PREPARE THE MAN

STUDENT TEXT
C1503-416M-ST

Computer Systems Department

DESCRIPTORS - INPUT/OUTPUT CONTROL FOR BUIC III SYSTEM

October 1967

Keesler Technical Training Center
Keesler Air Force Base, Mississippi

Designed For ATC Course Use

Computer Systems Department
KTTC, Mississippi

Study Guide & Student Text
October 1967

DESCRIPTORS, INPUT/OUTPUT CONTROL FOR THE BUIC III SYSTEM

CONTENTS

DESCRIPTORS, INPUT/OUTPUT CONTROL FOR THE BUIC SYSTEM

OBJECTIVE

To explain the basic control for transferring data between the Terminal Equipment and the Core Memory units within the BUIC System.

INTRODUCTION

This study guide is intended to define the format and control functions of the control words (DESCRIPTORS) that are used in the BUIC System to control the transfer of data between the Terminal Equipment and the Core Memory units. These control words will be utilized for Input Operations, where we transfer data from a selected Terminal Device into a Memory unit, and for Output Operations, where we transfer data from a selected Memory Unit to a Terminal Device.

GENERAL

The I/O module in the BUIC System works for the computer in much the same manner that a "Tool Chaser" works for a mechanic. When the mechanic needs a tool he sends his "Tool Chaser" after it, when the "Tool Chaser" brings it back the mechanic uses it. Once the mechanic is finished with the tool he has the "Tool Chaser" put it back. The program works much the same way. In our case when the computer needs data it sends the I/O module after it. The I/O module will get the data from a terminal device such as the Card Reader or Magnetic Tape for Input Operations, or from a Core Memory for Output Operations. When the I/O gets the data from these units it will place it where the computer has specified. For input operations this will be memory locations that have been specified by the program. For output operations this will be stored on magnetic tape units or magnetic drum units. Once the I/O module has gotten the required data and transferred it to the specified location, the I/O will notify the computer that the job has been accomplished. The computer which has been performing other tasks while waiting for the I/O to receive the data, will now come back and check to see that the I/O received the correct data the program wanted just as the mechanic would check to see that the "Tool Chaser" got him the correct size and type wrench. If everything was received correctly, then the computer would utilize the data that the I/O had transferred in accomplishing other tasks.

A computer does not have the ability to think or know what is to be accomplished like a human mechanic does. Each action that is to be performed by the computer system in accomplishing any task must be told to the computer by our program. This includes notifying the computer when to perform the operation, what terminal equipment to use, and what type operation (Input or Output) that is to be performed. This is accomplished by the use of control words in our program which we call Descriptors. These Descriptors are used to setup (initially condition) the equipment, initiate and control each operation of the terminal equipment, terminate the operation once the required operation has been completed, and to terminate the operation at any point in the operation if a malfunction occurs.

There are five types of control words that are used with the BUIC System. These are broken down into two general classifications. Three descriptors are classified as software descriptors (sometimes referred to as Program generated). This means that these three descriptors are physically made up by the programmer and placed in the program that will be operating the equipment. These three descriptors will be sent from the program that is stored in a specified memory to the I/O module to be used by the I/O in controlling the operation with the terminal equipment. The three software descriptors are the Set-Up descriptor, the Command descriptor and the Release descriptor. The function and format of each of these will be covered later in this study guide.

The remaining two descriptors are labeled hardware descriptors (sometimes referred to

as I/O formed descriptors). These two descriptors are formed in the I/O module by logic circuits and will be transferred to a specified memory unit where they are stored so that the program can check the results of the operation that was to be run. The two hardware descriptors are labeled the in-process descriptor and the result descriptor. The in-process descriptor is formed in the I/O module at the start of an operation to notify the program if the selected terminal equipment is available and if the operation did start. The result descriptor is formed within the I/O module at the end of an operation and sent back to a specified memory location to notify the program of the results of the operation. This allows a check of the operation to determine just how successful the operation was and if there was any malfunction that occurred during the operation. In forming the hardware descriptors, the I/O uses the software descriptor it received and modifies or updates the software descriptor and sends a copy of the updated descriptor to a specified memory location as the hardware descriptor.

