Publication Number C29-473R03

QSA / SSA

PROGRAMMING MANUAL

PERKIN-ELMER

Computer Systems Division
2 Crescent Place
Oceanport, N.J 07757

PAGE REVISION STATUS SHEET

PUBLICATION NUMBER C29-473

TITLE Quad Synchronous Adapter (QSA) Programming Manual

REVISION R03 DATE October 1979

PAGE	REV.	DATE	PAGE	REV.	DATE	PAGE	REV.	DATE
i/ii	R03	10/79						
1 thru 16	R03	10/79						
A-1 thru A-4	R03	10/79						
B-1 thru B-29/ B-30	R03	10/79						
C-1 thru C-29/ C-30	R03	10/79						

PREFACE

This manual provides programming information for the PERKIN-ELMER Quad Synchronous Adapter (QSA) Product Numbers M47-002 and M47-003, and the Single Synchronous Adapter (SSA) Product Numbers 47-106 and 47-107. The QSA or the SSA provides an interface between the Multiplexor Bus, Memory Access Multiplexor (MAM), or Selector Channel Bus of a PERKIN-ELMER digital system and up to four Two-Wire or Four-Wire Synchronous Data Sets. The manual includes an introduction to the adapters, and detailed information concerning configurations, data formats, programming instructions, sequences, and constraints, interrupts, initialization, and device addressing. Appendices to the manual provide a programmer's guide, and 16-bit and 32-bit programming examples.

QSA/SSA
PROGRAMMING MANUAL

1. INTRODUCTION

This document provides programming information for the Perkin-Elmer Single and Quad Synchronous Adapters. It assumes that the reader is familiar with the I/O programming structure of Perkin-Elmer Processors. One manual which will be of interest is the appropriate Processor User's/Reference Manual.

The Quad Synchronous Adapter (QSA) and the Single Synchronous Adapter (SSA) are designed to operate with synchronous modems. They contain the serial/parallel converters, character buffers, and control circuits for one or four modems (1 or 4), operating in either two-wire or four-wire mode. For technical information concerning the SSA or QSA, refer to the SSA Installation and Maintenance Manual, Publication Number 29-689, or the QSA Instruction Manual, Publication Number 29-411.

2. CONFIGURATION

The 381 mm (15") SSA, Marketing Number M47-106 and M47-107, or QSA, Marketing Number M47-002 and M47-003, may be configured on Models 6/16, 7/16, 8/16E, 7/32, 8/32, Series Sixteen, or Series 3200. They provide one or four double buffered interfaces to synchronous data sets, such as the Bell 201. The QSA does not include data set signal level conversion, but interfacing to a modem is made possible by using the appropriate Line Conditioning Module (LCM). Provision is also made to connect an SSA to an LCM for other than RS-232 signal level conversion.

When strapped for four-wire operation, an SSA has two and a QSA has eight consecutive device addresses, with each consecutive pair being the receive and transmit addresses for each line. The receive line is always assigned the even address and the transmit line address is odd. For two-wire operation, each line responds to the even address, unless two-wire address interleaving is installed. The address interleaving scheme enables the consecutive addressing of eight two-wire lines, on two Quad Synchronous Adapters. This scheme is described further in the "DEVICE ADDRESSING" section.

The SSA or QSA can also be configured under a Selector Channel (SELCH) for high speed data transfer, a Memory Access Multiplexer, or the Auto Driver Channel.

3. DATA FORMAT

Various line and control options are available on the SSA and QSA. They are described in the following paragraphs.

3.1 Data Set Operating Modes

Each data line on the SSA and QSA is provided with strap options to operate in two-wire or four-wire mode. Each board is provided with strap options to provide automatic deletion of leading SYNC characters, and Automatic Resynchronization after termination, during the Zero Bit Insert/Delete (ZBID) receive mode.

3.2 Transmission Rate

The transmission rate is fixed by the modem being used with the SSA or QSA. The interface is transparent to this rate which must not exceed the maximum of 1.6M baud. The modem must always supply the transmit and receive clocks. The Local Loopback mode of operation uses an internal test clock that is strappable to two approximate baud rates; 8k baud with the strap removed and 800 baud with the strap installed.

3.3 Method of Transmission

In non-ZBID mode, data is serial, synchronous by character and bit, using SYNC characters for character synchronization. Any 5, 6, 7, or 8 bit character in the range of X'01' to X'FE' (with or without parity) may be used, and is program selectable for each line.

The ZBID mode is required to accommodate several communication conventions, and is selected under program control. In this mode, the character length is fixed at eight bits per character and the Flag (F) character is hardware fixed at X'7E'.

4. PROGRAMMING INSTRUCTIONS

The Processor I/O Instructions are used to communicate with the SSA and QSA.

4.1 SSA/QSA Programming Instructions

4.1.1 Sense Status (SS or SSR)

The Sense Status instruction is used to determine if character transfers are complete and correct, and also to interrogate the associated modem status.

4.1.2 Output Command (OC or OCR)

The Output Command instruction is used to Enable or Disable interrupts, answer or disconnect calls, and to set the interface to the Receive or Send Mode.

4.1.3 Write Data (WD or WDR)

The Write Data instruction is used to load the output character into the Data Register, and to set the SYNC character register.

4.1.4 Read Data (RD or RDR)

The Read Data instruction is used to read an assembled character into the processor.

4.1.5 Acknowledge Interrupt

The Acknowledge Interrupt command provides two different types of commands. When bit 15 is reset, the command is recognized as the common command for setting the function mode of the addressed line (both Received and Transmit). When bit 15 is set, the command is the Receive/Transmit command. In four-wire mode, the command sent to the Receive line is interpreted as a Receive command and to the Transmit line as a Transmit command. In two-wire mode, command bit 14 (RQ2S) sets the Transmit or Receive mode.

4.2 Command Byte

Table 1 shows the interpretation of each bit in the Command byte.

TABLE 1 SSA/QSA COMMAND BYTE

	8	9	10	11	12	13	14	15
RECEIVE	DISABLE DISARM	ENABLE	LOCAL LOOP BACK	SYNSCH	READY (DTR)	SPECIAL	WRITE/ RQ2S	1
TRANSMIT	DISABLE DISARM	ENABLE		RESET DATA MODE				
COMMON			NO. OF DATA BITS SELECT			LINE MODE CONTROL		0

Disable When this bit is set and Enable is reset, any interrupts occurring for that device address are not passed to the processor but are queued.

Enable When this bit is set and Disable is reset, any interrupt occurring for that device address is presented to the processor.

Disarm

When both the Enable and Disable bits are set, no interrupts are generated or queued by the device address. All eight device addresses on the QSA and both device addresses on the SSA are set to this mode on initialization.

*Local Loopback

This command bit, if set, causes the transmit and receive lines to go to the Local Loopback mode. All control and data outputs to the modem are forced to their OFF state and all incoming control and data line signals are taken from the specified local loopback source line (see the "Local Loopback" section).

Local Loopback csnnot be used when the SSA or QSA is configured under a Selector Channel.

Synsch

This command bit, if set and issued to the Receive line (or two-wire line in receive mode), causes the line to go the SYNC/Flag Search (SYNSCH) mode. Incoming serial data is discarded and not presented to the processor until the correct synchronization sequence is detected. See the Synchronization section. A command with this bit reset does not change the current state of the line, but resets the OV and Parity Error bits of the Status byte, if set.

Reset Data Mode

A command with this bit set and the RQ2S bit set to the transmit line (or a two-wire line in the transmit mode) while in BISYNC mode (non-ZBID) forces the line to transmit continuous MARK characters (all bits set including parity is enabled), starting at the completion of the present character being shifted out (EOC). In ZBID mode, this command causes the line to transmit continuously the last contents of the transmit data holding register, without inserted zero bits, starting at the completion of the present character being shifted out (EOC). A command with this bit reset and RQ2S set causes no change to the transit line data mode. The transmit line returns to the data mode at the next EOC after a Write Data, while Data Mode is reset. In ZBID Mode, the SSA/QSA automatically inserts one leading flag character (X'7E') before the data frame is transmitted.

*Ready

When this bit is set, a command forces the Data Set Ready (DSR) control line to the modem to go to the ON level. This bit must be set on all Receive and Transmit commands to maintain the data mode. If this bit is reset, DSR is forced off, carrier and CL25 return to the idle state, and an existing call is terminated. Ready must be sent to answer an incoming call.

*Special

This command bit is used to control the state of special modem control signals, such as NEW SYNC and SPEED SELECT (CH/CI). The special control line is forced to the ON level by setting this bit. This bit is ignored if no special function is wired. (See the LCM Instruction Specification, 02-380A20 for wiring information.)

RQ2S

This command bit controls the state of the REQUEST-TO-SEND (RQ2S) control line to the modem. In a two-wire configuration, it also controls the state of the Transmit and Receive mode of the line.

In two-wire configurations, a command with this bit set switches the line into Write mode and forces the RQ2S control line active. Interrupts are disarmed from the receive logic and only the Transmit line status is returned. A command with this bit reset, switches the line into the Read mode and forces the RQ2S control line inactive. Interrupts are disarmed from the Transmit logic and only the Receive line status is returned. A subsequent write data to the line in the Read mode sets the programmable SYNC match character register.

In four-wire configurations, a command to the transmit line with this bit set forces the RQ2S. The program must, therefore, leave RQ2S active for at least two character periods after the last useful character is output to ensure that it has been transmitted. One method is to write two or more pad characters (e.g., X'FF') after the data in non-ZBID mode.

No. of Data Bits

These two bits control the number of data bits (does not include parity per character for non-ZBID mode. In ZBID mode, the SSA/QSA forces eight bits/character regardless of these two command bits. Characters are right-justified in the data byte, if less than eight bits per character. If parity is specified, the number of bits transmitted per character is equal to this value plus one.

Bit No. 10	Bit No. 11	Bits/Character
0	0	5 bits/character
0	1	6 bits/character
1	0	7 bits/character
1	1	8 bits/character

In a four-wire configuration, these command bits are common to both the transmit and receive side of the line. The last command to either side stands for both sides.

Line Mode Contrcl These two bits control the line mode as follows:

Bit No. 13	Bit No. 14	Line Mode
0	0	No parity
0	1	Zero bit Insertion/Detection
1	0	Odd parity (inserted-checked-deleted)
1	1	Even Parity (inserted-checked-deleted)

NOTE

The Data bit Select and Line Mode Control bits are undefined at initialization. The program must define them by an Output Command.

4.3 Status Byte

Table 2 displays the status bytes for both the transmit and received sides.

TABLE 2 SSA/QSA STATUS BYTES

	8	9	10	11	12	13	14	15
RECEIVE	OV	PARITY ERROR/ TERM.	SPECIAL	RING	BUSY	EXAMINE	CARRIER OFF	
TRANSMIT	OV				BUSY	EXAMINE	$\overline{CL2S}$	$\overline{DSR}$

OV Overflow. This bit is set if the SSA/QSA is not serviced at its data rate (within 1/2 of a character time).

Data was not read from the SSA/QSA before another was assembled in the Receive mode. In this case, a data byte was over written in the SSA/QSA.

Data was not supplied to the SSA/QSA in time, during transmission, and the previous character was re-transmitted in the non-ZBID mode. For the ZBID mode, the flag is set upon an overflow condition.

This bit, once set, remains set until reset by a command to the line with bit-15 set or by system Initialization.

Parity

Parity Error/Termination. During non-ZBID mode operation, this bit indicates that a parity error has been detected on the incoming character. During ZBID mode operation, this bit indicates that the hardware control logic has detected a flag or an abort condition.

This bit is reset by any command to the Receive line with bit 15 set. In non-ZBID mode, the next incoming data character with good parity resets this status bit.

Special

This bit is set when the special status function from the modem is active. This bit is always reset if no special function is wired in the associated LCM. See the LCM Installation Specification 02-380A20.

Ring

This bit is set when the RING control line from the data set is ON.

Busy

This bit is set in the Status Byte when the line is not ready to present a data byte to the processor (Receive) or to accept a data byte from the processor (Transmit). It is also set by the following conditions:

Receive Busy:

- The Carrier-ON (CO) line from the modem is at the OFF level.
- The receive data buffer has not been loaded from the character assemble register since the provious 'Data Request'.

Transmit Busy:

- The CLEAR-TO-SEND (CL2S) line from the modem is at the OFF level.
- A 'Write Data' is issued from the processor.

This bit is reset when the Receive Data Buffer is loaded or when the Transmit Data buffer is emptied.

Examine

This bit is set when OV or Parity Error/Term status bit is set.

Carrier

This bit, when set, indicates that the CO line from the Data Set is OFF.

$\overline{\text{CL2S}}$

Not Clear to Send. This bit, when set, indicates that the CL2S line from the modem is OFF.

$\overline{\text{DSR}}$

Data Set Not Ready. This bit, when set, indicates that the Ready (RDY) line from the modem is OFF.

5. SYNCHRONIZATION

Refer to Figures A1-4, A1-5, and A1-6 in Appendix A.

In the non-ZBID mode, the Receive line is synchronized with the incoming data upon reception of two consecutive SYNC characters. Ensuing leading Sync characters are discarded if the automatic SYNC delete option is installed; otherwise all data following synchronization is loaded into the buffer and presented to the processor. A Write Data to the Receive side of a four-wire line, or a Write Data to a two-wire line in Receive mode, sets the SYNC Match Character register. When the adapter is initialized, the contents of the SYNC Match Character register is undefined and must be set by the program.

In ZBID mode, the Receive line is synchronized with incoming data upon detection of a single flag character of X'7E', fixed by hardware. It remains in as 'flag received' state with no data presented to the processor, if the ensuing characters are flags. If an abort character, any character of 6 or more contiguous 1 bits, not equal to a flag (X'7E'), without inserted zero bits, (i.e., X'3F', X'FC', C'7F', X'FE', or X'FF') is detected following a flag, the line is put back into SYNC Search mode ready to establish synchronization, upon the reception of another flag (X'7E'). If a data character (neither flag nor abort) is detected immediately following the detection of a flag character, the line goes into the data mode, and data is presented to the processor. Once in the data mode, a flag or an abort character terminates the data frame and sets the Parity Error/Term status bit. If automatic resynchronization is installed, a termination flag character puts the line back to the 'flag received' state, ready to go into the Data mode on the next data byte; otherwise, the line goes into the SYNC Search mode, to be resynchronized by another flag.

The SYNC characters must be supplied as data to the transmit line in non-ZBID mode. After a data transmission is terminated by OV, the transmit line continues to send the last contents of the contiguous flag characters.

In ZBID mode, the transmit logic precedes the first data byte of a frame with a flag character, and after a frame transmission is terminated by OV, the transmit line sends multiple contiguous flag characters.

6. INPUT/OUTPUT

To set up a Read operation, the SYNC Match character register should be set as described in the "Synchronization" section. An Output Command with SYNSCH and DTR should be issued to the Receive line (RQ2S must be reset). At the next change of status (or interrupt) after carrier is ON, a zero status indicates that a character is available. A Read Data must be issued before the end of the character time.

To set up a write operation with a non-ZBID transmit line, an output command with RQ2S, DTR, and Reset Data Mode should be issued. (Reset Data Mode ensures that the line is at a constant MARK state.) At the next change of status (or interrupt) after CL2S goes active, a zero status indicates that the adapter is ready to accept a character. A Write Data initiating the data transfer may be issued at any time after an all-zero status occurs. After CL2S goes active, the transmit SSA/QSA generates an interrupt, if enabled, once every character period, regardless of whether or not a data transfer is in progress. OV will not set until a data transfer is initiated and the OV condition is detected. If RQ2S is reset, the communication link is lost.

A write operation using a ZBID transmit line requires a slightly different procedure. Refer to the section entitled "Zero-Bit Insertion/Deletion" for details.

NOTE

> All commands to a two-wire line which cause a line turn around (Write to Read or Read to Write) must disarm interrupts.

If the Auto SYNC Delete option is installed, a leading character with parity error (after synchronization is complete) is not deleted. Instead, the Receive line goes into the Data mode and the first character read is the SYNC character in error, with the "parity error" status bit set.

High Speed Data Transfer

In order to support high speed data sets, the SSA/QSA must be configured under a Selector Channel and operated in the two-wire mode. Data transfers must be in fixed block lengths. However, in ZBID operation, the SSA/QSA is able to receive variable data blocks because the ZBID termination flag status terminates the SELCH.

Since the SELCH is stopped by the termination status, the automatic resynchronization option is not available in this configuration.

Local Loopback

The Local Loopback feature provides on-line testing ability. When the Local Loopback command is issued, all data and control line signals to the modem go to the inactive state, and the receive logic gets data and control line signals from another transmit line (another line). The Local Loopback command is separate for each line, but the receive/transmit address pairs are not changeable. The QSA Local Loopback sources and destinations are specified in Figure 1. The control line loopback for SSA and QSA are shown in Figure 2.

For Local Loopback Baud rates refer to "Transmission Rates".

QSA TRANSMIT LINE	IS LOOPED INTO	QSA RECEIVED LINE
(X'BD') 2	→	0 (X'B8')
(X'BF') 3	→	1 (X'BA')
(X'B9') 0	→	2 (X'BC')
(X'BB') 1	→	3 (X'BE')

Figure 1 QSA Local Loopback Configuration

RQ2S	→	CL2S (of the transmit side; Carrier-ON (of the Receiver side)
DTR	→	DSR
X Special	→	R Special, Ring

Figure 2 SSA/QSA Control Line Loopback

7. PROGRAMMING SEQUENCE

Refer to Appendix A.

7.1 Originating and Receiving a Call

To manually originate a call, the operator depresses the TALK key and dials the desired number. When the call is answered, a carrier tone is heard and the operator should depress the Data key to complete the connection. The operator may now replace the hand-set and depress the AUTO key to return to automatic-receive, following this call. The 801 Auto Call Unit can be used to establish a call without manual attendance. Refer to the 801 Automatic Call Unit Interface Operations and Programming Manual, Publication Number 29-114.

To receive a call, the AUTO key on the auxillary hand set must be depressed. If not, manual answering is required. RING then interrupts the processor, and carrier comes ON.

7.2 Program Initialization

When initializing a call, the program commands the adapter with DTR, RQ2S, and XMIT-NO-OP. Clear-to-Send (CL2S) is set after a certain delay depending upon the modem used. Data transmission can begin after CL2S is active ($\overline{CL2S}$ -> 0).

