1401A DIGITAL I/O SUBSYSTEM

TABLE OF CONTENTS

INTRODUCTION

The 1401A Digital I/O Subsystem is used on GE-PAC* 3010 computer systems to direct monitoring and control of conditions external to the computer. These conditions are represented by a binary status (e.g., open or closed) which is regarded by the computer as information to be transferred between the processor and the actual circuits involved.

Digital input and output transfers by the subsystem involve several interacting functional modules. These modules are depicted by the block diagram in Fig. INT.1. As indicated by the diagram the subsystem has one controller module and optional numbers of input/output modules, termination cabinets, and interrupt modules.

The controller directs information between the processor and the input/output modules. There can be a maximum of 16 such modules attached to the controller. Each module interfaces with and is capable of selecting 128 circuits in eight-bit groups (bytes). Thus 16 bytes are addressable from a module via instruction from the controller. (Eight circuits/byte times 16 bytes equals 128 circuits.)

Each input and output circuit attaches to the subsystem at a termination cabinet, as shown in the diagram. These cabinets house assemblies containing electronics necessary to interface the circuits with their corresponding input/output module. This electronics is contained on Printed Wiring Boards (PWB's) that plug into the assemblies. Each assembly has PWB's of either input or output circuit electronics and there are eight possible PWB's per assembly. (The "high-density" termination assemblies contain 16 PWB's.) Output circuit PWB's contain relays of a holding or momentary closure type, or the electronics necessary to generate pulses, as determined by the specific application. The input circuit PWB's contain signal conditioning circuitry. Optional change detect PWB's are also available to provide both the signal conditioning circuitry for the input circuit and an interrupt signal to the interrupt module whenever the input changes status.

The controller, and interrupt module when used, plugs into the processor's multiplexer (Mux) Bus. This is actually a series of card slots (PWB connection points) whose connecting signal lines are re-

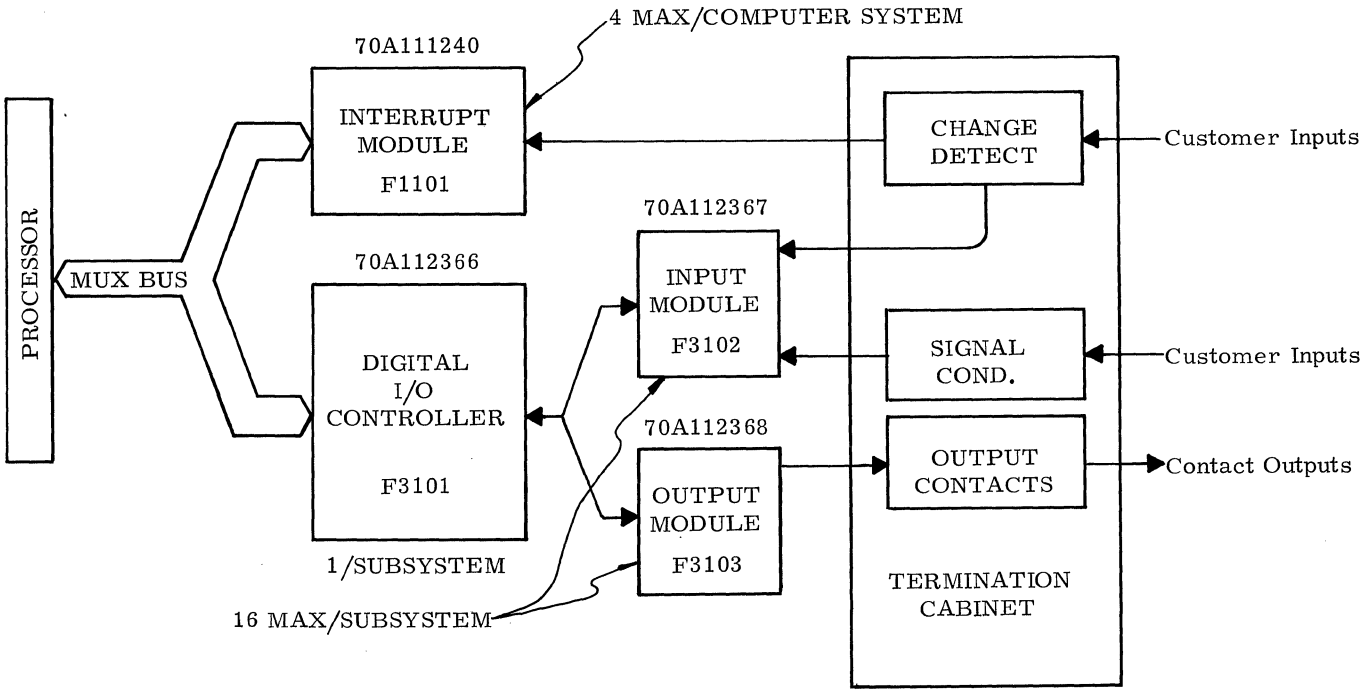


Fig. INT.1 General Block Diagram - 1401A Subsystem

*Registered Trademark of General Electric Company

ceived in parallel from the processor. The position that the device (controller, interrupt module, etc.) occupies determines the priority of its "ready" interrupt. The "device's" address is determined by "wire-wrap" jumpers on the PWB itself. It is by this address that information is directed to the desired device on the Mux Bus.

The I/O controller receives the necessary control lines and information from the processor to direct its digital input and output transfers. Input transfers direct the status of eight input circuits through their corresponding input module to the controller, from where it is transferred to the processor. Output transfers involve an eight-bit status that is transferred from the processor to the controller, from where the information is sent to the desired output module to select the desired output circuits.

Table INT. 1 indicates how to correlate the number of circuits that can be handled by the various areas of the subsystem. The numbers in parentheses pertain to the "high-density" version termination assemblies. Those numbers flagged by an asterisk indicate combined input and output capabilities. Either function's capability would be half the number.

REFERENCES

Publications

- GET-6047 3010 Reference Manual
- PCP-126 System Interface Manual

Logic Descriptions (See Appendix)

- 70A112366 Digital I/O Controller Maint. Spec.
- 70A112367 Input Module Maint. Spec.
- 70A112368 Output Module Maint. Spec.
- 70A111240 Eight-Line Interrupt Module Spec.

Logic Drawings

- 70D118021 I/O Controller
- 70D118011 Input Module
- 70D118012 Output Module
- 70B113366 Interrupt Module
- 70C181209 3010 System I/O

	BYTE	MODULE	CHASSIS	CONTROLLER	TERM. ASM.	TERM. CAB.
Bits (Circuits) Per	8	128	512	4096*	64 (128)	512 (1024)
Bytes Per		16	64	512*	8 (16)	64 (128)
Modules Per			4	32*	DNA	DNA
Chassis Per				8	DNA	DNA

Table INT. 1 Subsystem Configurations

GENERAL DESCRIPTION

Each digital I/O sequence begins with an initial transfer of directions to the controller. These directions are contained in an eight-bit command byte, transferred from the processor during the execution of an OC (Output Command) instruction. Specific bits of the command byte, when received by the controller, predetermine the interrupt status and operating mode for the subsequent transfers.

The interrupt status bits of the command byte determine how the controller is to handle subsequent external interrupts or controller ready conditions. It can be directed to either arm (enable), disarm (queue), or disable (ignore) such interrupt conditions. Since the controller's external interrupt line is not typically used in 3010 applications and the RTMOS-30 Digital I/O Multiplexer driver operates without interrupts, the interrupt line is typically disabled.

The operational mode portion of the command byte is used to "set up" the controller for either read (input) or write (output) transfers in a random or sequential mode. Once set up the controller logic operates in that mode until given a new command byte or initialized.

The OC instruction is followed by a WD (Write Data) instruction, which transfers an address byte to the controller. This address byte selects both a module and a byte address within that module. The type of module selected (input or output) is determined by the operational mode (read or write) selected by the prior command byte.

After receiving the initial OC and WD instructions the controller is set up, ready to respond to the subsequent transfer instructions from the processor in the manner predetermined by the command byte. The type of subsequent transfer instructions will depend upon the direction of transfer selected: Read Data instructions are used to transfer input status and Write Data instructions are used to transfer output status. The sequence of the subsequent transfer instructions is dependent upon the operational mode selected: random or sequential.

Random and sequential I/O operations differ somewhat in their sequence. They both require the initial set up just mentioned, but the actual sequence for subsequent transfers differs. This is best indicated by the flow chart shown in Fig. GDS. 1.