Before we take a look at the format and function of each of our descriptors there are some basic background facts we need to establish.

An operation is initiated by the computer utilizing the program which has been stored in a specified memory module(s). This program is a step-by-step sequence of instructions that will be performed during the running of the program. In the BUIC System, operations to transfer data between the Terminal Equipment and Core Memory is controlled by the program utilizing an instruction called a TIO (transmit to I/O) instruction. This instruction causes a descriptor control word to be obtained from a memory location and sent to an I/O module on either Bus A or Bus B, where the selected I/O will perform the required operation. This TIO instruction would be the same as our mechanic verbally telling the "Tool Chaser" to go get him a 1" open-end wrench from the tool box.

The computer in the BUIC System is able to go ahead and accomplish other tasks while waiting for the I/O module to transfer the requested data, once the operation has been started by the I/O module. This is much the same as the mechanic would do on his job while waiting for the "Tool Chaser" to return with the wrench.

There are three different states or conditions that the I/O module can be in:

1. NON-BUSY: This is a condition where the I/O is not performing a data transfer operation. This can be compared to a "stand-by" condition, where the I/O is waiting to be told to perform a task much as our "Tool Chaser" would be waiting for the mechanic to tell him what tool to go get next.

2. INACTIVE BUSY: In the Inactive Busy state an I/O module is not performing a data transfer operation and can not be instructed to perform an operation. However, from all appearance the I/O module looks as if it is actually performing an operaion. This is much like you have seen individuals who looked busy when the supervisor came around, yet they were not performing a specific task. The reason for the Inactive Busy state is for system flexibility and will become more apparent later in the course. The Inactive Busy state is sometimes referred to as a passive busy or just busy state. It means the I/O looks busy.

3. ACTIVE BUSY: In the Active Busy state the I/O module is performing a data transfer operation between a Terminal Device and a selected Memory Unit.

First Non-Busy I/O: (FNBI/O) This is a term used in the BUIC System to designate the lowest numbered non-busy I/O module on a bus. A bus is the inter-connecting lines between the I/O modules and the Core Memory units. In the present BUIC System we have two I/O modules on Bus A and two I/O modules on Bus B. I/O modules #1 and #2 are located on Bus A, while I/O modules #3 and #4 are located on Bus B. When we speak of the FNBI/O, we are referring to the lowest numbered I/O module

on the bus. For example: Assume that on Bus A, I/O #1 and I/O #2 are in the Non-Busy state; then I/O #1 would become the FNBI/O. If I/O #1 was busy then I/O #2 would become the FNBI/O.

SOFTWARE DESCRIPTORS: The three software descriptors which are transferred from a memory location to the I/O module as a result of the computer executing a TIO instruction will contain a 48 bit control word and a parity checking bit that is used to insure that a valid control word was transferred.

SETUP DESCRIPTOR: The setup descriptor is sent to all I/O modules on a bus to perform two functions: (1) To initially condition the I/O module by putting the I/O in a busy state after power has been applied to the system. (2) To set the Descriptor Base Address Register (DBAR) in each I/O module to specify the memory module and memory module address that the I/O module will send the in-process and result descriptors to after each operation is performed. This allows the program to check to see that each operation started and the results of the operation when the operation ends. This tells the program if the operation was completed successfully or if there was a malfunction during the operation.

The setup descriptor will normally be transferred after power is applied to the system to initially condition the equipment and set up the DBAR register. The other condition which requires the transmittal of a setup descriptor is when the type of program being utilized is changed to a different program. When the program is changed the DBAR for the new address specifying the location for the return of the in-process and result descriptors for the new program will be loaded into the DBAR register.