When receiving a call, the SSA/QSA interrupts should be enabled and the processor is interrupted by RING (dial-up line only). The program can then switch a two-wire line into receive mode and expect synchronization upon reception of SYNC/flag characters.

7.3 Non-ZBID Data Processing

Once a call is established, a character should be transferred, whenever the Busy status bit is reset (an interrupt is generated at this transition). If the program does not perform a data transfer within a character period, the Overflow status bit (OV) is set and a character is lost (Receive), or the last character is retransmitted (Transmit).

To transmit data, RQ2S should be set, and after $\overline{CL2S}$ resets, data should be sent when Busy changes from set to reset. The message must be preceeded by at least two SYNC characters (to ensure synchronization, three to six SYNC characters are usually sent).

To receive data, the SYNC Match Character register must be programmed to the desired SYNC character by a Write Data to the receive side. The line should be set in SYNSCH mode, and synchronization is established when two consecutive SYNC characters are matched. The program should stop the message of any leading SYNC characters, if the automatic SYNC delete option is not installed.

1. On a two-wire line, there is no delay to allow for all the characters to be sent when the line is switched from Write to Read (RQ2S 1 -> 0). In ZBID mode (see the "Zero bit Insertion/Deletion" section), the program should wait at least two character periods (two interrupts) after OV is set before switching the line. In non-ZBID mode, the program should wait at least two character times (two interrupts) after the last useful character.

2. For maximum throughput, for example, when the SSA/QSA is configured under a Selector Channel, all unused lines should be set to their Idle state by reseting DTR. RQ2S should not stay active any longer than necessary.

3. On a two-wire line, any command which causes a Read/Write or a Write/Read trnsition must have interrupts disarmed. Commands which cause a Write/Read transition must specify SYNSCH, for example in a Write followed by a Read sequence, terminate the Write with Disarm (e.g., X'DB' command), and start the Read with Disarm and SYNC SEARCH (e.g., X'D9' command). Following this sequence, a Read Enable may be issued. A similar sequence must be issued to terminate a Read and begin a Write regardless of any time lapse between line turnaround.

WARNING

FAILURE TO OBEY THE ABOVE RULES MAY LEAD TO INTERMITTENT SYSTEM LEVEL FAILURES, WHICH MAY NOT EASILY BE TRACEABLE TO THE SSA OR QSA.

Zero Bit Insertion/Deletion (ZBID)

An optional SSA/QSA feature is available to operate with zero bit insertion/deletion as required by SDLC, HDLC, and ADCCP protocols.

To initialize transmission in the ZBID mode, a Write Data specifying the initial line bit pattern (normally Flags, X'7E') must be issued, followed by a command with RQ2S, DTR, and Reset Data Mode set. As soon as CL2S goes active, the adapter transmits continuously the contents of the transmit holding data register (previously set up by the preceding Write Data) until the first Write Data is issued. The adapter then preceeds the first data byte with one flag character (X'7E'). If data is not supplied to the adapter within one character time after the previous character was written, the adapter terminates the data frame with continuous Flag characters and sets the OV status. The line resumes data transmission at the first Write Data after a command with Bit 15 set. Thus, several frames may be transmitted in a contiguous sequence (with the SSA/QSA generating interframe flag characters) by allowing OV to set after the first frame is sent, then issuing a command with RQ2S and DTR set and with Reset Data Mode not set, and then initiating the following frame with a Write Data.

In certain instances, the Transmit line must generate either a "frame abort" or "idle line" sequence. This is accomplished by issuing a Write Data of X'FF' to the adapter, immediately followed by a command with Reset Data Mode set along with RQ2S and DTR. In the case of "frame abort", the character must be followed by contiguous flags, accomplished by issuing a Write Data cf X'7E' (flag) after the next busy 1 -> 0 transition, and then immediately followed by another command with Reset Data Mode set along with RQ2S and DTR.

The ZBID Mode transmit line shutdown procedure is as follows:

1. After transmission of the last data frame is complete, OV should be allowed to set, thereby ending the frame with a Flag character.

2. As soon as OV status is set, issue a Write Data specifying the initial line pattern (normally Flags, X'7E') followed by a command with RQ2S, DTR, and Reset Data Mode set.

3. Following the next Busy 1 -> 0 transition, the line may be shutdown by issuing a command with interrupts disarmed and RQ2S reset.

   This allows time for the ending FLAG character to be transmitted before shutdown occurs.

To receive, the read logic must be placed in the SYNSCH mode. The SSA/QSA strips off the leading flag characters and interrupts (BSY 1 -> 0) the processor when data is assembled. Data transfer terminates upon detection of a flag character or abort character. The End flag or abort character is returned in the receive buffer so that the program can determine the condition of termination by reading it. The End flag is X'7E' and the abort characters are any combination of six or more contiguous one bits, not equal to a flag. For example, X'3F', X'FC', X'7F', X'FE', or X'FF' is acceptable. If the Automatic Resynchronization** is installed, the End/Abort flag is only available to the processor for one character period, after which time the Parity Error/Term status bit resets. If the End flag was detected, the line waits for data, or if an abort character was detected, the line goes into SYNSCH for flags. If ABORT characters appear in the interframe time fill (normally Flags), the SSA/QSA interrupts with the parity Error/Term status bit reset. Without automatic reSYNC, the line terminates further data transfers with the End/Abort flag in the buffer. It must be restarted with a SYNSCH command. Parity Error/Term is set upon termination but is reset by resynchronization. However, when an ABORT character is received while Automatic Resynchronization is installed, a second interrupt occurs one character following the initial ABORT character. This second character interrupt permits software to differentiate between an ABORT sequence (7 to 14 contiguous one bits) and an IDLE LINE sequence (15 or more contiguous one bits). As long as ABORT characters continue being received, these interrupts continue at one character intervals. If, during a period of continuous ABORT interrupts, a FLAG is received, the SSA/QSA RESYNC's on this FLAG and schedules an interrupt. The continuous ABORT interrupts may be stopped by a SYNSCH command.

**Automatic Resynchronization allows reception of more than one data frame without re-initialization of the line.

High Speed Data Transfer (SELCH)

When configured under a Selector Channel, the SSA/QSA can be operated with only one line active at a time. This is true if the line is configured in Four-wire or Two-wire mode.

To transmit in this configuration, the SELCH should be provided with the starting and ending buffer address. An Output Command should be issued to the line with RQ2S set, wait until $\overline{\text{CL2S}} \rightarrow 0$, and followed by the command for SELCH GO. The SELCH interrupts the processor when the the data transfer is completed.

To receive data, the SELCH should be set up with buffer addresses. When carrier is detected on the receive side (a two-wire line should be switched into the Read mode), an Output Command should place the line into the SYNSCH mode, followed by a SELCH GO, READ command. A SELCH interrupt is received upon completion of the data transfer. In ZBID mode, an End or Abort flag also causes the SELCH interrupt.

8. INTERRUPTS

The following are interrupt conditions for both four-wire and two-wire operations when enabled by Output Command and PSW Enable bits:

RECEIVE STATUS	TRANSITION
Busy	1 -> 0
Carrier OFF	1 -> 0,0 -> 1
Ring	0 -> 1
$\overline{\text{DSR}}$	0 -> 1

TRANSMIT STATUS	TRANSITION
Busy	1 -> 0
$\overline{\text{CL2S}}$	0 -> 1
Ring	0 -> 1
$\overline{\text{DSR}}$	0 -> 1

9. INITIALIZATION

The SSA/QSA is initialized by Power Fail of System Initialization, to the DISARM, Receive mode (Read) with RQ2S reset, Ready (DTR) reset , special reset, and Local Loopback reset. The status of the number of data bits code and the line mode control code after initialization are undefined.

10. DEVICE ADDRESS

The preferred device address range for the SSA/QSA is X'B8' through X'BF'. A QSA with all four lines strapped for four-wire, has eight consecutive device addresses, one for receive and one for transmit for each of the four lines. The SSA wired in the same configuration has two consecutive addresses. The even addresses are assigned to the Receive lines and the odd addresses to the Transmit lines. A line (without two-wire interleaving), strapped for two-wire operation, responds to the Receive line address, and all interrupts from a two-wire line returns the Receive line address only. Normally, the least significant address bit is not checked in the two-wire mode, but if the two-wire Address Interleaving option is installed, the least significant address bit is checked. This option may be used when the system has one or more pairs of QSA's, each having all four lines strapped for two-wire operation. The QSA address switch inverts the least-significant address bit before it is checked, thus allowing multiple QSA's to be configured with interleaved device addresses.

Table 3 gives an example of two-wire Address Interleaving. In this example, the system has two QSA's each with two-wire Address Interleaving installed. The Address switch of QSA 1 is set at X'B8' and that of QSA 2 is set at X'B9'.

TABLE 3 EXAMPLES OF TWO-WIRE QSA ADDRESS INTERLEAVING

QSA 1 Responds	QSA 2 Responds	Result
B8	No match	QSA 1, Line 0
No match	B9	QSA 2, Line 0
BA	No match	QSA 1, Line 1
No match	BB	QSA 2, Line 1
BC	No match	QSA 1, Line 2
No match	BD	QSA 2, Line 2
BE	No match	QSA 1, Line 3
No match	BF	QSA 2, Line 3

NOTE

When the SSA/QSA is on a 16-bit processor, Bits 6 and 7 of the device address field must be set to ZERO. Bits 0-5 are don't care bits.

11. SAMPLE PROGRAMS

Sample Programs for driving the SSA/QSA in both Full and Half Duplex modes are shown in Appendices B and C. The examples shown are as follows:

16 Bit Processor Programming Examples (Appendix B)

1. 2 wire - Non-ZBID
2. 2 lines - 4 wire-Non ZBID
3. 2 wire - ZBID
4. 2 lines - 4 wire-ZBID

32 Bit Processor Programming Examples (Appendix C)

1. 2 wire - Non ZBID
2. 2 lines - 4 wire-Non ZBID
3. 2 wire - ZBID
4. 2 lines - 4 wire-ZBID

APPENDIX A
PROGRAMMER'S GUIDE

RECEIVE	DISABLE ENABLE DISARM		LOCAL LOOP BACK	SYNSCH	READY (DTR)	SPECIAL	WRITE/ RQ2S	1
TRANSMIT	DISABLE ENABLE DISARM			RESET DATA MODE				
COMMON			NO. OF DATA BITS SELECT			LINE MODE CONTROL		0

Figure A-1. Command Byte

RECEIVE	OV	PRTY/ TRMN	SPECIAL	RING	BUSY	EXAMINE	CARRIER OFF	$\overline{DSR}$
TRANSMIT	OV				BUSY	EXAMINE	$\overline{CL2S}$	

Figure A-2. Status Byte

RECEIVE	TRANSMIT	TRANSITION
BUSY	BUSY	1 → 0
CARRIER-OFF	$\overline{CL2S}$	1 → 0(R), 0 → 1
RING	RING	0 → 1
$\overline{DSR}$	$\overline{DSR}$	0 → 1

R - RECEIVE SIDE ONLY

Figure A-3. Interrupts

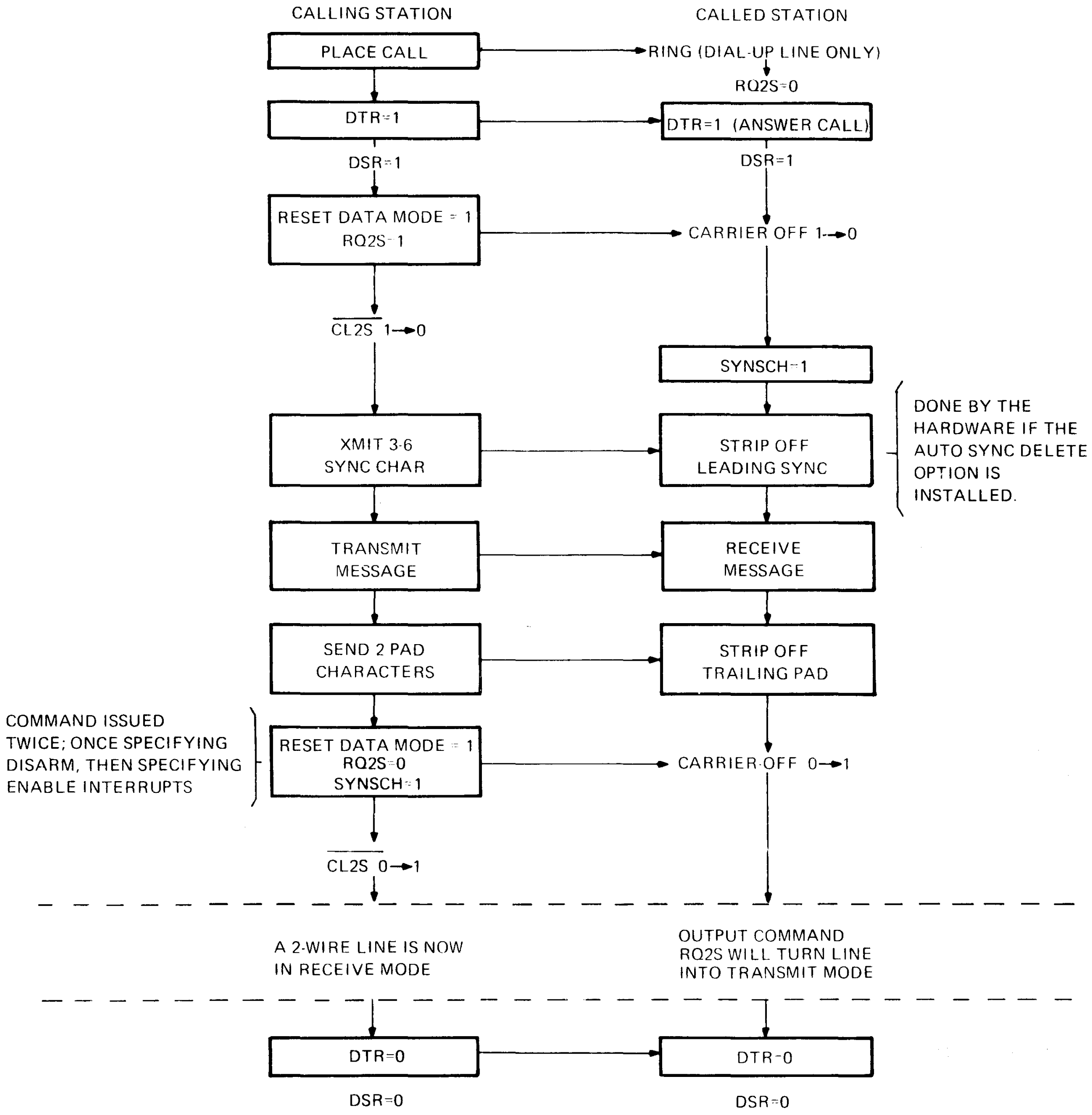


Figure A-4. Two-Wire Non ZBID Programming Sequence

NOTE

All commands causing line turnaround are issued twice, first specifying Disarm, then specifying Enable Interrupts.