The actual I/O instructions executed by the processor during an I/O operation can occur in either an RR or RX format. RR I/O instructions involve information exchanges between a processor general register and the controller. These exchanges are between core memory and the controller during I/O instructions of the RX format. Refer to publication GET-6047, the 3010 Reference Manual, for details on format and functions of these instructions.

Read Data (RD) instructions are used for the transfer of an eight-bit byte status to the processor. This status, of course, reflects the condition of eight input circuits and will be either placed in a processor register or transferred to a core address, as determined by the instruction format. The second operand portion of an RDR (Read Data Register instruction, which is in the RR format) would indicate a specific register as the recipient of the byte. For a normal RD instruction, which is in the RX format, the second operand specifies the core address that is to receive the data.

The status of digital input circuits are sensed by RD (Read Data) instructions from an input module's core memory location. Each of an input module's 16 core locations provides status information for eight connected circuits. When addressed, this status is transferred to a data register in the input module as described under the "Input Circuits" heading appearing later in this publication.

If the RD instruction is part of a random operation, the input data is actually sensed by the preceding WD instructions that addresses the byte at the input module. The sensed information is then held in a data register at the module until the RD instruction is executed, causing the data to be transferred to the processor. Its destination in the processor is determined by the instruction format (RR or RX) and the contents of the second operand.

During sequential operations, however, the Read Data instructions are not separated by Write Data instructions to provide new byte addresses. Instead, a RD instruction transfers a byte from a holding register, loaded by the WD instruction that precedes the initial input transfer, and the controller address register is incremented. This causes the controller to automatically access the next byte from the input module so that it will be in the controller's data register for the next RD instruction.

An output operation transfers a byte status to affect eight output circuits connected to an output module. In random operations the first WD instruction transfers the output module address and is followed by a second WD instruction to transfer the status to the specified byte address within that module.

Sequential output operations utilize WD instructions to transfer the status, as indicated by Fig. GDS. 1. Like the other I/O instructions, the WD can appear in two formats. The WDR instruction (Write Data - Register) causes a transfer of a byte from a register specified by the instruction's second operand field. The normal WD instruction's second operand specifies a core address as the origin of the byte.

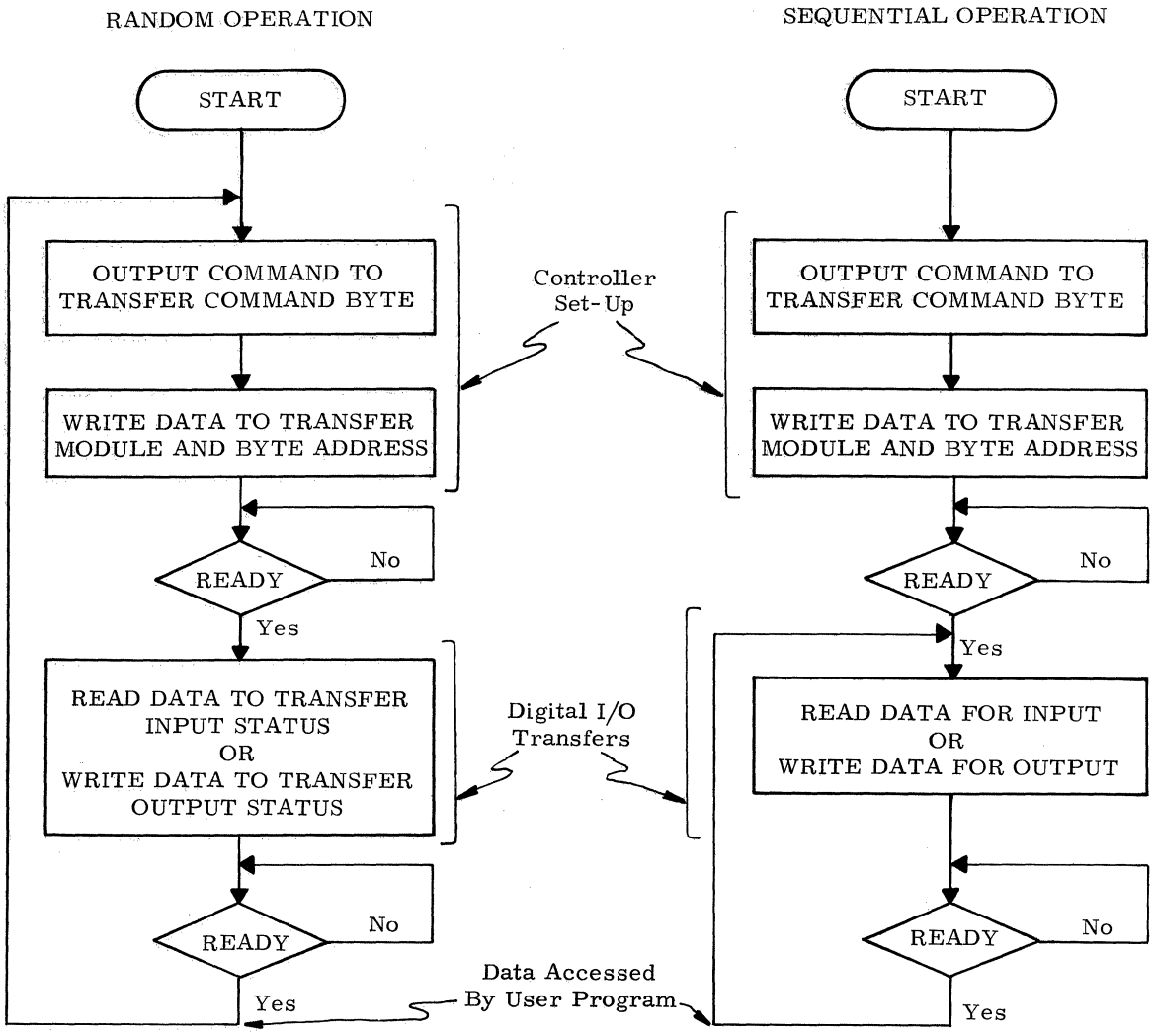


Fig. GDS. 1 Random/Sequential Flow Charts

The subsequent Controller Response descriptions cover the enabling of control lines to the controller in response to the instructions just mentioned. Each description is accompanied by a block diagram to indicate the controller logic affected by that control line. Detailed logic analysis for the gate and flip-flop functioning is covered within the individual maintenance specifications for the related modules. (See "References", mentioned earlier in this publication.)

CONTROL LINE RESPONSES

The following descriptions pertain to controller and control module (input and output) responses to information transfers during typical I/O operations. Logic involved in these transfers is depicted in the associated block diagrams and is discussed by accompanying text. Each description and block diagram covers the response to the enabling of a control line from the processor. These descriptions are of a general nature. See the corresponding Maintenance Specifications, called out under "References" in this publication's Introduction, for detailed descriptions of the logic.

The control line descriptions are discussed in the order of typical I/O operations. A controller "set-up", as shown in Fig. GDS. 1, requires an Output Command, followed by a Write Data instruction to select a Module and Byte address. Thus the order that control lines are enabled by I/O instructions during a controller set-up would be ADRS0, CMD0, ADRS0, DA0. The ADRS0 and CMD0 signals would occur during the Output Command instruction and the ADRS0 and DA0 signals would occur during the Write Data instruction, respectively.

The controller set-up would be followed by either a Write Data instruction to output data or a Read Data instruction to input data. These instructions' corresponding control lines, DA0 and DR0, are therefore the remaining two that are discussed. It should be

kept in mind that the DR0 - DATA INPUT and DA0 - DATA OUTPUT control line descriptions and their accompanying block diagrams presume that they have been preceded by instructions to set-up the controller.

ADRS0

The address line (ADRS0) is enabled at the beginning of each I/O instruction during Program-controlled I/O. This line is shown entering the left side of sheet 1 of the controller logic, 70D118021, at coordinate A-5, and is also depicted on this description's block diagram, Fig. GDS.2.

The information on the controller's Data Available Lines (DAL) is sampled during the period that ADRS0 is true. If the information at the DAL receivers compares with the controller's address, the SYN0 signal is sent back to the processor and the controller's address flip-flop sets. The processor disables the ADRS0 signal upon receipt of SYN0. The address flip-flop in the controller remains set until another ADRS0 signal is received accompanying an address different from that of the controller, or until the computer is initialized. Setting the address flip-flop enables the controller to receive further instructions from the processor.

