instruction and the contents of RB, RC and the value of the I/O Busy bit are as they were before the instruction was executed.

Note: Some adapters, because of the way they are designed, are never busy at the time an I/O instruction is initiated by the CPU, or else may never be busy for I/O instructions with certain values in the CF and DAC fields. In these

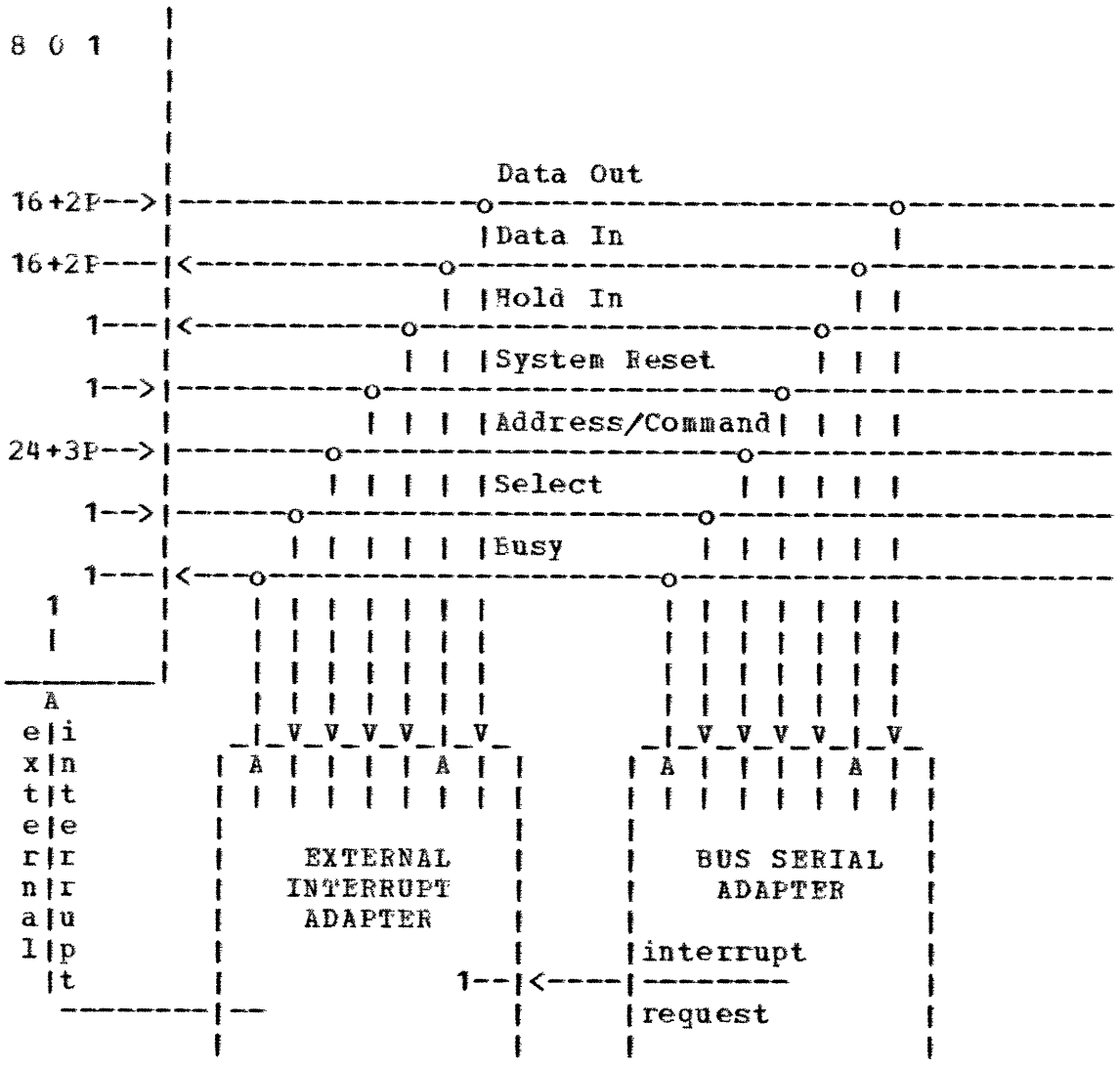
cases, it is not necessary to test by software the I/O Busy bit for the sucessful completion of the data transfer part of the I/O instruction.

I/O BUS.

The I/O Bus is described in terms of its lines and how they are sequenced.

LINES DEFINITION.

The following figure shows the logical lines between the 801 CPU and Adapters.




The above lines have the following use:

* Address/Command lines: They carry the address/command 24 bits which are generated by an I/O instruction (see above in I/O INSTRUCTIONS).

- Select line: This line is raised when the Address/Command and Data Out (if Write) lines become valid. It falls when the I/O instruction ends.

- Data Out lines:They carry two bytes of data from the 801 to the selected adapter when the instruction is IOW.

- Data In lines: They carry two bytes of data from the selected adapter to the 801 when the instruction is IOR.

- Hold In line: This line is raised by the selected adapter when it has (1) completed the IOW instruction or (2) raised the Data In lines to complete the IOR instruction or (3) raised the Busy line.

- Busy line: This line is raised by an adapter to signal that it cannot complete the data transfer part of the I/O instruction (see above in I/O INSTRUCTIONS). Except when I/O Check occurs, the 801 processor always sets the I/O Busy bit to the state of this line during an I/O instruction.

- I/O System Reset line: Raising this line causes a reset function to be performed by all adapters connected to the I/O Bus. The function performed will be adapter dependent and the effect of the reset on a CU or device connected to an adapter may also further depend on their type. I/O System Reset is initiated manually from the operator panel by pressing System Reset, Power On or IPL.

- interrupt request line: Each adapter may connect to the External Interrupt Adapter directly, presenting to it an interrupt request. There may also be interrupt request lines from non adapter sources.