When a setup descriptor is applied to one of the I/O buses (A or B) it is received by all I/O modules connected to the bus. All the I/O modules on the bus (busy and non-busy) check the setup descriptor for the correct parity and take the appropriate action as follows:

1. Setup descriptor received with correct parity.

a. All active busy I/O modules load the base address of the descriptor into their respective DBAR without interrupting the active operation. Does not return an in-process descriptor.

b. Non-busy I/O modules will load the base address into the DBAR register and are made inactive busy. Will return a partial in-process that contains only the DBAR setting in bits 1 through 11 and status code in bits 17, 18, and 19.

c. The first non-busy (lowest-numbered non-busy) I/O module (FNBI/O) returns a full in-process descriptor that is identical to the setup descriptor except that update I/O status is inserted into bits 17, 18, and 19. The in-process descriptors can now be examined by the program in control to verify the successful receipt of the setup descriptor. The in-process will be returned to the new DBAR address.

2. Setup descriptor received with incorrect parity (even number of 1's):

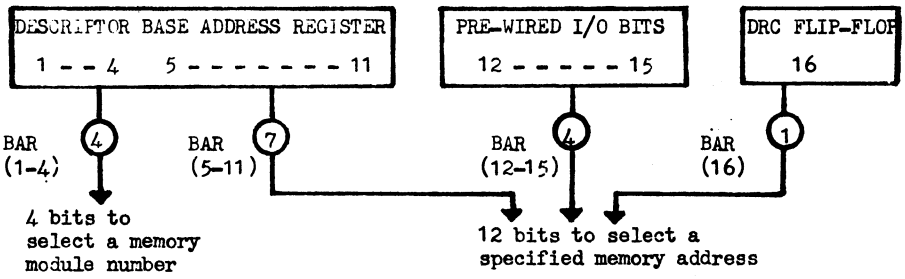
a. All active busy I/O modules on the bus ignore the new DBAR setting and do not load their DBAR register and continue the operation they are performing.

b. The non-busy and inactive busy I/O modules on the bus will ignore the new DBAR address and do not load the new address in their DBAR register. However, the non-busy I/O modules are set to the inactive busy state. The non-busy I/O's will return a partial in-process.

c. The FNBI/O will return a full in-process which is identical to the setup descriptor with status inserted in bits 17, 18, and 19 = 111. The in-process will be sent back to the original DBAR address. The format for the Setup Descriptor is shown on C1502-416M-SU, Descriptor Code Card.

The Setup Descriptor contains the 11 bit DBAR in bits 1 through 11 of the descriptor.

These 11 bits are used in conjunction with 4 bits (pre-wired into each I/O module) and the status of the Descriptor Return Control flip-flop (DRC) to provide 16 bits to select the memory and memory address location where the in-process and result descriptor will be returned.



DBAR Bits 1 through 4 will specify the the memory module (0_8 through 17_8) that the programmer has selected for storage of the in-process and result descriptors.

DBAR Bits 5 through 11 make up the 7 MSB of the 12 bit word location address.

DBAR Bits 12 - 15 (pre-wired in I/O module) are used in conjunction with DBAR bits 5 through 11 to make up the 11 MSB of the 12 bit word address. BAR 12 - 15 will be pre-wired to the I/O module #. For I/O #1, Bar 12 - 15 = 0001.

DRC The DRC flip-flop will be used with BAR 5 - 15 to form the LSB of the memory address. When an in-process descriptor is being returned the DRC will = 0. For a result descriptor the DRC will = 1.

BAR 12 - 16 will equal the I/O module # X2 for an in-process descriptor. When I/O #1 is being used BAR 12 - 16 will equal 0010. For a result descriptor, BAR 12 - 16 will equal the I/O # X2 + 1. For I/O # 1 this means BAR 12 - 16 will equal 0011.

The 5 LSB of the memory address location will always be the same for all programs. The 7 MSB can be changed by the programmer to specify the location the programmer wishes to utilize in his program.

It should be noted that each I/O module has only one location for sending its in-process descriptor to and one location for its result descriptor. After each operation the program makes a check of the location and this way the next operation can utilize the same location for the in-process or result descriptors.

Bits 13 through 31 will be all Ø's.

