

P A R T I I I

**Space Shuttle Advanced System/4 Pi
Input/Output Processor (IOP)**

**Principles of Operation for the
Bus Control Element**

Prepared for

Rockwell International Corporation
Space Division
12214 Lakewood Blvd.
Downey, CA 90241

26 April 1976

Under Purchase Order No.
MAJ7XMA-483019

IBM No. 6246556A



Federal Systems Division, Owego, New York 13827

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY FORM OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPYING, RECORDING, OR BY ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM.

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
1.0	BUS CONTROL ELEMENT-----	1
1.1	FORMATS-----	4
1.1.1	BUS FORMATS-----	4
1.1.2	ADDRESSING AND INSTRUCTION FORMATS-----	7
1.2	BCE REGISTERS-----	12
1.2.1	BCE PROGRAMMABLE REGISTER-----	12
1.2.2	BCE STATUS REGISTERS-----	13
1.3	BCE IMPLEMENTATION-----	17
2.0	GENERAL BCE OPERATION-----	19
2.1	HALT STATE-----	19
2.2	WAIT STATE-----	22
2.3	BUSY STATE-----	23
2.3.1	BUSY STATE OPERATING MODES-----	24
2.3.2	SUMMARY OF ERROR MODES-----	24
3.0	BCE INSTRUCTIONS-----	28
3.1	BCE REGISTER INSTRUCTIONS-----	29
3.2	BCE BRANCHING-----	36
3.3	BCE TRANSMISSION INSTRUCTIONS-----	42
3.3.1	TRANSMIT COMMAND INSTRUCTIONS-----	42
3.3.2	TRANSMIT DATA INSTRUCTIONS-----	42
3.3.3	TYPICAL BUS TIMING-GAPS BETWEEN OUTPUTS-----	43
3.3.4	ECHO BACK-----	44
3.3.5	ERROR MODES-DISABLED MIA-----	47
3.3.6	ERROR MODES-EXCESSIVE CONCURRENCY-----	47
3.4	BCE RECEIVE DATA INSTRUCTIONS-----	58
3.4.1	MIA-MIA BUFFER-BCE OPERATION-----	58
3.4.2	RECEIVE DATA SETUP-----	65
3.4.3	ACCEPTANCE OF FIRST INPUT-----	66
3.4.4	ERROR CHECKS-----	68
3.4.5	HANDLING OF GOOD INPUTS-----	68
3.4.6	ERROR TERMINATION-FAULTY INPUT-----	69
3.4.7	ERROR TERMINATION-TIME OUT-----	69
3.4.8	RECEPTION OF INTERMEDIATE WORDS-----	70
3.5	SPECIAL INSTRUCTIONS-----	77
4.0	LISTEN MODE-----	81
4.1	SAMPLE OPERATION IN LISTEN MODE-----	81
4.2	INITIALIZATION INTO LISTEN MODE-----	85
4.3	DIFFERENCES IN INSTRUCTION EXECUTION-----	85
<u>Appendix A</u>		
	IOP BCE INSTRUCTION REPERTOIRE-----	88

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY MANNER FOR NON-ACADEMIC PURPOSES ONLY

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
1.1	BUS Message Formats-----	6
1.2	BCE-Computed Main Memory Addresses-----	8
1.3	Basic BCE Short Instructions Formats-----	8
1.4	BCE Instructions-Long Formats-----	9
1.5	BCE Local Store Usage-----	18
2.1	BCE States-----	21
3.1	Bit Times of Listen Command-----	40
3.2	Wait For Index-----	41
3.3	Typical Transmit Sequence-----	45
3.4	MOUT Operation-----	57
3.5	MIA-MIA-Buffer-BCE Operation-----	60
3.6	#RDS, #RDLI, #RDL Setup-----	61
3.7	#MIN Setup-----	62
3.8	Receive Data Algorithm-First Input-----	63
3.9	Main Receive Loop-----	64
4.1	Listen Mode Configuration-----	83

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY MANNER, NOR USED IN ANY MANNER.

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

LIST OF TABLES

<u>Table</u>	<u>Page</u>
1.1 BCE Characteristics-----	3
1.2 BCE Status Register-----	14
2.1 Summary of BCE-Related Errors-----	25
3.1 Typical Gaps Between Command and Data Word-----	46
3.2 Time To First Look At Data-----	67
4.1 Differences in Instruction Executions Due to BCE Mode-----	87

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE).
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY MANNER, NOT PLACED IN ANY REPOSITORY

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

1.0 BUS CONTROL ELEMENT

The Bus Control Element (BCE) is a microprogrammed controller specifically tailored for management of I/O traffic on one of the Space Shuttle system busses. Within each IOP there is one BCE for each system bus, for a total of 24 BCE's. Each of these BCE's is capable of independent program execution, data buffering to and from memory, and communication with the MSC. Further, each BCE is connected to its own bus via its own Multiplexer Interface Adapter (MIA), which performs all parallel to serial and serial to parallel conversions. Table 1.1 summarizes the basic characteristics of a BCE.

The major purpose of a BCE is threefold:

- (1) Initiate transmission of commands to subsystems on the bus.
- (2) Handle data coming back from a commanded subsystem.
- (3) Fetch data to be sent to a commanded subsystem.

To handle these tasks there are two classes of instructions (transmit and receive), and two special operating modes (Command, and Listen) that are unique to the BCE.

The transmit instructions allow transmission of both commands and data to a subsystem. When transmitting data a BCE/MIA pair performs:

- (1) Update of main memory buffer addresses.
- (2) Conversion between 32 bit main memory data format and 25 + Sync bit bus data format.
- (3) Check on number of words to be transferred.

The receive commands allow a BCE to accept a stream of input data from a subsystem through its MIA. When receiving data a BCE performs:

- (1) Time outs on data arrival.
- (2) Error checks on incoming data.
- (3) Assembly into main memory 32 bit data format.
- (4) Maintenance of main memory buffer addresses.
- (5) Transferral of data to main memory.
- (6) Check on number of words to be received.

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY FORM OR BY ANY MEANS

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

The two operating modes that a BCE may be in influence the way the BCE uses its bus. In Command mode, a BCE is master of its bus, and is free to transmit both commands and data. This allows a BCE to command a subsystem, receive data from it, or transmit data to it. In Listen mode a BCE monitors its bus for directions on how to handle any data that might appear on the bus. In this mode a BCE may only receive data, and may not transmit either commands or data. This handles the common situation in the Space Shuttle where several IOP's and thus several BCE's may be connected to one bus. In such a situation only one BCE is allowed to issue subsystem commands, but all BCE's on that bus wish to receive copies of the resulting data. The listening mode allows the command BCE to tell the others what data to expect, and when to expect it.

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY MANNER, NOR BE USED IN ANY MANNER.

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

TABLE 1.1
BCE CHARACTERISTICS

Type -- Programmable I/O traffic controller
Number -- One per bus, 24 BCE's per IOP
Control Structure -- Microprogrammed

Programmable Registers (per BCE)

18 Bit Base Register (BASE)
18 Bit Program Counter (PC)
18 Bit Maximum Time Out Register (MTO)
5 Bit Interface Unit Address Register (IUAR)
1 Bit BCE/MSD Indicator Bit

Other BCE Registers (per BCE)

32 Bit Status Register
1 Bit Program Exception Register (part of STAT 1)
1 Bit Busy/Wait Bit (part of STAT 4)
1 Bit MIA Transmitter Enable
1 Bit MIA Receiver Enable
6 Bit Identify Register

Instruction Formats : 16 Bit Short/32 Bit Long/ 64 Bit Extended

Instruction Repertoire: 10 Short/5 Long/ 2 Extended

Addressing Space: 131,072 32 Bit Fullwords/262,144 16 Bit Halfwords

Addressing Modes: Immediate, PC relative, Base relative, Absolute

Special Operating Modes: Command, Listen

Bus Data Format: 25 + Sync Bit serial

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY MANNER OR PLACED IN ANY REPOSITORY

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

1.1 FORMATS

1.1.1 Bus Formats

There are four basic formats for data or commands carried over a system bus in serial form. These are pictured in Figure 1.1. They are all 28 bits long, with 3 bit times for sync, 24 bits of information, and 1 bit for word parity. These 28 bits are transmitted at a serial rate of 1 bit per microsecond.

On transmission from an IOP to a subsystem, a BCE provides the middle 24 bits of information and an indication of the type of sync to use -- either command or data sync. Command sync is used when the 24 bits of information are to be treated as a command to be obeyed by some subsystem. Data sync is used when a BCE has previously conditioned a device, via a ccommand, to accept a stream of data, and the word being put on the bus is a member of this stream.

Conversely, there are situations where a BCE may accept, through its MIA, words with either command or data sync. In such cases the MIA simply provides the type of sync that a word had, the 24 bits of information present in the word, and appropriate error signals. Command sync is recognized when a BCE is in the "Listening Mode" and is expecting a command from another BCE connected to the same bus. Data sync is accepted when a BCE is expecting to receive a stream of data from a subsystem on the bus. This subsystem was previously commanded to send this data by a message with command sync from some BCE on the bus.

The contents of the 24 bits of information in a bus word depends both on the sync type and the direction of transfer -- from a BCE or to a BCE. In all cases the upper 5 bits contain an interface unit address (IUA). When used in data words or commands to subsystems, these 5 bits identify the subsystem on the bus who either originated the data or is to accept the command or data. In Listen Mode commands, the BCE in one IOP sends to all other BCE's in the other IOP's connected to this bus a command that has a "Common IOP address". This special pattern is not used by any subsystem, and allows a listening BCE to distinguish between listen commands and commands to subsystems.

In data transfers the remaining 19 bits consist of 16 bits of data (one half of a standard 32 bit main memory fullword) and 3 bits nominally containing the pattern 101. In BCE to subsystem data transfers, this pattern never changes. However, in subsystem to BCE data transfers, any variation in the 101 pattern indicates to the BCE that the sending subsystem has encountered a problem, such as power surge, invalid commands, etc.

In commands sent by a BCE to a subsystem, the format of the lower 19 bits of information is subsystem dependent, and not discussed further here.

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED
REPRODUCED IN ANY FORM, FOR ANY PURPOSE, WITHOUT THE WRITTEN PERMISSION OF THE
WICHITA STATE UNIVERSITY LIBRARIES

MS 87-08

Box 42

FF 50

In Listen Mode commands sent from one IOP to another, the 18 bits of the information contain a 5 bit number representing a terminal or subsystem on the BCE's bus and an 8 bit index into a table of BCE branch addresses.

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIE

SPECIAL COLLECTIONS

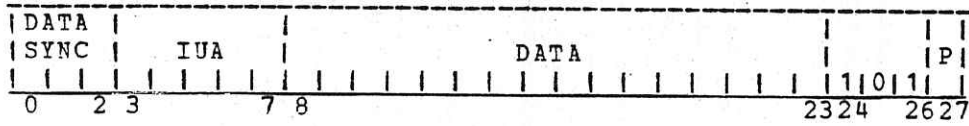
WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED
REPRODUCED IN ANY MANNER, OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPYING, RECORDING, OR BY ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM.

MS 87-08

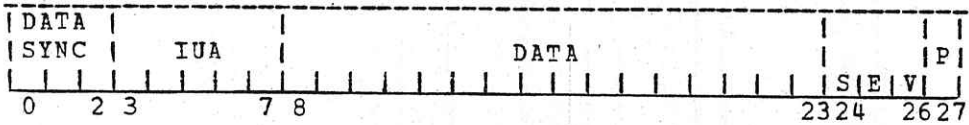
Box 42

FF 50

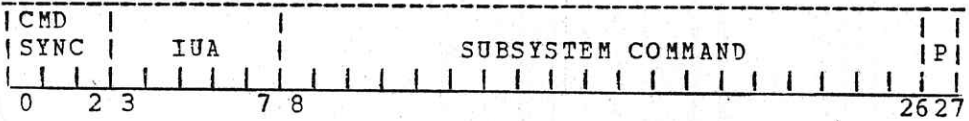
Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives



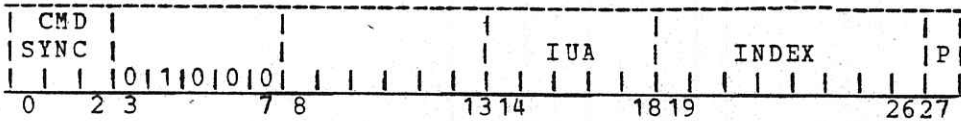
(a) DATA FORMAT -- FROM BCE TO SUBSYSTEM



(b) DATA FORMAT -- FROM SUBSYSTEM TO BCE



(c) COMMAND FORMAT -- BCE TO SUBSYSTEM



(d) COMMAND FORMAT -- BCE TO BCE (LISTEN MODE)

IUA = INTERFACE UNIT ADDRESS

Figure 1.1. Bus Message Formats

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE).
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED
 REPRODUCED IN ANY MANNER OR BY ANY MEANS ELECTRONIC OR MECHANICAL INCLUDING PHOTOCOPYING, RECORDING, OR BY ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

1.1.2 Addressing and Instruction Formats

A BCE may directly address up to 262,132 16-bit halfwords or 131,071 32-bit fullwords. To achieve this, all main memory addresses computed by the BCE are represented as 18-bit absolute numbers, as pictured in Figure 1.2. The upper 17-bits (bits 0 through 16) represent the fullword location, and the lowest bit (bit 17) the halfword portion of the addressed fullword. A 0 in this lower bit refers to bits 0 - 15 of the 32-bit fullword; a 1 refers to bits 16 - 31. When used as a fullword address, bit 17 is ignored. Thus, H'276' and H'277' refer to the same fullword.

There are four basic instruction formats used by the BCE and they are depicted in Figures 1.3 and 1.4.

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY:

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

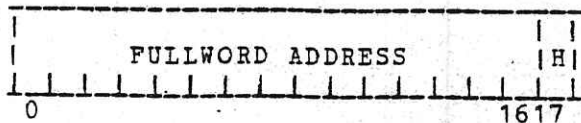
WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED
REPRODUCED IN ANY MANNER, NOR PLACED IN ANY REPOSITORY

MS 87-08

Box 42

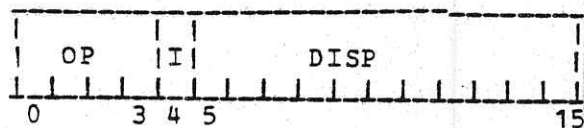
FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives



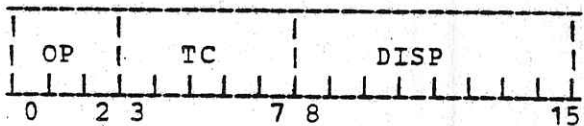
H = HALFWORD

Figure 1.2. BCE-Computed Main Memory Addresses



I = INDEX SPECIFICATION OR OPX

(a) SHORT FORMAT 1



(b) SHORT TRANSMIT/RECEIVE DATA FORMAT

Figure 1.3. Basic BCE Short Instruction Formats

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED OR

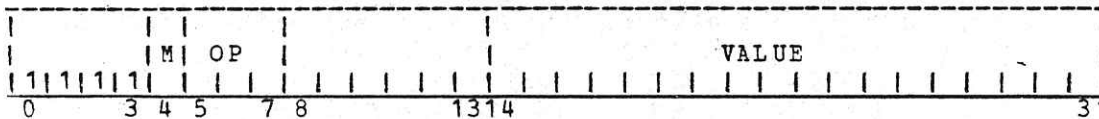
REPRODUCED IN ANY MANNER WITHOUT PERMISSION

MS 87-08

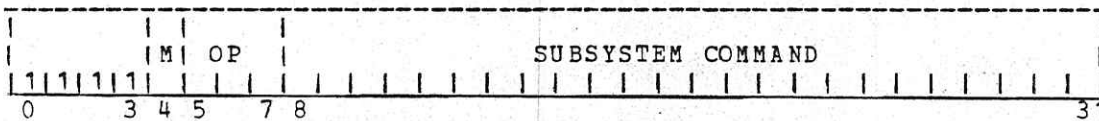
Box 42

FF 50

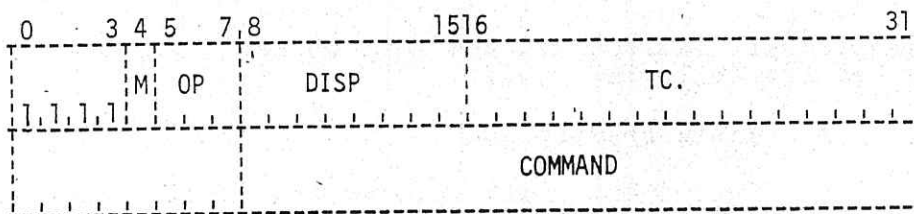
Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives



(a) STANDARD FORMAT



(b) COMMAND INSTRUCTION FORMAT



(c) EXTENDED FORMAT

Figure 1.4. BCE Instructions - Long Formats

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE PROTECTED BY COPYRIGHT LAW (TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARY SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE REPRODUCED IN ANY FORM OR BY ANY MEANS

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

Short format 1 is used primarily by instructions dealing with the BCE register. It has the following fields:

Field	Field Description
OP	This 4-bit field defines the basic operation to be performed by the BCE.
I	This bit serves either as an opcode extension or as an index mode specification in address generation.
DISP	This 11-bit field serves either as immediate data or as a PC relative address displacement.

Short format 2 is used by instructions that transmit and receive data. It has the following fields:

Field	Field Description
OP	This 3-bit field defines the basic operation to be performed (data read or write).
TC	Transfer Count. This field defines the number of inputs or outputs to be handled.
DISP	This 8-bit field serves as a displacement off of the BCE's base register in the computation of the main memory buffer addresses associated with the I/O data.

Most long format instructions use long format 1 (Figure 1.4). Long format provides the following fields:

Field	Field Description
OP	This 3-bit field defines the basic operation to be performed by the BCE.
VALUE	This 18-bit address is used either for immediate data or as an address.
M	This bit influences how the Value field is used.

Long format 2 is used by the Transmit Command instruction to provide 24 bits to be given to the MIA for command transmission.

Extended format 3 is used by the Message In/Out instructions to specify Displacement, Transfer Counts and Commands.

In many BCE instructions the direct addressing mode includes automatic indexing by twice the BCE's number. This allows BCE programs to be written in a table driven fashion, where the same BCE program can be used by many different BCE's, and yet each BCE will use a separate set of parameters in the programs execution.

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY
WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY MANNER

MS 87-08 Box 42 FF 50
Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

1.2 BCE REGISTERS

1.2.1 BCE Programmable Register

Each BCE contains several registers under direct program control. They are:

BASE --	An 18-bit Base Register
PC --	An 18-bit Program Counter
MTO --	An 18-bit Maximum Time Out Register
IUAR --	A 5-bit Interface Unit Address Register
Indicator --	A 1-bit indicator bit in the BCE/MSC Indicator Register.

The BASE is used in locating I/O buffers in main memory. All such buffers are base-relative, allowing the same BCE program to be used with different buffers by simply changing the Base before entering the common program segment.

The Program Counter is an 18-bit register indicating the main memory halfword or fullword location of the BCE instruction being executed by this BCE. It should be emphasized that there is a separate PC for each BCE, and that the contents of one PC need not have any relation to the contents of a different PC in another BCE.

The Maximum Time Out Register is used primarily during the reception of data to indicate the maximum time a BCE should wait for a subsystem to respond to a command with the beginning of a stream of input data. This time is defined as the "latency" of the subsystem. Failure of a subsystem to respond within this period of time results in a BCE declared error condition.