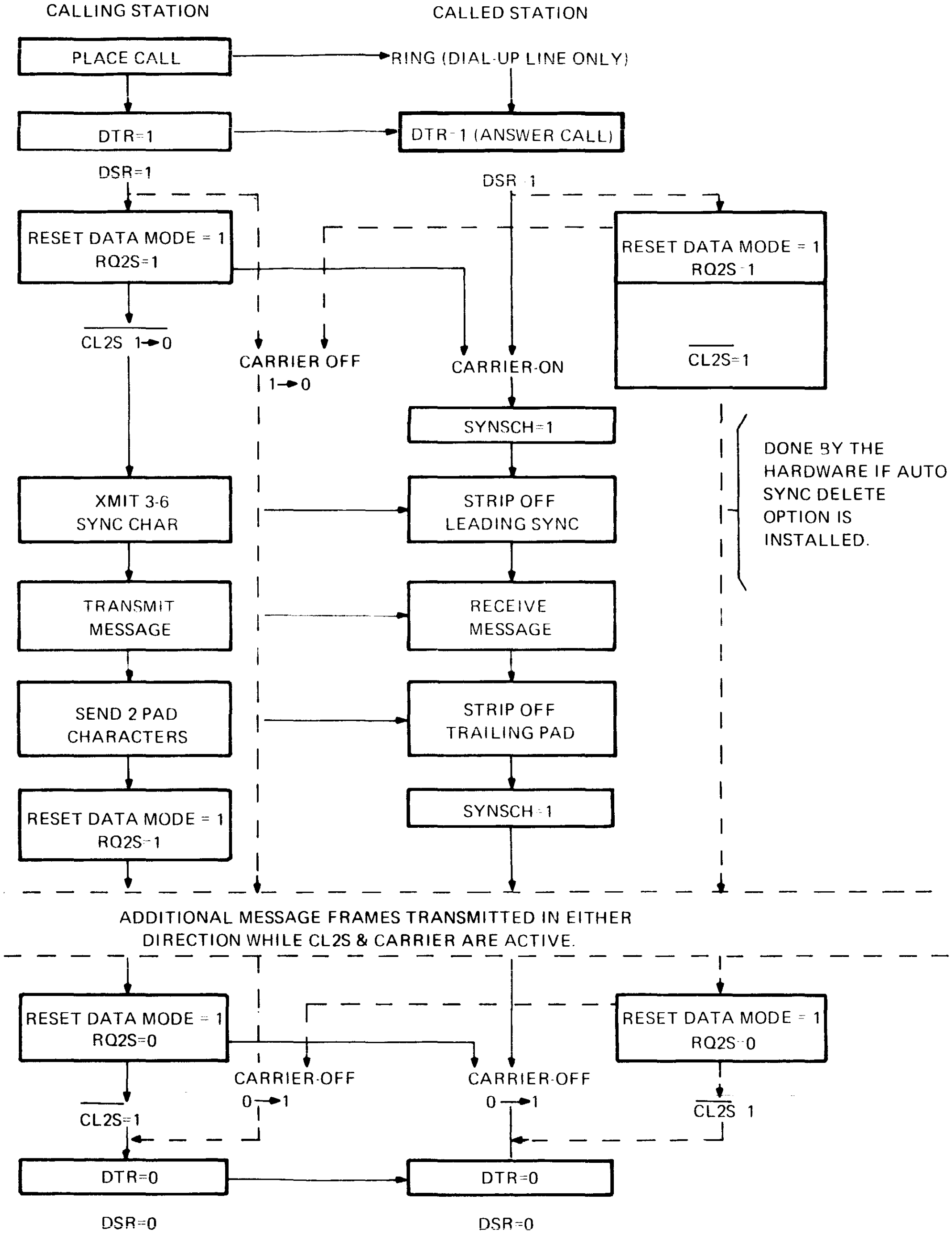


Figure A-5. Four-Wire Non-ZBID Programming Sequence

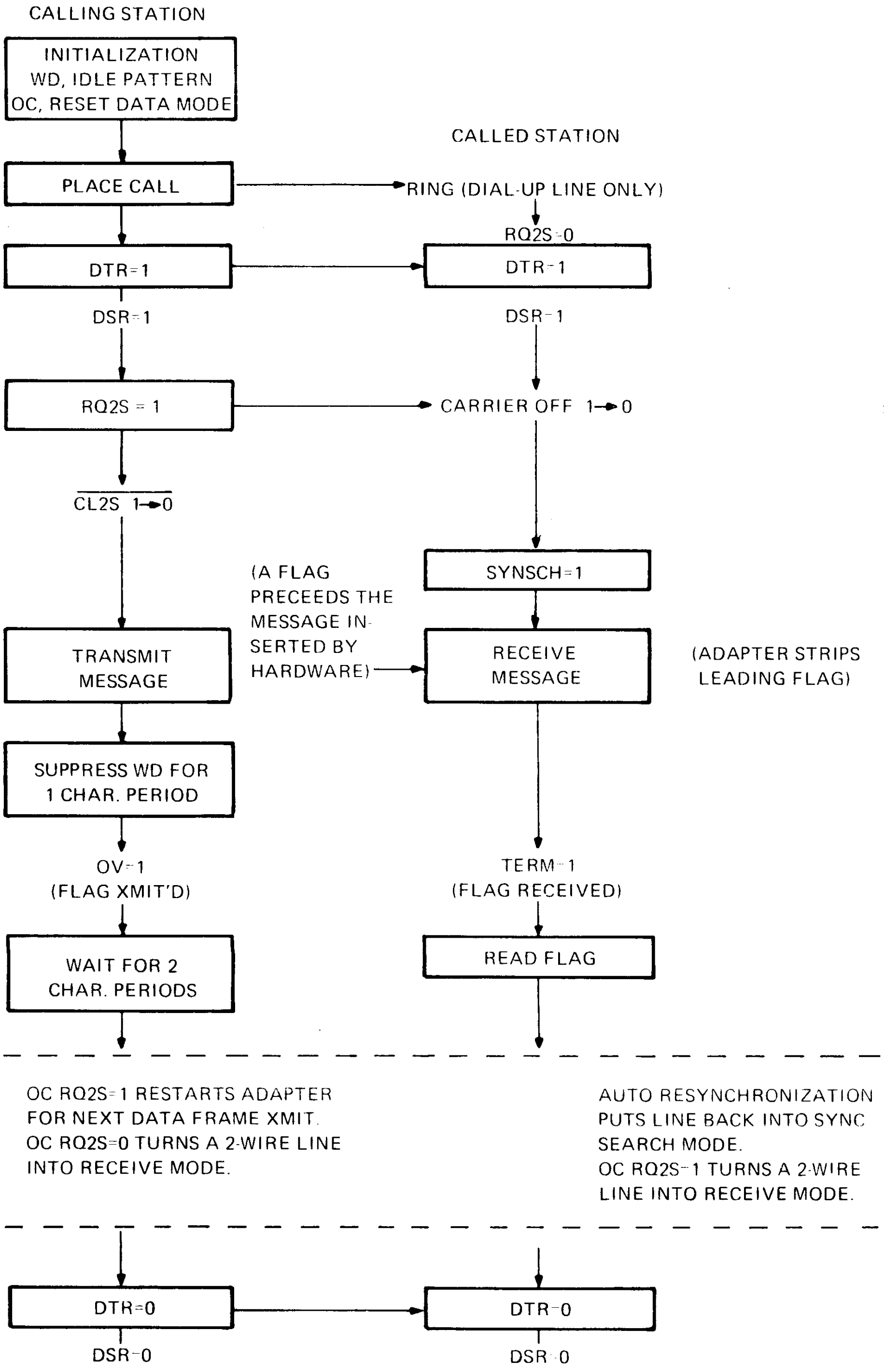
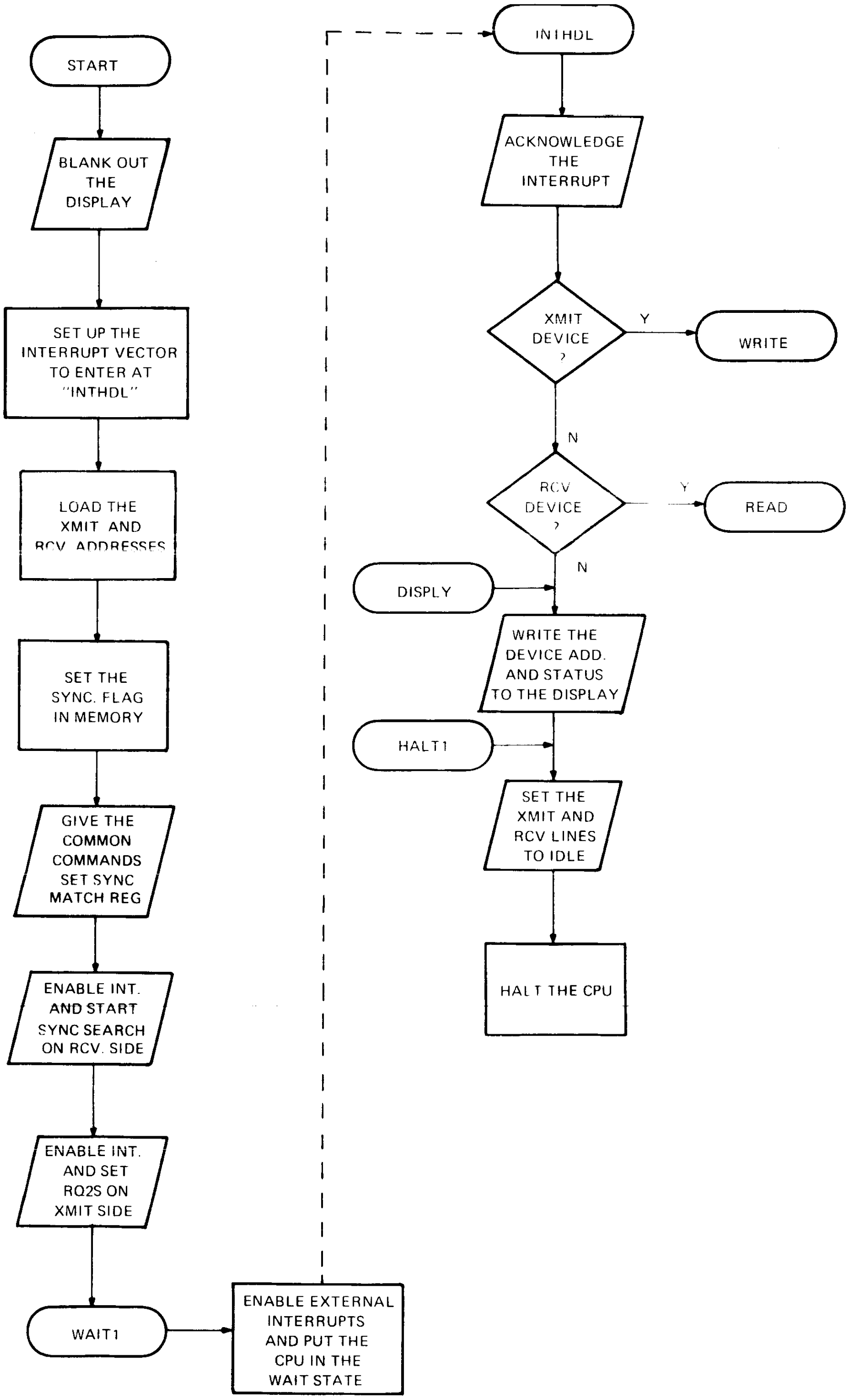


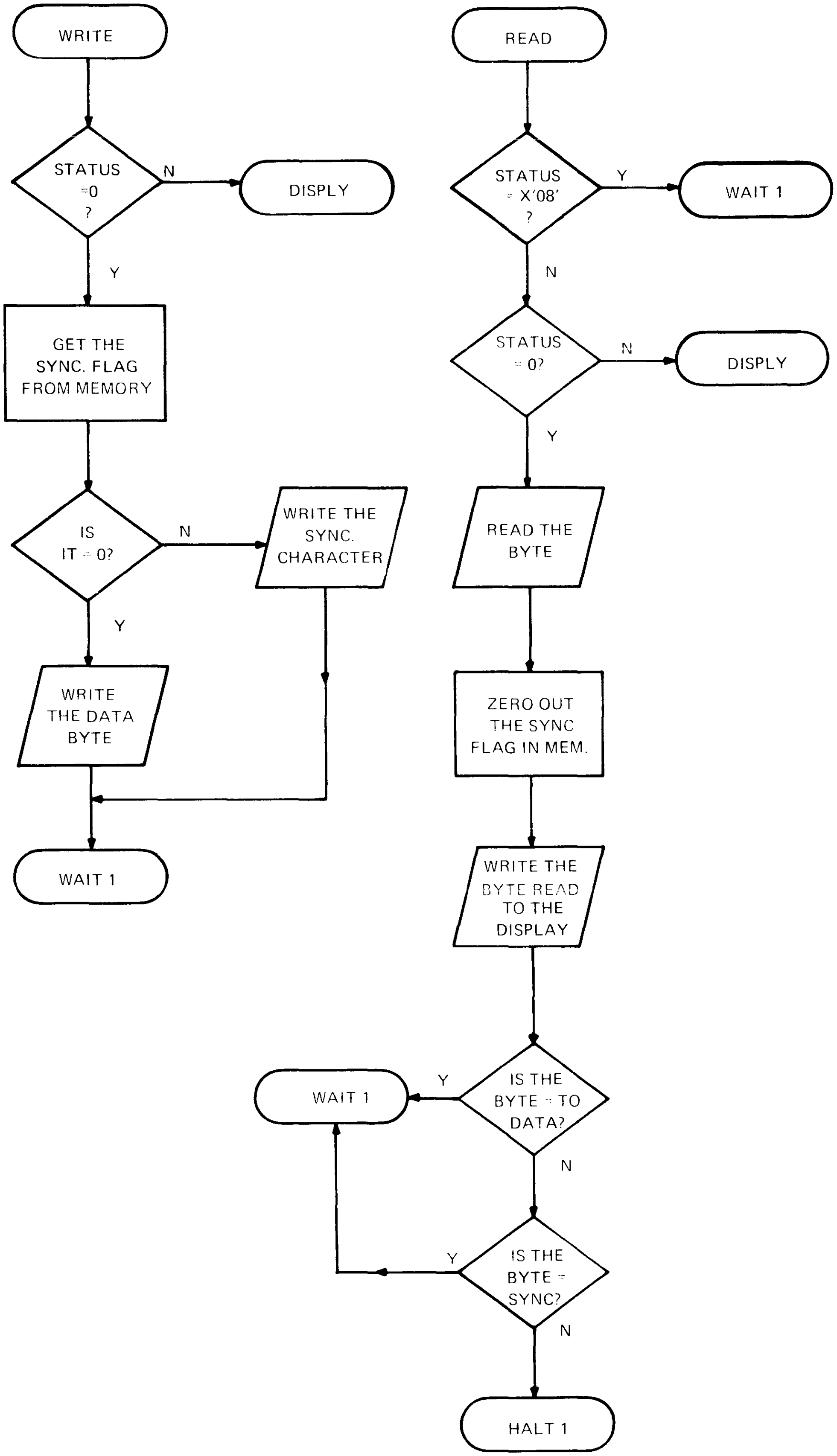
Figure A-6. ZBID Programming Sequence

APPENDIX B
16 BIT PROGRAMMING EXAMPLES


INTHDL
START
BLANK OUT THE DISPLAY
ACKNOWLEDGE THE INTERRUPT
XMIT DEVICE ?
Y
WRITE
SET UP THE INTERRUPT VECTOR TO ENTER AT "INTHDL"
N
LOAD THE XMIT AND RCV ADDRESSES
RCV DEVICE ?
Y
READ
N
DISPLY
WRITE THE DEVICE ADD. AND STATUS TO THE DISPLAY
SET THE SYNC. FLAG IN MEMORY
HALT1
GIVE THE COMMON COMMANDS SET SYNC MATCH REG
SET THE XMIT AND RCV LINES TO IDLE
HALT THE CPU
ENABLE INT. AND START SYNC SEARCH ON RCV. SIDE
ENABLE INT. AND SET RQ2S ON XMIT SIDE
WAIT1
ENABLE EXTERNAL INTERRUPTS AND PUT THE CPU IN THE WAIT STATE

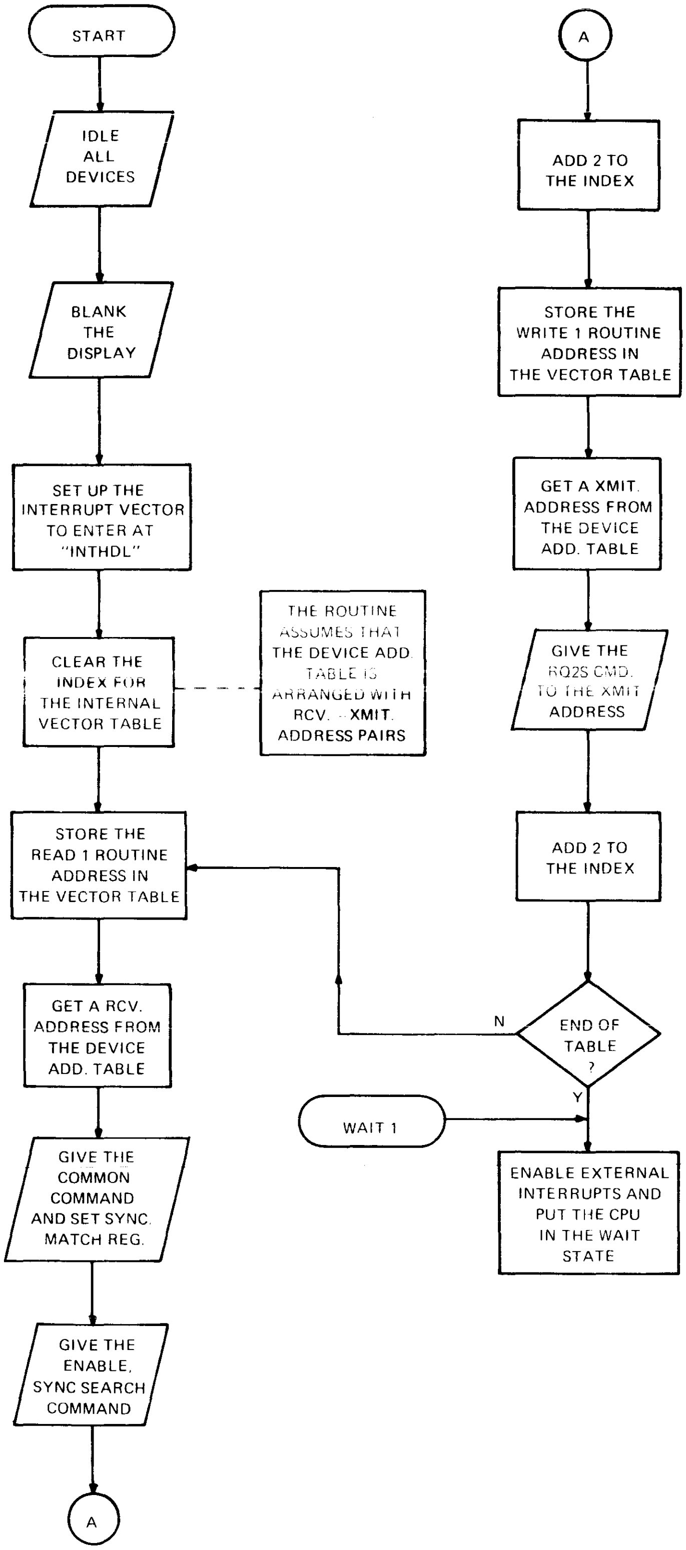
2-Wire non-ZBID

APPENDIX B (Continued)



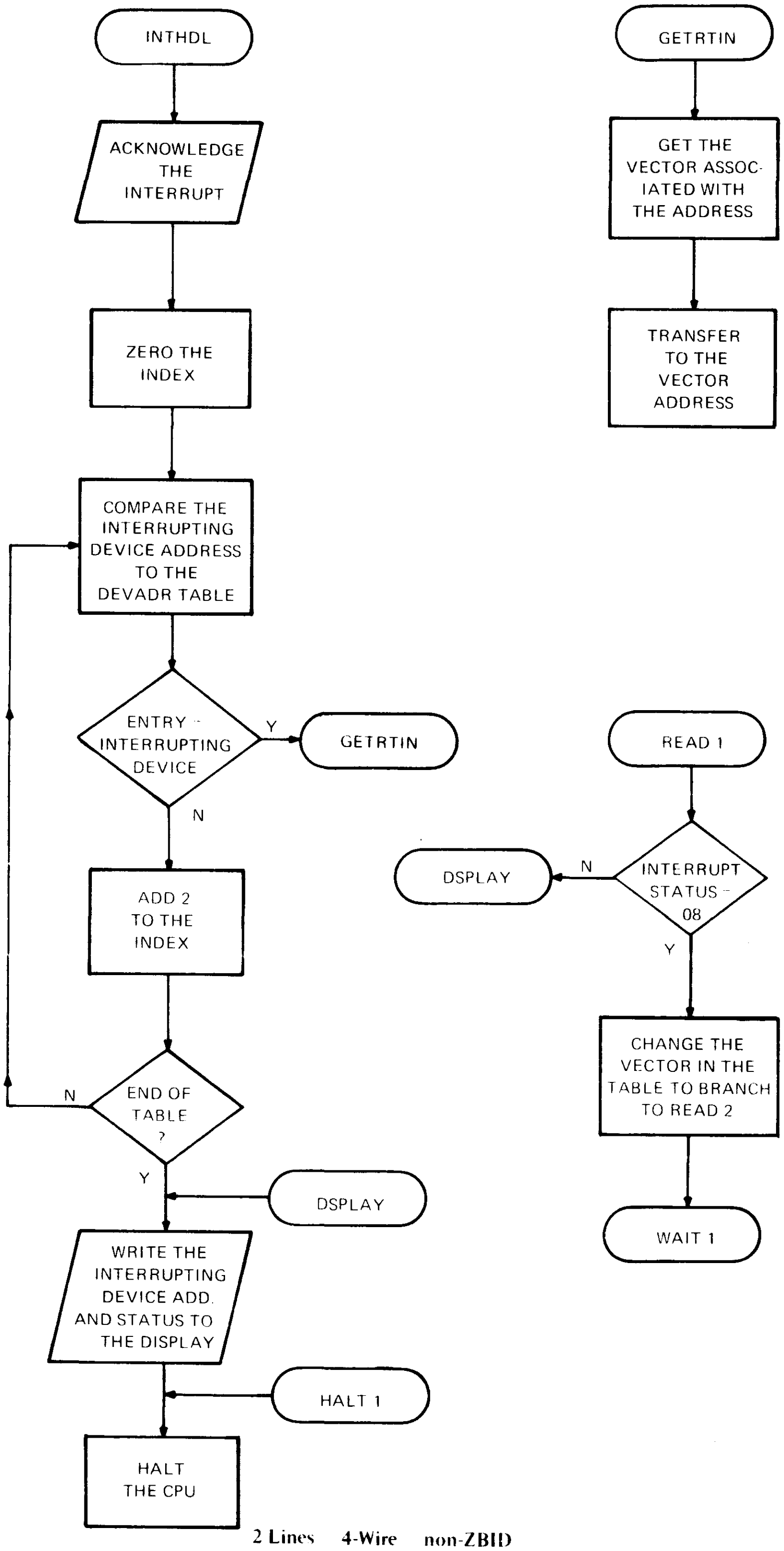
2-Wire non-ZBID

APPENDIX B (Continued)