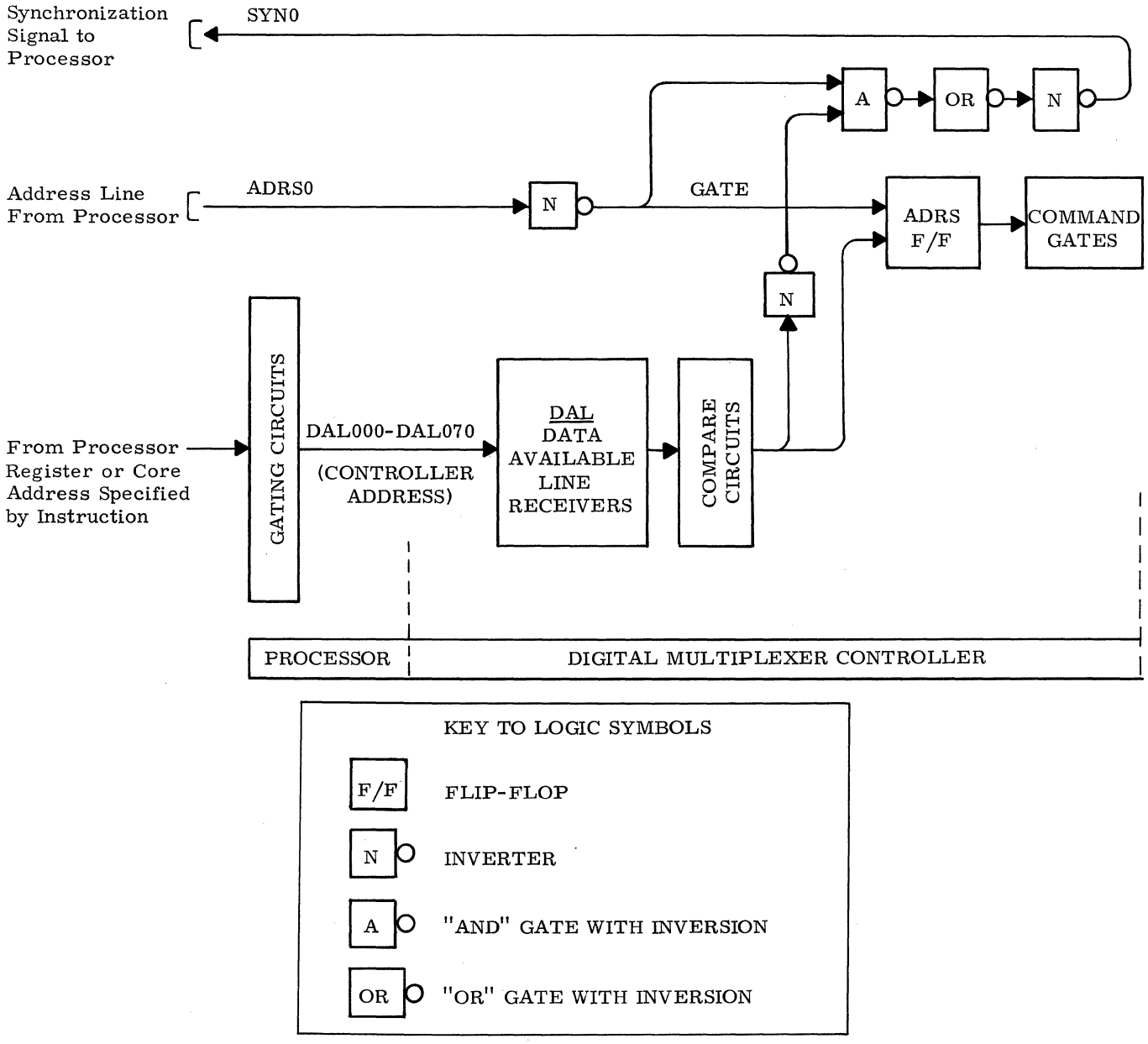


Fig. GDS. 2 Block Diagram - ADRS0 Line to Digital Controller (Enabled at the Beginning of All I/O Instructions to Controller)

CMD0

The CMD0 line is enabled during an Output Command (OC) instruction to transfer a command byte to the controller. During an OC instruction, the CMD0 line is preceded by the ADRS0 signal from the processor. The CMD0 signal is shown entering sheet 1 of the controller logic, 70D118021, at coordinate A-6. It is also depicted on this description's block diagram, Fig. GDS. 3.

CMD0 gates the command byte from the processor to the I/O controller. The significance of the command byte's bits is indicated by the block diagram's three control blocks, Interrupt Control, Random/Sequential Control, and Read/Write Control. The command byte organization is as follows:

0	1	2	3	4	5	6	7
INTERRUPT CONTROL				RAND	SEQ	WRT	RD

The controller sends the SYN0 signal back to the processor following receipt of CMD0. This causes the processor to disable CMD0. If the SYN0 signal is not returned to the processor within a 50 to 100 μs period, a time out signal would be generated to set a flag in the Program Status Word (PSW). This flag is software detectable, but not checked by the standard RTMOS-30 service routine.

DA0 (Module and Byte Address)

The DA0 signal is generated during write data type I/O instructions to transmit module and byte select information to the controller or to transfer data to a selected output module. This discussion pertains only to the response when the DA0 signal accompanies module and byte select information. The module and byte select information would follow the command byte transfer but precede the actual transfer of data.

DA0 is shown entering sheet 1 of the controller logic, 70D118021, at coordinate A-6. It is also shown on this description's block diagrams, Fig. GDS. 4 and GDS. 5. Fig. GDS. 4 illustrates the response if the DA0 signal follows a command byte specifying a write operation. Refer to Fig. GDS. 5 for the response to a DA0 signal following a command byte that has specified a read operation.

As indicated on both block diagrams, the first DA0 following CMD0 enables SAG1. SAG1 gates the information from the DAL receivers in the controller logic to its address register and enables the SYN0 line to the processor.

If the command byte has specified a write operation in either a sequential or random mode the subsequent DA0 pulses gate data to the selected output module. (Refer to the following "DA0-DATA OUTPUT" description.) Should the command byte have specified a read operation, this control line would be followed by DR0 signals. (Refer to "DR0-DATA INPUT" description.)

As shown in the block diagrams, the information in the address register is gated to the module select and byte address circuitry in the selected module. The selected module is determined by bits 0 through 3 of the information placed in the address register during this control period and by the status of the Write/Read flip-flop in the controller. The Write/Read flip-flop's status was determined by bits 6 and 7 of the command byte.

The selected module returns an acknowledge (ACK0) signal after being addressed. ACK0 is used to confirm that a module has been selected corresponding to the address sent. If there was no response the Device Unavailable flip-flop would set. The DU line can be sampled by the program if the controller receives a SRG0 signal. (Refer to the "SR0" description.)

DA0 (Data Output)

This description pertains to a DA0 signal accompanying output data to the I/O controller and should not be confused with the previous description of a DA0 signal accompanying module and byte address information to the controller. The DA0 signal is generated by a Write Data (WD) instruction. The ADRS0 line is enabled before the DA0 line when the instruction is executed.

The DA0 signal enters sheet 1 of the controller logic, 70D118021, at coordinate A-6. It is also shown in this description's block diagram, Fig. GDS. 6. As shown in the diagram, the controller's receipt of DA0 enables the return of the SYN0 signal to the processor. DA0 is inverted and combined with signal AD1 to yield DAG0. DAG0 enters sheet 2 of the controller logic where it is inverted and ANDed with the "set" output of the SA flip-flop, and with WR1, to enable OPDTG0. OPDTG0 is inverted, enabling OPDTG1 to gate information from the DAL receivers to the controller's data register. The output of the data register is gated to the selected output module's data input circuits.