- external interrupt line: This line is raised by the External Interrupt Adapter when some condition in the adapter indicates that an interrupt is required. There is only one external interrupt line per 801 CPU.

Note: Data Out, Data In and Address/Command lines has one additional parity line for each group of 8 data lines. When a parity error is detected by the 801 it forces completion of the instruction and causes an I/O Check Interrupt. When an adapter detects a parity error it does not raise Hold In thereby causing a time out (which see above).

Note: I/O System Reset line causes each adapter to set its Status Register to zero, to cancel an interrupt request and

to send a Reset Command to its CU or device (if any).

Note: The implementation may merge the Data In and the Data Out lines into 16 bidirectional Data lines.

SEQUENCE OF OPERATION.

The time required for the execution of I/O instructions is variable and depends on the speed of the adapter (see figure below).

In the machine cycle in which the I/O instruction is initiated, the CPU raises the Select line after putting data on the Address/Command lines and Data Out lines (if IOW).

If the selected adapter is busy and if the result of the instruction should be to set the I/O Busy bit in the Condition Register to 1, then the Busy line will be set to 1 before the adapter raises Hold In. Otherwise, Hold In signifies that the adapter has accepted the data on Data Out (if IOW) or has placed data on Data In (if IOR).

The CPU uses the data on the Busy line and Data In as appropriate for the instruction and then drops the Select line. Finally, the adapter drops Hold In, completing the I/O instruction.

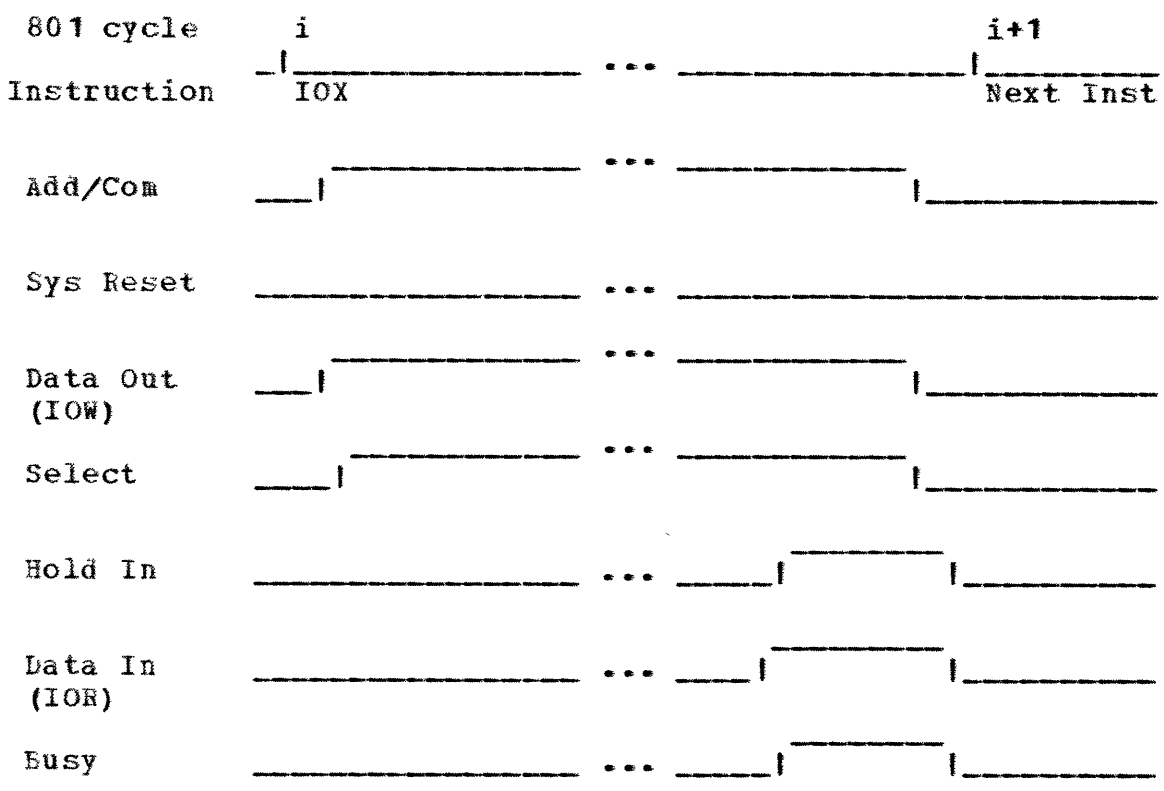
Note: A timer is started by the CPU at the beginning of all I/O instructions. The failure of an adapter to raise Hold In before the time out period has elapsed terminates the instruction and causes an I/O Check interrupt (see definition in ERROR IN I/O SUBSYSTEM). The time out period will be on the order of one micro-second.

Note: To avoid a potential ambiguity associated with the adapter responding with Hold In just as the time out period has elapsed, all adapters should be designed to respond in somewhat less time than the time out period.

Note: Deskewing of the data on the I/O Bus will be the CPU's responsibility.