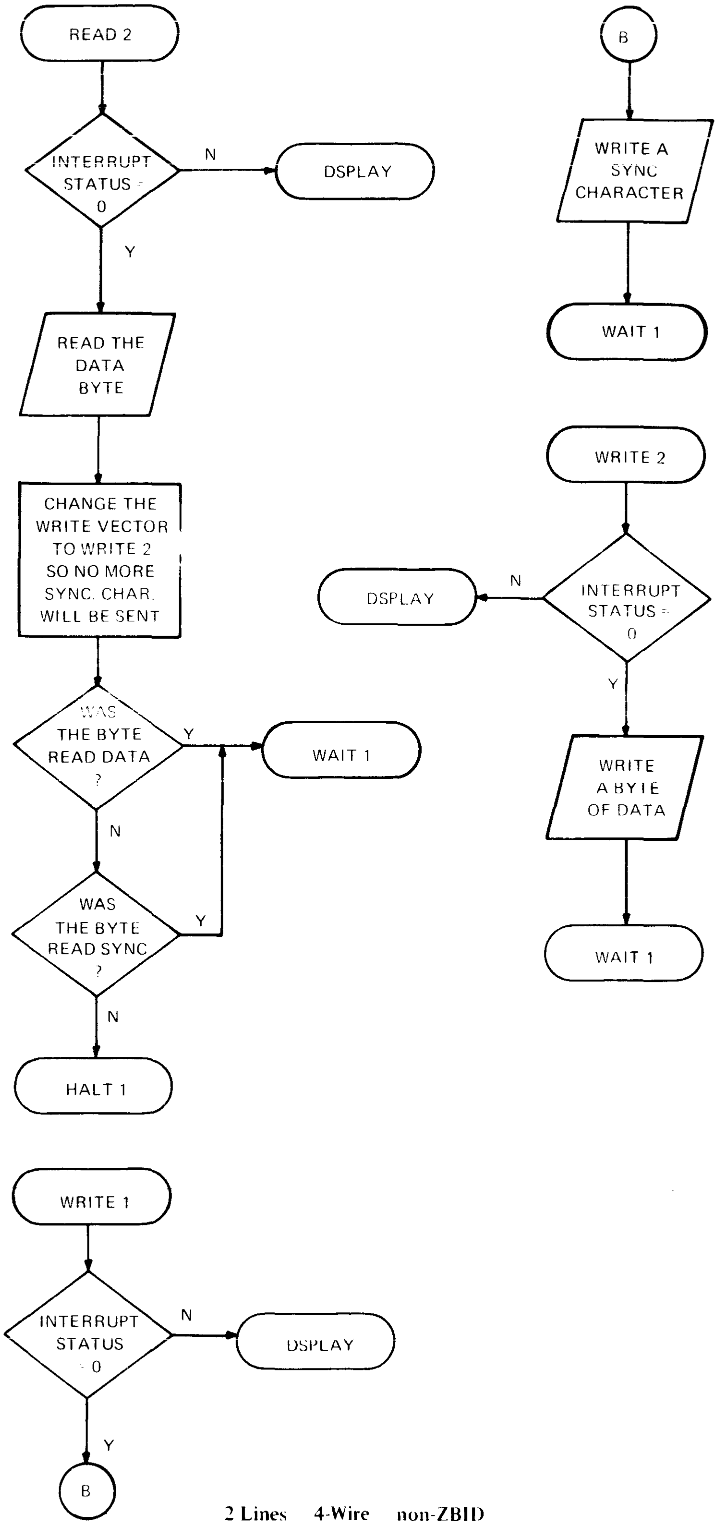

START
IDLE ALL DEVICES
BLANK THE DISPLAY
SET UP THE INTERRUPT VECTOR TO ENTER AT "INTHDL"
CLEAR THE INDEX FOR THE INTERNAL VECTOR TABLE
THE ROUTINE ASSUMES THAT THE DEVICE ADD. TABLE IS ARRANGED WITH RCV. - XMIT. ADDRESS PAIRS
STORE THE READ 1 ROUTINE ADDRESS IN THE VECTOR TABLE
GET A RCV. ADDRESS FROM THE DEVICE ADD. TABLE
GIVE THE COMMON COMMAND AND SET SYNC. MATCH REG.
GIVE THE ENABLE, SYNC SEARCH COMMAND
A
A
ADD 2 TO THE INDEX
STORE THE WRITE 1 ROUTINE ADDRESS IN THE VECTOR TABLE
GET A XMIT. ADDRESS FROM THE DEVICE ADD. TABLE
GIVE THE RO2S CMD. TO THE XMIT ADDRESS
ADD 2 TO THE INDEX
END OF TABLE ?
N
Y
WAIT 1
ENABLE EXTERNAL INTERRUPTS AND PUT THE CPU IN THE WAIT STATE

2 Lines 4-Wire - non-ZBID

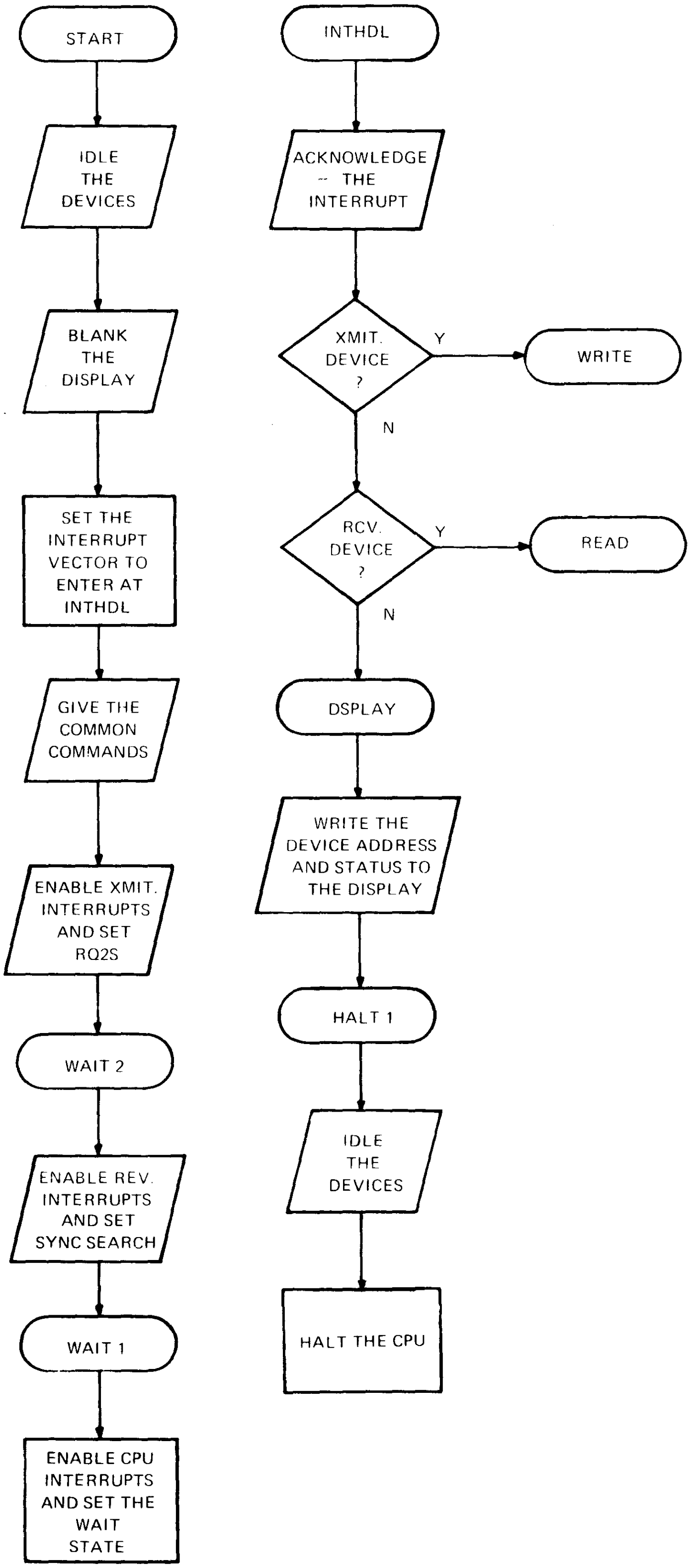
APPENDIX B (Continued)


INTHDL
ACKNOWLEDGE THE INTERRUPT
ZERO THE INDEX
COMPARE THE INTERRUPTING DEVICE ADDRESS TO THE DEVADR TABLE
ENTRY - INTERRUPTING DEVICE
Y
GETRTIN
N
ADD 2 TO THE INDEX
END OF TABLE ?
N
Y
DSPLAY
WRITE THE INTERRUPTING DEVICE ADD. AND STATUS TO THE DISPLAY
HALT 1
HALT THE CPU
GETRTIN
GET THE VECTOR ASSOC-IATED WITH THE ADDRESS
TRANSFER TO THE VECTOR ADDRESS
READ 1
INTERRUPT STATUS - 08
N
DSPLAY
Y
CHANGE THE VECTOR IN THE TABLE TO BRANCH TO READ 2
WAIT 1

2 Lines 4-Wire non-ZBID

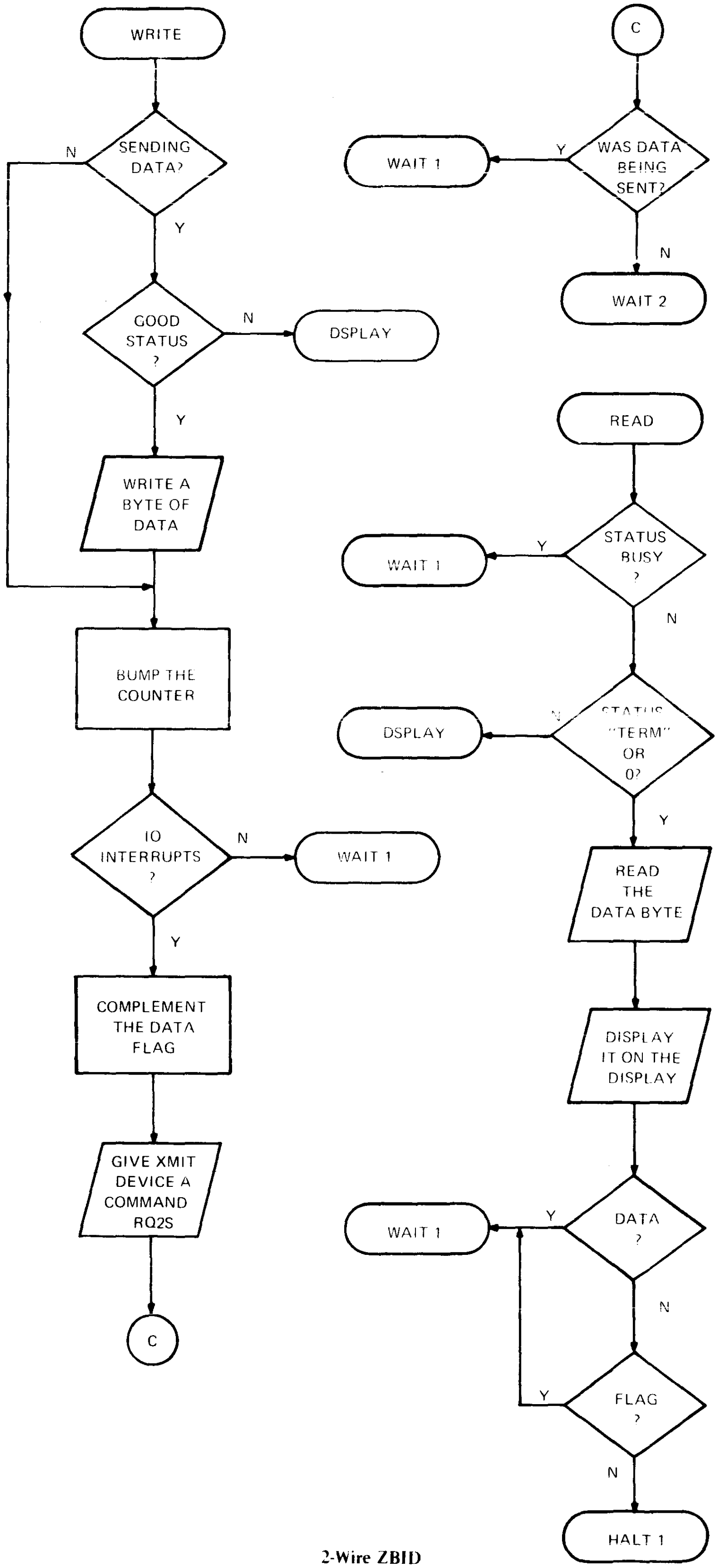


2 Lines 4-Wire non-ZBID

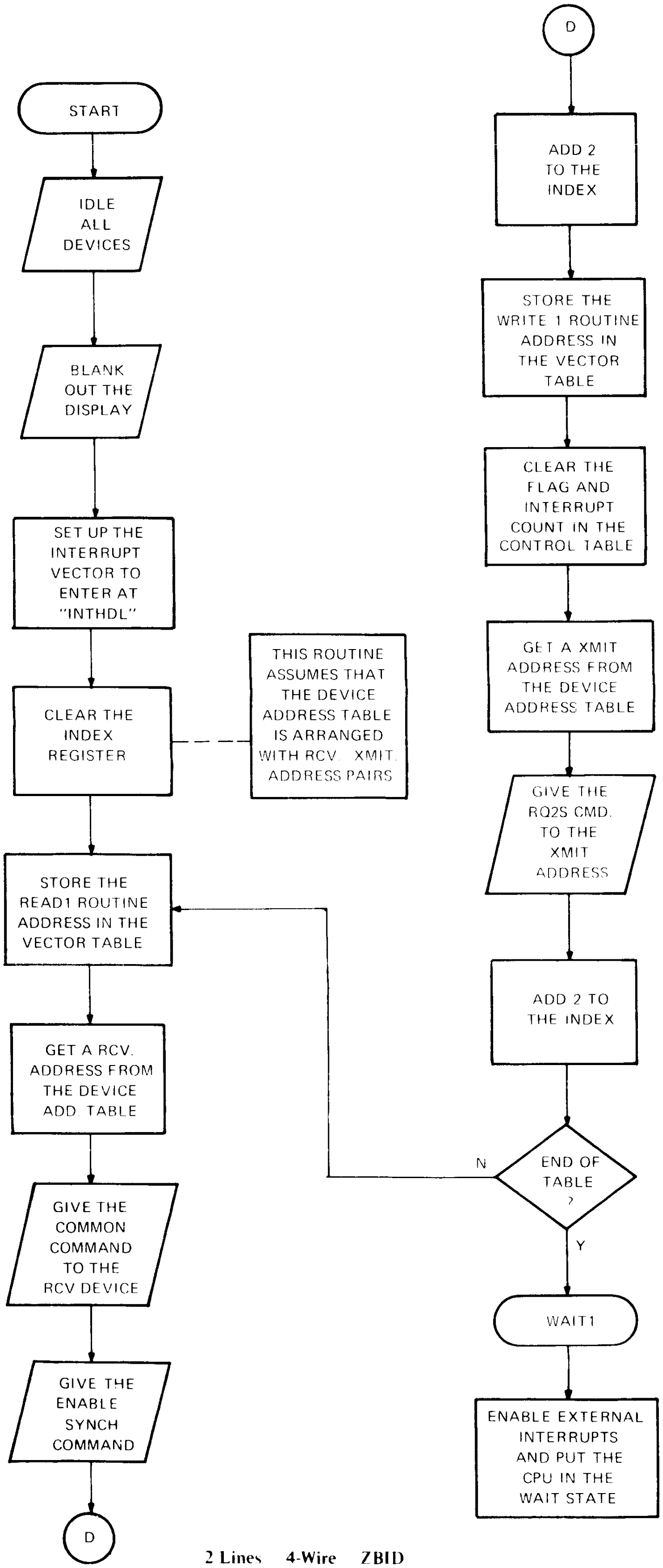

START
IDLE THE DEVICES
BLANK THE DISPLAY
SET THE INTERRUPT VECTOR TO ENTER AT INTHDL
GIVE THE COMMON COMMANDS
ENABLE XMIT. INTERRUPTS AND SET RQ2S
WAIT 2
ENABLE REV. INTERRUPTS AND SET SYNC SEARCH
WAIT 1
ENABLE CPU INTERRUPTS AND SET THE WAIT STATE
INTHDL
ACKNOWLEDGE -- THE INTERRUPT
XMIT. DEVICE ?
Y
WRITE
N
RCV. DEVICE ?
Y
READ
N
DSPLAY
WRITE THE DEVICE ADDRESS AND STATUS TO THE DISPLAY
HALT 1
IDLE THE DEVICES
HALT THE CPU