The OPDTG0 signal causes the controller's flip-flop to toggle "set". The flip-flop's "one" output is delayed before ANDing with WR1 to generate the data clock pulse, TDCK0. TDCK0 is inverted, then ANDed with ADS1 and the output select signal, OPS1, to enable ASC1. ASC1 enables clock signal CEN1G0 if bit 4 of the byte address is a "0", or CEN2G0 if bit 4 is a "1". The CEN1G0/CEN2G0 signal then combines with bits 5 through 7 of the byte address

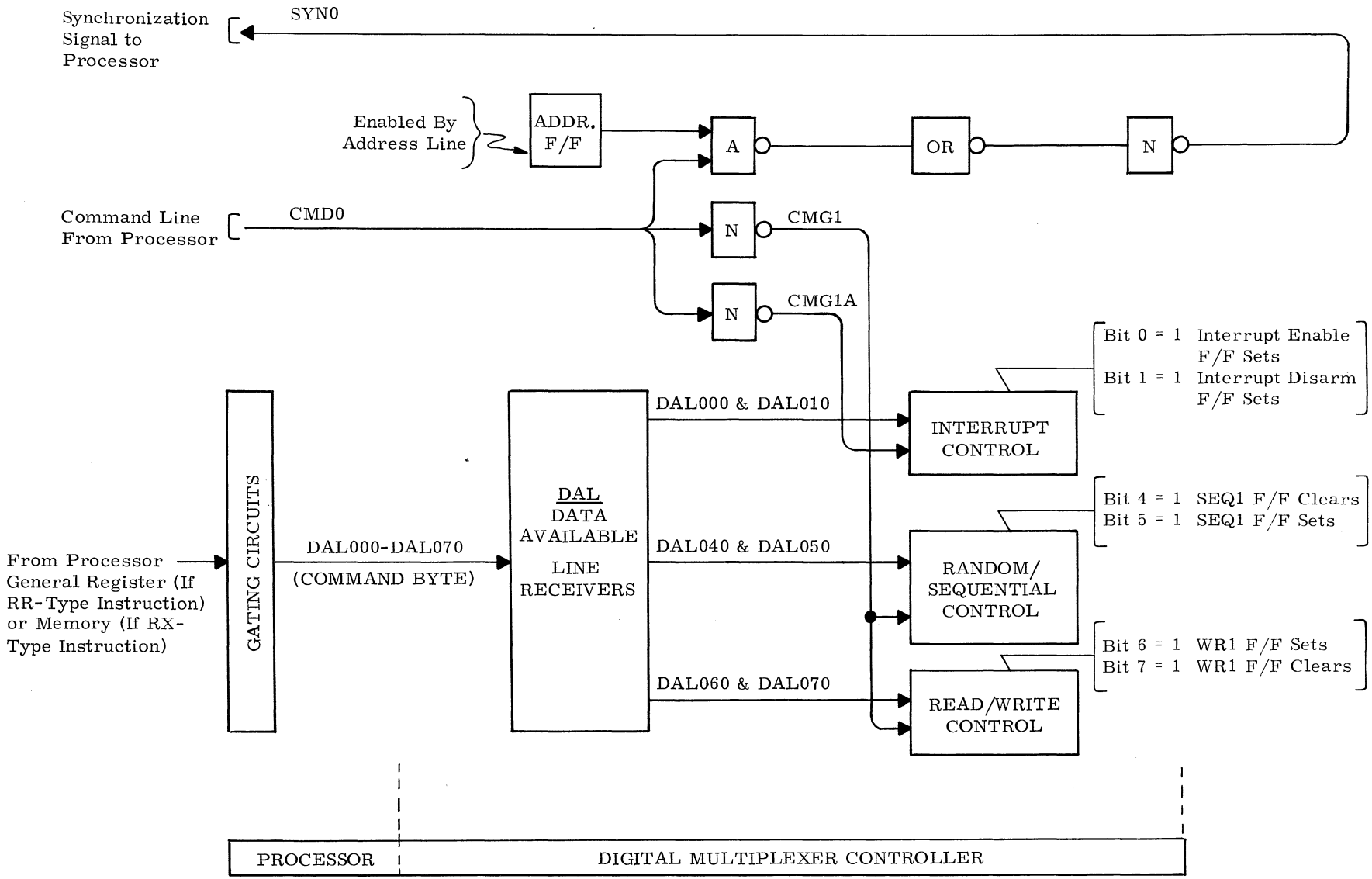


Fig. GDS.3 Simplified Block Diagram - CMD0 Line to Digital Mutliplexer Controller

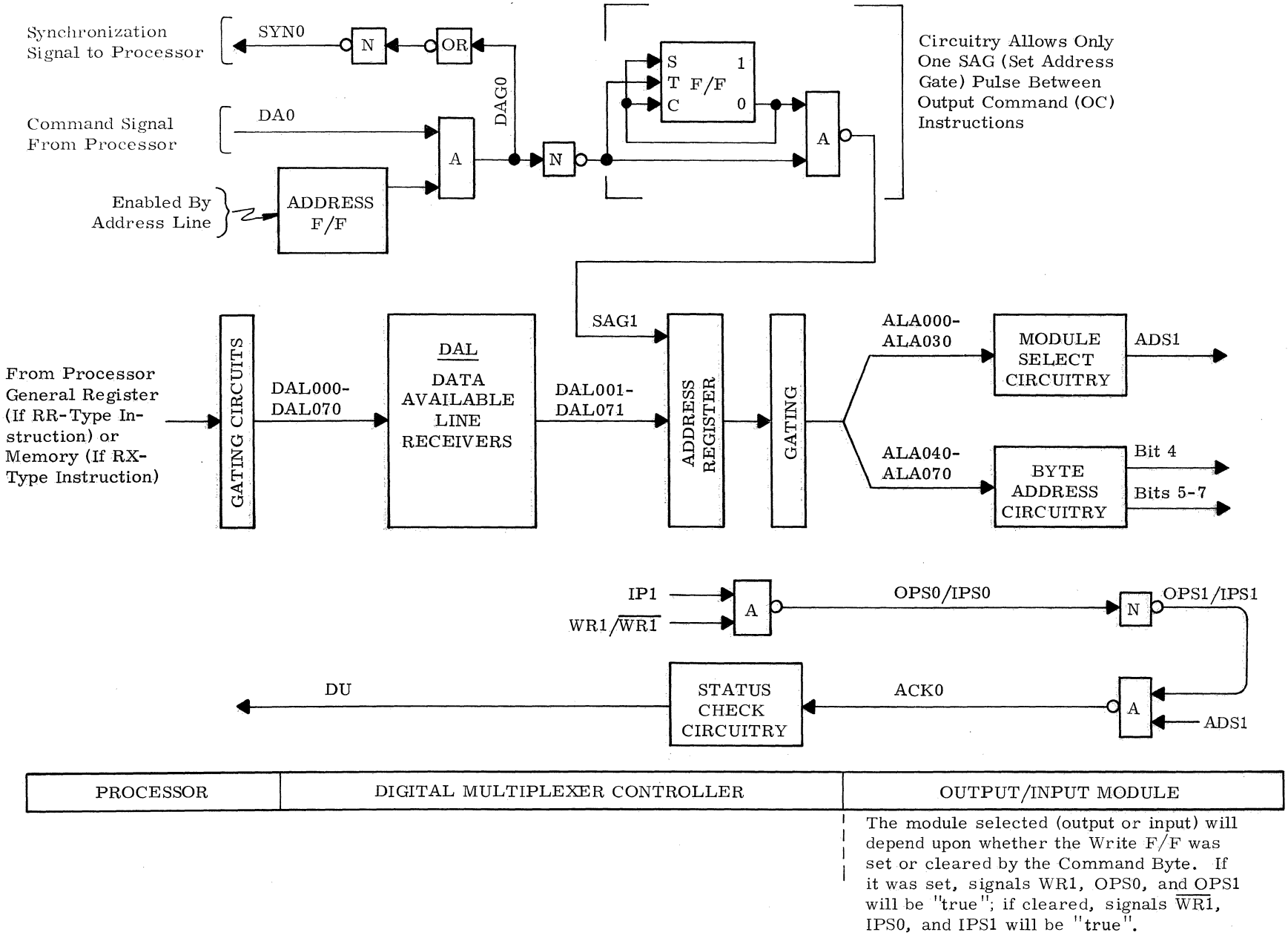


Fig. GDS.4 DA0 - Module and Byte Address (Output)