The Interface Unit Address Register contains the 5-bit subsystem address of the subsystem presently in communication with the BCE. This address is derived when a command is issued by a BCE (Figure 1.1(c)), and is used in the construction of data words to be sent to a subsystem (Figure 1.1(a)), and in checking for proper subsystem response when data words from a subsystem are being received (Figure 1.1(b)).

The BCE/MSC Indicator Register has 1 bit associated with each BCE. A BCE is free to set or clear this bit under program control. The MSC can read all such bits and monitor when various BCE's have set or cleared their bits. The MSC may also reset a BCE's Indicator bit via an @RBI instruction. This provides a means for BCE's to signal to the MSC the occurrence of various events such as execution of listening mode programs.

The detection of various errors will also set a BCE's Indicator bit to 1 (See Paragraph 2.2).

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED
REPRODUCED IN ANY MANNER OR IN ANY MEDIUM

MS 87-08

Box 42

FF 50

1.2.2 BCE Status Registers*

The modes and status of each BCE is recorded in the following registers:

- (1) BCE Busy/Wait Bit -- For BCE *i* this is bit *i* of the Busy/ Wait Register (STAT 4). A 1 in this bit indicates that the BCE is busy executing a program located in main memory. A 0 indicates it is not busy.
- (2) BCE Program Exception Bit -- For BCE *i* this is bit *i* of the Program Exception Register (STAT 1). A 0 indicates that the BCE has encountered some problem in the execution of a BCE program. A 1 indicates that no problem was encountered. The Status Register associated with that BCE contains a description of the cause of the problem.
- (3) Transmitter Enable Bit -- For BCE/MIA *i* this is bit *i* of the MIA Transmitter Enable Register. A 0 in this bit prevents the BCE from issuing a command or transmitting data over the bus to which its MIA is connected. This bit may be changed only by a PCO from the CPU.
- (4) Receiver Enable Bit -- For BCE/MIA *i* this is bit *i* of the MIA Receiver Enable Register. A 0 in this bit prevents the MIA from transferring any data or commands it receives on the bus to the BCE. This bit may be changed only by a PCO from the CPU.
- (5) BCE Status Register. Each BCE maintains a unique 32-bit status register describing what, if any, errors the BCE has encountered during the execution of a program. Table 1.2 describes the format used in these registers.
- (6) BCE Identity Register. Each BCE maintains a register that contains twice its own BCE number. This register is used in computing addresses during direct mode BCE instruction. The register is set while the BCE is leaving the Halt state and entering the Wait state, and is not altered by the BCE at any other time.

*PROGRAMMING NOTE

The BCE Status Registers should not be changed via PCO commands while the BCE is busy.

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY MANNER, NOR PLACED IN ANY REPOSITORY

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

TABLE 1.2
BCE STATUS REGISTER

			M	P	S	E	V			I	U	A			S						G	S	X		I	T	B	B			
																						A	T	M		T	O	O	T	A	I
0	2	3	4	5	7	8				12	13	14	15	16							21	22	23	24	25	26	27	28	29	30	31

BIT	
31 - 30	Reserved
29 - I	Illegal Opcode -- This BCE encountered an illegal instruction in the execution of a program.
28 - BA	Boundary Alignment Error -- This BCE encountered a long format instruction on an odd halfword boundary.
27 - BTO	Block Time Out--A Receive Data Instruction timed out while waiting for an interblock gap to end.
26 - TO	Time Out -- A Receive Data Instruction timed out while waiting for a data word to arrive. This time out occurred on data inputs other than the first.
25 - ITO	Initial Time Out -- A Receive Data Instruction timed out while waiting for the first input word to arrive.
24	Reserved
23 - XMT	Transmitter Disabled -- At some point in the execution of a Transmit Data instruction the MIA associated with this BCE had its transmitter disabled. This bit is also

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE).
COPY PROVIDED BY
WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS

MS 87-08 Box 42 FF 50
Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

set if the MIA was found busy when it was time to initiate transmission of a new data word.

- 22 - ST Self Test Error--A BCE Self Test Instruction has detected a fault in the BCE.
- 21 - GAP Gap of greater than 20 usec. occurred during execution of a transmit data instruction, or 5 usec during a MOUT.
- 20 - 16 Reserved.
- 15 - S Sync Error -- While executing a Receive Data Instruction, an input word with command sync was received. (See #RDS Instruction)
- 14 - 13 Reserved
- 12 - 8 Subsystem Address -- This field is the logical "OR" of the received subsystem addresses of all input words that were detected to have errors during execution of previous Receive Data Instrs.
- 7 - 5 SEV -- This field is the logical "OR" of the SEV bits of all input words that were detected to have errors during execution of previous Receive Data Instruction. The S and V bits were inverted before the "OR". Thus, any pattern other than 101 will be recorded in these bits.
- 4 - P Parity -- While executing a Receive Data Instruction, an input word with bad parity was detected.
- 3 - M Signature Mismatch -- While executing a Receive Data Instruction, a mismatch between the

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY MANNER WITHOUT
PERMISSION FROM THE LIBRARY

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

input's IUA and the BCE's IUAR. The input IUA was simultaneously 'OR'ed into bits 8 - 12 of the status word.

Reserved.

2 - 0

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR REPRODUCED IN ANY MANNER IN ANY MEDIUM OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPYING, RECORDING, OR BY ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM.

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

1.3 BCE IMPLEMENTATION

Any BCE implemented within an IOP consists of:

- (1) A segment of Local Store. This consists of 4 words from each of Banks A and B, and 8 words from Bank C.
- (2) A bit in the IOP Busy/Wait Register.
- (3) A bit in the IOP Program Exception Register.
- (4) A bit in the IOP Halt Register.
- (5) A bit in the BCE-MSI Indicator Register.
- (6) The associated MIA.
- (7) A buffer location in the MIA Buffer.
- (8) A bit in the MIA Transmitter Enable Register.
- (9) A bit in the MIA Receiver Enable Register.
- (10) One of the Microprogram Counters. This contains the ROM address of the next micro instruction to be executed for this BCE.

As described in the IOP Functional Description (IBM No. 74-A31-016;), the operation of all BCE's and the MSI is time-shared. Each BCE executes its next microinstruction from ROS at intervals of one every 16.5 microseconds. During other periods of time the only BCE-related operations still carried on are previously requested memory operations, MIA reception/transmission of data, and CPU directed PCI/O.

The general makeup of a BCE's Local Store segment is shown in Figure 1.5. All unlabelled words are used as temporary or working registers during instruction execution.

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)
COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY MANNER NOR PLACED IN ANY REPOSITORY

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

BANK A	BANK B	BANK C	WORD
			0
			1
PC	IH	IL	2
ID	MTO	BASE	3
		IUAR	4
		STATUS HIGH	5
		STATUS LOW	6
			7

PC = Program Counter
 IH = High half of Instruction Register
 ID = BCE Identify Register
 IL = Low Half of Instruction Register
 MTO = Max. Time Out Register
 BASE = Base Register
 IUAR = Interface Unit Address Register
 STATUS HIGH = High Half of Status Register
 STATUS LOW = Low Half of Status Register

Figure 1.5. BCE Local Store Usage

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY MANNER, NOR PLACED IN ANY PUBLIC DOMAIN

2.0 GENERAL BCE OPERATION

During normal operation a BCE can be in one of three states: Halt, Wait and Busy. In the Halt state the BCE is physically restrained from performing any operations; in the Wait state the BCE is awaiting a command to execute a program; and in the Busy state the BCE is executing BCE programs from main store. Figure 2.1 summarizes these states and the transitions between them.

Typical state transitions are as follows:

- (1) During any system or CPU-directed BCE reset, the BCE is in the Halt state.
- (2) Upon release from the Halt state, the BCE enters the Wait state.
- (3) A signal from the MSC initializes a BCE, and places it in the Busy state.
- (4) In the Busy state, the BCE is executing a program located in main memory. It exits the Busy state only upon execution of a Wait instruction, detection of an invalid instruction or operating error, or some reset signal. In all but the latter case, the BCE re-enters the Wait state; in the latter case it is forced to the Halt state.

The two bits that indicate the current state of BCE are its Halt bit (Bit i of the IOP Halt Register) and its Busy/Wait bit (Bit i of the IOP Busy/Wait Register). In addition, the BCE Program Exception bit indicates if the BCE has found an error during its last period of time in the Busy state. If it has found an error, the BCE Status Register contains a record of the exact error.

The following sections describe each state in detail.

2.1 HALT STATE

The primary purpose of the Halt state is two-fold:

- (1) Allow the external world to reset BCE operation to a known condition.
- (2) Upon the detection of very serious IOP faults (such as failure in the microstore) to isolate the BCE and prevent it from performing potentially erroneous operations.

Entry and exit from the Halt state are controlled by the value of a single "BCE Halt" status bit which is part of the IOP Halt Register. There is one such bit for each BCE. As long as this bit is set, the BCE microprogram counter is forced to point to a micro-

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW

(FILE IN U.S. CODE)
COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY MANNER NOR PLACED IN ANY INFORMATION SYSTEM

MS 87-08

Box 42

FF 50

instruction that performs no operation other than clear the BCE Busy/Wait bit. The Halt bit may be set to 0 at any time, and effectively terminates anything that the BCE is doing.

The signals that set a BCE Halt bit to 0 (Halt) include:

- (1) BCE/MSD disable discrete, Discrete Input 0, or a Master Reset PCO from the CPU.
- (2) Setting of the Fail Latch for this GPC by either the Redundancy Management voting logic or IOP detected serious errors such as ROS parity error.
- (3) A "Halt Processor" PCO from the CPU with a 1 in bit position i (when BCE i is to be halted).

The first two signal classes halt all BCE's.

Exit from this state to the Wait state occurs only when the following occurs:

- (1) BCE Disable Discrete is reset.
- (2) Fail Latch is reset.

and a CPU "Enable Processors" PCO with bit i (for BCE i) set to 1 is present. The PC I/O manual (IBM No. 6246556) describes the PCO that set/reset the Halt register.

Upon any exit from the Halt state, the BCE Program Exception bit is set to one (no errors). The BCE Status Register is cleared to all zeroes, and the BCE Identity Register is set to twice the BCE's number.

Although a BCE's Halt bit may be changed at any time, it does not affect the BCE's operation until the next time that the BCE executes a microinstruction. Consequently, to guarantee that a BCE has been halted, the BCE's Halt bit should be at 0 (halted) for a minimum of 18 usec. before being reset to 1 (enabled). This will guarantee that the BCE has truly been forced to the halt state for at least one BCE microcycle.

For similar reasons, a BCE should not be considered released from the Halt state and in the Wait state the instant its Halt bit is set to 1 (enabled). Release from the Halt state does not begin until the first BCE microcycle after the Halt bit has been reset, and continues for several BCE microcycles (about 100 usec.) as the BCE resets its internal registers and prepares to enter the Wait state. After this transition, the BCE will be in the Wait state, and will perform as described in the next section.

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR REPRODUCED IN ANY FORM OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPYING, RECORDING, OR BY ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM.

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

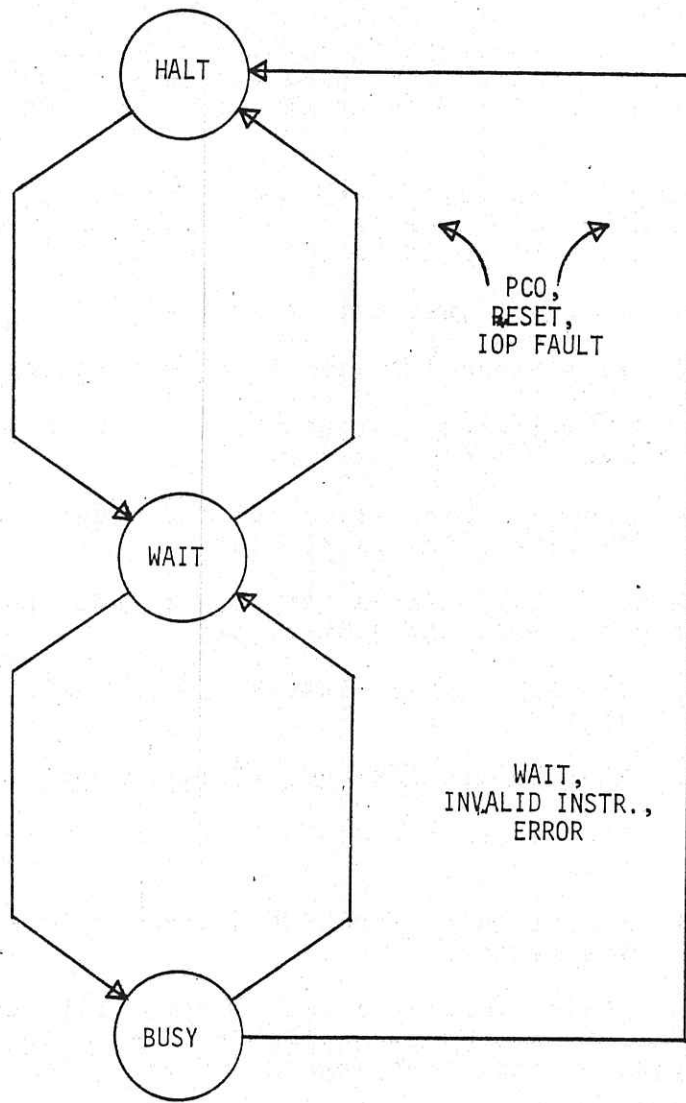


Figure 2.1. BCF States

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE PROTECTED BY COPYRIGHT LAW (TITLE 17 U.S. CODE). COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR REPRODUCED IN ANY MANNER, NOR IN ANY MEDIUM.

MS 87-08 Box 42 FF 50
Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

2.2 WAIT STATE

In the Wait state, a BCE is prepared to be told to perform a BCE program. This wait loop is implemented by a micro-routine that monitors the status of the BCE's Busy/Wait bit. The BCE remains in the Wait state as long as its bit is reset to the "Wait" value (0). When the MSC (via an @SIO) sets this bit to busy, the BCE transits to the Busy state. This transition consists of using the present value of the BCE's Program Counter as the starting address of a BCE program in main store.

Entry to the Wait state occurs either upon exit from the Halt state, as described above, or by a transition from the Busy state. A BCE performs this latter transition when any of the following occurs:

- (1) A "Wait" instruction is executed.
- (2) An instruction with an illegal opcode is encountered.
- (3) A valid long format instruction is found starting on an odd halfword boundary.
- (4) A significant error occurs. (such as in transmit or receive data instruction).

In the final three cases, a BCE performs the following actions before entering the Wait state:

- (1) It sets its bit in the Program Exception Register (STAT 1) to 0.
- (2) It sets its BCE-MSD Indicator bit to 1.
- (3) It records the cause of the error in its own BCE status register.
- (4) It leaves the Program Counter pointing to the offending instruction.

The BCE's Indicator bit is set, upon detection of an error, to aid in handling Listen mode programs where this bit is used to signal completion of some operation to the MSC (via a Repeat on Indicator instruction).

Exit from the Wait state is normally to the Busy state. This is done, by the MSC, via a sequence of MSC instructions that include:

- 1) A "Load BCE Base" instruction to set the BCE's Base register.
- 2) A "Load BCE Program Counter" instruction to set the BCE's Program Counter.

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

- 3) A "Start I/O" instruction to set the BCE's Busy Wait bit.

Note that the CPU can perform the equivalent of the first two functions via PCO's to the BCE's Local Store. Also, via PCO's, the CPU can reset the BCE's Status Register and Program Exception bit. The CPU cannot, however, directly set a BCE's Busy/Wait bit.

Note also that a BCE's Program Counter need not always be set before the MSC sets the BCE to busy, since the BCE Wait instruction (#WAT) when executed, leaves the PC pointing to the next sequential instruction. This next instruction may be programmed as a simple branch to the beginning of the next BCE program segment. In this case, the MSC need only execute an SIO instruction to restart the BCE at the next segment.

While a BCE is in the Wait state, the CPU may perform PCI/O activity without disturbing the BCE.

2.3 BUSY STATE

In the Busy state, BCE *i* is in the process of executing a program out of main store. A value of 1 in the *i*th bit of the IOP Busy/Wait Register indicates this condition.

The Busy state may be entered only from the Wait state, as described in the previous section. When the transition occurs, the BCE uses the value in its Program Counter to fetch the first instruction from memory and start executing it. The BCE continues executing instructions until either a Wait instruction or some kind of invalid instruction is encountered. In either case, the BCE transitions back to the Wait state, again as described in the previous section.

Although an MSC @SIO instruction will set a BCE's Busy/Wait bit to busy within the time frame of the instruction's execution on the MSC, the BCE will not necessarily begin immediate execution of the first instruction. The possible delays include:

- o The time required for the BCE to recognize that its Busy/Wait bit is set (up to 16.5 usec.)
- o The time required to bring out the PC and request the instruction word from memory (up to 33 usec) and,
- o Any time required for the memory read request to reach the DMA channels, be read from memory, and be returned to the BCE.

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED
 REPRODUCED IN ANY MANNER FOR ANY PURPOSES EXCEPT AS MAY BE PERMITTED BY THE LIBRARY

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

The memory read request time delays the BCE setup only when handling time exceeds 16 usec. This situation can occur when several BCE/MSB processors have requested memory reads simultaneously.

While the BCE is in the Busy state, the CPU may execute any PCI without problem. However, all PCO's that write into BCE Local Store should be carefully controlled since the MSB is not aware that this action has occurred, and the resulting BCE program execution may be unpredictable.

2.3.1 Busy State Operating Modes

Once in the Busy state a BCE may operate in one of two modes; Command and Listen. In the Listen Mode the BCE's MIA receiver is enabled, but the BCE's MIA transmitter is disabled. In Command mode both transmitter and receiver are enabled.

The BCE Mode affects the way certain BCE instructions are executed. In the Command Mode, a BCE is in command of its bus and is free to transmit commands and data over the bus. A BCE in the Listen Mode relies on a BCE in another IOP to indicate the operations to be performed and to issue the appropriate commands to subsystems on the bus to perform the operations. Consequently, a BCE in Listen mode will handle instructions that transmit commands in a different manner than a BCE executing the same program in the command mode. Additionally, a BCE in the Listen mode uses a Wait for Index instruction as an instruction to wait for some other BCE in another IOP to provide a signal as to what it should do. This signal is used by the Listening BCE in a table-look up process to determine an appropriate BCE program for execution.

Section 4 describes the differences between Command Mode and Receive Mode in greater detail.

2.3.2 Summary of Error Modes

Table 2.1 summarizes all the errors detectable during a BCE program and the resulting actions. Separate columns indicate which, if any status bits are set, and whether or not the BCE enters the wait state. If any status bits are set, the BCE Program Exception bit is also assumed set to 0, and the BCE/MSB Indicator bit is set to 1.