2-Wire ZBID

APPENDIX B (Continued)

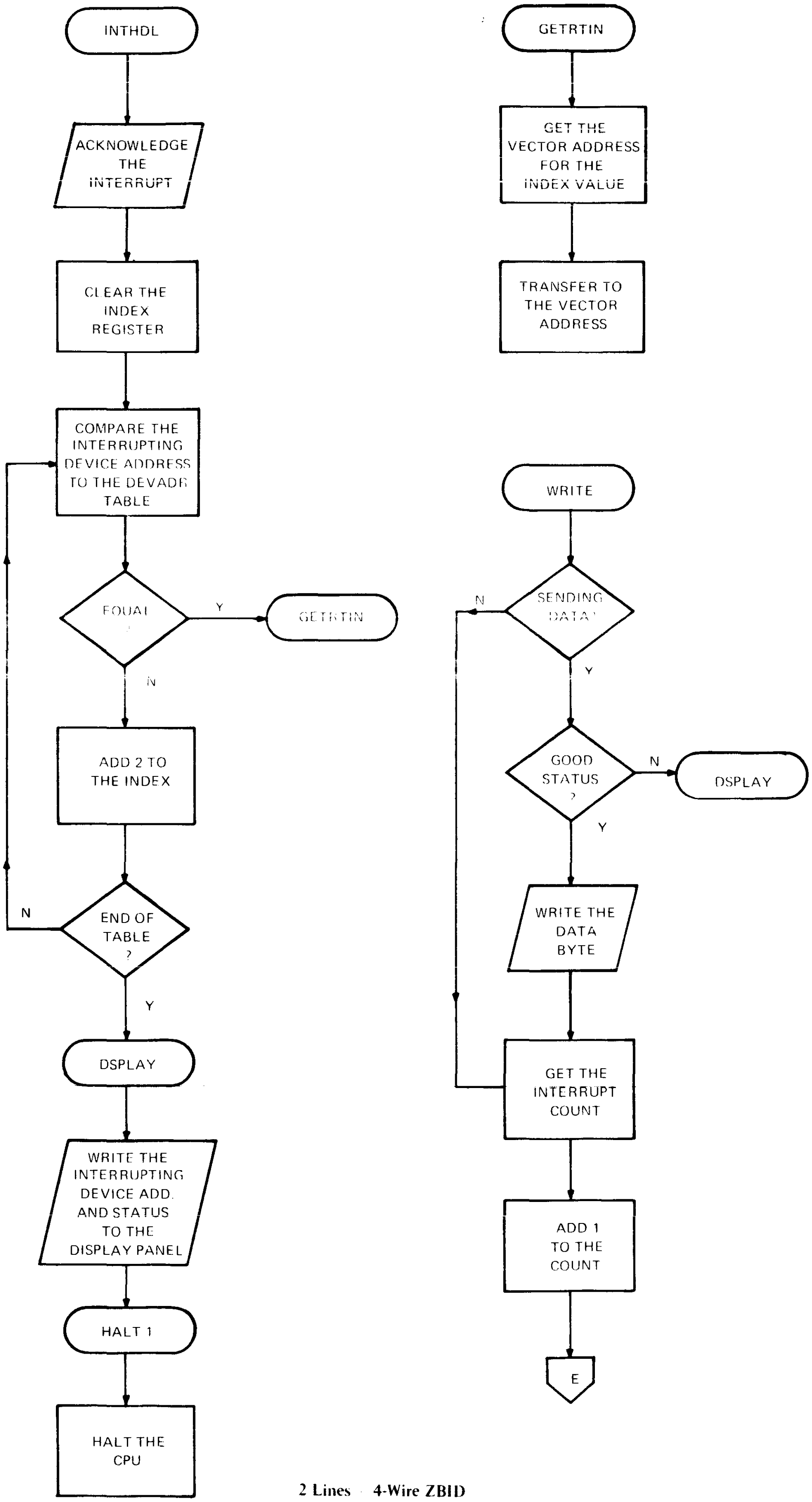


2-Wire ZBID


D
START
IDLE ALL DEVICES
BLANK OUT THE DISPLAY
SET UP THE INTERRUPT VECTOR TO ENTER AT "INTHDL"
CLEAR THE INDEX REGISTER
THIS ROUTINE ASSUMES THAT THE DEVICE ADDRESS TABLE IS ARRANGED WITH RCV. XMIT. ADDRESS PAIRS
STORE THE READ1 ROUTINE ADDRESS IN THE VECTOR TABLE
GET A RCV. ADDRESS FROM THE DEVICE ADD. TABLE
GIVE THE COMMON COMMAND TO THE RCV DEVICE
GIVE THE ENABLE SYNCH COMMAND
D
ADD 2 TO THE INDEX
STORE THE WRITE 1 ROUTINE ADDRESS IN THE VECTOR TABLE
CLEAR THE FLAG AND INTERRUPT COUNT IN THE CONTROL TABLE
GET A XMIT ADDRESS FROM THE DEVICE ADDRESS TABLE
GIVE THE RQ2S CMD. TO THE XMIT ADDRESS
ADD 2 TO THE INDEX
END OF TABLE ?
N
Y
WAIT1
ENABLE EXTERNAL INTERRUPTS AND PUT THE CPU IN THE WAIT STATE

2 Lines 4-Wire ZBID

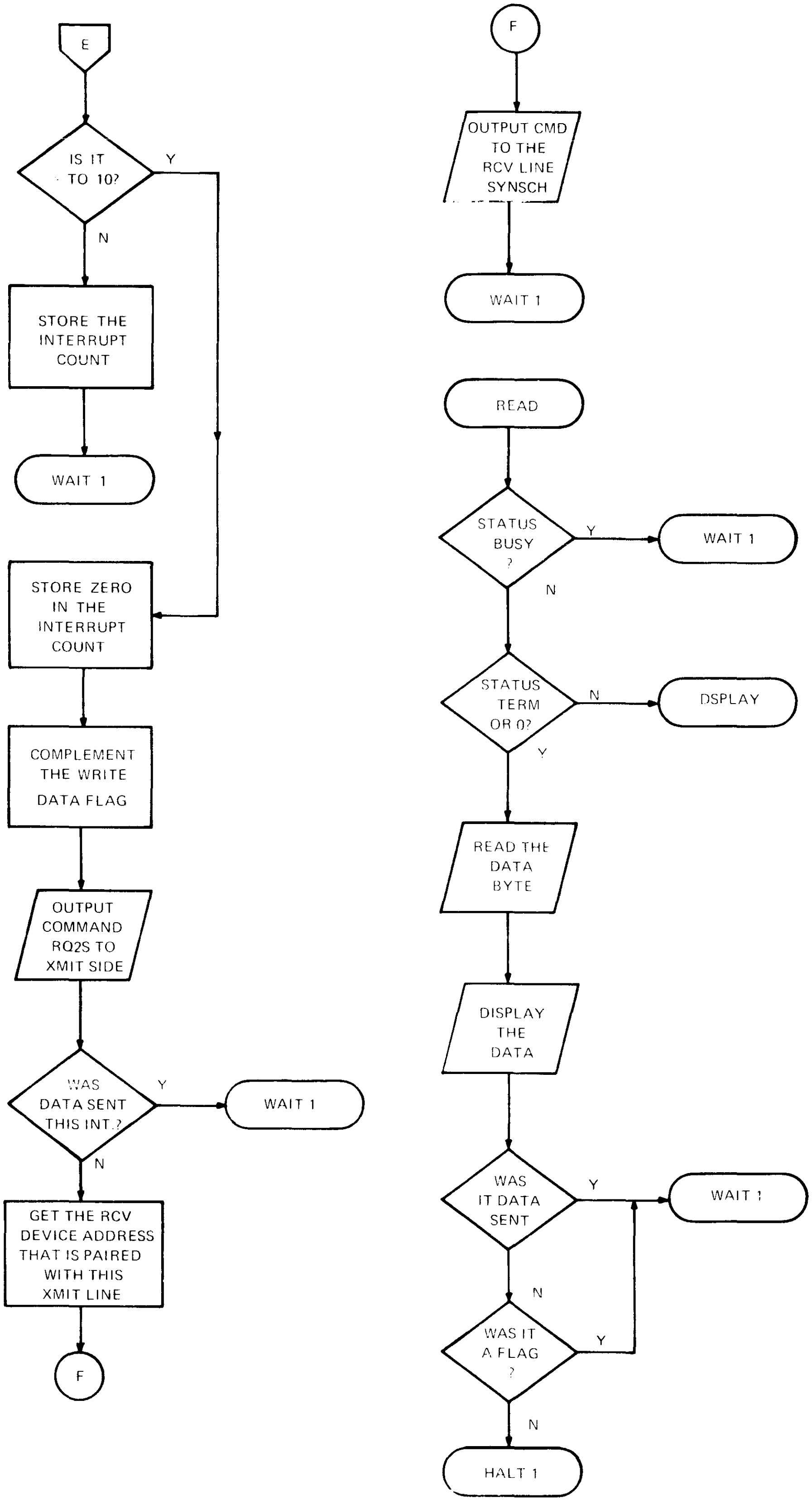
APPENDIX B (Continued)



INTHDL
ACKNOWLEDGE THE INTERRUPT
CLEAR THE INDEX REGISTER
COMPARE THE INTERRUPTING DEVICE ADDRESS TO THE DEVADR TABLE
EQUAL ?
Y
GETRTIN
N
ADD 2 TO THE INDEX
N
END OF TABLE ?
Y
DSPLAY
WRITE THE INTERRUPTING DEVICE ADD. AND STATUS TO THE DISPLAY PANEL
HALT 1
HALT THE CPU
GETRTIN
GET THE VECTOR ADDRESS FOR THE INDEX VALUE
TRANSFER TO THE VECTOR ADDRESS
WRITE
N
SENDING DATA ?
Y
GOOD STATUS ?
N
DSPLAY
Y
WRITE THE DATA BYTE
GET THE INTERRUPT COUNT
ADD 1 TO THE COUNT
E

2 Lines 4-Wire ZBID

APPENDIX B (Continued)


E
IS IT TO 10?
Y
N
STORE THE INTERRUPT COUNT
WAIT 1
STORE ZERO IN THE INTERRUPT COUNT
COMPLEMENT THE WRITE DATA FLAG
OUTPUT COMMAND RQ2S TO XMIT SIDE
WAS DATA SENT THIS INT.?
Y
WAIT 1
N
GET THE RCV DEVICE ADDRESS THAT IS PAIRED WITH THIS XMIT LINE
F
F
OUTPUT CMD TO THE RCV LINE SYNSCH
WAIT 1
READ
STATUS BUSY ?
Y
WAIT 1
N
STATUS TERM OR 0?
N
DSPLAY
Y
READ THE DATA BYTE
DISPLAY THE DATA
WAS IT DATA SENT
Y
WAIT 1
N
WAS IT A FLAG ?
Y
N
HALT 1