Bit 32 of the setup descriptor will always be a "1" bit. When the setup descriptor is loaded in the I/O module each syllable is processed through the control and parity checking register where it is checked and if the overall word parity is correct the DBAR is then loaded into the DBAR register. When loading the descriptor into the I/O module syllable D is loaded in first, then syllable C, B, A, parity. When syllable D is loaded into the control and parity register bit 44 would set bit position #8 to a "1" bit. We utilize bit 32 of setup to compliment the "1" bit back to a "0" when syllable C comes in so that when syllable A is loaded into the I/O whatever configuration of bit position #8 in the DBAR will be loaded into the DBAR register. If bits 1 through 11 of the DBAR register are not 0's when the last syllable of the descriptor word is entered, the DBAR setting specified in the setup descriptor would be modified.

Bits 33 through 43 will contain all 0's.

Bits 44 through 48 will contain the configuration of 10001 which is decoded within the I/O module to notify the I/O module that this is a setup descriptor. These five bits are called order code.

RELEASE DESCRIPTOR: The release descriptor illustrated in C1502 card is used to release a specified (addressed) busy I/O module from its busy state and set it non-busy. The I/O module has two different busy states in which it can exist; the "active" busy state, in which the I/O module is actively performing an operation, and the "inactive" busy state, in which the I/O module looks busy but is not performing an I/O operation. The release descriptor, by setting an inactive busy I/O module to the non-busy state, allows the I/O module to accept a new command descriptor immediately after receipt of the release descriptor in order to perform another operation. It should be noted that only one I/O module will receive a release descriptor at any one time on a bus. This makes this I/O module become the FNBI/O on the bus since all other I/O modules on the bus would be "active" busy or "inactive" busy. The release descriptor is also used to terminate an I/O operation in progress by releasing an active busy I/O module and setting it non-busy. Normally, a release descriptor will not be sent to an active busy I/O module as it would stop the operation that is in progress. The exception would be where a priority operation is needed to get data displayed on the Data Display console when all I/O's on the bus were tied up performing operations or a program error in sending the release descriptor at the wrong time. The Release Descriptor format is shown on C1502-416M-SU, Descriptor Code Card.

Bits 44 through 48 contain the descriptor type code of 10000. This is decoded within the I/O module as a release descriptor.

Bits 39 through 42 contain the specified I/O module number that will accept and take action on the release descriptor to make the I/O module non-busy. The remaining bits of the release descriptor are not used.

When a release descriptor is applied to one of the I/O buses (A or B), the descriptor will be received by the busy ("inactive" or "active") I/O module whose address is contained in bits 39 through 42 and by the FNBI/O on the bus. All other busy and non-busy I/O modules on the I/O bus will ignore the release descriptor. The addressed I/O module checks the parity of the release descriptor; if the parity is incorrect (even number of 1 bits), the addressed I/O will ignore the release descriptor and remain busy. If the parity is correct, the I/O module is made non-busy. If the I/O operation is terminated, the I/O module is made non-busy, and a result descriptor is sent back on the operation that was in progress. An in-process would not be sent back after the release descriptor.

The FNBI/O module also checks the parity of the release descriptor word. If the parity of the descriptor is incorrect, the FNBI/O returns an in-process descriptor which is a copy of the release descriptor with up date status of 111 in bits 17, 18, and 19 in order for the program to check the operation and see that there was an error.

It should be noted that only the FNBI/O will return an in-process descriptor following a release descriptor and that the in-process descriptor will be returned only if a parity error is detected. Therefore, if all I/O modules on the I/O bus are busy ("inactive" or "active") when the release is sent, no in-process descriptor will be returned to memory (whether a parity error exists in the release descriptor or not).

COMMAND DESCRIPTOR: The command descriptor is a coded instructor that is sent to the I/O module to initiate and control an Input/output operation with a selected terminal device. Once the command descriptor has been received with correct parity, the I/O module will select the specified terminal device and will control the data transfer operation without the aid of the computer module. This will allow the computer module to be utilized in performing other tasks within the program such as mathematical computations, performing other program routines, etc.