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED
REPRODUCED IN ANY FORM OR BY ANY MEANS, NOR PLACED IN ANY REPOSITORY

MS 87-08

Box 42

FF 50

TABLE 2.1

SUMMARY OF BCE-RELATED ERRORS

ERROR	Detected By	Status Bit Set?	Action	
			Wait State Entered?	Other Action
Illegal BCE Instruction	Microcode	Bit 29	Yes	
Long Format BCE Instruction on Odd Boundary	Microcode	Bit 28	Yes	
Bad Command Received in #WIX	Microcode/ MIA	None	No	Command ignored. Wait continued.
DMA Parity Error on Instruc- tion Read	Channel Logic	(see Illegal Instruction)		Write all 0's into Instruction register (illegal opcode) CPU interrupt.
DMA Parity Error, Microcode DMA Timeout, or Time Out Queue Overflow on Data read for Transmit Data Instruc- tion	Channel Logic	21	Yes	CPU Inter- rupt
DMA Parity Error, DMA Timeout, or Queue Overflow on other data reads	Channel Logic	None	No	CPU Inter- rupt. BCE does not receive data.
ROS Parity Error, Clock Failure, Termination Latch Set	IOP Hardware	No	No	BCE Reset to halt. CPU Interrupt

Self Test Bce Self Test Bit 22 No BCE Loops
Detected Instruction

Excessive Microcode 21 Yes
Concurrency Time out
resulting
gap in > 20 usec

Transmitter Microcode 23 Yes
Disabled Test
during
transmit
data

Transmitter Microcode 23 Yes
Busy too Test
long

Parity MIA 4 Yes Save IUA
Error on from data.
input data Data not
stored in
MIA Buffer.

Wrong SEV Microcode/ 5,6,7 Yes " "
bits Hardware

Command Microcode 15 Yes " "
Sync in (See #RDS)
Receive
Data

Wrong IUA Microcode/ 3 Yes " "
in incoming Hardware
data

Failure of Microcode 25 Yes
Subsystem time-out
to respond
with data
(1st word)

THIS MATERIAL MAY BE
REPRODUCED IN ANY FORM OR BY ANY MEANS
WITHOUT WRITTEN PERMISSION FROM THE
COPYRIGHT OWNER.

PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
REPRODUCED IN ANY FORM OR BY ANY MEANS
WITHOUT WRITTEN PERMISSION FROM THE
COPYRIGHT OWNER.

MS 87-08 Box 42 FF 50
Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

Excessive interword gap in incoming data

Microcode time out

26,27

Yes

Wrong bit encoding bit count Error.

MIA/Microcode time-out

26

Yes

Data is not stored in MIA Buffer. BCE will time out

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARY

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED, REPRODUCED, TRANSMITTED, OR PLACED IN ANY MEDIUM.

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

3.0 BCE INSTRUCTIONS

The following pages include descriptions of the instructions presently supported by each BCE. The format for the description of each instruction is as follows:

- o The general name of each instruction appears in the upper left of the first page describing that instruction.
- o The assembler abbreviation appears in the upper right hand corner.
- o The format of the instruction, including binary opcode assignments and field designations.
- o A table (where appropriate) relating addressing mode bits to their effect on parameters used in the instruction execution, and how these addressing modes are signalled to the assembler.
- o A textual description of the instruction and its uses.
- o The minimum instruction execution time will not be presented here as it is supplied in Document No. 74-A31-00009A.

Since the IOP is a multiprocessor with up to 24 BCEs and the MSC requesting memory access independently of each other, it is possible that a memory request from a BCE can be delayed due to the servicing of earlier requests from other IOP processors. If these delays become significant (become greater than about 16.5 usec) then the BCE will be forced to wait for the memory operation to complete in increments of 16.5 usec. This will of course increase the instruction times.

These descriptions of instructions are grouped into several separate classes with a short prologue at the beginning of each class. These classes include:

- Paragraph 3.1 - BCE Register Operations
- Paragraph 3.2 - Branching Instructions
- Paragraph 3.3 - Transmit Instructions
- Paragraph 3.4 - Receive Instructions
- Paragraph 3.5 - Special Instructions

3.1 BCE REGISTER INSTRUCTIONS

The BCE instruction set has several instructions that allow it to modify or store many of its registers. These registers include the Maximum Time Out register, the BCE-MSK Indicator bit, the Base register, and the Status register. In each case, the register referred to by the instruction is the one owned by the BCE executing that instruction.

The formats for these instructions are primarily 16 bits of the form of Figure 1.3, with the exception of the Load Base, which is a long format instruction (Figure 1.4). The Load Time Out and Load Base instructions use instruction bit 4 to indicate whether the addressing of the data to be loaded into the appropriate register is immediate or direct. In the first case, the data to be loaded is present in the instruction itself; in the latter, the instruction determines the address of the word containing data.

The Store Status instructions allow the specification of indexing by the current BCE number in the computation of the memory address to contain the status words. This permits the same instruction to be used by many BCE's since the index by processor number will construct a table of status words.

This group includes instructions to set or reset the BCE-MSK indicator bits. These bits have no meaning to the hardware, but can be read by MSC instructions, allowing them to be used as general-purpose flags by the BCE and MSC programs.

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR REPRODUCED IN ANY MANNER NOR PLACED IN ANY REPOSITORY

MS 87-08

Box 42

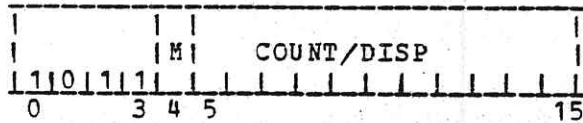
FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

LOAD TIME OUT REGISTER

#LTOI, #LTO

FORMAT



M	<u>Effective Count</u>	<u>INSTR. Format</u>
0	TIMEOUT ¹	#LTOI COUNT
1	(PC+DISP+2 x BCE#) ²	#LTO ADDRESS

NOTES:

1. Any value between 0 and 2047. This corresponds to time outs from 0 to 33.78 millisecc.
2. The lower 18-bits of the fullword addressed by PC(updated)+ displacement + 2 x BCE#. This allows any count between 0 and 262143, or 0 to 4.325 sec.

DESCRIPTION

This instruction loads the Maximum Time Out Register (MTO) with the Effective Count. This register is used by the Receive Data instructions to determine how long a BCE will wait for the first input word to arrive from a previously commanded subsystem. The resolution of this timeout count is 16.5 microseconds.

After loading the Maximum Time Out register, the BCE increments its program counter by 1 and begins execution of the next sequential instruction.

PROGRAMMING NOTE

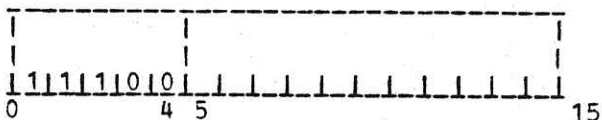
In direct mode, the computation of the effective address includes the number of the BCE executing the instruction. This allows many BCE's to execute the same BCE program, but, each one uses a time-out parameter best suited for the subsystem with which they are communicating. Note that two times the BCE number gives a fullword index.

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

RESET INDICATOR BIT

#RIB



INSTRUCTION FORMAT

#RIB

DESCRIPTION

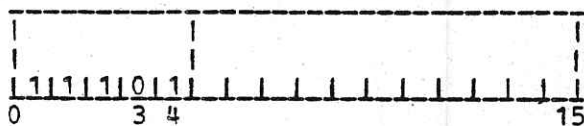
The BCE to MSC indicator bit associated with the BCE that is executing this instruction is reset to 0. This bit will not change to 1 until either a Set Indicator Bit (SIB) instruction is executed or, the BCE error terminates some instruction at a later time. After this bit is reset, the BCE's program counter is incremented by 1, and BCE program execution continues.

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY MANNER, NOR PLACED IN ANY INFORMATION SYSTEM

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

SET INDICATOR BIT

#SIB

INSTRUCTION FORMAT

#SIB

DESCRIPTION

The BCE to MSC indicator bit associated with the BCE that is executing this instruction is set to 1. This bit will not change to 0 until either the BCE executes a Reset Indicator (#RIB) instruction or the MSC executes a Reset BCE Indicator (@RBI) instruction with this BCE's number. After this bit is set, the BCE's program counter is incremented by 1, and BCE program execution continues.

PROGRAMMING NOTE

In indexing mode the computation of the effective address includes the number of the BCE that is executing this instruction. This allows many BCE's to use the same BCE program, and yet store the status from each BCE in a different location. Note that the BCE number is multiplied by 2 to give a fullword index.

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE).
COPY PROVIDED BY
WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
UNLESS OTHERWISE SPECIFIED, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY FORM OR BY ANY MEANS, ELECTRONIC OR MECHANICAL,
INCLUDING PHOTOCOPYING, RECORDING, OR BY ANY INFORMATION STORAGE
AND RETRIEVAL SYSTEM.

MS 87-08

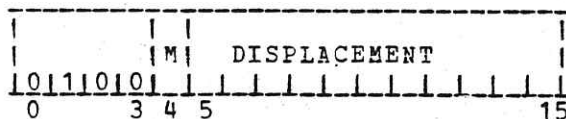
Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

STORE STATUS AND CLEAR

#SSC



M	Effective Address (EA)	INSTR.Format
0	PC + DISP.	#SSC ADDRESS
1	PC + DISP + 2XBCENO	#SSC ADDRESS (1)

NOTES: 1. PC is the updated Bus Control Element (BCE) Program Counter, i.e. the address of the next sequential instruction.

DESCRIPTION

This instruction stores the BCE's status register in the fullword location addressed by the Effective Address. The least significant bit of EA (the halfword address) is ignored. After initiating the store operation the BCE will then clear its status register and set its program execution bit (from STAT1) to 1 (GO). It will then increment its program counter by 1 and continue with execution of the next sequential instruction.

PROGRAMMING NOTE

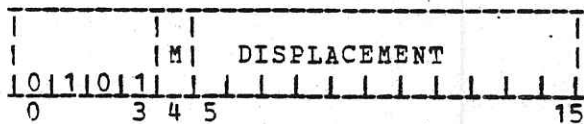
In indexing mode the computation of the effective address includes the number of the BCE that is executing this instruction. This allows many BCE's to use the same BCE program, and yet store the status from each BCE in a different location. Note that the BCE number is multiplied by 2 to give a fullword index.

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY MANNER NOR PLACED IN ANY REPOSITORY

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

STORE STATUS

#SST



M	Effective Address (EA)	INSTR.Format
0	PC + DISP.	#SST ADDRESS
1	PC + DISP + 2XBCENO	#SST ADDRESS (1)

NOTES: 1. PC is the updated Bus Control Element (BCE) program counter, i.e., the address of the next sequential instruction.

DESCRIPTION

This instruction stores the BCE's status register in the fullword location addressed by the Effective Address. The least significant bit of EA (the halfword address) is ignored. After initiating the store operation, the program counter is incremented by 1 and the next sequential instruction is begun.

PROGRAMMING NOTE

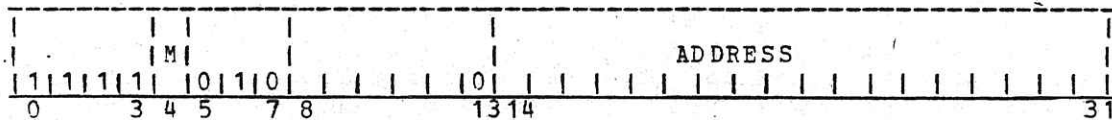
In indexing mode the computation of the effective address includes the number of the BCE that is executing this instruction. This allows many BCE's to use the same BCE program, and yet store the status from each BCE in a different location. Note that the BCE number is multiplied by 2 to give a fullword index.

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED
 REPRODUCED IN ANY FORM OR BY ANY MEANS

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

LOAD BASE REGISTER

#LBR



M	EFFECTIVE ADDRESS	INSTRUCTION FORMAT
0	Address	#LBR Address
1	(Address + 2 x BCE#)	#LBR@ Address

DESCRIPTION

The 18 bit effective address is loaded into the current Bus Control Elements (BCE) Base Register. The associated program counter is incremented by two.

PROGRAMMING NOTE

In direct mode the computation of the effective address includes the number of the BCE executing the instruction. This allows many BCE's to use the same program but, each BCE will get a different Base register. Note that twice the BCE number gives a fullword index.

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED,
 REPRODUCED IN ANY MANNER, OR PLACED IN ANY INFORMATION SYSTEM

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

3.2 BCE BRANCHING

The BCE instruction set includes instructions directing it to reset its own Program Counter and execute instructions at locations other than the next sequential one. As with the register class, these instructions affect only the PC register belonging to the BCE executing this instruction, and affects no other BCE.

This class of instruction includes an unconditional branch instruction and an instruction that allows a Listen Mode BCE to translate a Listen Command into a new Program Counter setting via a table lookup.

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED
 DEPOSITED IN THE NATIONAL ARCHIVES OR ANY OTHER REPOSITORY

MS 87-08

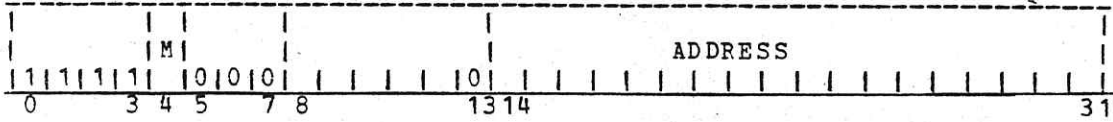
Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

BRANCH UNCONDITIONAL

#BU



M	Effective Address (EA)	INSTR. Format
0	ADDRESS	#BU ADDRESS
1	(ADDRESS + 2 x BCE#)	#BU@ ADDRESS

DESCRIPTION

The 18 bit effective address is stored into the current Bus Control Elements (BCE) Program Counter. The next instruction is found at this designated location, which may be on either a full or halfword boundary.

PROGRAMMING NOTE

In direct mode, the computation of the effective address includes the number of BCE executing the instruction. This allows many BCE's to use the same program and still retain the capability to branch to different segments as required for each BCE's operation.

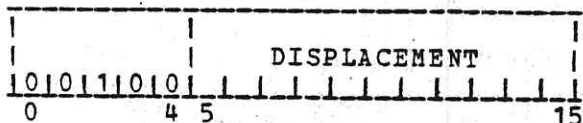
Note that twice the BCE number gives a fullword index.

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY FASHION, NOR PLACED IN ANY REPOSITORY

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

WAIT FOR INDEX
#WIX

FORMAT



INSTRUCTION FORMAT

#WIX Table

NOTE: Table = PC + Displacement, where PC is updated Program Counter, and Displacement is integer between -1024 and +1023.

DESCRIPTION

This instruction places the BCE in a state where it will monitor, through its MIA, the system bus to which it is attached for commands from other IOPs. When it receives such a command, it uses part of the command as an index into a table of branch addresses, and branches to the indicated location. This procedure allows one BCE in one IOP to signal to another BCE in a different IOP that it is time to perform some BCE program.

Figure 3.2 diagrams operation of this instruction. The starting address of the table of branch addresses (one address/fullword) is the sum of the updated PC (1 + address of present #WIX) and the 11 bit Displacement field. This address is rounded up by 1 if necessary to make it a fullword address (least significant bit = 0). After computing this address the BCE sets itself up to accept from its MIA a bus word termed a "Listen Command" that has command sync and an Interface Unit Address of 0 1 0 0 0 (in binary).

The BCE then goes into a tight loop of monitoring the MIA Buffer for a valid Listen Command. If at the entry to this loop, or at any time during this loop, the BCE finds that it is not in Listen Mode (i.e. its transmitter is enabled -- see Paragraph 4.1) then it will exit the loop and enter the Wait State. If it stays in Listen Mode, and finds a valid Listen Command, it exits the loop. The BCE then places bits 14 to 18 of the command in its Interface Unit Address Register (IUAR). It also adds the Index bits 19 to 26 to the Table address computed earlier. This 8 bit Index is right justified and padded on the left by ten zero's when it is added to the 18 bit Table address. Figure 3.1 diagrams the makeup of a Listen command.

This computed address is used to reference a fullword in memory which contains in bits 14 through 31 a branch address. These 18 bits are

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY MANNER NOR PLACED IN ANY REPOSITORY

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

then loaded into the BCE's PC, and execution of the indicated instruction begun.

PROGRAMMING NOTES

If a #WIX is executed with the MIA's transmitter enabled, execution of the #WIX is equivalent to that for a #WAT, i.e. it resets its Busy/Wait bit, updates its PC by 1, and goes into a loop until the MSC sets the bit back to 1.

If the BCE is in Listen Mode, then during the entire time that the BCE is waiting for a Listen Command the BCE's Busy/Wait bit stays set to Busy. Thus any attempt by the MSC to execute a @LBB, @LBP, or @SIO involving this BCE will not go through, and will result in an MSC error and the setting of appropriate bits in the MSC Status Register.

After execution of a #WIX, the BCE's IUAR has been set to bits 14 to 18 of the received Listen Command. This permits the BCE/IOP that placed the Listen Command on the bus to condition the listening BCE(s) to accept data only from a certain subsystem. Typically the commanding BCE sends this subsystem a command to return a stream of data, which will then be picked up properly by not only the commanding BCE but also those on the same bus that were "listening" to the command BCE. Paragraph 4.1 should be referenced for more complete detail.

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY
WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY FORM OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPYING, RECORDING, OR BY ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM.

MS 87-08 Box 42 FF 50
Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

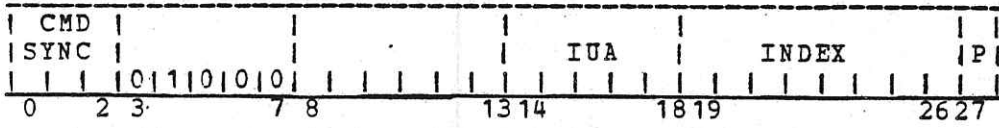


Figure 3.1. Bit Times of Listen Command

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE PROTECTED BY COPYRIGHT LAW (TITLE 17 U.S. CODE).

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE REPRODUCED

MS 87-08 Box 42 FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

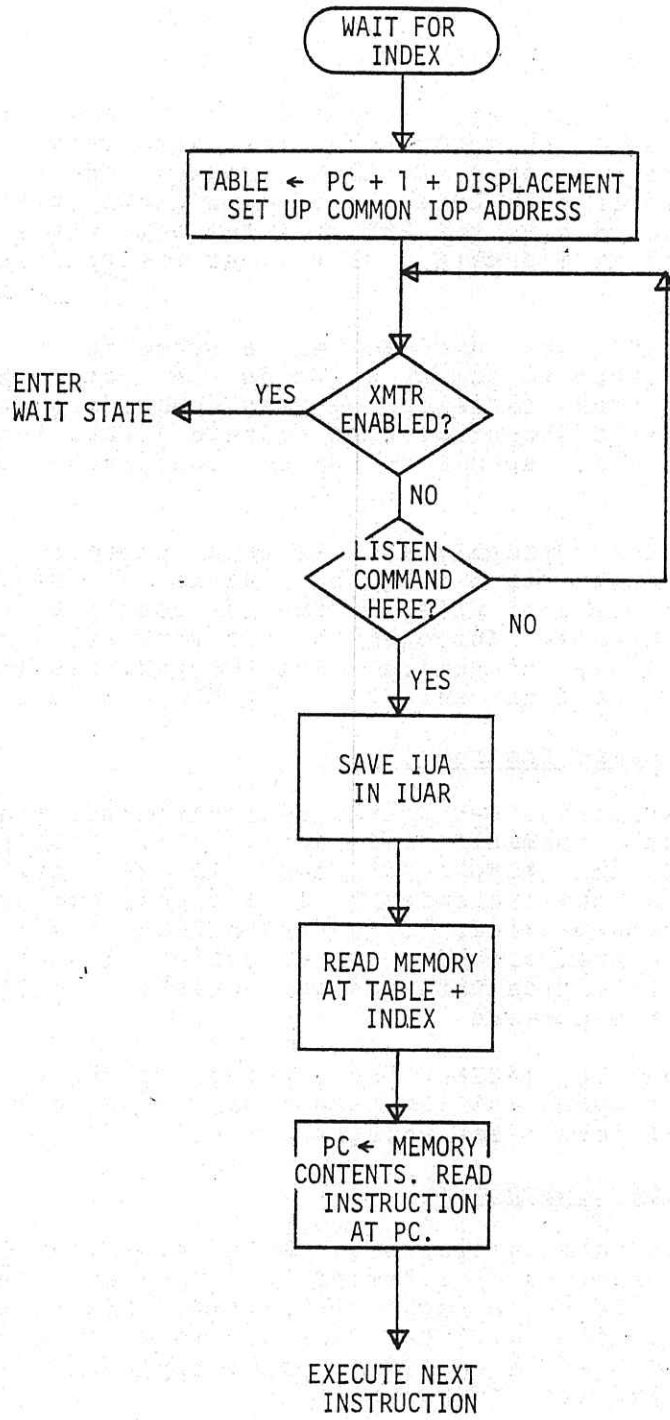


Figure 3.2. Wait for Index

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 TITLE 17 U.S. CODE,
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY MANNER, NOR PLACED IN ANY REPOSITORY

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

3.3 BCE TRANSMISSION INSTRUCTIONS

Each BCE in an IOP has the capability of directing its MIA to initiate the transmission of a word over the associated bus. These words may be either command or data words and when they are transmitted over the bus they have the formats shown in Figure 1.1(a) or (c). Out of these bits the BCE provides only the 24 information bits 3 through 26 and an indication of whether the word is a command or data word.