2 Lines 4-Wire ZBID

```
          QSA - 2-WIRE PROGRAMMING EXAMPLE - 16 BIT PROCESSOR        PAGE   1  12:53:08  07/19/78
    PROG= *NONE*     ASSEMBLED BY CAL 03-066R05-00   (32-BIT)
                                      1            SCRAT
                                      2            CROSS
                                      3            WIDTH 120
                                      4            TARGT 16
```

```
                        *
                        * *************************************************************
                        * THIS PROGRAM ILLUSTRATES A DATA TRANSFER BETWEEN TWO LINES ON  *
                        * THE QSA UNDER LOCAL LOOPBACK.
                        * A DATA BYTE IS TRANSMITTED CONTINUOUSLY, AND RECEIVED AND      *
                        * COMPARED ON THE RECEIVE END. THE BYTE IS DISPLAYED ON THE      *
                        * DISPLAY PANEL. IF A STATUS OR DEVICE NUMBER ERROR IS DETECTED  *
                        * THE PROCESSOR HALTS WITH THE DEVICE NUMBER AND THE STATUS      *
                        * DISPLAYED ON THE PANEL. IF THE ERROR OCCURS ON THE DATA, ONLY  *
                        * THE DATA IS DISPLAYED. PROGRAM IS EXECUTED UNDER INTERRUPT.    *
                        * *************************************************************
                        *                                                                *
        0000 0001       DISP      EQU   1                  DISPLAY ADDRESS REGISTER
        0000 0002       WORK      EQU   2                  WORK REGISTER
        0000 0003       RDEV      EQU   3                  RECV DEVICE ADDRESS REG.
        0000 0004       XDEV      EQU   4                  XMIT DEVICE ADDRESS REG.
        0000 0005       IDEV      EQU   5                  INTERRUPTED DEVICE ADDRESS REG.
        0000 0006       ISTA      EQU   6                  INTERRUPTED DEVICE STATUS REG.
        0000 0008       BUSY      EQU   8                  BUSY STATUS
0000R   4830 00CER                LH    RDEV,RADR          GET RECV DEVICE ADDRESS
0004R   DE30 00D6R                OC    RDEV,IDLE          IDLE BOTH DEVICES AND CLEAR
0008R   4840 00D0R                LH    XDEV,XADR          GET XMIT DEVICE ADDRESS
000CR   DE40 00D6R                OC    XDEV,IDLE          ANY QUEUED INTERRUPTS
0010R   2411                      LIS   DISP,1             DISPLAY PANEL ADDRESS
0012R   0722                      XHR   WORK,WORK
0014R   DE10 00D2R                OC    DISP,INCR          SET DISPLAY TO INCREMENT MODE
0018R   9A12                      WDR   DISP,WORK          CLEAR DISPLAY
001AR   9A12                      WDR   DISP,WORK
001CR   9A12                      WDR   DISP,WORK
001ER   9A12                      WDR   DISP,WORK
0020R   DE10 00D3R                OC    DISP,NORM          SET DISPLAY BACK TO NORMAL MODE
0024R   4020 0040                 STH   WORK,X'40'         SET UP LOW CORE
0028R   4020 0042                 STH   WORK,X'42'
002CR   4020 0044                 STH   WORK,X'44'
0030R   C820 0058R                LHI   WORK,INTHDL        SET UP INTERRUPT HANDLER ADDRESS
0034R   4020 0046                 STH   WORK,X'46'
0038R   C820 00FF                 LHI   WORK,X'FF'         SET SYNC FLAG
003CR   4020 00CCR                STH   WORK,SYNFLG
0040R   DE30 00D7R                OC    RDEV,COMMON        SET LINE MODE
0044R   DE40 00D7R                OC    XDEV,COMMON
                        *
                        *  NOTE :
                        *
                        *     IF THE NEXT TWO OUTPUT COMMANDS ARE BEING ISSUED TO
                        *     TURN A 2-WIRE LINE AROUND (FROM WRITE TO READ OR READ TO
                        *     WRITE) THE COMMANDS MUST BE ISSUED TWICE.  FIRST WITH
                        *     INTERRUPTS DISARMED, AND THEN RE-ISSUED WITH THE DESIRED
                        *     INTERRUPT CONDITION, UNLESS THE DISARM CONDITION IS DESIRED.
                        *
0048R   DE30 00D4R                OC    RDEV,ENBSCH        RECV ENABLE, SYNCSH
004CR   DA30 00D8R                WD    RDEV,SYNC          SET SYNC MATCH REGISTER
0050R   DE40 00D5R                OC    XDEV,ENBRQS        XMIT ENABLE, RQ2S
0054R   C200 00C4R      WAIT1     LPSW  WAIT
```

```
     QSA - 2-WIRE PROGRAMMING EXAMPLE - 16 BIT PROCESSOR       PAGE   3  12:53:08  07/19/78

     NON ZBID

                                  59  *
   0058R  9F56                    60  INTHDL   AIR    IDEV,ISTA            ACKNOWLEDGE INTERRUPT
   005AR  0554                    61           CLHR   IDEV,XDEV            WRITE?
   005CR  4330 007AR              62           BE     WRITE                YES - GO TO WRITE ROUTINE
   0060R  0553                    63           CLHR   IDEV,RDEV            READ?
   0062R  4330 0098R              64           BE     READ                 YES - GO TO READ ROUTINE
   0066R  DE10 00D2R              65  DISPLY   OC     DISP,INCR            NEITHER -
   006AR  9A16                    66           WDR    DISP,ISTA            DISPLAY STATUS
   006CR  9A15                    67           WDR    DISP,IDEV            DISPLAY DEVICE ADDRESS
   006ER  DE30 00D6R              68  HALT1    OC     RDEV,IDLE            IDLE LINES
   0072R  DE40 00D6R              69           OC     XDEV,IDLE
   0076R  C200 00C8R              70           LPSW   HALT                 HALT PROCESSOR
                                  71  *
   007AR  0866                    72  WRITE    LHR    ISTA,ISTA            WRITE ROUTINE
   007CR  4230 0066R              73           BNZ    DISPLY               EXIT IF STATUS NOT ZERO
   0080R  4820 00CCR              74           LH     WORK,SYNFLG          SYNC FLAG SET?
   0084R  4330 0090R              75           BZ     WRTDAT               NO - SEND DATA
   0088R  DA40 00D8R              76           WD     XDEV,SYNC            YES - SEND SYNC
   008CR  4300 0054R              77           B      WAIT1
   0090R  DA40 00D9R              78  WRTDAT   WD     XDEV,DATA            SEND DATA BYTE
   0094R  4300 0054R              79           B      WAIT1
                                  80  *
   0098R  C560 0008               81  READ     CLHI   ISTA,BUSY            READ ROUTINE
   009CR  4330 0054R              82           BE     WAIT1                WAIT IF CARRIER INTERRUPT
   00A0R  0866                    83           LHR    ISTA,ISTA
   00A2R  4230 0066R              84           BNZ    DISPLY               EXIT IF STATUS NOT ZERO
   00A6R  9B32                    85           RDR    RDEV,WORK            READ DATA
   00A8R  4060 00CCR              86           STH    ISTA,SYNFLG          RESET SYNC FLAG
   00ACR  9A12                    87           WDR    DISP,WORK            DISPLAY DATA READ
   00AER  D420 00D9R              88           CLB    WORK,DATA            IS IT DATA SENT?
   00B2R  4330 0054R              89           BE     WAIT1
   00B6R  D420 00D8R              90           CLB    WORK,SYNC            NO - IS IT SYNC?
   00BAR  4230 006ER              91           BNE    HALT1                NO - HALT
   00BER  4300 0054R              92           B      WAIT1
                                  93  *
   00C4R                          94           ALIGN 4
   00C4R  C000                    95  WAIT     DC     X'C000',A(WAIT1)
   00C6R  0054R
   00C8R  8000                    96  HALT     DC     X'8000',A(HALT1)
   00CAR  006ER
   00CCR  0000                    97  SYNFLG   DC     0
   00CER  00B8                    98  RADR     DC     X'B8'                RECV LINE ADDRESS
   00D0R  00BC                    99  XADR     DC     X'BC'                XMIT LINE ADDRESS
   00D2R  40                     100  INCR     DB     X'40'                INCREMENT
   00D3R  80                     101  NORM     DB     X'80'                NORMAL
   00D4R  79                     102  ENBSCH   DB     X'79'                ENABLE, LOCAL, SYNSCH
   00D5R  7B                     103  ENBRQS   DB     X'7B'                ENABLE,LOCAL,RESET DATA MODE,RQ2S
   00D6R  F1                     104  IDLE     DB     X'F1'                DISARM, IDLE
   00D7R  30                     105  COMMON   DB     X'30'                8 BITS/CHAR, NO PARITY
   00D8R  16                     106  SYNC     DB     X'16'                SYNC CHARACTER
   00D9R  55                     107  DATA     DB     X'55'                DATA CHARACTER
   00DAF                         108           END
```

```
    QSA - 2-WIRE PROGRAMMING EXAMPLE - 16 BIT PROCESSOR       PAGE    4  12:53:08   07/19/78

    NON ZBID

ASSEMBLED BY CAL 03-066R05-00   (32-BIT)

START OPTIONS:  *NONE*

   NO CAL ERRORS
   NO CAL WARNINGS
    2 PASSES

 ABSTOP    0000 0000
 ADC       0000 0002
 BUSY      0000 0008         24*    81
 COMMON    0000 00D7R        44     45    105*
 DATA      0000 00D9R        78     88    107*
 DISP      0000 0001         18*    29     31     32     33     34     35     36     65     66     67     87
 DISPLY    0000 0066R        65*    73     84
 ENBRQS    0000 00D5R        57    103*
 EMBSCH    0000 00D4R        55    102*
 HALT      0000 00C8R        70     96*
 HALT1     0000 006ER        68*    91     96
 IDEV      0000 0005         22*    60     61     63     67
 IDLE      0000 00D6R        26     28     68     69    104*
 IMPTOP    0000 00DAR
 INCR      0000 00D2R        31     65    100*
 INTHDL    0000 0058R        40     60*
 ISTA      0000 0006         23*    60     66     72     72     81     83     83     86
 LADC      0000 0001
 NORM      0000 00D3R        36    101*
 PURETOP   0000 0000R
 RADR      0000 00CER        25     98*
 RDEV      0000 0003         20*    25     26     44     55     56     63     68     85
 READ      0000 0098R        64     81*
 SYNC      0000 00D8R        56     76     90    106*
 SYNFLG    0000 00CCR        43     74     86     97*
 WAIT      0000 00C4R        58     95*
 WAIT1     0000 0054R        58*    77     79     82     89     92     95
 WORK      0000 0002         19*    30     30     32     33     34     35     37     38     39     40     41     42
                             43     74     85     87     88     90
 WRITE     0000 007AR        62     72*
 WRTDAT    0000 0090R        75     78*
 XADR      0000 00D0R        27     99*
 XDEV      0000 0004         21*    27     28     45     57     61     69     76     78
```

```
          QSA - 4-WIRE PROGRAMMING EXAMPLE - 16 BIT PROCESSOR        PAGE   1  12:54:54  07/19/78
PROG= *NONE*     ASSFMBLED BY CAL 03-066R05-00   (32-BIT)
                                        1           SCRAT
                                        2           CROSS
                                        3           WIDTH 120
                                        4           TARGT 16
```

```
QSA - 4-WIRE PROGRAMMING EXAMPLE - 16 BIT PROCESSOR       PAGE   2  12:54:54  07/19/78

2 LINES - NON ZBID

                     *
                     * ***************************************************************
                     * THIS SAMPLE PROGRAM ILLUSTRATES A DATA TRANSFER BETWEEN TWO    *
                     * 4-WIRE LINES UNDER LOCAL LOOPBACK ON
                     * THE QSA. A VECTOR TABLE IS SET UP TO HANDLE INTERRRUPTS FROM   *
                     * FOUR DIFFERENT DEVICES THE SAME DATA BYTE IS TRANSMITTED ON    *
                     * BOTH LINES AND IS DISPLAYED ON THE DISPLAY PANEL. IF A DATA    *
                     * ERROR IS DETECTED, THE PROGRAM HALTS WITH THE BAD DATA         *
                     * DISPLAYED. IF A STATUS OR ADDRESS ERROR OCCURS, THE PROGRAM    *
                     * HALTS WITH THE DEVICE ADDRESS AND STATUS DISPLAYED. PROGRAM    *
                     * IS UNDER INTERRUPT CONTROL AND STOPS ONLY ON AN ERROR.         *
                     * ***************************************************************
                     *
       0000 0001     DISP      EQU   1                      DISPLAY ADDRESS REGISTER
       0000 0002     WORK      EQU   2                      WORK REGISTER
       0000 0003     IDEV      EQU   3                      INTERRUPTED DEVICE ADDRESS REG.
       0000 0004     ISTA      EQU   4                      INTERRUPTED DEVICE STATUS REG.
       0000 0005     IDX       EQU   5                      INDEX REGISTER
       0000 0006     R1        EQU   6
       0000 0009     WORK1     EQU   9
       0000 0008     BUSY      EQU   8                      BUSY STATUS
0000R  0755                    XHR   IDX,IDX                CLEAR THE INDEX REGISTER
0002R  4825 00FCR    SETUP     LH    WORK,DEVADR(IDX)       GET THE FIRST DEVICE AND
0006R  DE20 0110R              OC    WORK,IDLE              IDLE THE DEVICE
000AR  2652                    AIS   IDX,2                  INCREMENT THE INDEX
000CR  C550 0008               CLHI  IDX,VECTOR-DEVADR      FINISHED?
0010R  2037                    BNES  SETUP                  NO - CONTINUE
                     *                                    YES - BLANK THE DISPLAY
0012R  2411                    LIS   DISP,1                 DISPLAY PANEL ADDRESS
0014R  0722                    XHR   WORK,WORK
0016R  DE10 010DR              OC    DISP,INCR              SET DISPLAY TO INCREMENT MODE
001AR  9A12                    WDR   DISP,WORK              CLEAR DISPLAY
001CR  9A12                    WDR   DISP,WORK
001ER  9A12                    WDR   DISP,WORK
0020R  9A12                    WDR   DISP,WORK
0022R  DE10 010CR              OC    DISP,NORM              SET DISPLAY BACK TO NORMAL MODE
0026R  4020 0040               STH   WORK,X'40'             SET UP LOW CORE
002AR  4020 0042               STH   WORK,X'42'
002ER  4020 0044               STH   WORK,X'44'
0032R  C820 0074R              LHI   WORK,INTHDL            SET UP INTERRUPT HANDLER ADDRESS
0036R  4020 0046               STH   WORK,X'46'
003AR  0755                    XHR   IDX,IDX                CLEAR INDEX
003CR  C820 009CR    SETVET    LHI   WORK,READ1             STORE READ ROUTINE ADDRESS
0040R  4025 0104R              STH   WORK,VECTOR(IDX)       INTO VECTOR TABLE
0044R  4865 00FCR              LH    R1,DEVADR(IDX)         GET RECV DEVICE ADDRESS
0048R  DE60 0111R              OC    R1,COMMON              SET COMMON CONTROL
004CR  DE60 010ER              OC    R1,ENBSCH              RECV ENABLE, SYNSCH
0050R  DA60 0112R              WD    R1,SYNC                SET SYNC MATCH REGISTER
0054R  2652                    AIS   IDX,2                  INCREMENT INDEX
0056R  C820 00D6R              LHI   WORK,WRITE1            STORE WRITE ROUTINE ADDRESS
005AR  4025 0104R              STH   WORK,VECTOR(IDX)       INTO VECTOR TABLE
005ER  4865 00FCR              LH    R1,DEVADR(IDX)         GET XMIT DEVICE ADDRESS
0062R  DE60 010FR              OC    R1,ENBRQS              XMIT ENABLE, RQ2S
```

```
QSA - 4-WIRE PROGRAMMING EXAMPLE - 16 BIT PROCESSOR       PAGE   3  12:54:54  07/19/78

2 LINES - NON ZBID
0066R  2652                            AIS    IDX,2                 INCREMENT INDEX
0068R  C550 0008                       CLHI   IDX,VECTOR-DEVADR     END OF TABLE ?
006CR  4280 003CR                      BL     SETVET
0070R  C200 00F4R              WAIT1   LPSW   WAIT
                               *
0074R  9F34                    INTHDL  AIR    IDEV,ISTA             ACKNOWLEDGE INTERRUPT
0076R  0755                            XHR    IDX,IDX
0078R  4535 00FCR              MATCH   CLH    IDEV,DEVADR(IDX)      SEARCH DEVICE ADDRESS TABLE
007CR  4330 0096R                      BE     GETRTIN               MATCH - GO TO ROUTINE
0080R  2652                            AIS    IDX,2
0082R  C550 0008                       CLHI   IDX,VECTOR-DEVADR     END OF TABLE ?
0086R  4280 0078R                      BL     MATCH
008AR  DF10 010DR              DSPLAY  OC     DISP,INCR             ERROR -
008ER  9A14                            WDR    DISP,ISTA             DISPLAY STATUS
0090R  9A13                            WDR    DISP,IDEV             DISPLAY DEVICE ADDRESS
0092R  C200 00F8R              HALT1   LPSW   HALT                  HALT
                               *
0096R  4825 0104R              GETRTIN LH     WORK,VECTOR(IDX)      GET ROUTINE ADDRESS
009AR  0302                            BR     WORK
                               *
009CR  C540 0008               READ1   CLHI   ISTA,BUSY             CARRIER ON INTERRUPT ?
00A0R  4230 008AR                      BNE    DSPLAY
00A4R  C820 00B0R                      LHI    WORK,READ2            RESET READ ROUTINE ADDRESS
00A8R  4025 0104R                      STH    WORK,VECTOR(IDX)      IN VECTOR TABLE
00ACR  4300 0070R                      B      WAIT1
                               *
00B0R  0844                    READ2   LHR    ISTA,ISTA             ZERO STATUS?
00B2R  4230 008AR                      BNZ    DSPLAY                NO - HALT
00B6R  9B32                            RDR    IDEV,WORK             READ A BYTE
00B8R  9A12                            WDR    DISP,WORK             DISPLAY IT TO PANEL
00BAR  D425 0114R                      CLB    WORK,DATA(IDX)        IS IT DATA ?
00BER  4330 0070R                      BE     WAIT1
00C2R  D420 0112R                      CLB    WORK,SYNC             IS IT SYNC BYTE?
00C6R  4230 0092R                      BNE    HALT1
00CAR  C890 00E4R                      LHI    WORK1,WRITE2          SET THE WRITE ROUTINE ADDRESS
00CER  4095 0106R                      STH    WORK1,VECTOR+2(IDX)   IN THE VECTOR TABLE
00D2R  4300 0070R                      B      WAIT1
                               *
00D6R  0844                    WRITE1  LHR    ISTA,ISTA             ZERO STATUS?
00D8R  4230 008AR                      BNZ    DSPLAY                NO - HALT
00DCR  DA30 0112R                      WD     IDEV,SYNC             WRITE SYNC BYTE
00E0R  4300 0070R                      B      WAIT1
                               *
00E4R  0844                    WRITE2  LHR    ISTA,ISTA             ZERO STATUS?
00E6R  4230 008AR                      BNZ    DSPLAY                NO - HALT
00EAR  DA35 0114R                      WD     IDEV,DATA(IDX)        WRITE A DATA BYTE
00EER  4300 0070R                      B      WAIT1
00F4R                                  ALIGN  4
00F4R  C000                    WAIT    DC     X'C000',A(WAIT1)      ENABLE INTERRUPT, WAIT
00F6R  0070R
00F8R  8000                    HALT    DC     X'8000',A(HALT1)      DISABLE INTERRUPT, WAIT
00FAR  0092R
00FCR  00B8                    DEVADR  DC     X'B8',X'BD',X'BC',X'B9'
```

```
      QSA - 4-WIRE PROGRAMMING EXAMPLE - 16 BIT PROCESSOR       PAGE   4  12:54:54  07/19/78

      2 LINES - NON ZBID

00FER  00BD
0100R  00BC
0102R  00B9
0104R  0000                110  VECTOR   DC     0,0,0,0
0106R  0000
0108R  0000
010AR  0000
010CR  80                  111  NORM     DB     X'80'                DISPLAY NORMAL MODE
010DR  40                  112  INCR     DB     X'40'                DISPLAY INCREMENT MODE
010ER  79                  113  ENBSCH   DB     X'79'                ENABLE, LOCAL, SYNSCH
010FR  7B                  114  ENBRQS   DB     X'7B'                ENABLE,LOCAL,RESET DATA MODE,RQ2S
0110R  F1                  115  IDLE     DB     X'F1'                DISARM, IDLE
0111R  30                  116  COMMON   DB     X'30'                8 BITS/CHAR, NO PARITY
0112R  16                  117  SYNC     DB     X'16'                SYNC CHARACTER
0114R                      118           ALIGN 4
0114R   0100               119  DATA     DC     X'0100'
0116R   0100               120           DC     X'0100'
0118R   8000               121           DC     X'8000'
011AR   8000               122           DC     X'8000'
011CR                      123           END
```

```
     QSA - 4-WIRE PROGRAMMING EXAMPLE - 16 BIT PROCESSOR       PAGE   5  12:54:54  07/19/78

     2 LINES - NON ZBID

 ASSEMBLED BY CAL 03-066R05-00   (32-BIT)

 START OPTIONS:  *NONE*

    NO CAL ERRORS
    NO CAL WARNINGS
     2 PASSES

  ABSTOP     0000 0000
  ADC        0000 0002
  BUSY       0000 0008          26*    79
  COMMON     0000 0111R         51     116*
  DATA       0000 0114R         89     104    119*
  DEVADR     0000 00FCR         28     31     50     57     60     66     69     109*
  DISP       0000 0001          19*    34     36     37     38     39     40     41     71     72     73     88
  DSPLAY     0000 008AR         71*    80     86     98     103
  ENBRQS     0000 010FR         58     114*
  ENBSCH     0000 010ER         52     113*
  GETRTIN    0000 0096R         67     76*
  HALT       0000 00F8R         74     108*
  HALT1      0000 0092R         74*    92     108
  IDEV       0000 0003          21*    64     66     73     87     99     104
  IDLE       0000 0110R         29     115*
  IDX        0000 0005          23*    27     27     28     30     31     47     47     49     50     54     56     57
                                59     60     65     65     66     68     69     76     82     89     94     104
  IMPTOP     0000 011CR
  INCR       0000 010DR         36     71     112*
  INTHDL     0000 0074R         45     64*
  ISTA       0000 0004          22*    64     72     79     85     85     97     97     102    102
  LADC       0000 0001
  MATCH      0000 0078R         66*    70
  NORM       0000 010CR         41     111*
  PURETOP    0000 0000R
  R1         0000 0006          24*    50     51     52     53     57     58
  READ1      0000 009CR         48     79*
  READ2      0000 00B0R         81     85*
  SETUP      0000 0002R         28*    32
  SETVET     0000 003CR         48*    61
  SYNC       0000 0112R         53     91     99     117*
  VECTOR     0000 0104R         31     49     56     60     69     76     82     94     110*
  WAIT       0000 00F4R         62     107*
  WAIT1      0000 0070R         62*    83     90     95     100    105    107
  WORK       0000 0002          20*    28     29     35     35     37     38     39     40     42     43     44     45
                                46     48     49     55     56     76     77     81     82     87     88     89     91
  WORK1      0000 0009          25*    93     94
  WRITE1     0000 00D6R         55     97*
  WRITE2     0000 00E4R         93     102*
```

```
        QSA - 2-WIRE PROGRAMMING EXAMPLE - 16 BIT PROCESSOR       PAGE   1  12:55:57  07/19/78
PROG= *NONE*     ASSEMBLED BY CAL 03-066R05-00   (32-BIT)
                                 1          SCRAT
                                 2          CROSS
                                 3          WIDTH 120
                                 4          TARGT 16
```

```
ZBID

                           *
                           * ***************************************************************
                           * THIS PROGRAM ILLUSTRATES A DATA TRANSFER BETWEEN TWO LINES    *
                           * ZBID MODE. 2-WIRE OPERATION IS ASSUMED. 10 DATA BYTES ARE
                           * SENT AND FOLLOWED BY 10 FLAGS. THE FLAGS ARE SENT BY HARDWARE. *
                           * THIS PATTERN CONTINUES UNTIL AN ERROR IS DETECTED. IF A STATUS *
                           * OR ADDRESS ERROR IS DETECTED. THE PROCESSOR HALTS WITH THE     *
                           * STATUS AND DEVICE ADDRESS DISPLAYED. IF IT IS A DATA ERROR,    *
                           * ONLY THE DATA BYTE IS DISPLAYED. PROGRAM IS EXECUTED UNDER     *
                           * INTERRUPT.                                                     *
                           * ***************************************************************
                           *
      0000 0001            DISP     EQU   1                 DISPLAY PANEL ADDRESS
      0000 0002            WORK     EQU   2                 WORK REGISTER
      0000 0003            RDEV     EQU   3                 RECEIVE DEVICE ADDRESS
      0000 0004            XDEV     EQU   4                 TRANSMIT DEVICE ADDRESS
      0000 0005            IDEV     EQU   5                 INTERRUPTED DEVICE ADDRESS
      0000 0006            ISTA     EQU   6                 INTERRUPTED DEVICE STATUS
      0000 0007            COUNT    EQU   7                 COUNT REGISTER
      0000 0008            R1       EQU   8                 INCREMENT REGISTER
      0000 0009            R2       EQU   9                 END VALUE REGISTER
      0000 0008            BUSY     EQU   8                 BUSY
      0000 00BB            TERM     EQU   X'BB'             INVERSE OF TERM STATUS
      0000 007E            ENDFLG   EQU   X'7E'             END FLAG
0000R 4830 00E6R                    LH    RDEV,RADR         GET RECV DEVICE ADDRESS
0004R DE30 00EFR                    OC    RDEV,IDLE         IDLE BOTH DEVICE ADDRESSES AND CLEAR
0008R 4840 00E8R                    LH    XDEV,XADR         GET XMIT DEVICE ADDRESS
000CR DE40 00EFR                    OC    XDEV,IDLE         ANY QUEUED INTERRUPTS
0010R 2411                          LIS   DISP,1            DISPLAY PANEL ADDRESS
0012R 0722                          XHR   WORK,WORK
0014R DE10 00EAR                    OC    DISP,INCR         SET DISPLAY TO INCREMENT MODE
0018R 9A12                          WDR   DISP,WORK         CLEAR DISPLAY PANEL
001AR 9A12                          WDR   DISP,WORK
001CR 9A12                          WDR   DISP,WORK
001ER 9A12                          WDR   DISP,WORK
0020R DE10 00EBR                    OC    DISP,NORM         SET DISPLAY BACK TO NORMAL MODE
0024R 4020 00E4R                    STH   WORK,FLAG         RESET FLAG INDICATOR
0028R 4020 0040                     STH   WORK,X'40'        SET UP LOW CORE
002CR 4020 0042                     STH   WORK,X'42'
0030R 4020 0044                     STH   WORK,X'44'
0034R C820 005ER                    LHI   WORK,INTHDL       SET UP INTERRUPT HANDLER ADDRESS
0038R 4020 0046                     STH   WORK,X'46'
003CR 2499                          LIS   R2,9              10 BYTES PER FRAME
003ER 2481                          LIS   R1,1              INCREMENT
0040R 0777                          XHR   COUNT,COUNT       CLEAR COUNTER
0042R DE30 00F1R                    OC    RDEV,COMMON       SET ZBID MODE
0046R DE40 00F1R                    OC    XDEV,COMMON
004AR DE40 00F0R                    OC    XDEV,WRMOD        PUT XMIT TO WRITE MODE
004ER DA40 00F3R                    WD    XDEV,FILL         ENSURE LINE STARTS UP WITH MARKS
0052R DE40 00EDR                    OC    XDEV,ENBRQS       XMIT LINE IN RQ2S
                           *
                           *  NOTE :
                           *
```

```
QSA - 2-WIRE PROGRAMMING EXAMPLE - 16 BIT PROCESSOR       PAGE   3  12:55:57  07/19/78

ZBID
                        *     IF THE NEXT COMMAND IS BEING ISSUED TO TURN A 2-WIRE LINE
                        *     AROUND FROM WRITE TO READ, THE COMMAND MUST BE ISSUED TWICE
                        *     FIRST WITH INTERRUPTS DISARMED,AND THEN REISSUED WITH THE
                        *     DESIRED INTERRUPT CONDITION, UNLESS THE DISARM CONDITION
                        *     IS DESIRED.
                        *
       0000 0056R       WAIT2    EQU   *
0056R  DE30 00ECR                OC    RDEV,ENBSCH          RECV LINE IN SYNSCH
005AR  C200 00DCR       WAIT1    LPSW  WAIT
                        *
005ER  9F56             INTHDL   AIR   IDEV,ISTA            ACKNOWLEDGE INTERRUPT
0060R  0554                      CLHR  IDEV,XDEV            WRITE?
0062R  4330 0080R                BE    WRITE                YES - GO TO WRITE ROUTINE
0066R  0553                      CLHR  IDEV,RDEV            READ?
0068R  4330 00B2R                BE    READ                 YES - GO TO READ ROUTINE
006CR  DE10 00EAR       DSPLAY   OC    DISP,INCR            NEITHER -
0070R  9A16                      WDR   DISP,ISTA            DISPLAY STATUS
0072R  9A15                      WDR   DISP,IDEV            DISPLAY DEVICE ADDRESS
0074R  DE30 00EFR       HALT1    OC    RDEV,IDLE            IDLE LINES
0078R  DE40 00EFR                OC    XDEV,IDLE
007CR  C200 00E0R                LPSW  HALT
                        *
0080R  4820 00E4R       WRITE    LH    WORK,FLAG            FLAG SET?
0084R  4230 0092R                BNZ   CONTIN               YES - SKIP WRITE DATA
0088R  0866                      LHR   ISTA,ISTA            STATUS ZERO?
008AR  4230 006CR                BNZ   DSPLAY               NO - ERROR
008ER  DA40 00F2R                WD    XDEV,DATA            WRITE A BYTE
0092R  C170 005AR       CONTIN   BXLE  COUNT,WAIT1          CONTINUE FOR 10 BYTES
0096R  0777                      XHR   COUNT,COUNT          CLEAR COUNTER
0098R  4820 00E4R                LH    WORK,FLAG            REVERSE FLAG INDICATOR
009CR  C720 00FF                 XHI   WORK,X'FF'
00A0R  4020 00E4R                STH   WORK,FLAG
00A4R  DE40 00EER                OC    XDEV,ENBRQ           RESET OV
00A8R  0822                      LHR   WORK,WORK            SET THE CONDITION CODE
                        *   WHERE FLAGS BEING SENT ON THE LINE
00AAR  4230 005AR                BNZ   WAIT1                NO - DATA WAS ON THE LINE
00AER  4300 0056R                B     WAIT2                YES - PUT THE RCV SIDE INTO SYNC
                        *                                      SEARCH
                        *
00B2R  C560 0008        READ     CLHI  ISTA,BUSY            CARRIER-ON INTERRUPT?
00B6R  4330 005AR                BE    WAIT1                YES - WAIT
00BAR  C360 00BB                 THI   ISTA,TERM            STATUS ZERO OR TERM?
00BER  4230 006CR                BNZ   DSPLAY               NO - ERROR
00C2R  9B32                      RDR   RDEV,WORK            READ A BYTE
00C4R  9A12                      WDR   DISP,WORK            DISPLAY THE BYTE
00C6R  D420 00F2R                CLB   WORK,DATA            IS IT DATA SENT?
00CAR  4330 005AR                BE    WAIT1                YES - WAIT
00CER  C520 007E                 CLHI  WORK,ENDFLG          IS IT THE END FLAG?
00D2R  4330 005AR                BE    WAIT1                YES - WAIT
00D6R  4300 0074R                B     HALT1                NO - HALT
00DCR                            ALIGN 4
00DCR  C000             WAIT     DC    X'C000',A(WAIT1)
00DER  005AR
```

```
     QSA - 2-WIRE PROGRAMMING EXAMPLE - 16 BIT PROCESSOR       PAGE   4  12:55:57  07/19/78

     ZBID

00E0R  8000              111  HALT    DC    X'8000',A(HALT1)
00E2R  0074R
00E4R  0000              112  FLAG    DC    0
00E6R  00B8              113  RADR    DC    X'B8'            RECV LINE ADDRESS
00E8R  00BC              114  XADR    DC    X'BC'            XMIT LINE ADDRESS
00EAR  40                115  INCR    DB    X'40'            INCREMENT
00EBR  80                116  NORM    DB    X'80'            NORMAL
00ECR  79                117  ENBSCH  DB    X'79'            ENABLE, LOCAL, SYNSCH
00EDR  7B                118  ENBRQS  DB    X'7B'            ENABLE,LOCAL,RESET DATA MODE,RQ2S
00EER  6B                119  ENBRQ   DB    X'6B'            ENABLE, LOCAL, RQ2S
00EFR  F1                120  IDLE    DB    X'F1'            DISARM, IDLE
00F0R  FB                121  WRMOD   DB    X'FB'            DISARM,LOCAL,READY,WRITE,RESET D.M.
00F1R  32                122  COMMON  DB    X'32'            ZBID MODE
00F2R  55                123  DATA    DB    X'55'            DATA BYTE
00F3R  FF                124  FILL    DB    X'FF'            MARK CHARACTER
00F4R                    125          END
```

```
    QSA - 2-WIRE PROGRAMMING EXAMPLE - 16 BIT PROCESSOR       PAGE   5  12:55:57  07/19/78

    ZBID

ASSEMBLED BY CAL 03-066R05-00   (32-BIT)

START OPTIONS:  *NONE*

   NO CAL ERRORS
   NO CAL WARNINGS
    2 PASSES

 ARSTOP     0000 0000
 ADC        0000 0002
 BUSY       0000 0008        27*     98
 COMMON     0000 00F1R       51      52     122*
 CONTIN     0000 0092R       82      86*
 COUNT      0000 0007        24*     50      50      86      87      87
 DATA       0000 00F2R       85     104     123*
 DISP       0000 0001        18*     34      36      37      38      39      40      41      74      75      76     103
 DSPLAY     0000 006CR       74*     84     101
 ENBRQ      0000 00EER       91     119*
 ENBRQS     0000 00EDR       55     118*
 ENBSCH     0000 00ECR       66     117*
 ENDFLG     0000 007E        29*    106
 FILL       0000 00F3R       54     124*
 FLAG       0000 00E4R       42      81      88      90     112*
 HALT       0000 00E0R       79     111*
 HALT1      0000 0074R       77*    108     111
 IDEV       0000 0005        22*     69      70      72      76
 IDLE       0000 00EFR       31      33      77      78     120*
 IMPTOP     0000 00F4R
 INCR       0000 00EAR       36      74     115*
 INTHDL     0000 005ER       46      69*
 ISTA       0000 0006        23*     69      75      83      83      98     100
 LADC       0000 0001
 NORM       0000 00EBR       41     116*
 PURETOP    0000 0000R
 R1         0000 0008        25*     49
 R2         0000 0009        26*     48
 RADR       0000 00E6R       30     113*
 RDEV       0000 0003        20*     30      31      51      66      72      77     102
 READ       0000 00B2R       73      98*
 TERM       0000 00BB        28*    100
 WAIT       0000 00DCR       67     110*
 WAIT1      0000 005AR       67*     86      94      99     105     107     110
 WAIT2      0000 0056R       65*     95
 WORK       0000 0002        19*     35      35      37      38      39      40      42      43      44      45      46      47
                             81      88      89      90      92      92     102     103     104     106
 WRITE      0000 0080R       71      81*
 WRMOD      0000 00F0R       53     121*
 XADR       0000 00E8R       32     114*
 XDEV       0000 0004        21*     32      33      52      53      54      55      70      78      85      91
```

```
       QSA - 4-WIRE PROGRAMMING EXAMPLE - 16 BIT PROCESSOR       PAGE   1  12:57:05  07/19/78
PROG= *NONE*     ASSEMBLED BY CAL 03-066R05-00   (32-BIT)
                                         1           SCRAT
                                         2           CROSS
                                         3           WIDTH 120
                                         4           TARGT 16
                                         5           PROG  QSA - 4-WIRE PROGRAMMING EXAMPLE - 16 BIT PROCESSOR
```