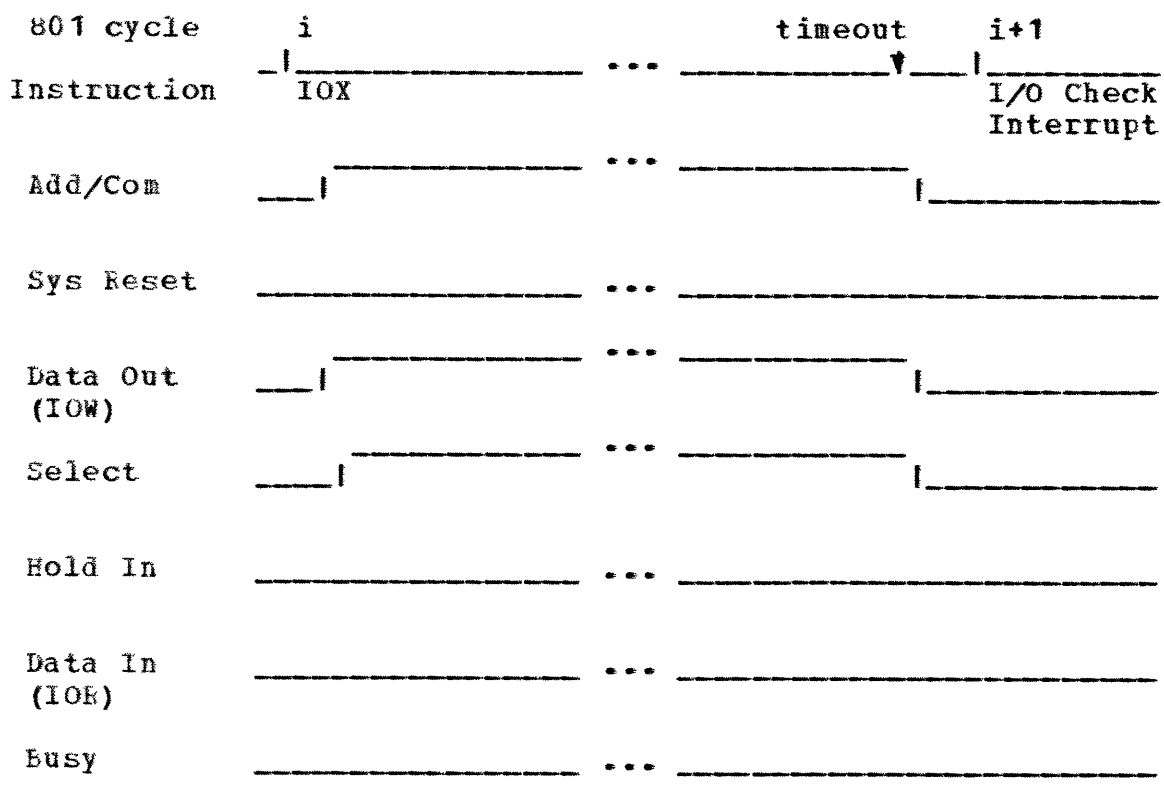
Case 1 : Adapter responds within time out period.

The diagram below shows the sequence of operations in the case where the adapter responds within the time out period of an IOX instruction.




Case 2 : Adapter does not respond within time out period.

The diagram below shows the sequence of operations in the case where the adapter failed to respond within time out period.




ADAPTER VS. CPU SPEED.

The first case above demonstrated the use of the Hold In and Busy lines. These lines together with the timer permit some flexibility in adapter design in that some adapters can legitimately take longer to respond to an I/O instruction than the minimum possible response time. Such a situation can arise when the adapter determines that its busy condition will be cleared in a short period of time and decides to wait until it is before responding with Hold In. Likewise, the adapter may contain local memory that is slower than the cache but faster than main memory, and the adapter stores or fetches from this local memory without using the busy mechanism at all. Obviously, the maximum response time must be less than the time out value. The 801 waits for Hold In to come up and cannot take an interrupt other than I/O Check (time out). The actual response times implemented are functions of the 801 machine speed, technologies and adapter requirements.

ERROR IN THE I/O SUBSYSTEM.

There are three class of errors in the I/O subsystem. The first class of error is a hardware malfunction of the bus itself, which is recognized by the 801 or the adapter or both. The second class of errors are invalid adapter addresses or DACs due to program errors. The third class of errors occur in or between the BSA, DMA, CU or device. These errors do not directly affect the I/O Bus and are not detected as part of the I/O instruction.

Sufficient status is available so that the class of error and its probable cause can be diagnosed and recovered from where possible.

CLASS 1 - HARDWARE ERROR ON THE BUS.

This type of error is detected through the hardware parity checker of the CPU or the Adapter or both. As always, an even multiple of errors will not be detected. Only an odd multiple of errors is detected. There is no hardware correction facility.

If the CPU detects a parity error, it sets a parity check bit and causes an I/O Check interrupt to occur as it completes the I/O instruction (this bit is made available as part of the status of the I/O Check interrupt). Thus, programming can determine the nature and extent of the problem perhaps even localizing the problem to a particular adapter. In general, solid I/O Bus parity errors offer little hope for recovery.

If an adapter detects the parity error, it sets the parity error bit of its local status register and does not respond to the I/O instruction. This causes an I/O Check Interrupt because of time out (this time a bit is set as part of the status of the I/O Check interrupt to indicate time out). Programming will assume the adapter is in difficulty and attempt to read the adapter status. This may or may not work depending on the extent of the error. If it does, the problem can then be localized. As before, a solid I/O Bus parity error offers little hope of recovery.

CLASS 2 - INVALID ADDRESSES OR COMMANDS.

This class of errors are typically due to program errors if we exclude undetected hardware errors.

In the case of an invalid address, no adapter is selected on the bus. So the Hold In signal does not come back. As

before, time out occurs with the resultant I/O Check interrupt.

In the case of an invalid command, the error is detected by the selected adapter which sets a local invalid status and does not respond to the I/O instruction causing an I/O Check interrupt. The setting of the local invalid status bit distinguishes this error from the invalid adapter address error.

CLASS 3 - ERRORS NOT DETECTED AS PART OF I/O INSTRUCTIONS.

This class of errors does not cause an immediate I/O Check interrupt. Instead, the adapter when it encounters an error, sets the appropriate local error status and raise an interrupt request to the External Interrupt Adapter. This will eventually lead to an I/O Check or external interrupt or to programming discovering the error on the next IOR of the adapter's status register. If at the time of the error, the interrupt for this adapter is allowed, then the external interrupt will occur. If the interrupt is not allowed, then the external interrupt will occur when the adapter's interrupt is allowed or an I/O Check interrupt will occur if next I/O instruction requires a status register whose contents are zero or programming will recognise the error in the status register, whichever comes first. Since the program that requested the I/O that suffered the error may not be the program that was interrupted or the one that discovered the error, the error must be asynchronously communicated to it. While difficult, this is a normal programming problem and the usual techniques will be used to recover.