Command words are used to tell a subsystem to perform some action such as, get setup to accept N words of data that will be transmitted later. Since there can be many subsystems on a bus, each command contains a 5-bit Interface Unit Address (IUA), bits 3-7 of a bus command word, that specifies which subsystem should obey the command.

Data words are typically sent after a command, and contain the actual information that the BCE wants transferred to the subsystem. For each bus word this information consists of a 16-bit halfword from GPC memory. Surrounding this word are sync bits, a 5-bit IUA, the pattern 101, and parity. The BCE provides the MIA with just the IUA, the 16-bit data, and 101. The MIA adds the rest.

3.3.1 Transmit Command Instructions

The BCE instruction set contains instructions that direct the transmission of both commands and data. The Transmit Command instructions (#CMDI and #CMD) provides all 24 bits of command information needed for the transmission of a single command word. The Message Out Instruction provides both a command and the location of a stream of data to be transmitted. The execution of such instructions also sets the BCE's Interface Unit Address Register (IUAR) to whatever IUA is specified in the command.

The Message In (#MIN) instruction combines both the transmission of a command and the reception of a stream of returning data. For additional detail see Paragraph 3.4.8.

3.3.2 Transmit Data Instruction

The Transmit Data instructions (#TDS, #TDLI, #TDL, and #MOUT) specify the starting address of a buffer in memory and the number of halfwords that are to be taken sequentially from this buffer and given to the MIA for transmission as data. The subsystem to which this data is directed is specified in the IUAR, which typically was set during the transmission of the last command.

All buffer addresses for data transmission are Base register relative. The instructions contain only a displacement (sometimes assumed to be zero) that is added to the Base to compute the buffer starting address. This implementation is particularly useful when I/O

buffer areas are formatted in exactly the same way (for example for identical buffers containing torquing data for a set of identical gyros). The same BCE program can be used to output the data from any of these buffers by appropriate initialization of the BCE's Base register before the BCE is started at that program. Beside the BCE itself, both the MSC (via a @ LBB) and the CPU (via a PCO) can set any BCE's Base.

Base relative addressing also allows simultaneous use of the same BCE program by several BCE's. Initializing each BCE's Base register differently allows each BCE to transmit different data, while using the same BCE program.

Each Transmit Data instruction specifies the number of memory halfwords to be transferred. In all cases, the actual binary field used to specify this transfer count is treated as a positive integer that is numerically 1 less than the number of halfwords that will be transferred. Thus, a count of 0 corresponds to a transfer of 1 halfwords; a count of 262,143 corresponds to a 262,144 transfer.

3.3.3 Typical Bus Timing - Gaps Between Outputs

A typical sequence of BCE instructions that transmit a set of data to a subsystem consists of either a Transmit Command instruction followed by a Transmit Data instruction or a single Message Out instruction. The resultant sequence of activity as seen on the bus is pictured in Figure 3.3. Each bus word requires 28 microseconds for transmission, and is separated from the next bus word by a short period of time during which the bus is inactive. These periods of time are called "inter-word gaps". On Figure 3.3, the gap between the i th and $i+1$ st data words is denoted as $g(i)$, where the gap between the command and first data word is $g(0)$.

Under normal operating conditions the gaps between data words ($g(i)$, $i \geq 1$) transmitted by a BCE is fixed at 5. The length of gap $g(0)$ depends on the instruction sequence used to transmit the command. For a Message Out instruction this gap is fixed at 5 microseconds. For a Transmit Command instruction followed by a simple transmit data instruction (non #MOUT), the gap $g(0)$ depends on the exact type of Transmit Data instruction used, the number of instructions between the Transmit Command and the Transmit Data, whether the first data word comes from an even or odd halfword, and how long it takes to make all memory references between execution of the Transmit Command and the Transmit Data. Assuming no delay due to memory contention (See Error Modes - Excessive Concurrency), the gap $g(0)$ for such cases can be estimated from:

Time to execute all instructions between the Transmit Command instruction and the Transmit Data instruction

+

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY

ICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

PERMISSION TO REPRODUCE THIS MATERIAL MAY NOT BE COPIED
WITHOUT PERMISSION FROM THE ARCHIVES

ICHITA STATE UNIVERSITY LIBRARIES

MS 87-08

Box 42

FF 50

Time to set up the Transmit Data Instruction

- 12 usec

The time to set up any Transmit Data instruction is the time required to compute the address of the first word of data, to save the transfer count and to request and receive this first word. Table 3.1 summarizes some typical initial gaps and assumes no intervening instructions and no delays due to memory contention.

3.3.4 Echo Back

Whenever a MIA transmits something it echos back into the MIA Buffer a copy of what is sent out. This copy stays in the MIA buffer until either it is overwritten by something else received or transmitted by the MIA or it is removed by the BCE during a Receive Data Instruction.

Note that if one or more data words are transmitted, the last data word remains in the MIA buffer. A subsequent read data instruction would mistake this word from the first data word and probably time out waiting for what it thought was the second word. At least the data would be skewed one word to the right and the last word would be lost.

This problem can be avoided by entering the halt state between the transmit and the receive instructions or by issuing a read of one word and discarding the data and then issuing the desired read command.

The above situation does not create a programming problem in most BCE operations, because in the usual case the word in the buffer when the data read is executed will be a command word, which the read microprogram will inspect and discard. One case in which the programmer must take into account the problem is that of mass memory write operations, where the IOP transmits 512 data words and then receives a Search Complete Word (SCW) which was automatically transmitted by the mass memory without a command word intervening. In this case the read of one word must be made to clear the buffer, then a read with delay to await the SCW. The problem has also been reported to occur in DEU dump responses. The same avoidance technique is effective here.

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITH WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED OR REPRODUCED IN ANY MANNER NOR PLACED IN ANY REPOSITORY

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY MANNER NOR PLACED IN ANY REPOSITORY

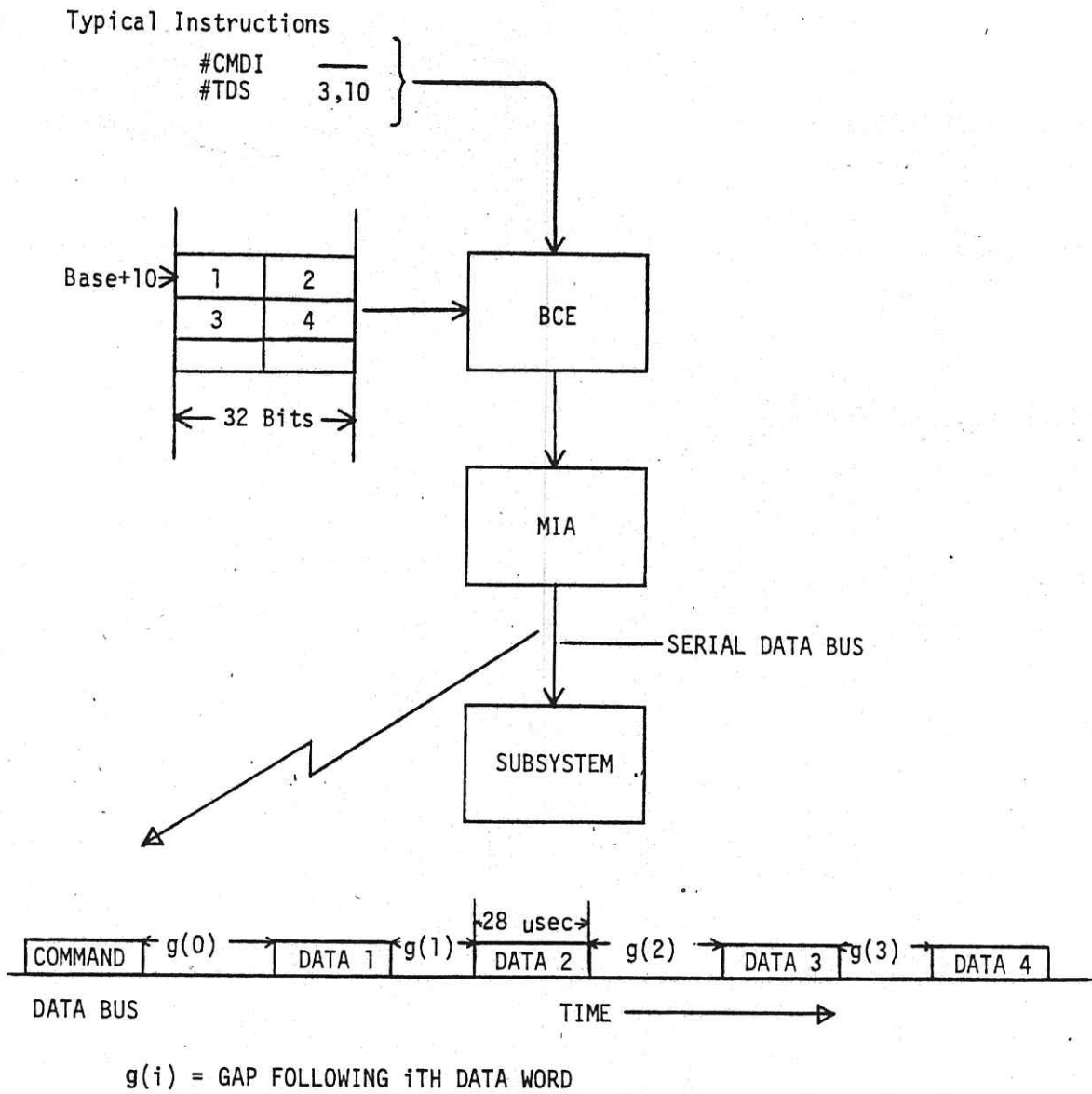


Figure 3.3. Typical Transmit Sequence

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

TABLE 3.1

TYPICAL GAPS BETWEEN COMMAND AND DATA WORD

<u>Type of Transmit Data</u>	<u>Assuming 1st Data from Even Addressed Halfword</u>	<u>Assuming 1st Data from Odd Addressed Halfword</u>
#TDS	21.5	38
#FDLI	21.5	38
#TDL	38	54.5
#MOUT	5	N.A.

All Times in Microseconds

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY
WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS

MS 87-08 Box 42 FF 50
Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

3.3.5 Error Modes - Disabled MIA

There are two aspects of general IOP operation that are outside the control of an individual BCE but that affects the execution of a Transmit instruction. These are the condition of the MIA's transmitter and receiver (enabled or disabled), and the amount of traffic through the IOP DMA channel due to PC I/O and DMA requests from the MSC and other BCE's. The status of the MIA is controlled by the Transmitter Enable and Receiver Enable bits in the Control Monitor. These bits are set/reset strictly by the CPU via a PCO. The transmitter must be enabled before a BCE will attempt to transmit either commands or data. If the transmitter is disabled at any time during the execution of a Transmit Data instruction (an error mode), the BCE executing that instruction will terminate the transmission of the rest of the data stream, set its Program Exception bit to 0 and bit 23 of its Status register and its BCE-MSI Indicator bit to 1, and enter the Wait State. It is up to the MSI or CPU, by analysis of the Program Exception register and Status Registers, to detect this error and perform appropriate recovery.

A related error mode occurs if the MIA is still busy when the BCE has decided it is time to transmit another word of data. The normal timing of a BCE makes this impossible unless either the MIA or BCE has failed. Therefore, the termination is the same as for a disabled transmitter.

Execution of a Transmit Command instruction, with the transmitter disabled, will not cause an error condition. This allows the Listen Mode BCE's to execute the same programs segments that the BCE/IOP in command executes. For example, a BCE must tell a subsystem to return some data but a listen BCE whose transmitters are disabled should ignore such commands and simply continue to the following receive instructions.

Note that this is particularly true of the Message In instruction, where a Command mode BCE will transmit the command and a Listen Mode BCE will go directly to the receive loop.

3.3.6 Error Modes - Excessive Concurrency

The IOP is functionally a multiprocessor with at least 25 separate processors (24 BCE's, 1 MSI), all of which can be generating memory requests independently. Since there is only one DMA channel and it can handle only one request at a time, these requests may stack up, delaying the honoring of the later requests for significant periods of time. If these requests were for data for a Transmit Data instruction (#TDS, #TDLI, or TDL), and they were delayed beyond the time at which the BCE wished to transmit them, then there could be significant periods of dead time on the bus between separate transmission of bus data words. These dead times are called "inter-word gaps", and if they become excessive they will cause the subsystem at the receiving end to time out and declare the BCE to be at fault.

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S.C.OBIE)
COPY PROVIDED BY
WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
UNIVERSITY OF SEOPHODOR
REPOSITORY
UNIVERSITY OF SEOPHODOR
REPOSITORY
UNIVERSITY OF SEOPHODOR
REPOSITORY

MS 87-08
Box 42
FF 50
Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

Consequently, a BCE in the middle of a Transmit Data loop will allow a gap of at most 20 microseconds to occur between data words. If a data request is not honored by the time a gap of 20 microseconds has occurred since completion of the last data word transmission, then the BCE executing the instruction terminates the transmission, sets its Program Exception bit, its BCE-MSI Indicator, and bit 21 of its Status register, and enters the Wait State. An MSI or CPU routine must then handle recovery or retry as deemed appropriate.

A Message Out instruction provides more time for data read requests to be handled and thus is less susceptible to excessive concurrency than any of the simple Transmit Data instructions. Consequently, a #MOUT instruction will error terminate in the above fashion if it cannot maintain a 5 usec gap between all outputs in the message including those between the command word and the first data word.

If the DMA Channel discovers a parity error when it is honoring a memory request for a Transmit Data instruction, it interrupts the CPU and terminates the request. This is reflected in the BCE by the failure of the data to arrive on time, and consequently by eventual execution of the time-out sequence described above.

Note that there is no time out check on the first data request for #TDS, #TDLI, or #TDL.

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE).

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY MANNER, IN ANY FORM, OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPYING, RECORDING, OR BY ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM.

MS 87-08

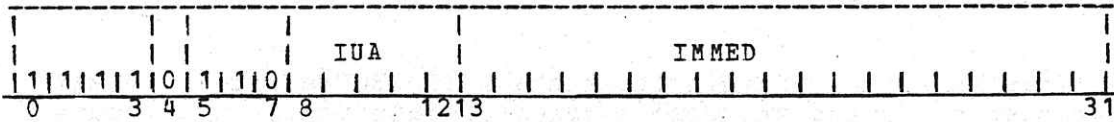
Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

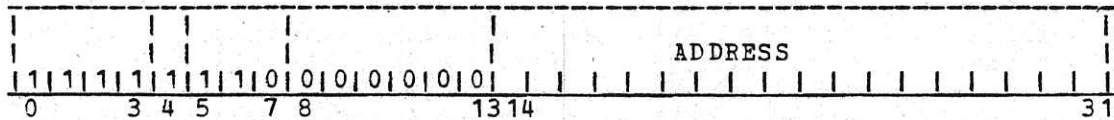
TRANSMIT COMMAND

#CMDI, #CMD



INSTRUCTION FORMAT (BIT 4 = 0) 1,2

#CMDI IUA, IMMED



INSTRUCTION FORMAT (BIT 4 = 1)

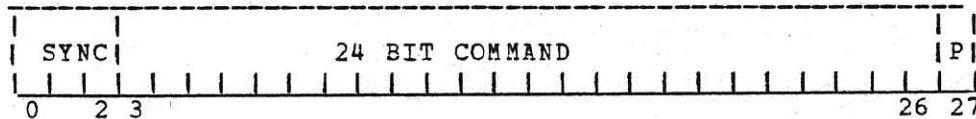
#CMD ADDRESS³

NOTES:

1. IUA is the Interface Unit Address, a 5 bit quantity in the range of 0 to 31.
2. Immed is a 19 bit constant.
3. Address + 2 x BCE# is the main storage location of a fullword containing a 24 bit command, right justified.

DESCRIPTION

This instruction is used to send 24 bit commands to a subsystem on the serial bus connected to the current BCE's MIA. In immediate mode (#CMDI) the command is found immediately in bits 8 thru 31 of the instruction. In direct mode (#CMD) the command is the lower 24 bits of the main store fullword computed from the address field (Bits 14 thru 31) of the instruction. (The halfword addressing bit, bit 31, is ignored). The format of the commands sent out by the MIA is:



SYNC = COMMAND SYNC

P = PARITY

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY MANNER, AND NO PART MAY BE REPRODUCED OR
 TRANSMITTED IN ANY FORM OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPYING, RECORDING, OR BY ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM.
 MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

The actual transmission of the command is conditional on the BCE's associated MIA's Transmitter being enabled and the MIA not being busy. If either condition is false the command is not sent, but no error condition is set.

After execution of this instruction the BCE's program counter is incremented by 2, and the next sequential instruction is executed.

PROGRAMMING NOTES:

The start of the actual transmission of the command by the MIA begins about 16 usec before the end of the instruction. Thus execution of the next instruction following a #CMDI or #CMD (Transmit Command) begins before the MIA has completed transmission of the command. If this next instruction is also a #CMDI, the MIA will still be busy when the #CMDI wishes to transmit its command, and transmission of the command will be suppressed. The #CMDI will then wait 16.5 microseconds and try again. If the MIA is no longer busy and the transmitter is enabled, the command is sent and the next instruction is processed. If either condition is false, the command is aborted, but no error condition is set, and the next instruction will be executed.

Either instruction can be used to send a "Listen Command" to a BCE on the same bus but in another IOP. In either case the 24 bits of command information consists of 01000 (the common IOP address), a don't care bit, and an 18 bit absolute address.

For the #CMD instruction, the address of the command word includes the number of the BCE executing the instruction. This allows many BCE's to execute the same BCE program, but at the same time it allows each BCE to transmit a different command. Note that twice the BCE number gives a fullword index.

MIA page hardware does not allow for TRANSMIT ENABLE going active less than or equal to one microsecond before the #CMDI or #CMD. If this one microsecond limitation is violated, proper transmission of the command word cannot be ensured.