Prior to sending a command descriptor to one of the I/O buses (A or B), a release descriptor would have been sent to a specified I/O module to make it non-busy so that it would be able to receive the command descriptor. The selected I/O module would be the only I/O on the bus in a non-busy state, thus making it the FNBI/O on the bus.

When a command descriptor is transmitted to the I/O bus (A or B) it will be received by the FNBI/O only; all other I/O modules will ignore the command descriptor. If all I/O modules on the I/O bus are busy, the transmission of the command descriptor will be unsuccessful.

When the FNBI/O receives the command descriptor, the parity of the command descriptor is checked, and the FNBI/O will attempt to engage the designated terminal device so that the I/O operation is initiated. If the parity is incorrect or if the specified terminal device is not available, appropriate I/O status information is inserted into the in-process descriptor before it is returned to the descriptor list in memory. If the command descriptor is received with correct parity and the terminal device is available, an in-process descriptor will be returned to the descriptor list with appropriate status so the program can check and see that the operation was initiated successfully. Whether the command descriptor is received satisfactory by the FNBI/O or not, the I/O module is placed in the busy state. Once the command descriptor is received satisfactory and the terminal device is available, the FNBI/O will initiate and control the operation as specified by the coded information in the different sections of the command descriptor. (C1502 card). The I/O operation now proceeds at a rate determined and controlled by the terminal device in use and continues until a condition for terminating the operation occurs (normal completion, malfunction occurs, or a release descriptor). The I/O module then releases the terminal device and returns a result descriptor to the descriptor list in memory which contains status bits that indicates whether or not the operation was completed successfully. The Command Descriptor format is shown on C1502-416M-SU, Descriptor Code Card.

Bits 1-12, indicates the number of words to be transferred. Maximum of 4096 words per command descriptor. As each word is transferred the word count is down counted by 1. Normal termination of operation in word count = 0. The signal to down count the word counter is developed in the I/O module as each word is transferred.

Bits 13-16, indicates the number of records to be transferred. The maximum number of records for a command descriptor is 16. As each record is transmitted the record count is down counted by 1. The record count is used on input operations only. When the record count is down counted to 0 during an input operation, regardless of word count, the operation will be terminated. The signal to down count the record count is developed in the terminal device and sent to the I/O module to down count the record count.

Bits 17 through 20 of the command descriptor are not used.

Bits 21 through 24 are used to specify the memory module where the data will be transferred to or taken from. It should be noted that the address for the memory modules will be address 0 through 7 for BUIC III configuration.

Bits 25 through 36 contains the starting word location in the specified memory module for the first word to be transferred. As each word is transferred through the I/O module the word location address is up counted by 1 to indicate the new address for the next data word to be transferred.

It should be noted that if the word address is maximum (7777_8) and a word is transferred, the next word location will be address $0000_{(8)}$ and the memory module number would be increased to the next higher number module so that the next data transfer would go to memory address $0000_{(8)}$ of the next memory module.

Bit 32 is not used in the command descriptor.

Bit 38 is used when a priority (A) is needed to allow the I/O module a higher priority in obtaining access to the bus when simultaneous command descriptors are made by different I/O modules for access to the bus. Priority A operations are normally given for only the higher speed terminal devices to reduce the possibility of I/O terminations due to "data too slow" device error status.

The I/O module has a higher priority access to the buses for transfer of data with the memories than the central computer modules.

Bits 39 through 43 designate which terminal device is to be used in the I/O operation. These five bits, in conjunction with bit 44 of the operation type, specify any one of 64 possible terminal devices. In the maximum complement of terminal devices, not more than 32 output devices and 32 input devices can be utilized within the BUIC III System. The total input or output device can be comprised of simple (one-way) and complex (two-way) terminal devices. Each simple device will be assigned a device number. Each complex terminal device will have two device numbers associated with it, one number for input operation and a different number for output operation. For example, the "Flexowriter" unit is a complex device. When used for output operation the device number for the Flexowriter is 0. (Bits 39 through 44 = 000000). For an input operation the Flexowriter is device number 1. (Bits 39 through 44 = 000001).