## ADAPTER FUNCTIONAL DESCRIPTION.

The adapter's function is to attach CUs and devices to the I/O bus. This function can be more or less complex depending upon the type of CU and/or device which is to be served. The following describes the common function to be implemented in all adapters attaching to the 801 I/O Bus.

## FUNCTIONS COMMON TO ALL ADAPTERS.

Adapters will vary in functional capability but all will implement a Status Register and the functions described below.

### Status Register.

The 16 bit Status Register can be read or written with IOR or IOW, respectively, with a CF = 00 and DAC = 0000 (both bits are always given for CF, full hexadecimal of 2 bytes is always given for DAC). The IOR and IOW are unconditionally executed by the adapter even though there may be a pending error condition in the adapter and the adapter never returns Busy to these instructions. The following bits have been reserved:

- Bit 0 : interrupt request latch.
- Bit 1 : bus parity error.
- Bit 2 : invalid command.
- Bit 3 : busy.
- Bit 4-15 : adapter dependent.

### Reset Interrupt Request Latch.

The Interrupt Request latch is set to zero by an IOW instruction with CF = 00 and DAC = 0001. The contents of RB (abbreviated (RB)) put on Data Out are ignored by the adapter, and the adapter is never busy to this IOW.

Adapter Reset.

The adapter performs its reset function when the I/O System Reset line is raised or when executing an IOW instruction with CF = 00 and DAC = 0002. The contents of RB put on Data Out are ignored by the adapter, and the adapter is never busy to this IOW.

BUS SERIAL ADAPTER.

The Bus Serial Adapter (BSA) is an adapter designed to permit Programmed I/O with a CU connected to the BSA over a serial transmission link.

DESCRIPTION.

Provided the BSA is not busy, an IOW with CF = 01 results in the transmission of the DAC field and (RB) to the CU. This transmission occurs after the completion of the IOW and CPU execution proceeds while the BSA is transmitting to the CU. The CU interprets the DAC and returns either a one byte acknowledgement if the DAC field specified a Write to CU or else it returns 2 bytes if the DAC implies a Read from CU. The BSA stores the 1 or 2 bytes in its Data Received Register which may then be read by an IOR instruction.

In addition, the BSA permits the transmission of one byte control commands to the CU.

The BSA registers and commands are defined below:

BSA Status Register.


As for all adapters, the BSA Status Register is read or written by I/O instructions with a CF = 00 and a DAC = 0000. The significance of the bits in the BSA Status Register are:

- bit 0 : interrupt request latch.
- bit 1 : bus parity error.
- bit 2 : invalid command.
- bit 3 : busy bit.
- bit 4 : transmission error.
- bit 5 : time out.
- bits 6 - 15 : reserved.

Bit 0 is set when an error condition is detected after the completion of the IOR or IOW instruction.

The busy bit is set to one by IOW instructions which cause transmission on the Serial Link and is reset to zero when the CU reply has been received without transmission error.

A time out period limits the time the BSA will wait for the CU reply (e.g. 30 microseconds). If the time out occurs, bits 0 and 5 are set to 1. Bits 0 and 4 are set to 1 if a Serial Link transmission error is detected.

Write to CU.

An IOW with a CF = 01 results in the transmission of the DAC field and (RB) to the CU provided the BSA Status Register is zero when the instruction is issued. If the busy bit is 1 but all other bits in the BSA Status Register are zero, the I/O Busy bit in the Condition register is set to 1 and no data transfer occurs. If bit 0, 4 or 5 are 1, the adapter causes an I/O Check interrupt by not raising Hold In. The busy bit is set to 1 by the instruction and reset when the CU reply is received.

Write Control.

An IOW with CF = 00 and DAC = 80xx causes the single byte xx to be transmitted to the CU provided the BSA Status Register is zero when the instruction is issued. Otherwise, the instruction is identical to Write to CU above.

Data Received Register.


This register will contain the one or two byte CU reply to IOW specifing a Write to CU or a Write Control. As specidied in the DAC or CONTROL transmitted to the CU, the CU replys with one or two bytes unless it detects a transmission error in which case its reply is always one byte. The BSA always stores what it receives; it does not know what to expect. Programming must know the size of the reply; except for errors, no indication is given. The Write to CU or Write Control that caused the CU to respond with two bytes can be thought of as a Read from CU.

The register may be read with a CF = 01 and DAC = 0000. The I/O Busy bit in the Condition Register is set to the value of the busy bit in the BSA Status Register; no data transfer occurs if its value is one.

The adapter causes an I/O Check interrupt if any bit other than the busy bit in the BSA Status Register is non zero when this instruction is executed.

BSA Reset.

A BSA reset is initiated by the system reset line or by an IOW with CF = 00 and DAC = 0002. The effects of the reset are:

01. Any serial transmission in progress at the time the instruction is issued is aborted and the BSA assumes control of the Serial Link.

02. The reset control byte is sent to the CU (no response is expected from the CU). The Serial Link is left so that the BSA is the next to send.

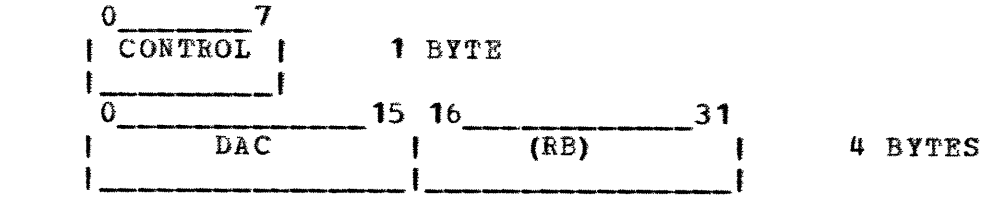
03. The BSA Status Register is set to zero (including the interrupt request latch).