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED
FROM ANY EDITION AND ARCHIVAL REPOSITORY

MS 87-08 Box 42 FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

TRANSMIT DATA SHORT

#TDS

		COUNT					DISPLACEMENT										
11010																	
0 2 3		7 8					15										

INSTRUCTION FORMAT

#TDS COUNT, DISPLACEMENT

NOTES: Count has range 0 to 31. (1 less than the number of transfers)
 Disp. has range of 0 to 255.

DESCRIPTION

This instruction directs the Bus Control Element (BCE) to transmit a number of 16 bit memory halfwords (determined by the COUNT) through the BCE's MIA to a subsystem attached to the MIA's serial bus. The location of the first halfword is the sum of the BCE Base Register and the Displacement field. This may be any halfword location. Succeeding output quantities come from succeeding halfwords in memory. Each halfword is assembled by the BCE into the following format and given to the MIA for transmission. (The MIA adds the sync and parity).

DATA		IUA					DATA Halfword															
SYNC		ADDRESS																			P	
0 2 3		7 8					23 24 26 27															

P = PARITY

The Interface Unit address (IUA) is a copy of the current contents of the Interface Unit Address Register (IUAR) which in turn was loaded during the execution of the last #CMD instruction performed by the BCE executing this #TDS. The value in the IUAR is thus the last subsystem to which this BCE has sent or attempted to send a command.

The number of bus words actually sent is 1 more than the number in the Count Field. Thus a Count of 0 causes one memory halfword to be transmitted; a count of 31 corresponds to 32 half words.

Completion of data transmission is followed by incrementation of the BCE's program counter by one, and execution of the following instruction.

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE).
 COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE REPRODUCED

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

If at any time during execution of this instruction, the BCE is not able to keep the time gap between intermediate data words on the bus to less than 20 microseconds, then bit 21 of the status register is set, the BCE's Program Exception bit (STAT1) is set to 0, the BCE-MSI Indicator is set to 1, the present instruction terminated, and the Wait State entered. The cause of such gaps may be due to memory contention that prevented a data read from being completed in time, or a parity error when the DMA channel attempted to read the data from memory for the BCE.

The instruction will also be error terminated (with bit 23 in the Status Register set to 1) if the MIA transmitter is either disabled or found busy when it was time to transmit a data word. Either situation represents an error condition.

PROGRAMMING NOTE

To insure correct status after execution of this instruction, the MTO register must be set to zero before execution.

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY

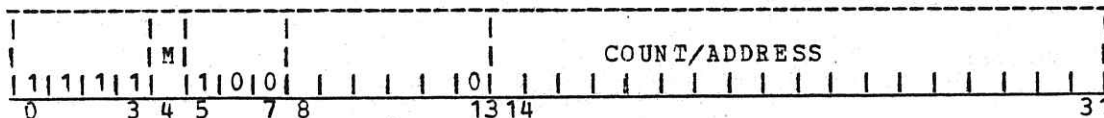
WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY MANNER WITHOUT THE WRITTEN PERMISSION OF THE

MS 87-08 Box 42 FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

TRANSMIT DATA LONG

#TDLI, #TDL



M	EFFECTIVE TRANSFER COUNT	INSTRUCTION FORMAT
0	Bits 14 thru 31 of instruction (Count Field)	#TDLI Count
1	Bits 14 thru 31 of fullword addressed by address field plus 2 x BCE#.	#TDL Address

DESCRIPTION

Execution of this instruction is the same as that for Transmit Data Short, with the following exceptions:

1. The displacement from the base is zero. The base must then point to the beginning of the buffer.
2. The count of input words to be transmitted can range from 0 to 262143, and may be specified by either bits 14 thru 31 of the instruction (#TDLI) or by bits 14 thru 31 of the main storage fullword addressed by bits 14 thru 31 of the instruction (#TDL). In the second case the least significant bit of the address (the halfword selection) is ignored. In either case the number of memory halfwords actually transmitted is 1 more than the Effective Transfer Count.
3. After completion of the instruction, the BCE's program counter is incremented by 2.

PROGRAMMING NOTE

For the #TDL instruction the address of the transfer count includes the number of the BCE executing this instruction. This allows many BCE's to execute the same program while at the same time allowing each BCE to transmit a different number of outputs.

To insure correct status after execution of these instructions, the MTO register must be set to zero before execution.

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED
 OR REPRODUCED IN ANY MANNER OR PLACED IN ANY REPOSITORY

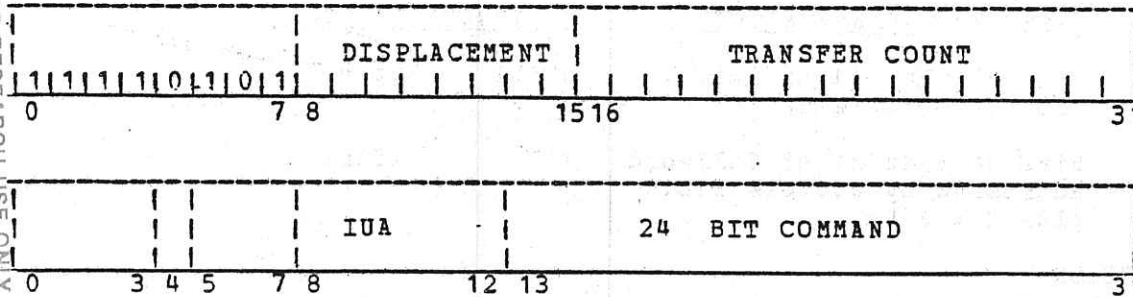
MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

MESSAGE OUT

#MOUT

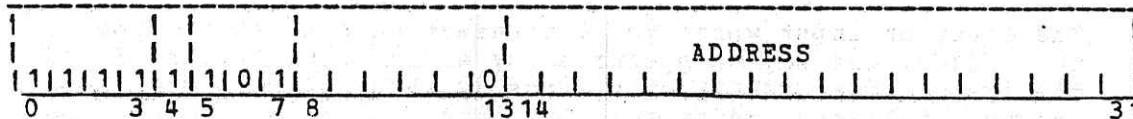
INSTRUCTION FORMAT (Bit 4 = 0)

FORMAT (2 Words)



#MOUT Displacement, Transfer Count
 #MOUTC IAU, Command

FORMAT (1 Word)



INSTRUCTION FORMAT (Bit 4 = 1)

#MOUT@ Address

NOTE: Displacement, Transfer Count found at Address +2 x BCE #;
 Command found at Address + 48 + 2 x BCE #

DESCRIPTION

This instruction initiates the transmission of a command followed immediately by a stream of data. As such it performs in one instruction the same functions as a Transmit Command (#CMDI or #CMD) followed by a Transmit Data (#TDS, #TDLI or #TDL). The command followed by the data is termed a "message".

Message Out comes in two formats, either of which allows independent specification of the command to be sent, the number of data words to be sent, and the location of the data buffer in main memory. The first format consists of a 2 word instruction. The second word contains all 24 bits of the command to be sent. The first word contains an 8 bit Base relative displacement and a 16 bit Transfer Count. The Displacement may be any positive integer between 0 and 255, and is added to the present contents of the Base register to form the address of the data buffer. Unlike #TDS, #TDLI, and #TDL

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY MANNER, NOR PLACED IN ANY MEDIUM

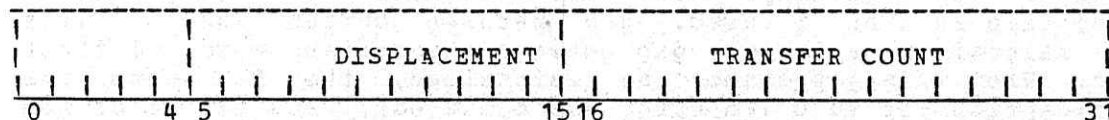
MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

this address is assumed to a fullword address -- the least significant bit of the address is ignored. Consequently, the first data word to be sent out will come from the upper half (bits 0 to 15) of the selected fullword.

In this format the Transfer Count may be any positive integer between 0 and 65535, and represents one less than the actual number of data halfwords to be taken from memory and transmitted as data words on the BCE's bus.

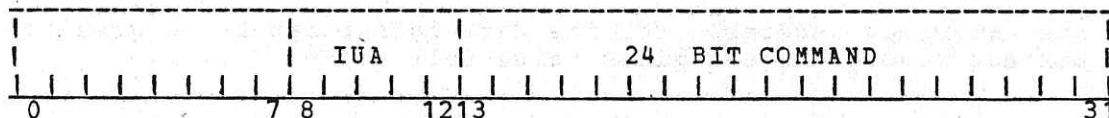
The second format allows automatic indexing by BCE number into two tables providing basically the same information found in the first format. Each table has 24 entries, one per BCE. Twice the number of the BCE executing this instruction is used as an index so that the table entries are each fullwords.

The first table starts at ADDRESS +2, and contains the Displacement and Transfer Count. For BCE i the table entry at ADDRESS + $2 \times i$ is formatted as:



This allows a range of displacement between 0 and 2047 and a transfer count between 0 and 65535. As before, however, the sum of the Base and Displacement is treated as a fullword address.

The second table starts at ADDRESS + 50, and contains the 24 bit command. For BCE i the table entry at ADDRESS + $48 + 2 \times i$ is formatted as:



Operation of #MOUT is diagrammed in Figure 3.3. The Transfer Count and Displacement are derived from the instruction, and are followed by requests to read the memory words containing the commands and the first fullword of data. After the command word has been read from memory, the 5 bit IUA in the command (bits 8 to 12 of the memory word containing the command) is saved in the BCE's IUAR, and the command given to the MIA for transmission.

After command transmission, the BCE goes into a loop that transfers data words to the MIA for transmission at a rate of 1 per 33 usec. These data words consist of the 5 bit IUA found in the IUAR and

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT
LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS

THESE WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY FORM OR BY ANY MEANS WITHOUT THE WRITTEN PERMISSION

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

a half-word of data from memory assembled as in Figure 1.1(a). When appropriate, the BCE also requests that new fullwords be read out of memory into the BCE. These words contain data that is to be transmitted later in the sequence. When the specified number of data words has been given to the MIA, the BCE exits this loop, updates the PC, and begins the next instruction.

As with any of the Transmit Data instructions, if the BCE ever attempts to transmit a word but finds either its transmitter disabled or its MIA busy at a time when it should be idle, the BCE will terminate the #MOUT, set its Program Exception Bit to 0 (an error has occurred), set its Status Register Bit to 1, set its BCE-MISC indicator, and enter the Wait State.

Also as in a Transmit Data instruction, a BCE will terminate a #MOUT if it cannot maintain a controlled interword gap between output words. However, the nature of #MOUT allows memory requests, and data requests in particular, to be made much earlier than they could in a Transmit Command/Transmit Data instruction sequence. This reduces the effects of memory contention from other BCEs to the point where only very exceptional conditions will interfere with a #MOUT transmission. Consequently, a BCE will terminate a MOUT only when it cannot maintain an even 5 usec. gap between output words. This includes maintaining a 5 usec. gap between the command word and first data word. When this gap cannot be maintained, the BCE sets its Program Exception Bit to 0 (an error has occurred), sets bit 21 of the Status Register sets its BCE-MSC indicator, and enters the Wait State.

As with any of the Transmit Data instructions, the BCE will terminate in the same fashion if there is a problem in the DMA interface when one of its data read requests is being handled, such as a parity error or DMA hangup. The data from memory will never reach the BCE, and when the BCE reaches the point where that data is to be transmitted, it will detect the absence of the data and terminate.

PROGRAMMING NOTE

The starting address for the data buffer can be no greater than the maximum memory address minus three full words.

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR REPRODUCED IN ANY MANNER, NOR PLACED IN ANY MEDIUM.

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

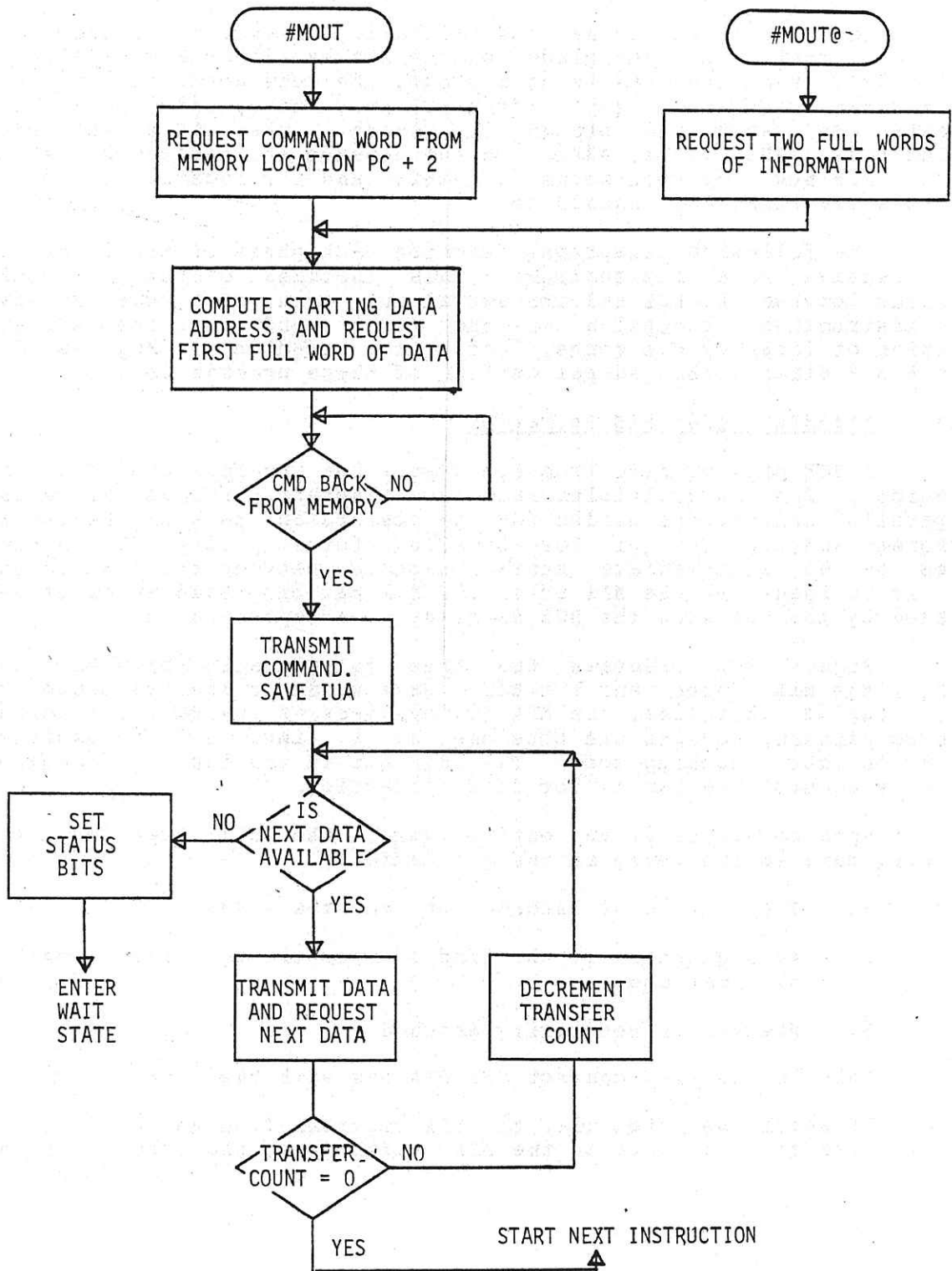


Figure 3.4. MOUT Operation

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY MANNER, NOR FOR ANY PURPOSES OTHER THAN

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

3.4 BCE RECEIVE DATA INSTRUCTIONS

Each BCE in an IOP has the capability of accepting, from its MIA, data that has been placed on the bus by either a subsystem or another GPC. When the data is on the bus, the data words are in the format shown in Figure 1.1(b). After BCE processing, only the 16 bits of data are saved and placed in memory. The BCE Receive Data Instructions (#RDS, #RDLI, #RDL, and the Message In instruction #MIN) tell the BCE how many such words to receive and the location in memory where the resulting data should go.

The following paragraphs describe each phase of the execution of a Receive Data instruction. This includes operation of the interface between the BCE and its serial bus, setup of a BCE Receive Data instruction, reception of the first word of a data stream, reception of intermediate words, and error handling. Figures 3.5 through 3.8 diagram the general outline of these procedures.

3.4.1 MIA-MIA Buffer-BCE Operation

A BCE obtains data from its system bus through its MIA and MIA Buffer. The MIA (Multiplex Interface Adapter) performs the serial to parallel conversions needed for the conversions between the serial bus format and the internal IOP parallel format. The MIA Buffer serves as an intermediate storage register between the MIA and the BCE. It is loaded by the MIA after the MIA has processed an input and unloaded by the BCE when the BCE is ready to accept data.

Figure 3.5 diagrams the time relationship between the bus/MIA, the MIA Buffer, and the BCE. Data words on the bus occupy 28 usec. During this time, the MIA (independent of the BCE) recognizes the sync pattern, accepts the bits one at a time, and accumulates parity on the incoming word. The 28th bit of the input is compared with this accumulated parity for fault detection.

Upon reception of an entire word, the MIA transfers the following data to its entry in the MIA Buffer:

1. The 24 bits of information from the word,
2. An indication of the kind of sync the word had (command or data) and,
3. Whether or not parity matched.

This is the last contact the MIA has with the data.

It should be noted that the MIA requires from between 1 to 5 usec to transfer a word to the MIA buffer after the receipt of the last bit.

THIS MATERIAL MAY BE
FOR RESEARCH USE ONLY
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
COPY PROVIDED BY
AT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY FORM OR BY ANY MEANS WITHOUT THE WRITTEN PERMISSION OF THE
WICHITA STATE UNIVERSITY LIBRARIES

MS 87-08 Box 42 FF 50
Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

It should also be noted that words with too few or too many bits, or invalid electrical encoding on the bus are not passed from the MIA to the MIA buffer.

The BCE operates asynchronously to the MIA, and samples the MIA Buffer to ascertain when a new data input has arrived. It samples the MIA Buffer only during either execution of a Receive Data or Wait for Index instruction. This latter operation is described in Section 4.