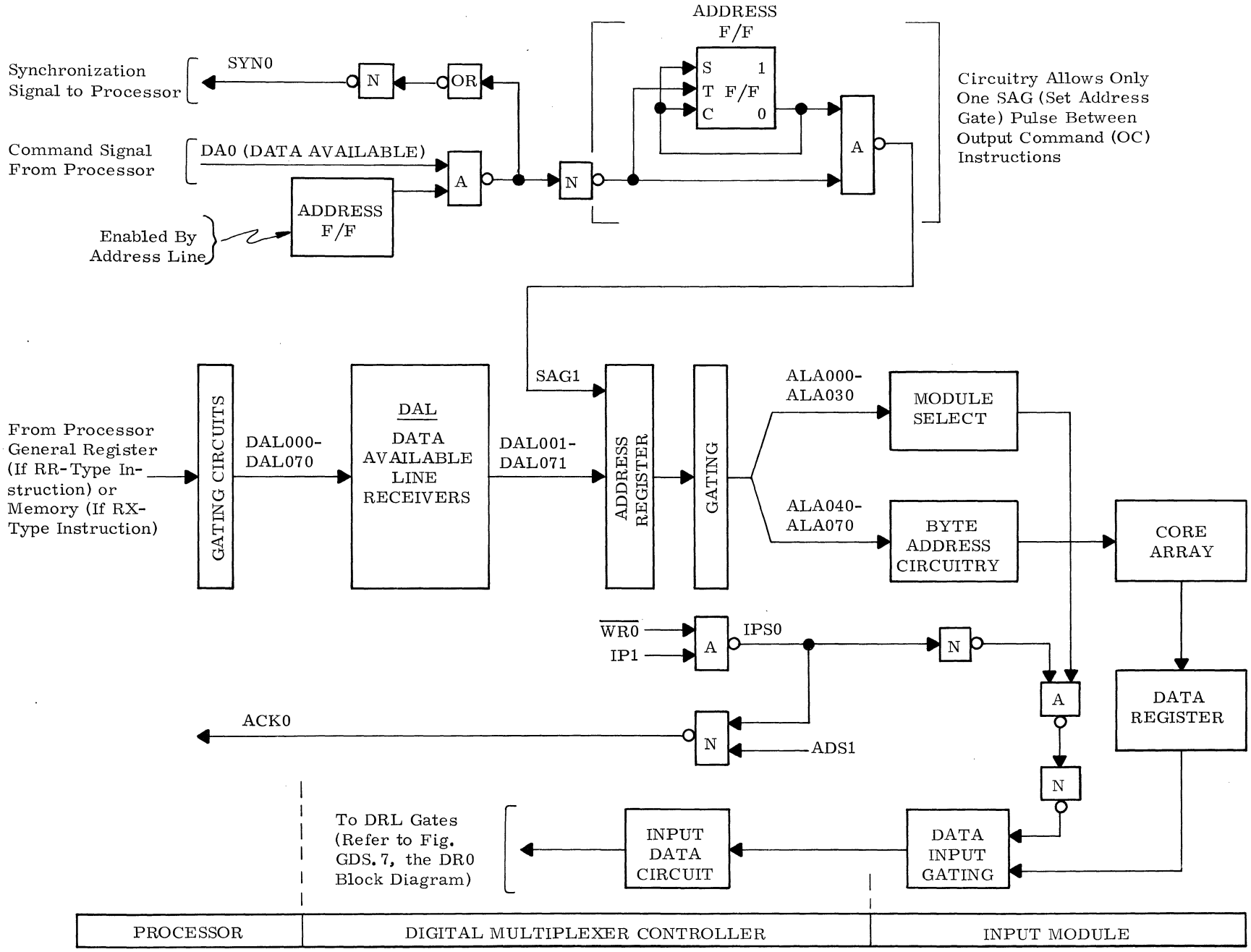


Fig. GDS. 5 DA0 - Module and Byte Address (Input)

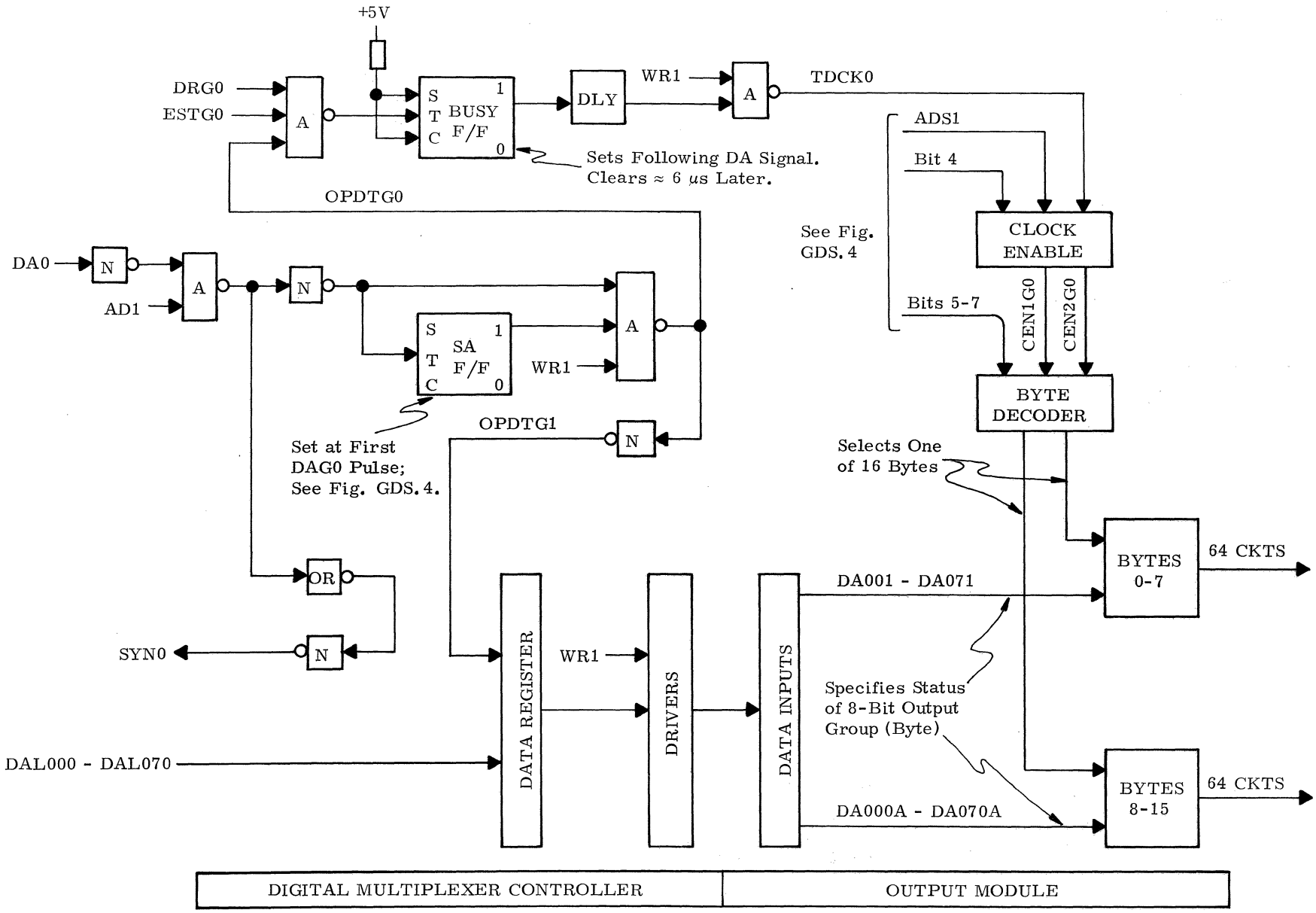


Fig. GDS. 6 DA0 - Data Output Control Line

in the byte decoder logic to enable one of the 16 output byte addresses. Thus the selected byte receives the information from the data input lines, causing the corresponding output circuits to respond in the manner determined by the type of output PWB connected. (Refer to "Output Circuits" in this publication.)

DR0 (Data Input)

The DR0 signal is generated by a Read Data (RD) instruction.

DR0 enters sheet 1 of the controller logic, 70D118021, at coordinate A-7. It is also shown on this block diagram's Fig. GDS.7. As shown in the diagram, DR0 is inverted before ANDing with AD1. The resultant signal, DRG0, enables the SYN0 line back to the processor, and is inverted, allowing the data from the input module to be gated to the DRL gates. The input data was read from the input module's core memory and gated to its data register during the DA0 pulse described by "DA0 - module and byte address". That operation was illustrated by block diagram Fig. GDS.5.

The data at the DRL gates is gated into the processor during the period DR0 is enabled. DR0 is disabled upon receipt of SYN0.

SR0 (Status Read)

The SR0 signal is sent to the I/O controller to request a readback of its test line status. These test lines and their corresponding bit positions in the readback byte are as follows:

0	1	2	3	4	5	6	7	← Bit Position
			EI	BSY	EX		DU	← Bit Significance

EI - The EI bit is set when an external interrupt is received by the controller. The external interrupt flip-flop is subsequently cleared by the controller's receipt of a CMD0 line from the processor. (Refer to CMD0 control line description in this publication.)

BSY - The BSY bit is set when the controller is handling data for transfer. Its status is transferred to bit 12 of the PSW (Program Status Word) during a status read.