BSA/CU SERIAL LINK PROTOCOL.

Except for initialization and certain errors, the Serial Link protocol is alternating half duplex. The BSA always is the next to send when the link is quiet. The CU is next to send in response to a BSA request.

It is important to remember that the physical link may well be full duplex.

The BSA transmits either one or four bytes of data to the CU in the two formats shown below:




Each 8 bit byte is followed by its parity bit.

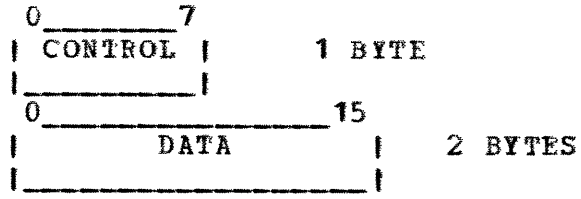
This CONTROL byte is the second byte of the DAC when the I/O instruction is IOW with CF = 00 and DAC = 80xx (Write Control).

One byte is sent as a result of the Write Control instruction or an I/O adapter reset operation. While the encoding of control information into this byte may be CU dependent, certain byte values have been assigned for the DMA/CU Serial Link protcol (see below) and future code point assignments should respect these assignments. In particular, RESET has the encoding of 76.

The four byte format accomadates the instruction Write to CU.

As part of the alternating half duplex protocol, the BSA starts a timer after completing each of its transmissions. The time it will wait for a CU reply is limited by this time out period. If the time out period elapse, bits 0 and 5 of the BSA Status Register are set and the standard time out actions occur.

The CU in response to a BSA message transmits either one or two bytes of data in the two formats shown below:




Each 8 bit byte is followed by its parity bit.

The CU responds with two bytes of data when the CONTROL byte or DAC field implies a Read from CU, provided no error was detected in the received message.

The one byte format is sent to either acknowledge (40) the correct reception of data when no reply is necessary or whenever a transmission error was detected (80). The BSA will store the one or two byte reply in the Data Received Register. One byte replies are checked by the BSA for the transmission error code (80), If the reply is two bytes it only checks the parity bits associated with the bytes.

Note: The engineering implementation must have a mechanism which allow the BSA and CU receivers to (1) accomplish bit synchronism and (2) determine which message format has been transmitted. One possible implementation is to transmit a bit clock signal along with the data and to mark the begining and end of messages by dropping clock bits.

DIRECT MEMORY ATTACHMENT.

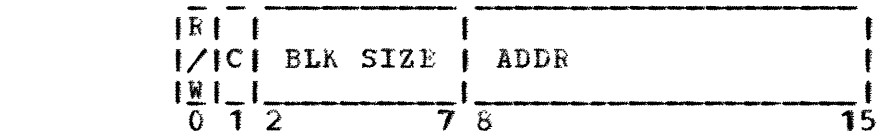
The Direct Memory Access (DMA) is used to transfer data directly between the main memory and a control unit. Before the data transfer begins, the DMA and CU must be appropriately initialized by the CPU via Programmed I/O. A start command is then issued to the DMA and the data transfer proceeds without CPU action. The data end condition is recognized by either the CU or the DMA but the completion of the data transfer is signalled to the CPU by a DMA interrupt. The CU may also signal a subsequent device end by means of an interrupt.

The DMA as presently defined includes the BSA capability as a subset of its capabilities. Except for invalid commands, a DMA and a BSA are interchangeable so far as programs designed for BSA operation are concerned. It should be noted that in system configurations that do not include a switch, the inclusion of BSA capability is necessary, whereas in switched configurations it is possible to leave it out.

DESCRIPTION.

In what follows, only the DMA functions beyond those in a BSA are described. I/O instructions addressing these registers always complete the data transfer part of the instructions and always set the busy bit to zero. The DMA contains the registers and functions defined below:

Control Register.




The Control Register may be read or written by IOR or IOW with CF = 00 and DAC = 0003.

The R/W bit determines the direction of the data transfer: R is to the DMA, W is from the DMA. The transfer of data on the Serial Link occurs in blocks of 1,2,4,8,16 or 32 bytes. Bits 2-7 select one of these six block sizes: bit 2 = 1 corresponds to a block size of 1, bit 2 = 1 corresponds to 2, etc. One and only one of these 6 bit must be set to 1, C is the chaining bit. When C = 1, the CU will remain

connected to the DMA and will not generate an additional interrupt at the normal end of the operation if it has already generated one at the end of the data transfer.

Address Register.


Byte 1 of the Control Register concatenated on the right of the Address Register specify the location of the first byte in main memory involved in the data transfer. This 24 bit address must have its low order 5 bits set to zero (viz. I/O transfers must start on a cache line boundary). Subsequent bytes are used in ascending order of address from the original. As each block is transfered, this address is incremented by the number of useful bytes transferred. If the last transfer (because the count has gone to zero or an error occurs) of a read is less than a full cache line, a full line is placed in memory with the useless bytes set to zero.

Except for errors, the exact count of bytes is always transfered for a write. A full line is always fetched from memory even if it is not fully used because the count was not a multiple of the cache line size. As with all writes, these unused bytes are not changed in memory.

The Address Register may be read or written by IOR or IOW with CF = 00 and DAC = 0004.

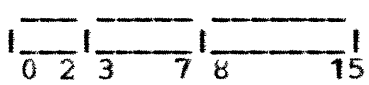
Count Register.


The Count Register may be read or written by IOR or IOW with CF = 00 and DAC = 0005.