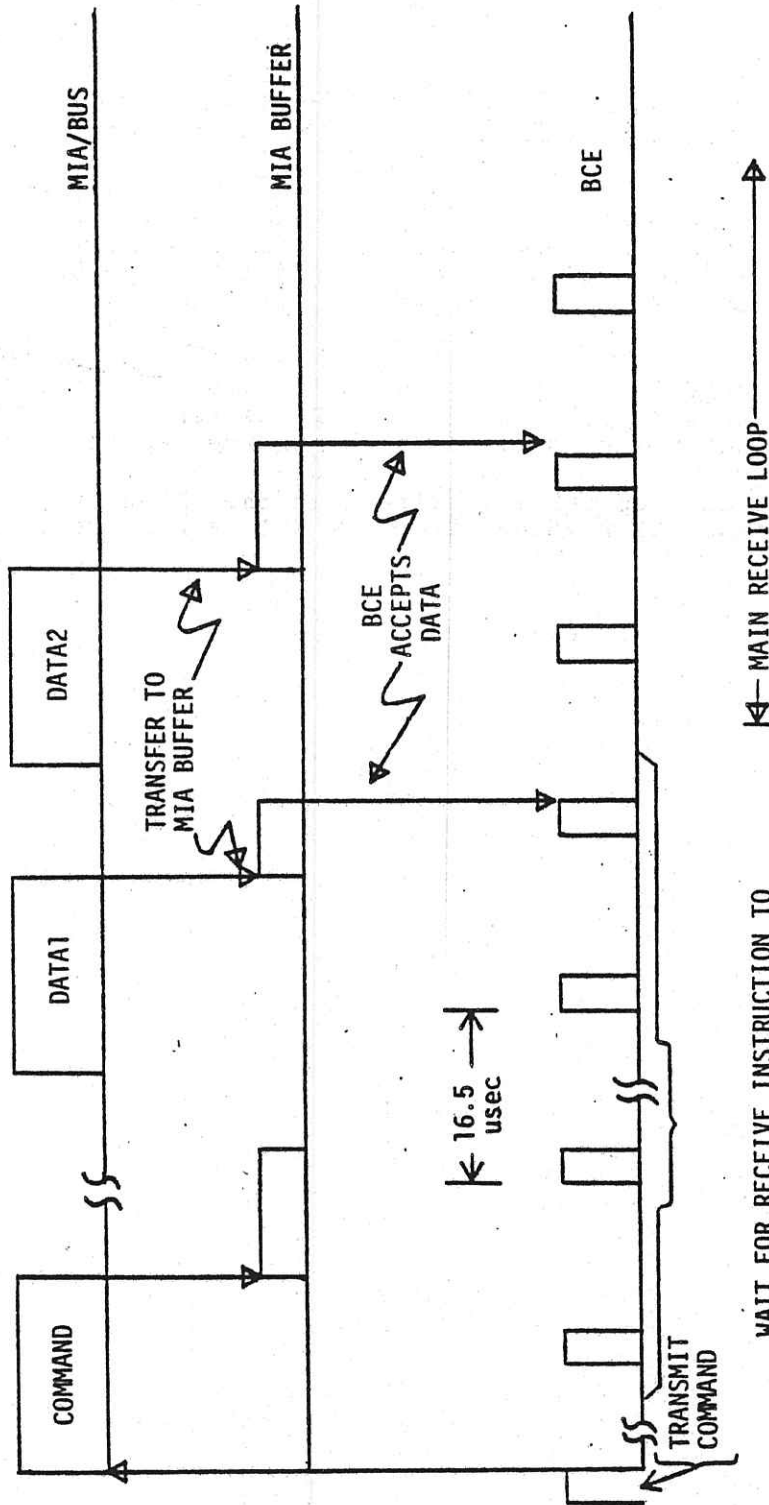
In either case the sampling process occurs at most once every 16.5 usec. When an entry is found in the MIA buffer, the BCE removes it and performs whatever series of error checks are deemed necessary.

Note that once an entry is placed in the MIA buffer it stays there until either the BCE removes it or the MIA overwrites it with a new value.

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIB
SPECIAL COLLECTIONS
PERMISSION, THIS MATERIAL MAY NOT
BE REPRODUCED OR TRANSMITTED IN ANY FORM
WITHOUT PERMISSION

MS 87-08 Box 42 FF 50
Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives



WAIT FOR RECEIVE INSTRUCTION TO ARRIVE FROM MEMORY (IF NOT A #MIN), SET UP FOR RECEIVE, AND START WAITING FOR DATA.

Figure 3.5. MIA-MIA-Buffer-BCE Operation

FOR RESEARCH USE ONLY.
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE).
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY MANNER OR PLACED IN ANY REPOSITORY.

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

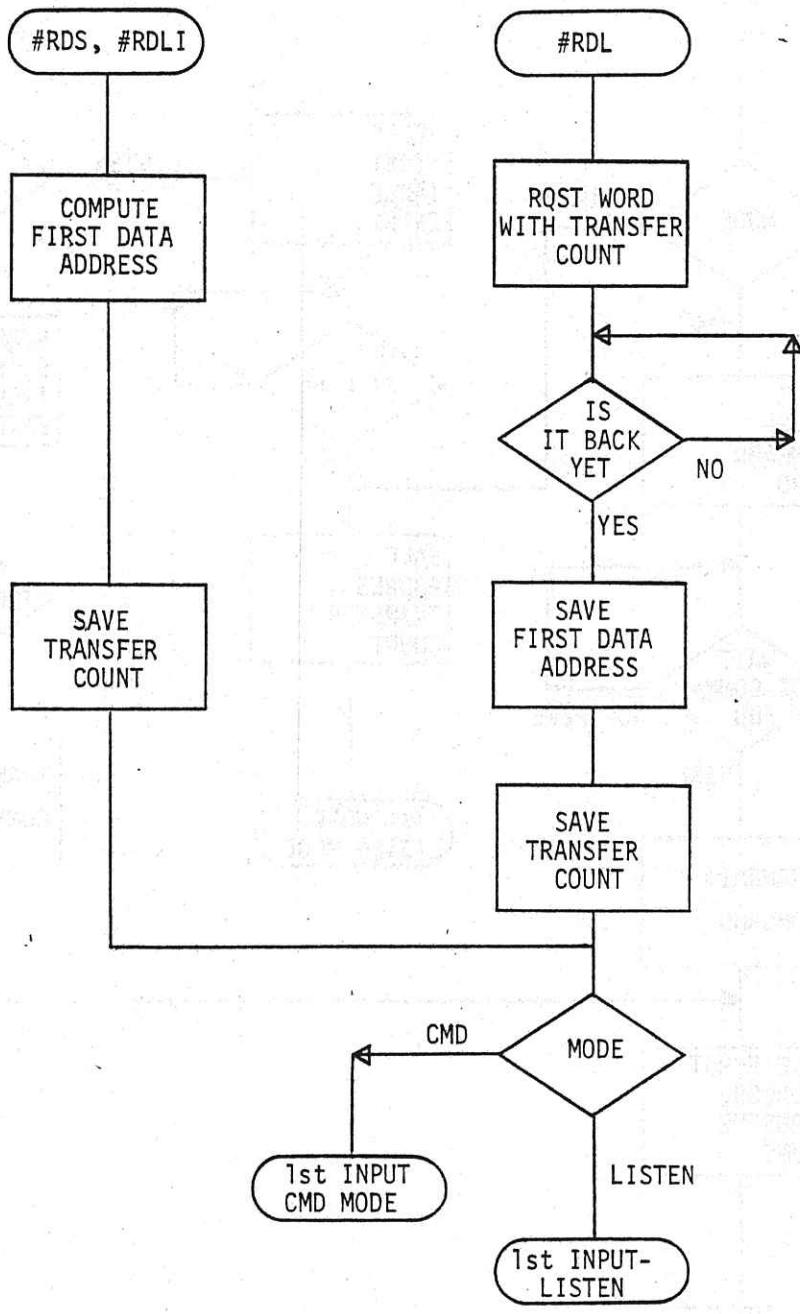


Figure 3.6. #RDS, #RDLI, #RDL Setup

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE),
 COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY MANNER, NOR PLACED IN ANY REPOSITORY.

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

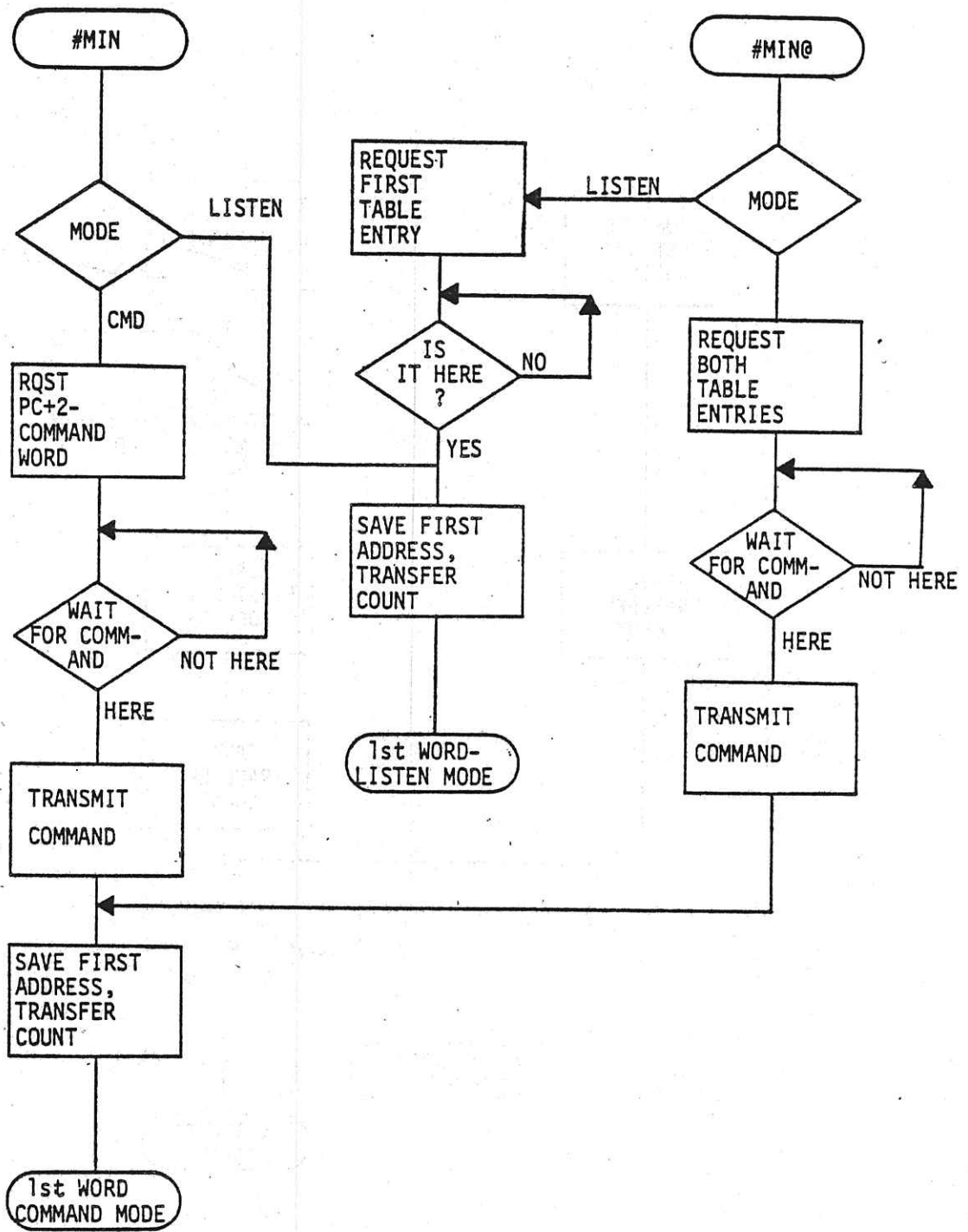


Figure 3.7. #MIN Setup

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY FORM OR BY ANY MEANS

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

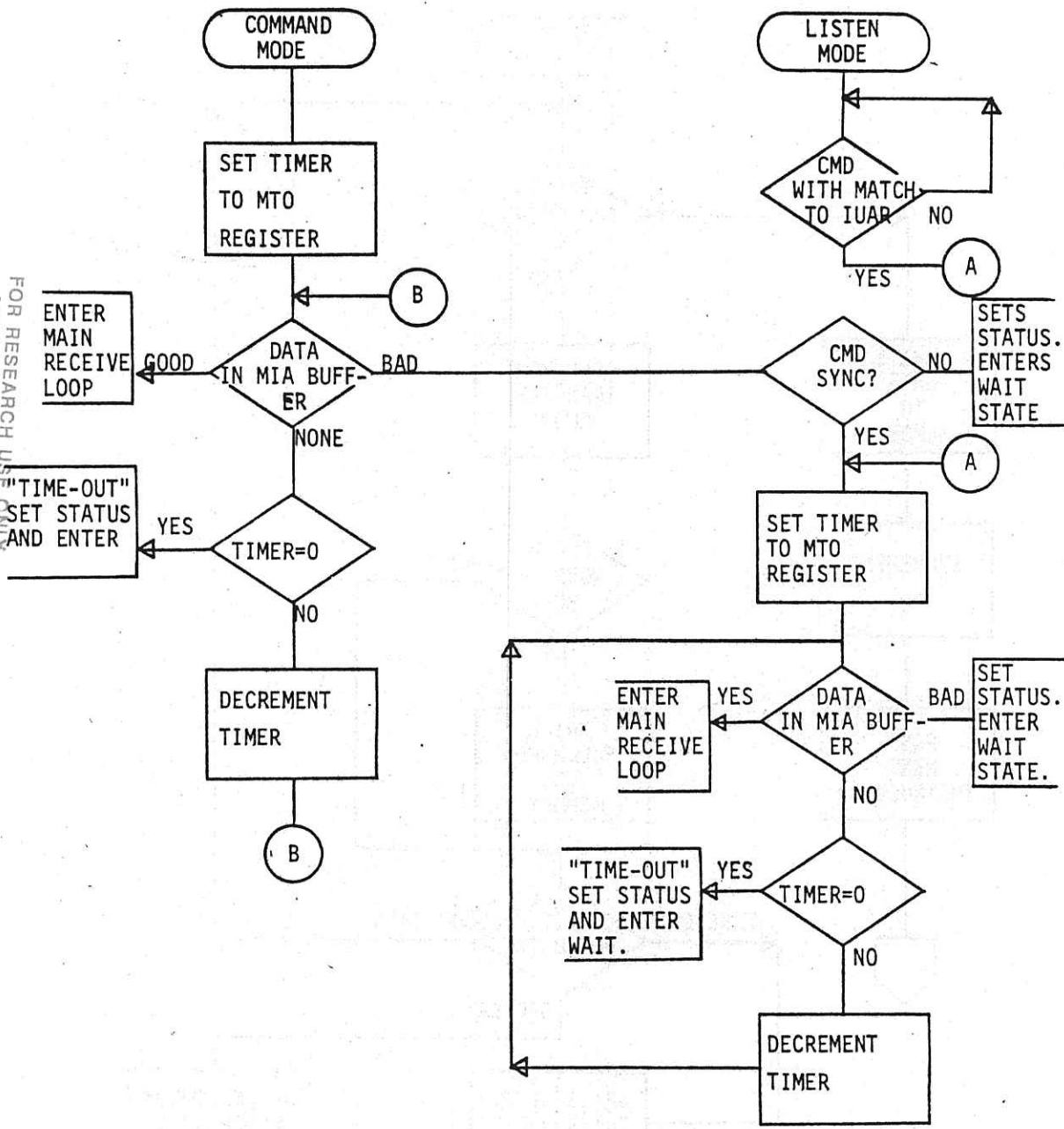


Figure 3.8. Receive Data Algorithm-First Input.

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE).
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE
 REPRODUCED, ANY-FASHION, NOR PLACED IN ANY

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

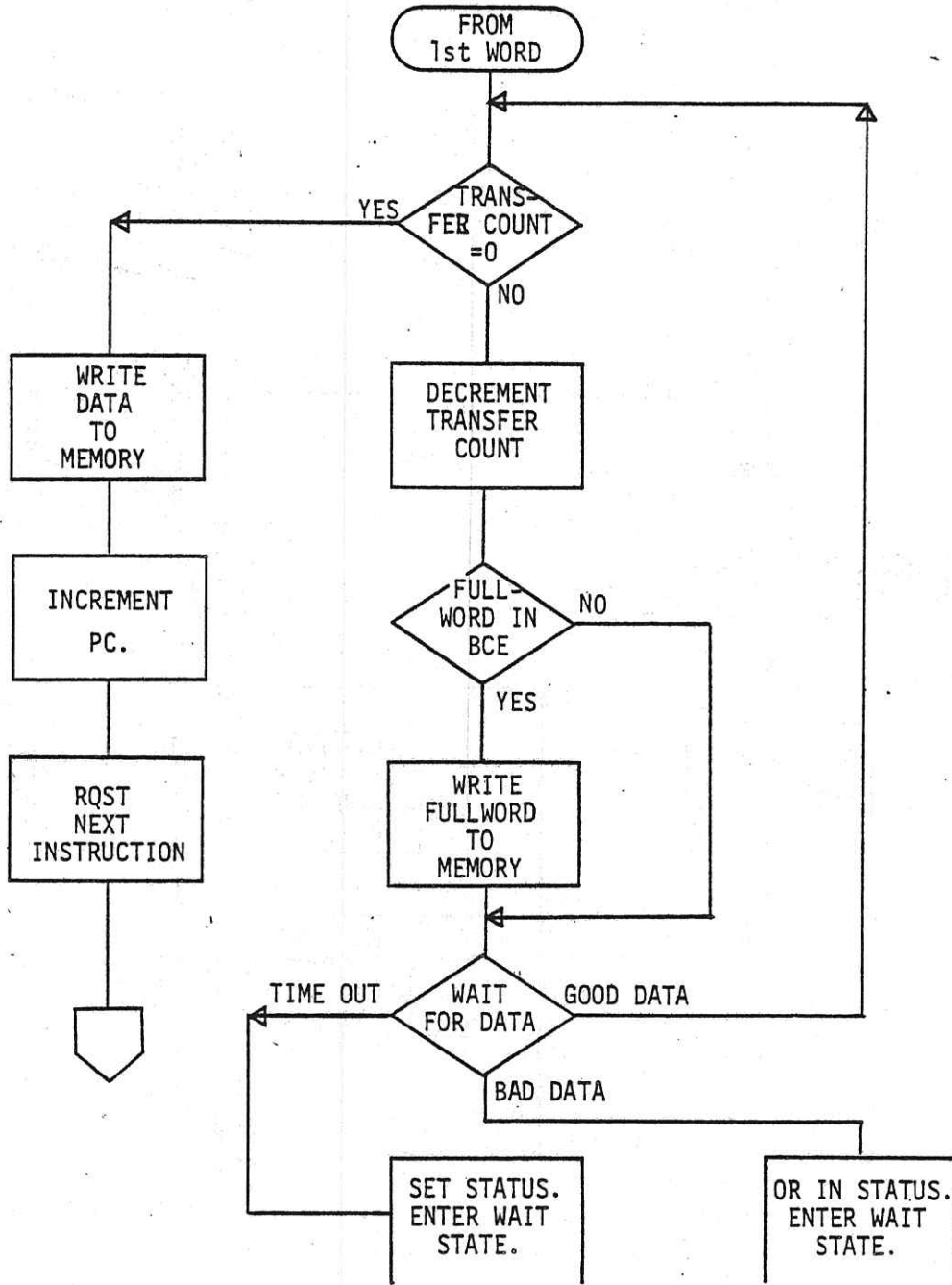


Figure 3.9. Main Receive Loop

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION MATERIAL MAY NOT BE COPIED
 REPRODUCED IN ANY MANNER

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

3.4.2 Receive Data Setup

All Receive Data instructions specify:

1. The starting address of a buffer in memory in which the incoming data should be placed
2. The number of input words to be received.

Additionally, the Message In instruction specifies the command that should be sent to the subsystem before data should be expected in return.

The subsystem from which this BCE expects to receive data is specified in the IUAR.

This register could have been set in a variety of ways; including:

- 1) A CPU PCO while the BCE was in Wait State
- 2) For a command mode BCE, the execution of an instruction that included transmission of a command word (#CMDI, #CMD, #MIN or #MOUT).
- 3) For a Listen Mode BCE, reception of a Listen command while the BCE was executing a Wait for Index instruction.

All buffer addresses for data reception are Base register relative. The instructions contain only a displacement (sometimes assumed to be zero) that is added to the Base to compute the buffer starting address. This implementation is particularly useful when I/O buffer areas are formatted in exactly the same way (for example for identical buffers to contain data from a set of identical gyros). The same BCE program can be used to input the data to any one of these buffers by appropriate initialization of the BCE's Base register before the BCE is started at that program. Both the BCE itself (via an #LBR), the MSC (via an LBB) and the CPU (via an PCO) can set any BCE's Base.