C1502-416M-SU, Descriptor Code Card contains the device number (in decimal) assigned for each terminal device and the descriptor format (in octal) for selecting each terminal device in the BUIC NACC System.

Bit 44 of the command descriptor defines the type of operation to be performed (input or output). Bit 44 = Ø indicates an output operation, Bit 44 = 1 indicates an input operation.

Bit 45 of the command descriptor defines the type of device to be used (simplex or complex). Bit 45 = Ø indicates a simplex device, Bit 45 = 1 indicates a complex device.

The operation type area (bits 44 & 45) of the command descriptor designate the following type of operation:

a. 00 = output operation with simplex device

b. 01 = output operation with complex device

c. 10 = input operation with simplex device

d. 11 = input operation with complex device

Bits 46 through 48 of the command descriptor indicate the specified operation to be performed. These three bits of the operation code enables eight different variations to be specified for a device when needed. However, most terminal devices do not use all of the operation codes, and the meanings of the different codes vary with the device as pointed out in C1502-416M-SU, Descriptor Code Card.

Two limitations exist with regard to operation codes for simplex input terminal devices. The first limitation is that operation codes 000 and 001 cannot be used, because together with bits 44 and 45 (10) they form the description type codes of the release descriptor (10000 for bits 44 through 48) and the setup descriptor (10001 for bits 44 through 48). Any other combination in bits 44 through 48 is decoded by the I/O module as a command descriptor. The second limitation is that the operation code is sent from the I/O module to the terminal device over the output data lines, and a simplex input device has no output data lines from the I/O module to the terminal equipment. Operation codes 010, 100, 101, 110, and 111 are all permissable for the simplex input device operation and will result in the same basic operation to be performed. Usually the simplex input device operation code is given only as 010.

At the termination of the operation (normal termination of word counter or record counter equal 0 or malfunction status termination) the I/O module will return a result descriptor to the specified memory location in order that the program can analyze the operation to determine if the operation was successful.

HARDWARE-GENERATED DESCRIPTORS: The hardware-generated descriptors, the in-process and result descriptors, provide the program with a means of checking to see if the operation specified was initiated and if the operation was successfully completed or if there was a malfunction during the operation.

IN-PROCESS DESCRIPTOR: The in-process descriptor is formed by an I/O module after the reception of a setup, release or command descriptor and is returned to the memory module address specified by the setup descriptor DBAR bits in conjunction with the four pre-wired I/O module bits and the DRC flip-flop within the I/O module. The memory address location for the in-process descriptor is referred to as the "A" list location of the descriptor list in core memory. The format of the descriptor list is shown on C1502-416M-SU, Descriptor Code Card.

In-Process Descriptors following the receipt of setup descriptor are illustrated on C1502-416M-SU, Descriptor Code Card.

The FNBI/O will return a full in-process descriptor to the appropriate "A" list in memory. This will be a copy of the setup descriptor with I/O module status placed in bits 17 through 19:

a. Bits 17 through 19 = 001 Setup received correctly. I/O module is now "inactive" busy and will remain until released by a release descriptor.

b. Bits 17 through 19 = 111 Setup received with parity error from memory. I/O module is "inactive" busy and will remain until released by a release descriptor.

All other non-busy I/O modules (NBI/O) will return an in-process descriptor that contains only the DBAR setting in bits 1 through 11 and I/O module will become "inactive" busy and remain until released by a release descriptor.

If all I/O modules on the bus are busy upon receipt of a correct setup descriptor all I/O modules will load the new DBAR address and do not return an in-process descriptor. If all I/O modules are busy upon receipt of an incorrect setup descriptor, all I/O modules ignore the setup descriptor and no in-process descriptor is returned to the "A" list in memory. In-process Descriptor following receipt of a release descriptor is illustrated on C1502-416M-SU, Descriptor Code Card.