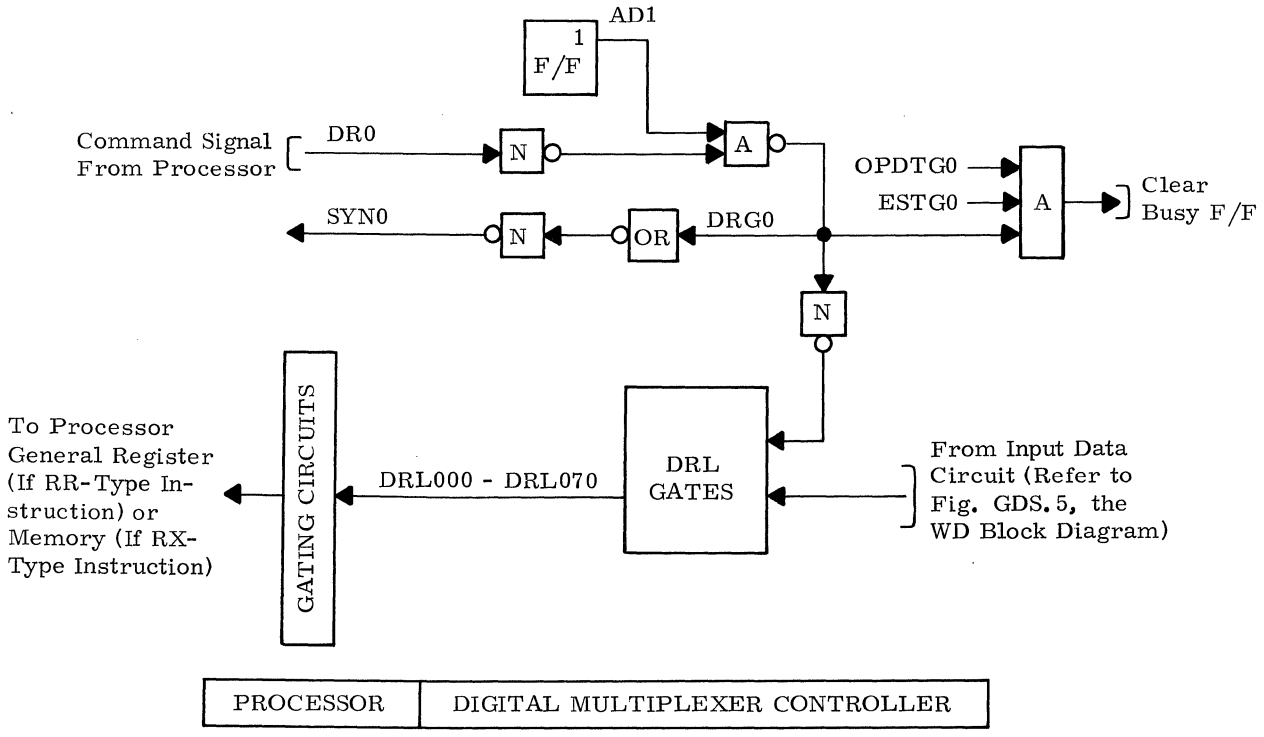


Fig. GDS.7 Block Diagram - DR0 Line to Controller

EX - The Examine bit is set when an external interrupt has occurred. This line is equivalent to the bit 3 line (EI). Its status is transferred to bit 13 of the PSW during a status read.

DU - The Device Unavailable line is set if a non-existent input or output module has been addressed. Its status is transferred to bit 15 of the PSW during a status read.

SR0 is shown entering sheet 1 of the controller logic, at coordinate A-7. It is also shown on this description's block diagram, Fig. GDS. 8. SR0 is issued by a sense status instruction and is preceded by an ADRS0 signal.

The controller's address flip-flop "set" output ANDs with the inverted SR0 signal as shown in the block diagram. The AND gate's output is inverted, yielding signal SRG0. SRG0 enables the controller logic to send a SYN0 signal and the status of several flip-flops back to the processor.

I/O CONTROLLER INTERRUPT

The I/O controller records the occurrence of two types of interrupts (ready or external) if its interrupt control logic has been "armed". If the logic has also been "enabled", the interrupt's occurrence is transferred to the processor in the form of signal ATN0. (The arm and enable status is determined by the status of bits 0 and 1 of the command byte, mentioned under the "CMD0" description.) The RTMOS-30 driver does not utilize the interrupt function, but it is still described here in the event it is used by special application software.

A controller "ready" interrupt is generated whenever the controller's busy flip-flop clears, indicating that the controller is ready to receive new data for transfer.

The external interrupt can be used for the detection of a change in an external circuit, as determined by the individual application.

The processor can determine which type of interrupt (ready or external) occurred by reading the controller's status byte to determine if an external interrupt occurred. (Refer to "DR0 - Status Read" description in this publication.)

ATN0, the interrupt line to the processor is depicted on block diagram, Fig. GDS. 8. As shown, the "ready" and "external" interrupts are ORed together to "set" the interrupt queue flip-flop, if the flip-flop is not held "clear" by the disarm logic. The "set" output of the flip-flop, if not held to "0" by the disable logic, then enables the ATN0 line to the processor.

The processor responds to an interrupt by raising the RACK0 line, shown entering sheet 1 of the controller logic, 70D118021, at coordinate A-8. The highest priority device controller having an interrupt will intercept the signal and prevent it from being "daisy-chained" to the lower priority controllers. The controller's ATN0 line will remain "true" until it intercepts this pulse or until a CMD0 signal is sent to disable the signal. When the controller intercepts the signal it generates signal ATSYN1, toggle "clearing" the interrupt queue flip-flop and transmitting the controller's address over the DRL gates to the processor.

INTERRUPT MODULE

The Model 4DP3000AF1101 Interrupt Module consists of one printed wiring board and is optionally used with the 1401A Digital I/O Subsystem. It interconnects with the 1401A subsystem in the manner shown by Fig. INT. 1, and is connected to the processor in the same manner that other device controllers connect to the Mux Bus.

The interrupt module's purpose in this configuration is to control and direct "change detect" interrupts from digital input circuits. Change detect interrupts should not be confused with controller "ready" interrupts, already mentioned under the I/O controller description. The interrupt module enables, queues, or inhibits change detect interrupts according to instructions from the processor. Each interrupt occurs as a result of a change in its corresponding digital input line's status and will interrupt the processor if enabled by the module. Queued interrupts are recorded by the module until such time that the interrupt line is enabled by instruction from the processor. Inhibited interrupts are lost and therefore have no effect upon the processor's operation.

Control status for each of the module's eight interrupt lines is established by information contained in the module's arm and mask registers. Information is placed in the registers by processor-executed I/O instructions. Both registers are eight bits in length, each bit controlling one of the eight interrupt lines.