Base relative addressing also allows simultaneous use of the same BCE program by several BCE's. Initializing each BCE's Base register differently allows each BCE to receive different data into different buffers while using the same BCE program.

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY MANNER NOR PLACED IN ANY REPOSITORY

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

3.4.3 Acceptance of First Input

Once past setup, a BCE executing a Receive Data instruction will begin watching its MIA Buffer for bus inputs from the MIA. A BCE in Command mode will immediately start monitoring the MIA Buffer for inputs with data sync. A BCE in Listen Mode, however, will first await an input with command sync and an IUA that matches the BCE's IUAR. As demonstrated in Section 4, this procedure synchronizes the BCE/IOP that is commanding a subsystem to return data with a Listen Mode BCE (in another IOP) that is waiting for the data to return. Once a Listen BCE receives such a command it will begin waiting for the first data word.

A BCE in either mode will not wait indefinitely for the first input with data sync to arrive. Instead, once it starts waiting for a data word, it initializes a timer to the value contained in the BCE's Maximum Time Out (MTO) register. This timer is then decremented every 16.5 usec until either the timer reaches 0, or the BCE finds something in the MIA buffer. If the timer reaches zero first, the BCE terminates the reception and enters the Wait state as described in paragraph 3.4.7 (with bit 25 in the Status Register set to 1).

Thus, if the MTO register has a value of N in it, the BCE will lock at most $N + 1$ times into the MIA Buffer before declaring a timeout.

Table 3.2 lists each Receive Data instruction and the time from the start of the instruction until the point where the BCE first looks at the MIA Buffer and initializes the timer to MTO.

THIS MATERIAL MAY BE
 FOR RESEARCH USE ONLY
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED
 REPRODUCED IN ANY FORM OR BY ANY MEANS, NOR PLACED IN ANY REPOSITORY

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

TABLE 3.2

TIME TO FIRST LOOK AT DATA

Instruction	Mode	Minimum Time from Start of Instruction to First Look at MIA Buffer
#RDS, #RDLI	Command	33 usec
#RDL	Command	49.5 usec 3
#MIN	Command	82.5 usec 1,3
#MIN @	Command	99 usec 1,3
#RDS, #RDLI	Listen	49.5 usec 2
#RDL	Listen	66 usec 2,3
#MIN	Listen	66 usec 2
#MIN @	Listen	66 usec 2,3

1. This is 65.5 usec after the BCE has handed the command to the MIA for transmission.
2. This is 16.5 usec after the BCE has found a command in the MIA Buffer.
3. These times may be increased if necessary read requests are not handled within about 16 usec after they are made.

3.4.8 Reception of Intermediate Words

After successful reception of the first word of a sequence, a BCE will continue to look into its MIA Buffer for additional data inputs until it has accepted the number indicated in the original Receive Data instruction. As with the first input, the BCE will not wait indefinitely. Unlike the first word, however, the time out detection procedure is normally fixed and is not adjustable by the programmer. A separate report describes this procedure in detail. If the BCE does not find data in the specified time, the BCE error terminates with Status Register bit 26 set to 1.

The only exception to the above error termination is if a timeout occurs and the number of inputs left to be received is an integer multiple of 512. If it is, the BCE will set a timer to MTO (as in the acceptance of the first input) and continue waiting. This procedure allows a BCE to recognize the inter-block gaps that occur between every 512 words of an input stream from the Mass Memory or Display Units. A timeout after this point has been reached will cause the BCE to error terminate with Status Register bit 27 set to 1.

Upon recognition of a data input before time runs out, a BCE will perform the same error checks applied to the first input. Any discrepancy regardless of the BCE's mode, will cause the BCE to error terminate.

Data that meets all the above conditions is saved by the BCE for transmission to memory, as described in paragraph 3.4.5.

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

3.4.6 Error Termination - Faulty Input

A BCE that has detected a faulty data input will immediately terminate the reception and will enter the Wait state. Before it enters the Wait State, it also performs the following tasks:

1. Into the high half of the BCE Status Register (bits 0 to 15) the BCE or's in the following pattern.

		M		P		SEV		IUA						S	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

WHERE:

M = 1 if there was a mismatch between the received interface unit address and the IUAR.

P = 1 if there was a detected parity error

SEV = the SEV bits from the input with the S and V bits inverted. Thus, a valid SEV pattern -101- forms 000 in the above pattern.

IUA = the actual Interface Unit address from the input.

S = 1 if the input had a command sync.

2. The BCE's Program Exception bit is set to 0 (error)
3. The BCE's BCE-MSI Indicator bit is set to 1.
4. BCE Local Store register A1 is left with a value equal to the number of inputs not yet received (i.e., the number of inputs not written to memory).
5. BCE Local Store register A0 is left pointing to the last halfword to receive good data. If the error is on the first input, A0 points to the beginning of the buffer.
6. The BCE's Busy/Wait bit is reset to Wait.

3.4.7 Error Termination - Time Out

A BCE that has waited the prescribed period of time for a data input will also error terminate to the Wait State in exactly the same fashion that is used on faulty inputs except that the appropriate BCE Status Register bit (either 25, 26, or 27) is set to 1.

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR

REPRODUCED IN ANY MANNER WITHOUT THE WRITTEN PERMISSION OF

MS 87-08

Box 42

FF 50

3.4.4 Error Checks

Upon receipt of an input from the MIA Buffer, the BCE performs a series of error checks, including:

1. Does the input word's IUA (bits 3-7 of Figure 1.1) match that contained in the BCE's IUAR (bits 13-17 of the BCE's Local Store cell C5).
2. Did the MIA detect a parity error.
3. Are any of the input word's SEV bits other than 101.
4. Was the sync data and not command?

If all these error checks are satisfied, then the data is accepted. If any one condition is not satisfied, the input is not accepted as a valid data word.

If the input is not the first input, the BCE error terminates. If it is the first data input, the resultant processing is a function of the BCE's mode. A BCE in Listen Mode will treat any discrepancy as an error and the BCE will terminate in the manner described in Paragraph 3.4.6. On the other hand, a BCE in command mode will check to see if the first input had command sync, and if so, it will ignore the input, reset the timer to MTO, and await the first input. This procedure allows a BCE that has just transmitted a command to skip over the copy of that command that has been echoed back by the MIA into the MIA Buffer. Note, however, that a second input with command sync will be treated as an error.

If the problem with the input is other than command sync, the BCE will error terminate regardless of its mode.

3.4.5 Handling of Good Inputs

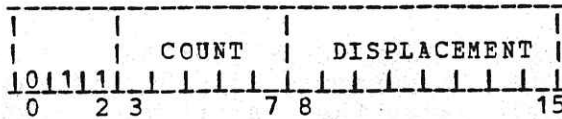
When a BCE has found an input in the MIA buffer that satisfies all of the above conditions, it extracts the 16 bits of data for storage in memory. If the data is destined for an even halfword location, and if further data is expected, this halfword is saved until the next input arrives. When it does, the two halfwords are combined and written as a fullword to memory.

If the data is the first to be received, and it is destined for an odd halfword location, or it is the last input to be received, and is destined for an even halfword, the BCE will do a halfword write to store array the single 16 bits of data.

A BCE is ready to accept a new input 33 usec after recognizing the previous one. This time includes the above processing.

RECEIVE DATA SHORT

#RDS_

INSTRUCTION FORMAT

#RDS COUNT, DISP.

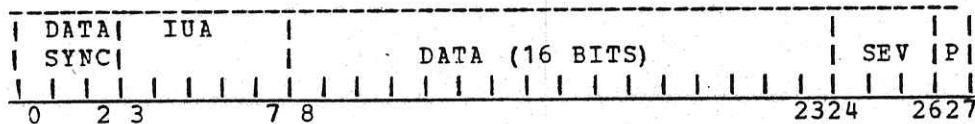
NOTES: Count has range of 0 to 31. (1 less than number of transfers)

Displacement has range of 0 to 255.

DESCRIPTION

This instruction directs the Bus Control Element (BCE) to accept a number of input words from its associated MIA, do a variety of checks on these words, and assemble them into 32 bit memory fullwords and place them in memory.

As received by the MIA each input word has the following format:



P = PARITY

IUA = Interface Unit Address

Bits 0-7 and 24-27 are used only in the validity checks; bits 8-23 represent the 16 bits of data that are to be saved in main memory.

Execution of this instruction is as described in paragraph 3.4.2 and in general proceeds as follows:

1. The input transfer count is extracted from the instruction. This count represents 1 less than the number of input words to be accepted, and these represent the number of 16 bit main memory halfwords accepted. Thus a transfer count of 0 corresponds to acceptance of 1 input. A transfer count of 31 corresponds to acceptance of 32 inputs.
2. The main memory address for the first input word is computed from the sum of the present contents of the current BCE's base register and the displacement field, bits 8 thru 15.

This displacement may have any value in the range 0 to 255. Note that this first address is the main memory halfword address for the first piece of data, and may be either even or odd.

3. The BCE monitors its MIA Buffer for data inputs. The procedure used is as described in Paragraph 3.4.2. If any errors are detected, the BCE terminates the instruction and enters the Wait State. The BCE's Program Exception Bit, BCE-MSI Indicator bit, and Status Register are also set to indicate the source of the problem.
4. Upon successful reception of all specified data words, and their transfer to memory, the BCE increments its Program Counter by 1 and begins the next instruction.

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE REPRODUCED

MS 87-08

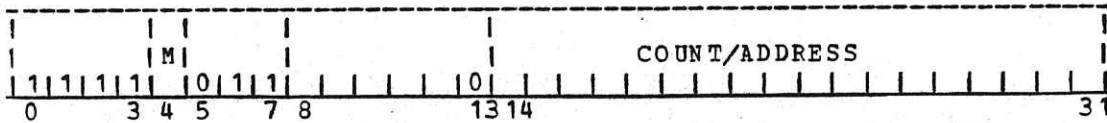
Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

RECEIVE DATA LONG

#RDLI, #RDL



M	EFFECTIVE TRANSFER COUNT	INSTRUCTION FORMAT
0	Bits 14 thru 31 of instruction (Count Field)	#RDLI Count
1	Bits 14 thru 31 of fullword addressed by address field plus 2 x BCE #	#RDL Address

DESCRIPTION

Execution of this instruction is the same as that for Receive Data Short, with the following exceptions:

1. The displacement from the base is zero. The base must point to the beginning of the buffer.
2. The effective count of input words to be received can range from 0 to 262143, and may be specified by either bits 14 thru 31 of the instructions (#RDLI) or by bits 14 thru 31 of the main storage fullword addressed by bits 14 thru 31 of the instruction (#RDL). In the second case, the least significant bit of the address (the halfword selection) is ignored. In either case the effective transfer count is one less than the number of words to be received.
3. After completion of the instruction, the BCE's program counter is incremented by 2.
4. The setup time for #RDL is at least 49.5 usec, and may be longer if contention for memory causes the access of the word containing the transfer count to take longer than 16.5 usec.

For the #RDL instruction the address of the transfer count includes the number of the BCE executing the instruction. This allows many BCE's to execute the same program while at the same time allowing each BCE to receive a different number of inputs. Note that twice the BCE number is a fullword index.

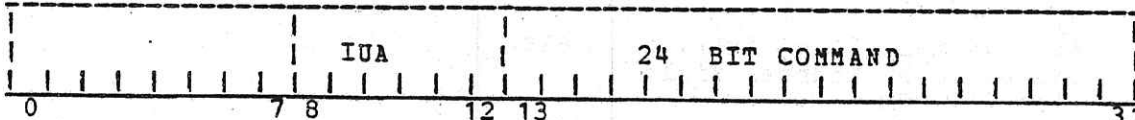
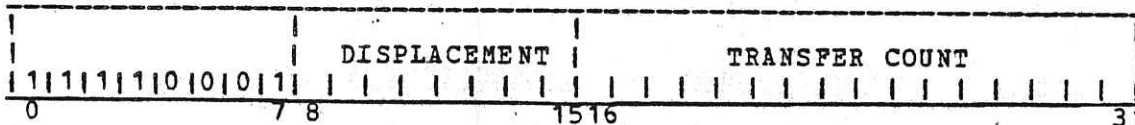
FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED
 OR REPRODUCED IN ANY MANNER.

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

#MIN

MESSAGE IN

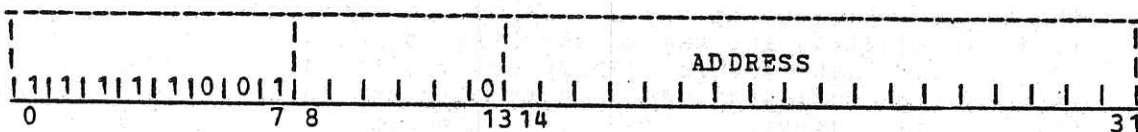
FORMAT (2 Words)



INSTRUCTION FORMAT (Bit 4 = 0)

#MIN Displacement, Transfer Count
 #MINC IUA,COMMAND

FORMAT (1 Word)



INSTRUCTION FORMAT (Bit 4 = 1)

#MIN@ Address

NOTE: The displacement and transfer count are found at address +2XBCE#. The command is found at address +48+2XBCE#.

DESCRIPTION

This instruction initiates the transmission of a command to a subsystem, and the acceptance of a stream of data returning in response from that subsystem. As such, it performs in one instruction the same functions as a Transmit Command (#CMDI or #CMD) followed by a Receive Data (#RDS, #RDLI, or #RDI). The command followed by the returning data is termed a "message".

Message In comes in two formats, either of which allows independent specification of the command to be sent, the number of input words to be accepted, and the location of the data buffer in memory where the incoming data should be stored. The first #MIN format consists of a two word instruction. The second word contains all 24 bits of the command to be sent. The first word contains an 8

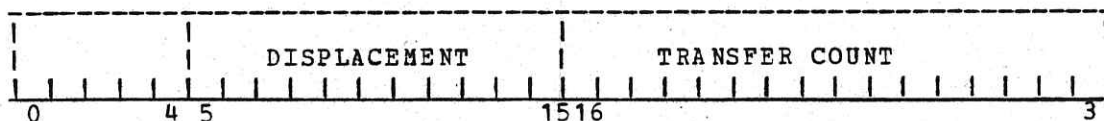
FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
 REPRODUCED IN ANY MANNER.

MS 87-08
 Box 42
 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

bit Base relative displacement and a 16 bit Transfer Count. The Displacement may be any positive integer between 0 and 255, and is added to the present contents of the Base Register to form the address of the data buffer. The Transfer Count may be any positive integer between 0 and 65535, and represents one less than the actual number of data half-words to be received.

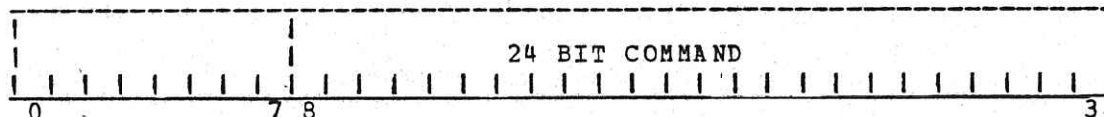
The second #MIN format allows automatic indexing by BCE number into two tables providing basically the same information found in the first format. Each table has 24 entries, one per BCE. Twice the number of the BCE executing this instruction is used as an index so that the table entries are each fullwords.

The first table starts at ADDRESS + 2, and contains the Displacement and Transfer Count. For BCE i the table entry at ADDRESS + $2 \times i$ is formatted as:



This allows a range of displacement between 0 and 2047 and a Transfer Count between 0 and 65535.

The second table starts at ADDRESS + 50, and contains the 24 bit command. For BCE i the table entry at ADDRESS + $48 + 2 \times i$ is formatted as:



Execution of a #MIN depends on the current mode of the BCE -- Command or Listen. In Command mode a BCE is assumed master of the bus connected to its MIA. Consequently, to receive data from some subsystem on that bus, the BCE must first issue a command telling the subsystem what type of data to return. The 24 bit command field in the #MIN contains the command. After issuing the command the BCE prepares to accept returning data.

In Listen mode, a BCE is not master of its bus, and must rely on some other BCE in another IOP to issue the command to the subsystem. A BCE in Listen mode that executes a #MIN thus does not transmit the command part of the #MIN, but instead goes directly into a MIA input monitor loop. However, like a Receive Data instruction, a #MIN executed in Listen mode will not expect to see data. Instead it will wait until it sees that some other BCE/IOP has issued a command and only then prepare to accept input data. This difference allows multiple BCE/IOPs connected to the same bus to use the same

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY FORM OR BY ANY MEANS.

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

routine to receive the same data from a single subsystem, but with only one BCE actually transmitting the commands to the subsystem.

An additional difference between the execution of a #MIN in Command or Listen mode is the setting of the BCE's IUAR. In Command mode the BCE's IUAR will be set from bits 8 to 12 of the instruction word containing the command. In Listen mode, the IUAR is unchanged. It is assumed that the Listen command sent by the command BCE (see #WIX instruction) contained the subsystem address that is to be involved in the following communications. Paragraph 4.1 -- Listening Mode -- should be seen for more detail.

In either mode, once the command has been handled the BCE enters the same reception algorithm used for all the Receive Data instructions (see #RDS).

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY MANNER WITHOUT THE WRITTEN PERMISSION OF THE

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents

Wichita State University Libraries, Special Collections and University Archives

3.5 SPECIAL INSTRUCTIONS

The BCE instruction set contains several instructions not falling under any of the previously defined categories. These include an instruction to delay the BCE for specifiable periods of time, and an instruction to cause the BCE to enter the Wait State.

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED
REPRODUCED IN ANY FORM NOR PLACED IN ANY REPOSITORY

MS 87-08

Box 42

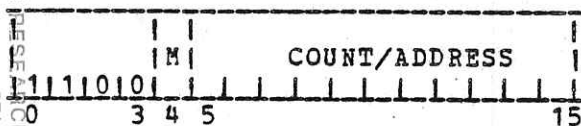
FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

#DLYI, #DLY

DELAY

FORMAT



Effective Count

INSTR.Format

TIMEOUT¹

#DLYI Timeout

1 (PC+DISP+2XBCE#)²

#DLY Address

Notes:

1. Any value between 0 and 2047. This corresponds to delays from 0 to 33.78 millisecc.
2. The Lower 18-bits of the fullword addressed by PC(updated)+ Displacement. This allows any count between 0 and 262143, or 0 to 4.325 sec.

DESCRIPTION

This instruction simply delays the execution of the next instruction. The time period delayed is a function of the Effective Count, with a resolution of 16.5 microseconds per count.

At the end of this delay the program counter is incremented by one, and the next instruction is executed.

PROGRAMMING NOTE

Each count of 1 represents a delay of 16.5 microseconds, the execution time of a BCE micro instruction. Each count of 2 represents a delay of 33 microseconds, the minimum time for a word transmission over a serial bus.

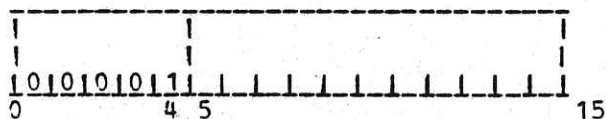
For the #DLY instruction, the address of the word containing the delay includes the number of the BCE executing the instruction. This allows many BCE's to execute the same program while still retaining the capability to delay for different periods. Note that twice the BCE number is a fullword index.

WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 COPY PROVIDED BY
 (TITLE 17 U.S. CODE)
 PROTECTED BY COPYRIGHT LAW
 THIS MATERIAL MAY BE
 FOR RESEARCH USE ONLY
 WITHOUT WRITTEN PERMISSION. THIS MATERIAL MAY NOT BE COPIED
 REPRODUCED IN ANY MANNER NOR PLACED IN ANY REPOSITORY

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

WAIT

#WAT

INSTRUCTION FORMAT

#WAT

DESCRIPTION

This instruction causes the Bus Control Element (BCE) that is executing it to leave the busy state and enter the wait state. The BCE's Busy/Wait bit (in STAT4) is set to 0 (WAIT). The BCE's Program Counter is incremented by 1, but no further instructions are executed. The BCE is reset to the busy state by the Master Sequence Controller (MSC) only. Once in the Wait State, a BCE performs no actions other than the monitoring of its Busy/Wait bit for a command to re-enter the Busy State.

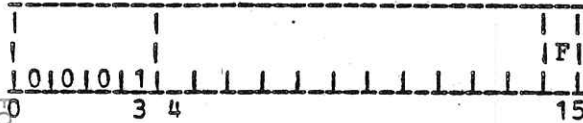
Paragraph 2.2 describes transitions to and from the Wait State in detail.

PROGRAMMING NOTE

While in the Wait State the BCE alters none of its programmer-visible registers. Thus the CPU is free to change, via PCI/0, any BCE register.

INSTRUCTION - SELF TEST

#STP

INSTRUCTION FORMAT

#STP FLAG

DESCRIPTION

This instruction initiates execution of a special micro program to perform self tests on the hardware supporting the Bus Control Element that is executing this instruction. These tests include checks of:

- o BCE Local Store
- o BCE Data Flow Operations
- o Ability of BCE to read and write from memory
- o MIA wrap capability
- o MIA Buffer

A flag of 0 causes all but the last two tests to be performed. A flag of 1 causes all tests to be run.

If an error is detected, the BCE's Program Exception bit is set to 0, bit 22 of the BCE Status Register is set to 1 (Self Test failure). The PC is incremented by 1, and the next instruction is executed.

Successful completion of this instruction causes the BCE to increment its PC by 1 and continue with the next instruction.

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE)
 COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED

REPRODUCED IN ANY MANNER FROM THE ARCHIVES

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

4.0 LISTEN MODE

In the Space Shuttle configuration there can be up to five separate CPU/IOP pairs connected to the same set of busses. When running in redundant mode, all CPU/IOPs want to receive exactly the same copies of all input data from relevant sensors so that the resulting outputs should all match. This presents problems. First, since a sensor in normal operation is tied to only one bus, only one IOP at a time can send it the requisite commands to return data. Any attempt by two or more IOPs to send commands over the same bus at the same time results in interference and garbling of the bus signals. Thus, if each IOP is to send a command to the same sensor telling it to return data, there must be some definite time sequence that prevents conflicts. Such sequencing is not only difficult to achieve, but also makes it impossible to guarantee that exactly the same data reaches each CPU/IOP.

The BCE Listening Mode solves this problem by allowing only one BCE/ IOP to place subsystem commands on a bus, but at the same time allow it to inform all other BCE/IOPs connected to the bus that a certain subsystem is to be commanded and that they should set up to handle the returning data. This technique not only eliminates the sequencing problem mentioned above, but also allows all GPC/IOPs to receive exactly the same data at exactly the same time.

The following paragraphs describe Listen Mode, first by example and then in detail.

4.1 SAMPLE OPERATION IN LISTEN MODE

Figure 4.1 diagrams a configuration of redundant IOPs that will use a Listen Mode to accept data from a single subsystem, Subsystem M. Also included is a reference BCE program segment. The BCE in GPC 1 is assumed to be configured in Command Mode. The equivalent BCEs in GPCs 2, 3, and 4 are in Listen Mode. All BCEs are assumed in Wait State initially.

Before a BCE in Listen Mode can accept a command from another BCE in a separate GPC, it must be executing a #WIX instruction -- Wait for Index. This instruction causes a BCE in Listen Mode to monitor its bus for a command from another GPC. The only way a BCE can be executing a #WIX is if its MSC has loaded the PC (via a @LBP or CPU PCO) and started it (via a @SIO) at a memory location containing a #WIX. Once a BCE has been started in this fashion, it will return to the Wait State only when it has either executed a standard #WAT instruction or it has encountered an error. Any attempt by the MSC to change the operation of the BCE after it has been started will be aborted, with error flags set in the MSC's status register.

Since the operational mode of a BCE in an operational sequence will very rarely be changed, the MSC in a GPC will initialize its Listen Mode BCEs once after a reconfiguration, and will leave them alone unless they execute a #WAT or error terminate. For this example

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED
REPRODUCED IN ANY MANNER NOR PLACED IN ANY REPOSITORY

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

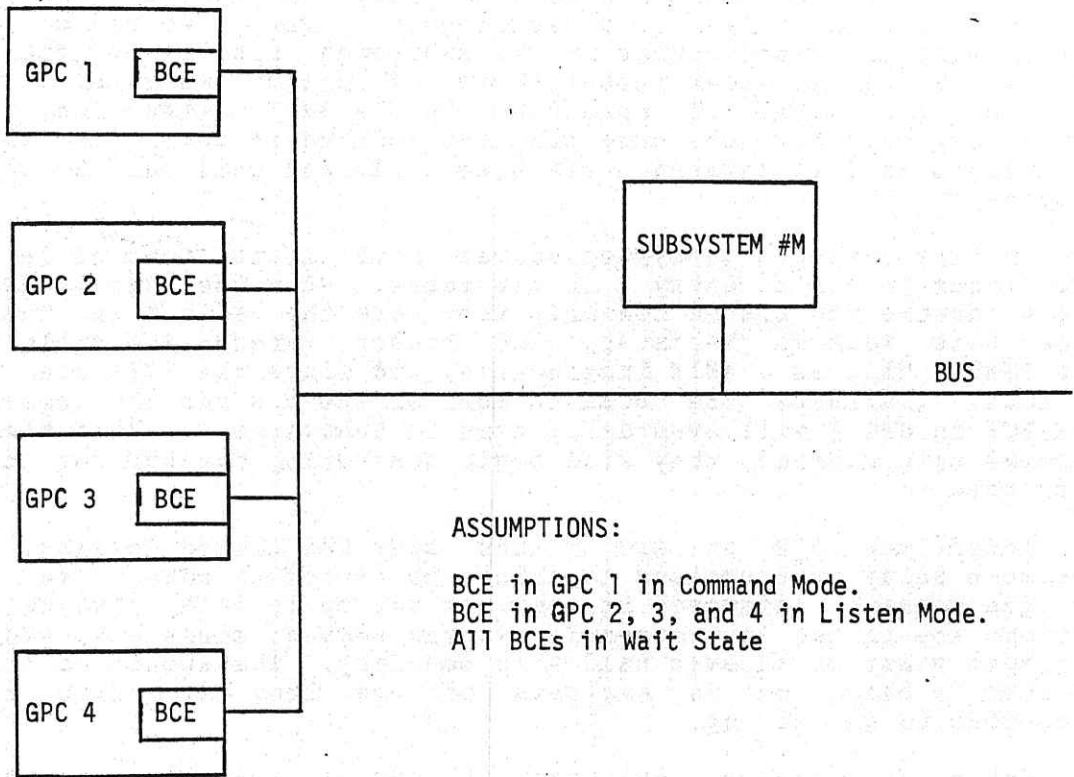
we assume that the BCEs in GPCs 2, 3 and 4 have been initialized to execute a "WIX TABLE" instruction in the near past, and are now watching their bus for a Listen Command.

At some point in time the MSC in GPC 1 will determine that it is time to receive data from Subsystem M and that its BCE is in Command Mode. The MSCs in the other GPCs will also come to the conclusion at about the same time that it is time for Subsystem M to give data, but since their BCEs are in Listen Mode, they do nothing active to get it. To get the needed data, the MSC in GPC 1 initializes its BCE to execute the program at location "START". When the MSC executes a @SIO, the BCE in GPC 1 enters the Busy State and begins execution at location START. The first instruction then tells the BCE to place on the bus a "Listen Command". This command is meant for all BCEs on the bus that are in Listen Mode and are executing a #WIX. It is distinguished from all other traffic on the bus by a unique Interface Unit Address (IUA) not used by any subsystem. This IUA is termed a "Common IOP Address", and in binary is 0 1 0 0 0.

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT
(TITLE 17 U.S. CODE)
COPY PROVIDED BY
WICHITA STATE UNIVERSITY LIBR
SPECIAL COLLECTIONS
WITH PERMISSION THIS MATERIAL MAY NOT
BE REPRODUCED OR TRANSMITTED IN ANY FORM OR BY ANY MEANS

MS 87-08 Box 42 FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives



REFERENCE PROGRAM SEQUENCE

START	#CMDI	IUA = Common IOP Address, Device # = M, Index = 2.
	#DLYI	
PGM2	#DLYI	
	#MIN	IUA = M, . . .
	#WIX	Table
TABLE	DC	A (PGM0)
	DC	A (PGM1)
	DC	A (PGM2)
	.	
	.	
	.	

Figure 4.1. Listen Mode Configuration

FOR RESEARCH USE ONLY
 THIS MATERIAL MAY BE
 PROTECTED BY COPYRIGHT LAW
 (TITLE 17 U.S. CODE).
 COPY PROVIDED BY
 WICHITA STATE UNIVERSITY LIBRARIES
 SPECIAL COLLECTIONS
 WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED
 REPRODUCED IN ANY FORM OR BY ANY MEANS

MS 87-08 Box 42 FF 50
 Dr. James E. Tomayko Collection of NASA Documents
 Wichita State University Libraries, Special Collections and University Archives

When a BCE in Listen Mode detects a word on the bus that has command sync and the Common IOP Address, they accept it and use the information to set themselves to a new program. The information in this word consists of the number of the subsystem that will originate the information, and an index into a table of branch address. The beginning of the table is specified by the #WIX instruction. We assume that all GPCs have the same programs located in memory so that all listening BCEs will reference the same table and pull out the same branch address.

In this case the subsystem number in the Listen Command is M, and the index points to entry 2 in the table. When the BCEs in GPCs 1, 3 and 4 receive the Listen Command, they save the value M in their Interface Unit Address Register, and branch through the table to location PGM2. This is a #MIN instruction, and since the BCEs are in Listen Mode, instructs the BCEs to monitor the bus for the command that the BCE in GPC 1 will eventually send to Subsystem M. When these BCEs receive this command, they will begin monitoring the bus for the returning data.

After the BCE in GPC 1 has sent the Listen Command, it executes some delay instructions to allow the Listening BCEs time to receive the command, interpret it, and get set up in #MIN. Two delay instructions are listed in the sample program segment since both #CMDI and #MIN must start on an even half word boundary. The amount of time delayed must be based on an analysis of how long it takes the Listening BCEs to get set up.

After the delays, the BCE in GPC 1 executes the #MIN instruction. Even though it is the same instruction executed by the Listening BCEs, the fact that the BCE is in Command Mode causes a different execution. The BCE will place on the bus the command to Subsystem M, telling it to return the appropriate data. The BCE will then prepare itself to handle this data.

When Subsystem M places its data on the bus, all four BCEs are prepared to accept it, and consequently receive exactly the same data at exactly the same time.

After successful reception of the data from Subsystem M, all BCEs execute the #WIX instruction. For the command BCE this is equivalent to a #WAT, and the BCE re-enters the Wait State where it will await another command from its own MSC. For the Listening BCEs, the #WIX instruction simply places them back into a loop where they are monitoring the bus for another Listen Command.

The above sequence obviously can be elaborated to include multiple #MINS from the same subsystem, instructions to change the Base or Time Out Register, or do other functions, such as set the BCE-MSD indicator bits to indicate completion of a BCE Listen Program. The extensions, however, are application dependent, and do not change the basic operation described above.

One of the important aspects of this example is that the software stored in each computer is identical. The commanding and listening BCEs use exactly the same instructions to receive the same data from the same subsystem. Only the mode of the BCEs was different.

4.2 INITIALIZATION INTO LISTEN MODE

Listen Mode is a mode of BCE operation, not a state. It affects how a BCE interprets instructions, but does not of itself represent a completely different state of BCE execution.

A BCE is in Listen Mode when its MIA transmitter is disabled and its receiver enabled. It may be changed at any time without harm while a BCE is in Halt state. Once a BCE is in Busy State, changing the mode of that BCE may cause indeterminant BCE actions. It is recommended practice that a BCE be halted before a mode change, and then enabled after the mode change is complete. This will prevent any possible indeterminant BCE activity.

The signals that may change the MIA's status are as follows:

1. CPU PCO's allow selective set/reset of each MIA's transmitter and receiver.
2. XMIT Disable Discrete disables all MIA transmitters.
3. Input discrete 13 disables MIA transmitters 10-13.
4. Input discrete 14 disables MIA transmitters 14-17 and 20-23.

4.3 DIFFERENCES IN INSTRUCTION EXECUTION

A BCE enters the Busy State only from Wait and only when the MSC has executed a @SIO. This transition is independent of the BCE's mode. The distinction between Listen and Command mode becomes significant only after the BCE has entered the Busy State. As demonstrated in the previous example, this distinction reflects itself in how various instructions are executed. Table 4.1 summarizes those BCE instructions that are influenced by mode. Each difference is discussed separately in the following paragraphs.

The Wait for Index instruction (#WIX) was designed for Listen Mode. In Command Mode #WIX is equivalent to a Wait instruction -- a #WAT. In Listen Mode a #WIX places the BCE in a loop where it is monitoring its MIA for a Listen Command (Interface Unit Address = 0 1 0 0 0). When it receives such a command it branches through a table of addresses. The BCE stays in the Busy State during the entire sequence, and will leave the Busy State only if it is halted, it receives a Listen Command that directs it eventually to a #WAT instruction, or it receives a Listen Command, starts executing a program, but is stopped by an error from further execution.

FOR RESEARCH USE ONLY
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY FORM OR BY ANY MEANS ELECTRONIC OR MECHANICAL
INCLUDING PHOTOCOPYING, RECORDING, OR BY ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives

It should be stressed that a BCE will monitor its bus for a Listen Command only when (1) the BCE is in Listen Mode and (2) it is executing a #WIX.

In Listen Mode a BCE's transmitter is not enabled. Consequently, it is not able to transmit either commands or data. This affects the execution of #CMDI, #CMD, #MIN, #MOUT, #TDS, #TDLI, and #TDL.

Any attempt by a BCE to execute a #MOUT, #TDS, #TDLI, or #TDL while in Listen Mode will be met by error terminating the BCE, i.e. setting Status Bit 23, resetting the Program Exception Bit to 0, and entering the Wait State. A BCE in Listen Mode should never be told to transmit data.

The instructions #CMDI, #CMD, and #MIN are not error terminated when executed by a BCE in Listen Mode. Instead, #CMDI and #CMD are treated as no-operations-no error indicators are set, and the BCE continues with the next instruction. Likewise in #MIN the transmission of the command is suppressed -- the BCE branches into its receive mode. The reason for this difference is to allow the same program segments to be used by BCEs in both Command and Listen Modes to receive the same data, as demonstrated in the example of Figure 4.1.

Execution of #RDS, #RDLI, #RDL, and the receive part of #MIN also differ between Command and Listen Mode, primarily in accepting the first data input. In Command Mode, when these instructions enter the receive algorithm, they immediately start waiting for the first data word, and timing out its arrival by decrementing a timer that had been set to the value in the MTO Register. If the first input the BCE finds in the MIA Buffer is a command, the BCE will discard the command, reset the timer, and start looking for data again. This allows the BCE to ignore the command that was sent to the subsystem but was also echoed back by the MIA to the BCE's MIA Buffer.

A BCE in Listen Mode goes through a different sequence. Rather than immediately waiting for data, the BCE waits (indefinitely) for a bus word having command sync, good parity, etc., and an IUA that matches the BCE's IUAR. This command should be the command being sent by the other BCE/IOP in Command Mode to the subsystem, telling it to return data. The BCE uses the arrival of this command as a signal to set its timers and start watching for arrival of the first input of data. This allows the Listening BCE to accurately time out the first data word, and detect if that word is excessively delayed in its arrival.

After arrival of the first word all BCEs use the same reception algorithm.

FOR RESEARCH USE
THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17 U.S. CODE)
COPY PROVIDED BY
WICHITA STATE UNIVERSITY LIBRARIES
SPECIAL COLLECTIONS
UNIVERSITY OF WASHINGTON PERMISSION THIS MATERIAL MAY NOT BE COPIED OR
REPRODUCED IN ANY MANNER WITHOUT THE WRITTEN PERMISSION OF THE ARCHIVAL REPOSITORY

MS 87-08

Box 42

FF 50

TABLE 4.1
DIFFERENCES IN INSTRUCTION EXECUTIONS
DUE TO BCE MODE

<u>BCE INSTRUCTION</u>	<u>EXECUTION IN COMMAND MODE</u>	<u>EXECUTION IN LISTEN MODE</u>
#WIX	Enter Wait State	Wait for Listen Command (Stay Busy)
#CMD, #CMDI	Transmit Command	No-operation
#MIN	Transmit Command, Receive Data	Wait for Command, Receive Data
#RDS, #RDLI, #RDL	Receive Data	Wait for Command, Receive Data
#MOUT	Transmit Command, Transmit Data	Set Error Bits, Enter Wait State
#TDS, #TDLI, #TDL	Transmit Data	Set Error Bits, Enter Wait State

APPENDIX A

IOP BCE INSTRUCTION REPERTOIRE

<u>NAME</u>	<u>MNEMONIC</u>	<u>FORMAT</u>	<u>PAGE</u>
<u>BCE Register Operations</u>			29
Load Time Out	#LTOI, #LTO	Short 1	30
Reset Indicator Bit	#RIB	Short 1	31
Set Indicator Bit	#SIB	Short 1	32
Store Status and Clear	#SSC	Short 1	33
Store Status	#SST	Short 1	34
Load Base	#LBR	Long	35
<u>BCE Branching</u>			36
Branch Unconditional	#BU	Long	37
Wait for Index	#WIX	Short 1	38
<u>BCE Transmit Instructions</u>			42
Transmit Command	#CMDI, #CMD	Long	49
Transmit Data Short	#TDS	Short 2	51
Transmit Data Long	#TDLI, #TDL	Long	53
Message Out	#MOUT	Long 2	54
<u>BCE Receive Instructions</u>			58
Receive Data Short	#RDS	Short 2	71
Receive Data Long	#RDLI, #RDL	Long	73
Message In	#MIN	Long 2	74
<u>Special Instructions</u>			77
Delay	#DLYI, #DLY	Short 1	78
Wait	#WAT	Short 1	79
Self Test	#STP	Short 1	80

NOTE: See Figure 1.3 and 1.4 for Formats.

FOR RESEARCH USE ONLY

THIS MATERIAL MAY BE

PROTECTED BY COPYRIGHT LAW

(TITLE 17 U.S. CODE)

COPY PROVIDED BY

WICHITA STATE UNIVERSITY LIBRARIES

SPECIAL COLLECTIONS

WITHOUT WRITTEN PERMISSION, THIS MATERIAL MAY NOT BE COPIED OR

MS 87-08

Box 42

FF 50

Dr. James E. Tomayko Collection of NASA Documents
Wichita State University Libraries, Special Collections and University Archives