It should be noted that an in-process descriptor following a release descriptor is sent back by the FNBI/O module only and only if a parity error existed. The in-process will be a copy of the original setup descriptor with I/O module status of parity error inserted in bits 17 through 19 of 111. Bits 20, 37, and 38 will contain 000 which is used for inserting status detected by a terminal device.

If all I/O modules are busy upon receipt of a release descriptor, no in-process des-

criptor will be returned to memory (whether a parity error exists in the release descriptor or not). When the release descriptor is received by the specified I/O module ("inactive busy") with correct parity, the I/O operation is terminated, the I/O module is made non-busy, and a result descriptor will be returned to the descriptor list. An in-process descriptor will not be returned from an "active" busy addressed I/O module.

It should be noted that under normal operation, a release descriptor is sent to an "inactive" busy I/O module to make the addressed I/O module non-busy. If the release descriptor is received with correct parity by the addressed I/O module there is no in-process descriptor returned to the descriptor list in memory.

IN-PROCESS DESCRIPTOR FOLLOWING COMMAND DESCRIPTOR: The in-process descriptor is formed by an I/O module after the receipt of a command descriptor and is returned by the I/O module to its corresponding "A" list location in memory. The I/O module uses a copy of the command descriptor and places appropriate status information in the copy of the command descriptor that is sent to memory as the in-process descriptor.

Normally, the I/O module that the command descriptor is going to be sent to will be the I/O module that just received a release descriptor so that this module will be the FNBI/O module on the bus. Upon receipt of the command descriptor the FNBI/O module is made busy and returns an in-process descriptor to the appropriate "A" list in memory. The parity of the command descriptor is checked by the FNBI/O module. If parity is correct, the FNBI/O will attempt to engage the designated terminal device so that the operation is initiated. If the specified terminal device is not available, or if the command descriptor was received with incorrect parity, approriate I/O status information is inserted into the in-process descriptor before it is returned to the descriptor list. The three possible I/O status codes that can appear in an in-process descriptor following a command descriptor are:

a. Bit 17, 18 and 19 = 000 Command descriptor received satisfactorily and terminal device is available so that operation will be initiated.

b. Bit 17, 18 and 19 = 001 Command descriptor received satisfactorily. However, the terminal device was not available. The terminal could have been busy with another I/O module or not ready. The I/O module will become "inactive" busy and remain so until released by a release descriptor.

c. Bits 17, 18 and 19 = 111 Parity error in command descriptor word received from memory. The I/O module will become "inactive" busy and remain so until released by a release descriptor.

The format for the In-process Descriptor following a Command Descriptor is shown on C1502-416M-SU, Descriptor Code Card.

RESULT DESCRIPTOR: The result Descriptor is formed in the active busy I/O module by updating the original command descriptor stored in the I/O module descriptor register so that at all times the descriptor word shows the extent of completion of the operation. A result descriptor is returned to the appropriate "C" list location of the descriptor list in memory whenever an operation is terminated. A result descriptor will occur under the following three conditions:

a. Normal completion of the operation. Word counter or Record counter = 0.

b. Existence of an error status condition during the operation. Parity error in a data word being transferred or equipment malfunction.

c. Release of an I/O operation in progress by a release descriptor.

The result descriptor codes that can appear are illustrated on C1502-416M-SU. Note that status can be detected by the I/O module and is placed in I/O status bits (17, 18 and 19) and can be detected by the terminal device and is placed in T.E. status bits (20,

37 and 38)). When an I/O operation is terminated, the I/O module inserts status information into the I/O or T.E. status bits of the descriptor word, releases the terminal device and returns the result descriptor to the "C" list in memory.

The format of a result descriptor is basically the same as the command descriptor except for the status bits. If an I/O operation reaches normal completion or if it is released prior to completion, the word count and record count areas of the result descriptor indicates the extent of completion of the operation by specifying the number of words and records remaining to be transferred. The memory module address and current memory address areas of the result descriptor will normally indicate the address in memory where the data word would be removed from or stored in when the terminating error occurred. The format of the result descriptor following a command descriptor is illustrated on C1502-416M-SU, Descriptor Code Card.