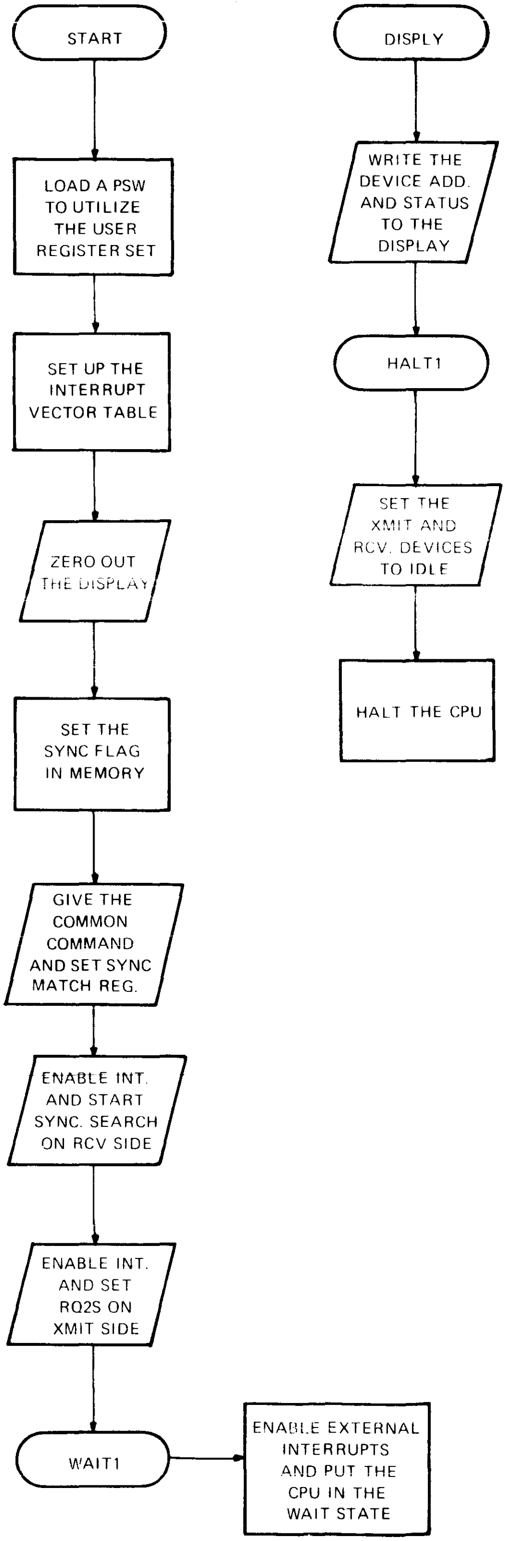
```
QSA - 4-WIRE PROGRAMMING EXAMPLE - 16 BIT PROCESSOR       PAGE   2  12:57:05  07/19/78

2 LINES - ZBID

                      * *************************************************************
                      * THIS SAMPLE PROGRAM ILLUSTRATES A DATA TRANSFER BETWEEN TWO    *
                      * 4-WIRE LINES RUNNING IN ZBID MODE UNDER LOCAL
                      * LOOPBACK. 10 DATA BYTES ARE SENT BY THE PROGRAM, AND THE 10    *
                      * FLAGS ARE SENT BY THE HARDWARE, DUE TO THE OVERFLOW CONDITION  *
                      * GENERATED AND CONTROLLED BY THE TRANSMIT SIDE. THIS PATTERN    *
                      * OF 10 DATA BYTES AND 10 FLAGS IS OUTPUT UNTIL AN ERROR IS      *
                      * DETECTED, OR PROGRAM EXECUTION IS TERMINATED BY THE USER.      *
                      * THROUGHOUT THE OPERATION OF THE PROGRAM THE DATA RECEIVED      *
                      * IS DISPLAYED ON THE DISPLAY PANEL. IF THE PROCESSOR HALTS      *
                      * ONE OF TWO ERRORS HAVE BEEN DETECTED. IF ONLY ONE DATA BYTE    *
                      * IS DISPLAYED WHEN THE PROCESSOR HALTS THEN A DATA ERROR WAS    *
                      * DETECTED, AND THE BYTE DISPLAYED IS THE BYTE IN ERROR. IF TWO  *
                      * BYTES ARE DISPLAYED, A STATUS ERROR OR AN UNEXPECTED INTERRUPT *
                      * WAS DETECTED. IN THIS CASE THE LEFTMOST BYTE IS THE DEVICE     *
                      * ADDRESS, AND THE OTHER BYTE IS THE DEVICE STATUS.              *
                      *                                                                *
                      *                                                                *
                      * *************************************************************
      0000 0001       DISP     EQU   1                   DISPLAY ADDRESS REGISTER
      0000 0002       WORK     EQU   2                   WORK REGISTER 0
      0000 0003       IDEV     EQU   3                   INTERRUPTING DEVICE ADDRESS
      0000 0004       ISTA     EQU   4                   INTERRUPTING DEVICE STATUS
      0000 0005       IDX      EQU   5                   INDEX REGISTER
      0000 0006       WORK1    EQU   6                   WORK REGISTER 1
                      *
0000R 0755                     XHR   IDX,IDX             CLEAR THE INDEX REGISTER
0002R 4825 0124R      SETUP    LH    WORK,DEVADR(IDX)    GET THE FIRST DEVICE AND
0006R DE20 0143R               OC    WORK,IDLE           IDLE THE DEVICE
000AR 2652                     AIS   IDX,2               INCREMENT THE INDEX
000CR C550 0008                CLHI  IDX,VECTOR-DEVADR   FINISHED?
0010R 2037                     BNES  SETUP               NO - CONTINUE
                      *                                 YES - BLANK THE DISPLAY
0012R 2411                     LIS   DISP,1              LOAD THE DISPLAY PANEL ADDRESS
0014R 0722                     XHR   WORK,WORK           SET ZERO DATA
0016R DE10 013DR               OC    DISP,INCR           SET THE DISPLAY TO INCREMENTAL MODE
001AR 9812                     WHR   DISP,WORK           ZERO OUT THE
001CR 9812                     WHR   DISP,WORK           DISPLAY PANEL
001ER DE10 013CR               OC    DISP,NORM           SET THE DISPLAY TO NORMAL MODE
0022R 4020 0040                STH   WORK,X'40'          SET UP LOW CORE
0026R 4020 0042                STH   WORK,X'42'
002AR 4020 0044                STH   WORK,X'44'
002ER C820 007ER               LHI   WORK,INTHDL         SET UP THE INTERRUPT HANDLER ADDRESS
0032R 4020 0046                STH   WORK,X'46'
0036R 0755                     XHR   IDX,IDX             CLEAR THE INDEX REG
0038R C820 00F2R      SETVET   LHI   WORK,READ           STORE THE READ ROUTINE ADDRESS
003CR 4025 012CR               STH   WORK,VECTOR(IDX)    INTO THE VECTOR TABLE
0040R 4865 0124R               LH    WORK1,DEVADR(IDX)   GET THE RECEIVE DEVICE ADDRESS
0044R DE60 013FR               OC    WORK1,COMMON        SET UP THE COMMON CONTROL
0048R DE60 013ER               OC    WORK1,ENBSCH        RCV. ENABLE, SYNC. SEARCH
004CR 2652                     AIS   IDX,2               INCREMENT THE INDEX
004ER C820 00A2R               LHI   WORK,WRITE          STORE THE WRITE ROUTINE ADDRESS
0052R 4025 012CR               STH   WORK,VECTOR(IDX)    INTO THE VECTOR TABLE
```

```
QSA - 4-WIRE PROGRAMMING EXAMPLE - 16 BIT PROCESSOR       PAGE   3  12:57:05  07/19/78

2 LINES - ZBID

                         *
0056R  0722                       XHR   WORK,WORK            CLEAR WORK
0058R  4025 0134R                 STH   WORK,CONTROL(IDX)    CLEAR CONTROL FLAG
005CR  4025 0132R                 STH   WORK,CONTROL-2(IDX)  CLEAR THE BYTE COUNT
                         *
                         *
0060R  4865 0124R                 LH    WORK1,DEVADR(IDX)    GET THE XMIT DEVICE ADDRESS
0064R  DE60 0144R                 OC    WORK1,WRMOD          PUT XMIT TO WRITE MODE
0068R  DA60 0145R                 WD    WORK1,FILL           ENSURE LINE STARTS UP WITH MARKS
006CR  DE60 0140R                 OC    WORK1,ENBRQS         ENABLE XMIT, RQ2S
0070R  2652                       AIS   IDX,2                INCREMENT THE INDEX
0072R  C550 0008                  CLHI  IDX,VECTOR-DEVADR    MORE LINES TO ACTIVATE ?
0076R  4280 0038R                 BL    SETVET               YES
007AR  C200 011CR        WAIT1    LPSW  WAIT                 NO - WAIT
                         *
007ER  9F34              INTHDL   AIR   IDEV,ISTA            ACKNOWLEDGE THE INTERRUPT
0080R  0755                       XHR   IDX,IDX              CLEAR THE INDEX
0082R  4535 0124R        MATCH    CLH   IDEV,DEVADR(IDX)     SEARCH DEVICE ADDRESS TABLE
0086R  233B                       BES   GETRTIN              MATCH - FIND THE ROUTINE
0088R  2652                       AIS   IDX,2                BUMP THE INCREMENT
008AR  C550 0008                  CLHI  IDX,VECTOR-DEVADR    FINISHED ?
008ER  2086                       BLS   MATCH                NO - CONTINUE
0090R  DE10 013DR        DSPLAY   OC    DISP,INCR            YES - ERROR
0094R  9A14                       WDR   DISP,ISTA            DISPLAY THE STATUS
0096R  9A13                       WDR   DISP,IDEV            DISPLAY THE DEVICE ADDRESS
0098R  C200 0120R        HALT1    LPSW  HALT                 HALT THE CPU
                         *
009CR  4825 012CR        GETRTIN  LH    WORK,VECTOR(IDX)     GET THE SUBROUTINE ADDRESS
00A0R  0302                       BR    WORK                 BRANCH
                         *
                         *
       0000 00A2R        WRITE    EQU   *
00A2R  4825 0134R                 LH    WORK,CONTROL(IDX)    IS THE FLAG SET ?
00A6R  4230 00B4R                 BNZ   CONTIN               YES - SKIP THE WRITE DATA
00AAR  0844                       LHR   ISTA,ISTA            NO - IS THE STATUS ZERO ?
00ACR  4230 0090R                 BNZ   DSPLAY               NO - ERROR
00B0R  DA30 0142R                 WD    IDEV,DATA
00B4R  4825 0132R        CONTIN   LH    WORK,CONTROL-2(IDX)  GET THE BYTE COUNT
00B8R  2621                       AIS   WORK,1               INCREMENT BY 1
00BAR  C520 000A                  CLHI  WORK,10              FINISHED ?
00BER  4330 00CAR                 BE    CONTIN1              YES - SET UP TO SEND FLAG
00C2R  4025 0132R                 STH   WORK,CONTROL-2(IDX)  NO - SAVE COUNT
00C6R  4300 007AR                 B     WAIT1                RETURN
00CAR  0722              CONTIN1  XHR   WORK,WORK            CLEAR THE COUNT
00CCR  4025 0132R                 STH   WORK,CONTROL-2(IDX)  STORE IN CONTROL TABLE
00D0R  4825 0134R                 LH    WORK,CONTROL(IDX)    GET THE FLAG
00D4R  C720 00FF                  XHI   WORK,X'FF'           CHANGE THE FLAG
00D8R  4025 0134R                 STH   WORK,CONTROL(IDX)    STORE IT BACK
00DCR  DE30 0141R                 OC    IDEV,ENBRQ           RESET THE OVERFLOW BIT
00E0R  0822                       LHR   WORK,WORK            WHERE FLAGS BEING SENT
00E2R  4230 007AR                 BNZ   WAIT1                NO - DATA
                         *                                  YES -
00E6R  4825 0122R                 LH    WORK,DEVADR-2(IDX)   GET THE ADDRESS OF THE RECEIVE LINE
```