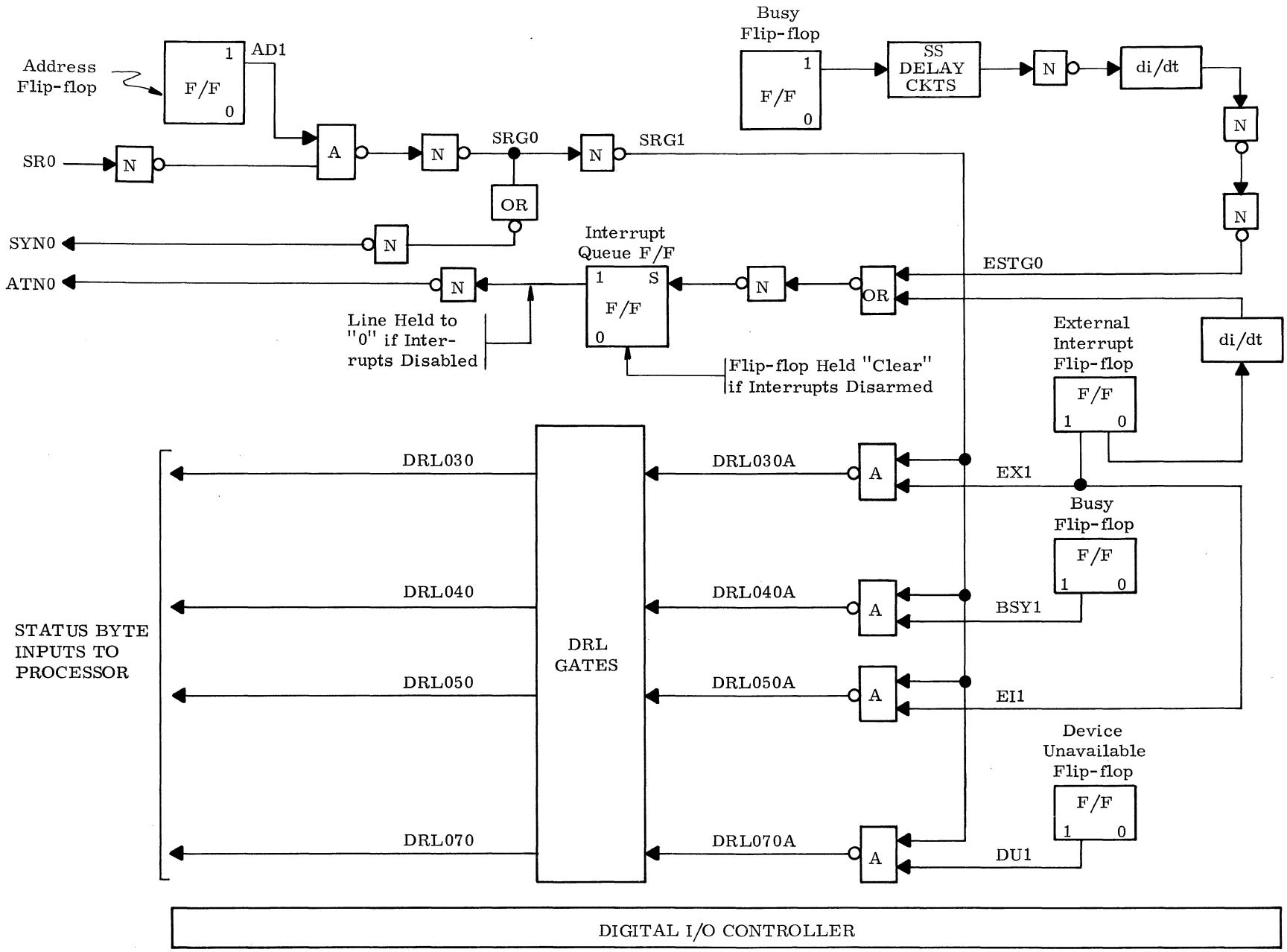


Fig. GDS.8 SR0 - Status Read Control Line

An arm register bit either arms (allows to be queued) or disarms its corresponding interrupt line as determined by the bit's logical "1" or logical "0" status. A logical "0" bit in the mask register masks (prevents interrupt transfer to processor) its corresponding interrupt line and a logical "1" bit unmasks the line. The arm/disarm and mask/unmask functions can be translated to enable, queue, or inhibit as specified by the following Table.

	DISABLE		QUEUE	ENABLE
Arm Register	"0"	"0"	"1"	"1"
Mask Register	"0"	"1"	"0"	"1"

Those interrupts armed by the module enable an interrupt line (ATN0) to the processor. The processor responds with a Request Acknowledge signal (RACK0) that is daisy-chained through the module's priority logic until it encounters its highest priority interrupt signal. This will disable the RACK0 signal from being transmitted through the remaining device controllers and the module will transmit a SYN0 signal back to the processor. If the acknowledged interrupt has not been masked by the module logic, its address will be transmitted back to the processor. The transmitted interrupt address is used by the processor to access a service routine which responds to the enabled interrupt, if in the interrupt enable mode.

Each of the module's interrupt lines has a unique device address. These addresses are sequential, starting at the address of the interrupt module to which they are assigned. The module address can be X'20', X'28', X'30', or X'38'. Interrupt addresses for a module having device address X'20', for example, would be sequential from X'20' through X'27'. A module with device address X'28' would have interrupt addresses X'28' through X'2F'. Modules with device addresses X'30' and X'38' would have interrupt lines with addresses X'30' through X'37' and X'38' through X'3F', respectively.

INPUT/OUTPUT CIRCUITS

The remaining text of this publication pertains to the actual input and output circuits being monitored or controlled by the Digital I/O Subsystem. Digital input circuits are opened and closed by customer equipment (switches, relays, etc.). The digital output circuits are opened or closed by computer control. In either case the circuits involve printed wiring boards (PWB's) that plug into the computer's termination cabinet at the point where the customer's input or output circuit is connected. The circuits and the different PWB types involved are discussed here. G. E. drawing 70C181209, the 3010 System I/O Logic, provides a convenient reference for the discussion.

INPUT CIRCUITS

Each input module connected to the Digital I/O Multiplexer Controller provides circuitry to monitor 128 input circuits. These circuits are addressable in 8-bit groups (bytes) by the controller, upon instruction from the processor. A digital input operation causes the addressing of the byte in the module and the subsequent transfer of its status to the processor.

The status of a digital input circuit is indicated by whether or not an input line's core switched states during addressing. A typical input circuit is illustrated by Fig. I/OC.2. If the contact is closed there will be current flowing through the core winding designated I_I in the input module box on Fig. I/OC.2. The current combines with the bias current (I_B) at the core, as indicated by point A on Fig. I/OC.1, which is not sufficient to cause the core to switch from the state designated Flux (-) to the state designated Flux (+) on the B axis. Even the addition of the X and Y drive currents, during readout, only takes the magnetic current state to point C on the axis. This is not sufficient to cause the core to switch states, and the readout circuitry regards this as indication of a logical "1" status.

If the contacts open, no input circuit current flows. This is indicated by point B on the axis. When the X and Y drive currents are added to this the magnetic current state moves to point D, the Flux (+) state. When the X and Y address currents are removed the bias current causes the magnetic state to revert back to point B, the Flux (-) state. This switching of magnetic states is detected during readout by the input module's readout circuitry as being a logical "0".

Signal Conditioning

Signal conditioning circuitry for digital input circuits is contained on printed wiring boards (PWB's) that plug into termination assemblies in the termination cabinet. Each of these PWB's has conditioning circuitry for eight input circuits. The conditioning circuitry can be either standard (filters) or change detect. Standard signal conditioning provides for different voltage levels (+28V or +125V) and signal characteristics (75 μs, 1 ms, and 22 ms time constant filters). Change detect PWB's can be plugged into a digital input circuit in place of the standard signal conditioning to provide both signal conditioning and change detect circuitry. Both categories are discussed in the following text.