This register is initialized at the beginning of an operation with the number of bytes to be transferred. On write operations after each block is sent to the CU, the count is decremented by the number of useful bytes in the block. The number of useful bytes can be less than the blocksize in the case when the initial count is not

divisible by the blocksize. In this case, the last block has unused bytes. The CU is informed of the number of such bytes by the residual count in the control field associated with the Serial Link protocol (see below). On read operations, after each block transfer between the DMA and main memory, the count is decremented by the number of useful bytes in the block. (32 bytes). The number of useful bytes in the last 32 bytes can be less than 32 in the case when the intial count is not divisible by 32 or when the CU recognizes the data end condition before the DMA count has gone to zero.

CU Control Register.




The last control byte sent by the CU to the DMA is saved in the CU Control Register.

Bits 0 - 2 are related to the Serial Link protocol and are defined below.

On normal ending conditions, the last control byte will specify in bits 3 - 7 the residual count, an integer between 0 and 31. On a write operation, the residual count is the number of useful bytes in the last block sent to the CU that were not used by the CU. On a read, the residual count is the number of bytes in the last block sent to the DMA that are not useful.

Bits 8 - 15 are not used and always set to zero.

The CU Control Register can be read or written by IOR or IOW with CF = 00 and DAC = 0006.

DMA Status Register.


The DMA Status Register can be read or written by IOR or IOW with CF = 00 and DAC = 0000.

The DMA Status Register bits have the following meaning:

- Bit 0 : interrupt request latch.
- Bit 1 : bus parity error.
- Bit 2 : invalid command.
- Bit 3 : busy.
- bit 4 : transmission error.
- Bit 5 : time out.
- Bit 6 : incorrect length indicator.
- bit 7 : check end.
- bit 8 : working.
- bits 9 - 15 : reserved.

For DMA operations that are same as BSA operations, bits 0 to 5 are the same as described above in the BSA section. Except for the busy bit, these bits are also effected by exclusive DMA operations as described below:

Bit 0 is set at the end of DMA data transfers and when errors are detected. The data transfer is terminated when errors are detected and the appropriate bits in the DMA Status Register are set to indicate the type of error.

Bit 4 is set when transmission errors on the Serial Link are detected. The DMA signals the CU if it detects the error and vice-versa.

Bit 5 is set when the time out period associated with the Serial Link protocol elapses. This time out has a different value than the BSA/CU time out.

The incorrect length indicator is set on a read operation when the number of bytes sent by the CU is greater than the initial count.

Check end is set whenever any kind of error not described above is detected.

Bit 8 is set to 1 by a sucessfully executed DMA Start and is reset to zero at the end of the data transfer operation.

DMA Reset.

A DMA Reset is initiated by the system reset line or by an IOW with CF = 00 and DAC = 0002. The effects of the reset are:

01. Any serial transmission in progress at the time the instruction is issued is aborted and the DMA assumes control of the Serial Link.

02. The RESET control byte is sent to the CU (no response is expected from the CU). The Serial Link is left so that the DMA is the next to send.

03. The DMA Status Register is set to zero (including the interrupt request latch).

DMA Start

An IOW with CF = 00 and DAC = 0007 causes a GO AHEAD or a GO AHEAD WITH CHAINING control byte (depending on the value of the C bit in the Control Register) to be sent to the CU and the data transfer between main memory and the CU to begin. The contents of RB are ignored. If the DMA is busy or working, this IOW completes with the I/O Busy bit set but no other action is taken.

DMA Halt

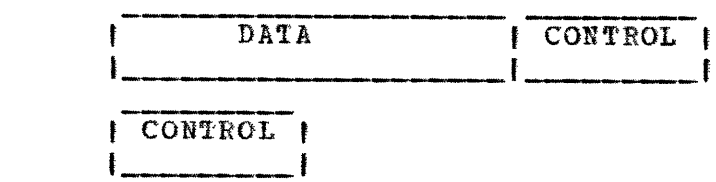
An IOW with CF = 00 and DAC = 0008 cause a HALT control byte to be sent to the CU and the data transfer is terminated after the CU acknowledgement of the HALT control byte. The interrupt request latch is then set.

If bit 8 in the DMA Status Register is zero when the IOW is issued by the CPU, the interrupt request latch is set but no other action is taken.

DMA/CU SERIAL LINK PROTOCOL.

A DMA/CU Serial Link is used to transfer blocks of data between main memory and a CU. Except for certain error recovery and start up procedures, the DMA/CU Serial Link operates with an alternating half duplex protocol. Data is sent on the link in paired transmissions; the DMA sends data and then the CU responds.

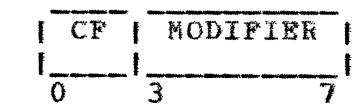
The data formats used by the DMA and the CU are the same and are shown below:




Each 8 bit byte has 1 parity bit associated with it in either format.

DATA in the first format may have lengths of 1, 2, 4, 8, 16 or 32 bytes. However for any particular DMA/CU connection, the block length will be constant for all transfers in either direction.

CONTROL has the following format:




The following command fields (CF) are defined:

- 001 DATA END
- 010 ACKNOWLEDGE TRANSMISSION
- 100 TRANSMISSION ERROR
- 111 CONTROL PREFIX

The modifier for the first three commands represent the residual count, an integer between 0 and 31 which is the number of unused bytes in the last block of data transmitted or received.

For the CONTROL PREFIX 111, the following modifiers are defined:

- 11000 HALT
- 01100 RESET
- 00110 GO AHEAD
- 10100 GO AHEAD WITH CHAINING

DATA END is a signal that the normal end of the data transfer operation has been detected, and is sent in both transmission formats. The HALT command is sent to the CU

when an error other than a transmission error is detected by the DMA hardware during a data transfer operation. GO AHEAD and GO AHEAD WITH CHAINING signal that the sender is ready to proceed further with the data transfer operation, and are sent in both transmission formats.

As a part of the alternating half duplex protocol, the DMA starts a timer after completing each of its transmissions. The time it will wait for a CU reply is limited by this time out period. If the time out period elapses, bits 0 and 5 of the DMA Status Register are set and the standard time out actions occur.

Transfer of data from DMA to CU (writing).

First the 801 initializes the DMA and the CU (see DMA description), establishes the path, and sends one of the GO AHEAD commands to the DMA. The following sequence takes place on the Serial Link:

| | DMA sends: | CU sends: |
|---|---|---|
| 1 | GO AHEAD/(chaining) | . |
| 2 | . | ACKNOWLEDGE TRANSMISSION |
| 3 | Data/GO AHEAD | . |
| 4 | . | ACKNOWLEDGE TRANSMISSION |
| | ... | ... |
| | If DMA count=0 then | |
| n | Data/DATA END(res count) | . |
| n+1 | . | ACKNOWLEDGE TRANSMISSION |
| | If DMA count>0 & CU count=0 then | |
| n | Data/GO AHEAD | . |
| n+1 | . | DATA END(res count) |

Note: If the first GO AHEAD has the chaining option, at the end of this transfer the CU must stay connected waiting for a GO AHEAD which will start another transfer.

At the end of a transfer the DMA raises an interrupt to the 801. The CU does not raise an interrupt if the chaining option has been used when the device end condition occurs. If the chaining option was not used then the CU behavior is CU dependent. The CU always keeps the ability to signal errors other than transmission errors by raising an interrupt to the 801.

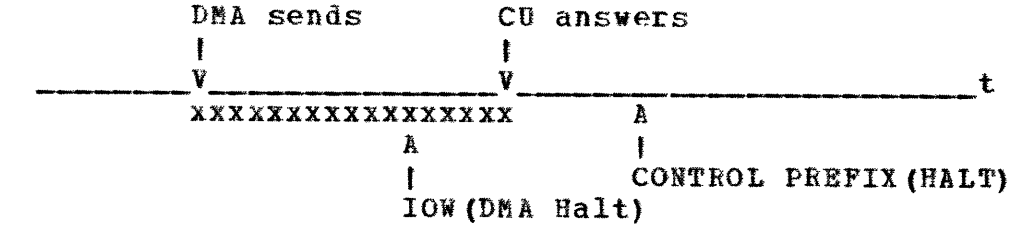
Transfer from CU to DMA (reading).

The same initialization as above takes place. Then the following sequence happens on the Serial Link:

| | DMA sends: | CU sends: |
|---|---|---|
| 1 | GO AHEAD/(chaining) | . |
| 2 | . | Data/ACKNOWLEDGE |
| TRANSMISSION | | |
| 3 | GO AHEAD | . |
| 4 | . | Data/ACKNOWLEDGE |
| TRANSMISSION | | |
| | ... | ... |
| | If DMA count>0 & CU count=0 then | |
| n | GO AHEAD | . |
| n+1 | . | Data/DATA END(res count) |
| | If DMA count=0 then | |
| n | DATA END | . |
| n+1 | . | ACKNOWLEDGE TRANSMISSION |

Use of HALT.

As described previously, HALT is issued by the DMA to terminate a transfer when it receives a DMA Halt from the 801. Since the IOW specifying DMA Halt may occur while the DMA is transfering data because of previous I/O instructions and the alternating protocol must be respected, the DMA cannot issue HALT on the Serial Link at any time but must wait for its next opportunity to send. However the DMA can accept a DMA Halt at any time.




The above figure shows that HALT cannot be sent in the 'x' regions. HALT must be acknowledge by the CU and then the DMA will request interrupt. If no acknowledgement is received, normal time out processing occurs and a request interrupt will also occur.

SWITCH.

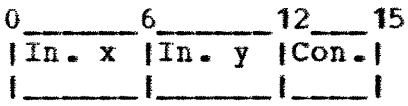
The Switch is a cross point switch of Serial Links. It can be inserted in any Serial Link. A switch's primary function is to permit fewer BSAs and DMAs than Control Units. It can also be used to connect any two of its inputs thereby permitting two CUs to communicate directly.

DESCRIPTION.

The Switch is attached to the I/O Bus and is controlled through Programmed I/O. The Switch has a maximum of 64 inlets. All inlets are identical and can be connected either to a BSA, a DMA or a CU. The Switch can establish a serial, transparent and bidirectional connection. The maximum number of simultaneous connections is 16. A connection is established through a junctor. Any junctor can connect any inlet to any other inlet.

The Switch is attached to the bus through the Switch Adapter (SA). This adapter does not implement the busy line.

When a connection is defined, it must be encoded into the 2 bytes of data of an I/O instruction according to the following format:




Bits 0 to 5 define the address of one inlet and bits 6 to 11 the address of the other inlet. Bits 12 to 15 define one junctor out of the 16.

When the program wants to establish a connection between 2 inlets, it must first make sure that none of these inlets are already connected to some other inlet, then it must find a free junctor and then issue an IOW with the appropriate DAC field and the above data. If there is no free junctor, it must first free one. If an inlet is still connected to some other inlet it must send the appropriate disconnect command. It is the programs responsibility to make sure that a link is free before it is disconected.

The Switch is controlled always using CF = 00. The DAC is used the following way:

- DAC = 0000 is used to read or write the Switch Status Register.
- DAC = 0001 is used in an IOW instruction and resets the interrupt request latch of the Switch Status

Register.

- DAC = 0002 resets the Switch Status Register to 0.
- DAC = 0003 Establishes the connection defined by the data field of the IOW instruction.
- DAC = x004 in an IOW instruction disconnects the inlets which are connected to the junctor the address of which is the x value of the DAC field. In an IOR instruction, this instruction reads the inlets which are connected to the junctor the address of which is the x value of the DAC field. The read data have the format defined here above.

The status register contains only bits 0 to 2 which are common to all adapters. Other DAC fields will be defined later to check the Switch.