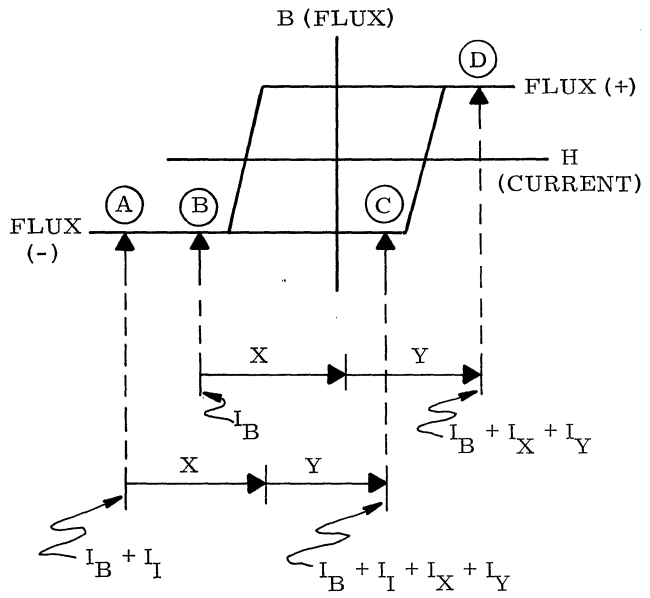


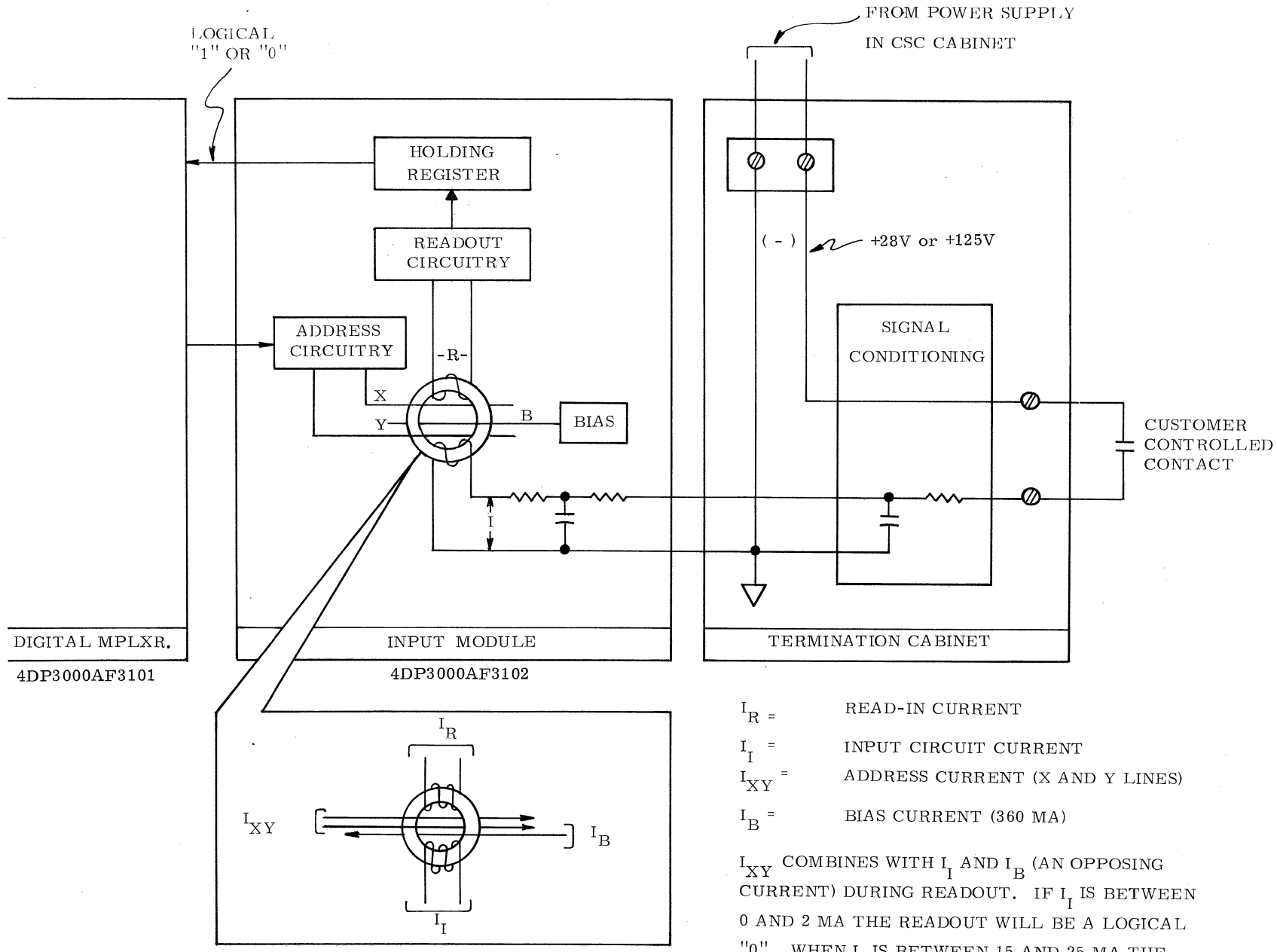
Fig. I/OC.1 Core Switching

STANDARD

The standard signal conditioning PWB's for digital input circuits contain low-pass filtering and current-limiting circuitry. These PWB's are available in three different delay constants at two different voltage levels, as specified by the following:

PWB TYPE	VOLTAGE	DELAY CONSTANT
PX3600ICFE1	28V	75 μs
PX3600ICFF1	28V	1 ms
PX3600ICFG1	28V	22 ms
PX3600ICFE2	125V	75 μs
PX3600ICFF2	125V	1 ms
PX3600ICFG2	125V	22 ms

The delay constants specified are nominal time periods required by the input circuits to respond to a status change. For example, a circuit with a signal conditioning PWB specifying a 75 μs delay constant would not indicate to the input module that the circuit



I_R = READ-IN CURRENT

I_I = INPUT CIRCUIT CURRENT

I_{XY} = ADDRESS CURRENT (X AND Y LINES)

I_B = BIAS CURRENT (360 MA)

I_{XY} COMBINES WITH I_I AND I_B (AN OPPOSING CURRENT) DURING READOUT. IF I_I IS BETWEEN 0 AND 2 MA THE READOUT WILL BE A LOGICAL "0". WHEN I_I IS BETWEEN 15 AND 25 MA THE READOUT IS A LOGICAL "1".

Fig. I/OC.2 Digital Input Circuit

had changed status unless it retained the status for approximately 75 μs or longer. This provides effective filtering for the input circuit since signals of less duration are not recognized.

CHANGE DETECT

Change detect PWB's can be inserted in place of standard signal conditioning PWB's to provide both a line for monitoring and a signal to indicate when that line's status has changed, if the change detect function is implemented. When used, the input line can be monitored in the same manner as an input circuit with standard signal conditioning, and the change detect signal is transferred to the interrupt module (see earlier description). If enabled by the interrupt module (per instruction from the processor), the signal interrupts the processor's operation.

Change detect PWB's are available in 28V or 125V levels, with delay constants of 500 μs or 22 ms. These delay constants pertain only to the change detect signal; the input line for these circuits have a standard 75 μs delay constant. PWB types corresponding to these characteristics are as follows:

PWB TYPE	VOLTAGE	DELAY CONSTANT
PX3600ICDE1	28V	500 μs
PX3600ICDF1	28V	22 ms
PX3600ICDE2	125V	500 μs
PX3600ICDF2	125V	22 ms

The change detect interrupts are typically implemented on a "per assembly" basis with one interrupt module handling eight interrupt lines for the subsystem. There is one interrupt line from each change detect PWB. This line is actually an "OR" of the PWB's eight input circuits and will be enabled if any one of the eight lines changes status. The standard wiring of an assembly connects all interrupt lines from the change detect PWB's, resulting in one change detect interrupt line for the assembly. This line would be enabled whenever there is a change in status of any circuit connected to the assembly. This configuration can be altered by rewiring for special applications.

OUTPUT CIRCUITS

Each output module connected to the Digital I/O Multiplexer provides circuitry to control 128 output circuits. The status of these circuits is determined by information sent from the processor to the controller in 8-bit groups (bytes).

An output circuit's status can assume one of two states. The state assumed when the circuit is enabled, however, is dependent upon the type of printed wiring board (PWB) used as output circuits for the 8-bit group. Depending upon the PWB type, the circuit (when enabled) will either close, pulse for a program-determined duration, or pulse monentarily. If the circuit is not enabled there will be no response. (This would result in an open circuit for the latching output type of PWB and no pulses from the other two circuit types.)

Fig. I/OC.3 illustrates a typical latching digital output circuit. The latching output shown is one of eight on a PWB. The latching output PWB can be either a PX3600IHSK1 or a PX3600IHSL1. IHSK1 PWB's have 100 VA mercury-wetted relays and the IHSL1 PWB's contain 10 watt dry-contact relays.

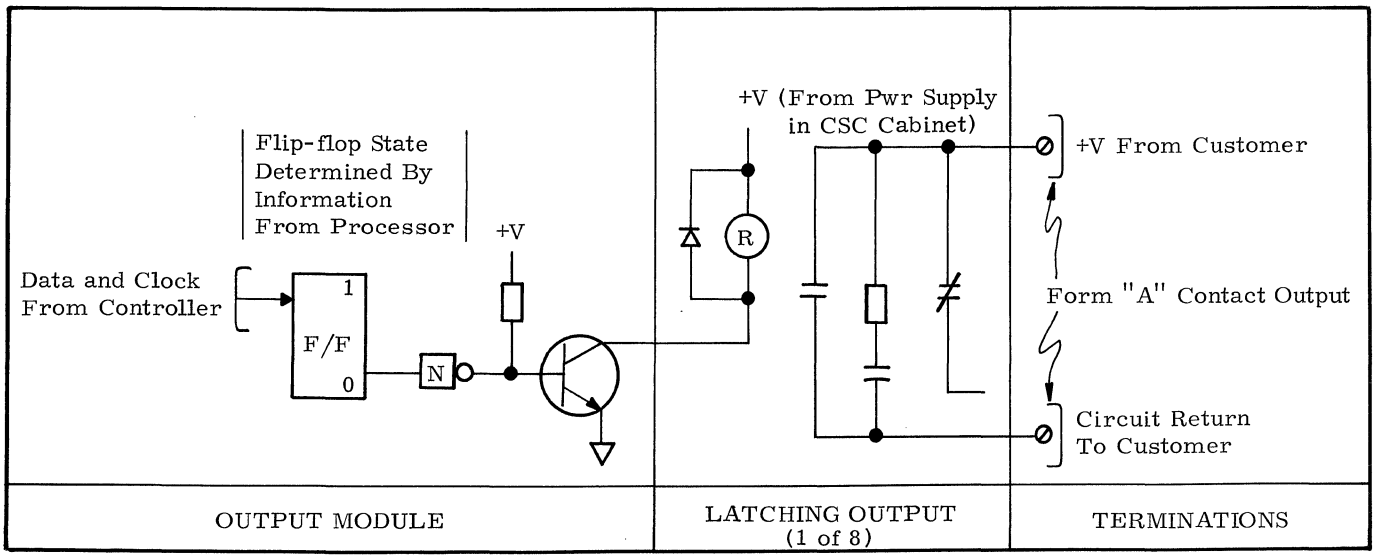


Fig. I/OC.3 Typical Latching Digital Output Circuit

APPENDIX