The Switch is never busy to Programmed I/O.

EXTERNAL INTERRUPT ADAPTER.

External Interrupt Adapters (EIA) can be tailored for specific application environments. The following describes an External Interrupt Adapter which will be suitable for the 801 prototype system.

DESCRIPTION.

The External Interrupt Adapter shown in the figure below provides for 16 interrupt sources. Fourteen of them are external to the EIA, one is a Program Control Interrupt (PCI) and one is a Timer Interrupt (TI). The Interrupt Request Vector (IRV) identifies which sources are requesting an interrupt and the AND of the IRV and the programmable mask are bit ORed to set the External Interrupt Request Latch (EIRL) and hence the 801's single external interrupt line. There is no hardware priority between the sources; priority is under software control.

The Clock associated with the Timer Interrupt mechanism is compared with the time set in the Interrupt Time register and when they are equal the timer interrupt latch is set. The 32 bit clock has a one microsecond resolution and can be read by two IOR instructions. The IOR to the lower order 2 bytes of the clock besides being a normal read, causes the higher order 2 bytes of the clock to be moved to the clock backup register. The IOR for the higher order 2 bytes is satisfied by the value in the backup register.

The EIA is never busy to Programmed I/O instructions and the EIA's registers may be accessed as described below:

Interrupt Request Vector.

The 16 bit Interrupt Request Vector (IRV) may be read using an IOR with CF = 00 and DAC = 0003.

Interrupt Mask.

The 16 bit Interrupt Mask may be read with an IOR or written with an IOW using a CF = 00 and DAC = 0004. A zero in any bit position masks off the corresponding interrupt in the IRV.

Interrupt Time Register.

The 32 bit Interrupt Time Register is compared with the Clock and bit 5 in the EIA Status Register is set to one when they are equal. The lower (higher) order 16 bits of this register may be read or written using a CF = 00 and DAC = 0005 (0006).

Clock.

The 32 bit Clock has a one micro-second resolution. When its lower order 16 bits are read by an IOR (CF = 00, DAC = 0008), the higher order 16 bits of the clock are transferred to the Clock Back-up Register. The entire 32 bit clock is initialized to zero by the adapter reset function.

Clock Back-up Register.

The 16 bit Clock Back-up Register may be read with an IOR with CF = 00 and DAC = 0008.

EIA Status Register.

The EIA Status Register is read or written using CF = 00 and DAC = 0000.

- Bit 0 : interrupt request latch.
- Bit 1 : bus parity error.
- Bit 2 : invalid command.
- BiT 3 : not used.
- Bit 4 : programmed control interrupt latch.
- Bit 5 : timer interrupt latch.
- Bit 6 - 15 : not used.

Bit 0 is set by the OR of the bit wise ANDs of the Interrupt Mask and the IRV and is reset by an IOW with CF = 00 and DAC = 0009. The PCI bit, bit number 4, may be set to 1 (0) by an IOW with CF = 0 and DAC = 000A (000B).

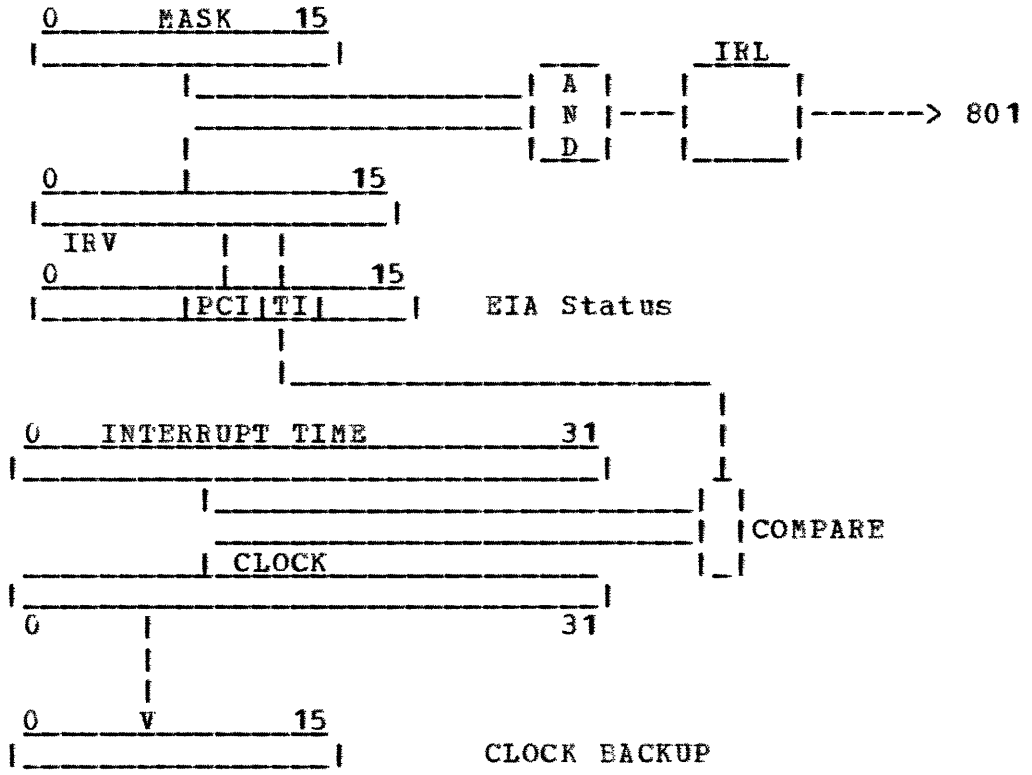
EIA Reset.

An EIA reset is initiated by the system reset line or by an IOW with CF = 00 and DAC = 0002. The effects of the reset are:

01. EIA Status Register set to zero.

02. Clock set to zero.

03. Interrupt Mask set to zero.




External Interrupt Adapter Registers