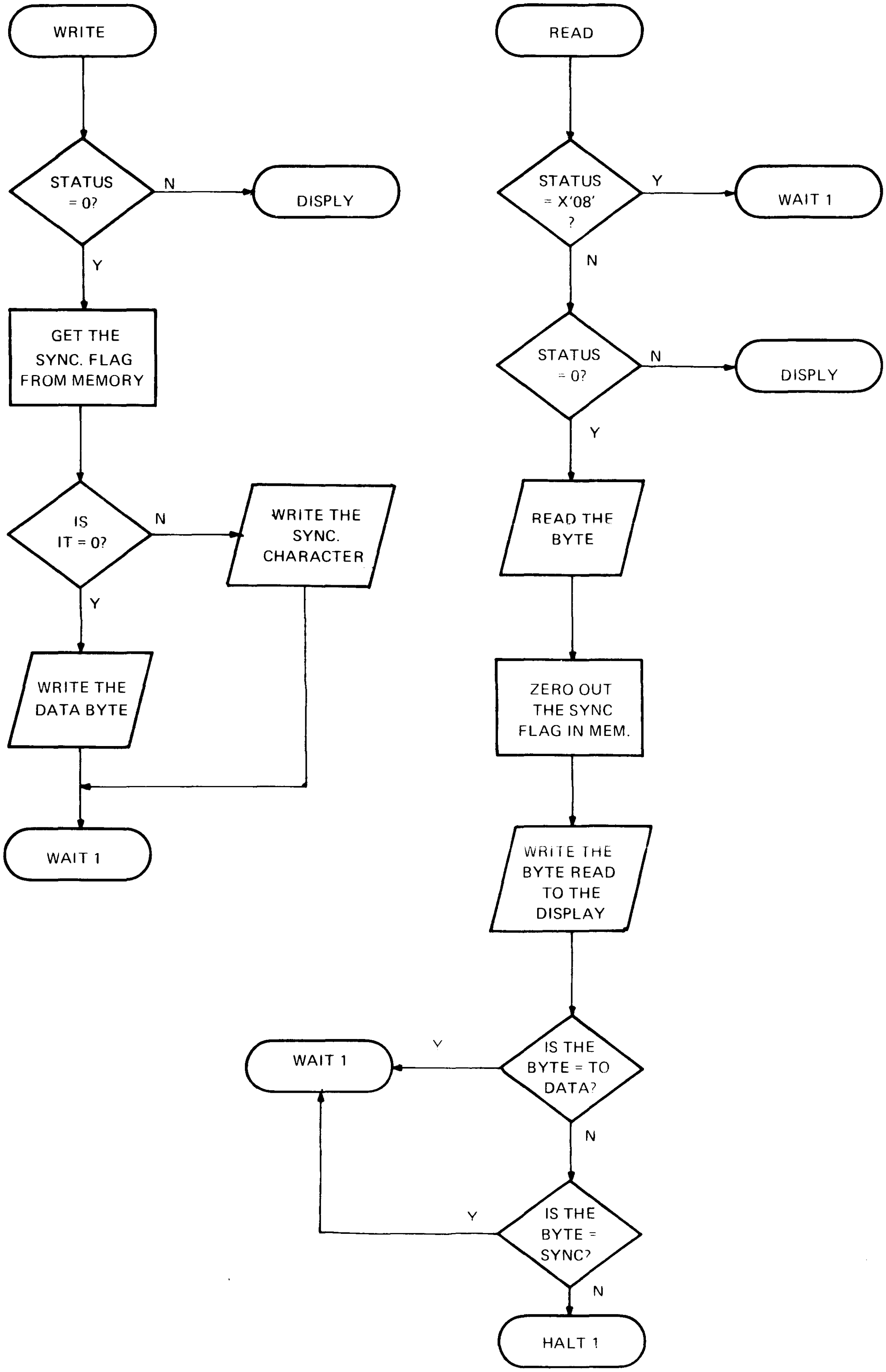
```
                          113  *                                  ASSOCIATED WITH THIS TRANSMIT LINE
00EAR  DE20 013ER         114            OC     WORK,ENBSCH        PUT THE RECEIVE SIDE INTO SYNC.
                          115  *                                  SEARCH
00EER  4300 007AR         116            B      WAIT1
                          117  *
       0000 00F2R         118  READ      EQU    *
00F2R  C540 0008          119            CLHI   ISTA,BUSY           CARRIER ON INTERRUPT ?
00F6R  4330 007AR         120            BE     WAIT1               YES - WAIT
00FAR  C340 00BB          121            THI    ISTA,TERM           NO - IS IT TERMINATE OR ZERO STATUS ?
00FER  4230 0090R         122            BNZ    DSPLAY              NO - ERROR
0102R  9B32               123            RDR    IDEV,WORK           YES - DISPLAY
0104R  9A12               124            WDR    DISP,WORK           THE DATA READ
0106R  D420 0142R         125            CLB    WORK,DATA           WAS DATA RCVED. ?
010AR  4330 007AR         126            BE     WAIT1               YES
010ER  C520 007E          127            CLHI   WORK,ENDFLG         NO - WAS A FLAG RCV'ED.
0112R  4330 007AR         128            BE     WAIT1
0116R  4300 0098R         129            B      HALT1
                          130  *
011CR                     131            ALIGN 4
                          132  *
011CR  C000               133  WAIT      DC     X'C000',A(WAIT1)
011ER  007AR
0120R  8000               134  HALT      DC     X'8000',A(HALT1)
0122R  0098R
                          135  *
0124R  00B8               136  DEVADR    DC     X'B8',X'BD',X'BC',X'B9'
0126R  00BD
0128R  00BC
012AR  00B9
012CR  0000               137  VECTOR    DC     0,0,0,0
012ER  0000
0130R  0000
0132R  0000
0134R  0000               138  CONTROL   DC     0,0,0,0
0136R  0000
0138R  0000
013AR  0000
013CR  80                 139  NORM      DB     X'80'               DISPLAY NORMAL MODE
013DR  40                 140  INCR      DB     X'40'               DISPLAY INCREMENTAL MODE
013ER  79                 141  ENBSCH    DB     X'79'               ENABLE,LOCAL,SYNC. SEARCH
013FR  32                 142  COMMON    DB     X'32'               ZBID MODE
0140R  7B                 143  ENBRQS    DB     X'7B'               ENABLE,LOCAL,RESET DATA MODE,RQ2S
0141R  6B                 144  ENBRQ     DB     X'6B'               ENABLE,LOCAL,RQ2S
0142R  55                 145  DATA      DB     X'55'               DATA BYTE
0143R  F1                 146  IDLE      DB     X'F1'
0144R  FB                 147  WRMOD     DB     X'FB'               DISARM,LOCAL,READY,WRITE,RESET D.M.
0145R  FF                 148  FILL      DB     X'FF'               MARK CHARACTER
       0000 0008          149  BUSY      EQU    8                   BUSY STATUS BIT
       0000 00BB          150  TERM      EQU    X'BB'               COMPLEMENT OF TERM STATUS
       0000 007E          151  ENDFLG    EQU    X'7E'               END FLAG
0146R                     152            END
```

```
    QSA - 4-WIRE PROGRAMMING EXAMPLE - 16 BIT PROCESSOR       PAGE   5  12:57:05  07/19/78

    2 LINES - ZBID

ASSEMBLED BY CAL 03-066R05-00   (32-BIT)

START OPTIONS:  *NONE*

   NO CAL ERRORS
   NO CAL WARNINGS
    2 PASSES

 ABSTOP    0000 0000
 ADC       0000 0002
 BUSY      0000 0008       119     149*
 COMMON    0000 013FR       55     142*
 CONTIN    0000 00B4R       93      97*
 CONTIN1   0000 00CAR      100     103*
 CONTROL   0000 0134R       62      63      92      97     101     104     105     107     138*
 DATA      0000 0142R       96     125     145*
 DEVADR    0000 0124R       34      37      54      66      71      77      80     112     136*
 DISP      0000 0001        26*     40      42      43      44      45      82      83      84     124
 DSPLAY    0000 0090R       82*     95     122
 ENBRQ     0000 0141R      108     144*
 ENBRQS    0000 0140R       69     143*
 ENBSCH    0000 013ER       56     114     141*
 ENDFLG    0000 007E       127     151*
 FILL      0000 0145R       68     148*
 GETRTIN   0000 009CR       78      87*
 HALT      0000 0120R       85     134*
 HALT1     0000 0098R       85*    129     134
 IDEV      0000 0003        28*     75      77      84      96     108     123
 IDLE      0000 0143R       35     146*
 IDX       0000 0005        30*     33      33      34      36      37      51      51      53      54      57      59      62
                            63      66      70      71      76      76      77      79      80      87      92      97     101
                           104     105     107     112
 IMPTOP    0000 0146R
 INCR      0000 013DR       42      82     140*
 INTHDL    0000 007ER       49      75*
 ISTA      0000 0004        29*     75      83      94      94     119     121
 LADC      000C 0001
 MATCH     0000 0082R       77*     81
 NORM      0000 013CR       45     139*
 PURETOP   0000 0000R
 READ      0000 00F2R       52     118*
 SETUP     0000 0002R       34*     38
 SETVET    0000 0038R       52*     72
 TERM      0000 00BB       121     150*
 VECTOR    0000 012CR       37      53      59      71      80      87     137*
 WAIT      0000 011CR       73     133*
 WAIT1     0000 007AR       73*    102     110     116     120     126     128     133
 WORK      0000 0002        27*     34      35      41      41      43      44      46      47      48      49      50      52
                            53      58      59      61      61      62      63      87      88      92      97      98      99
                           101     103     103     104     105     106     107     109     109     112     114     123     124
                           125     127
 WORK1     0000 0006        31*     54      55      56      66      67      68      69
 WRITE     0000 00A2R       58      91*
 WRMOD     0000 0144R       67     147*
```

APPENDIX C
32 BIT PROGRAMMING EXAMPLES

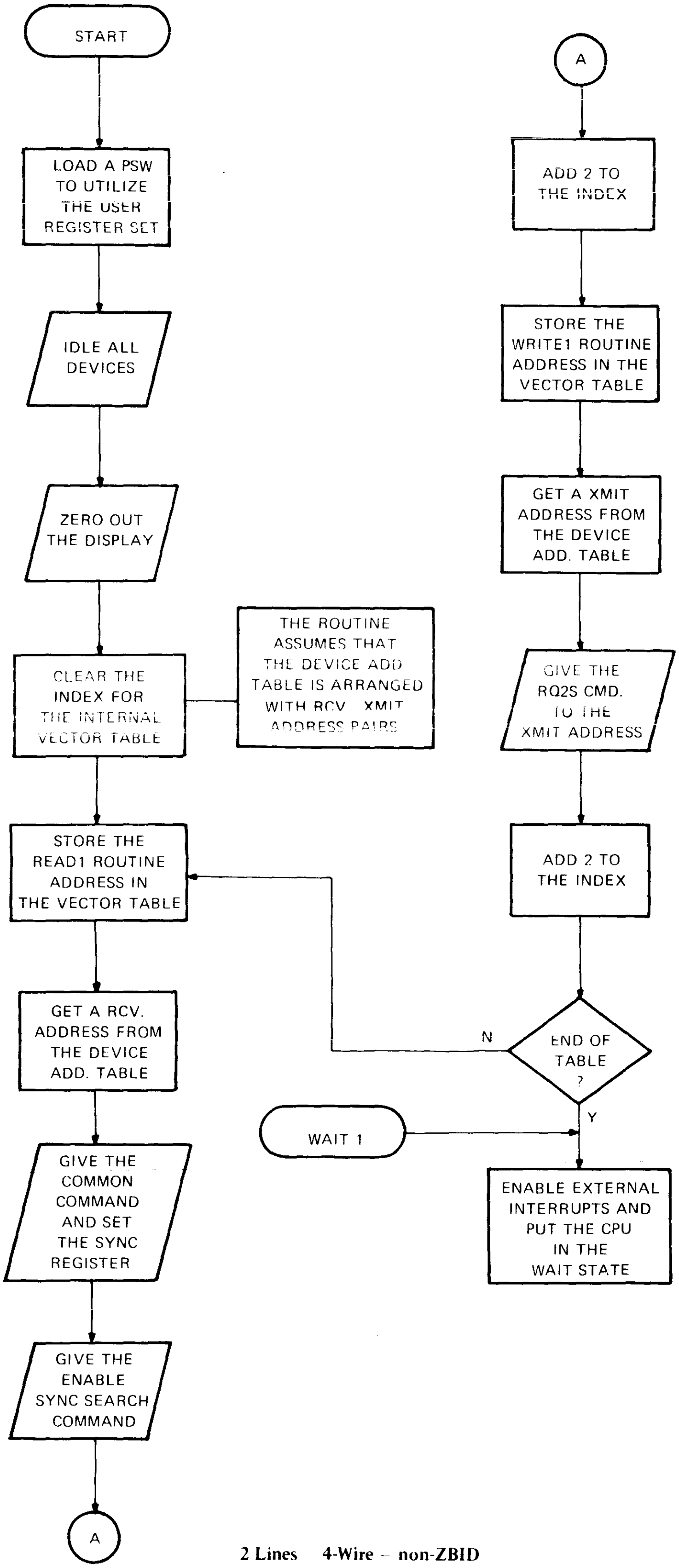

START
LOAD A PSW TO UTILIZE THE USER REGISTER SET
SET UP THE INTERRUPT VECTOR TABLE
ZERO OUT THE DISPLAY
SET THE SYNC FLAG IN MEMORY
GIVE THE COMMON COMMAND AND SET SYNC MATCH REG.
ENABLE INT. AND START SYNC. SEARCH ON RCV SIDE
ENABLE INT. AND SET RQ2S ON XMIT SIDE
WAIT1
ENABLE EXTERNAL INTERRUPTS AND PUT THE CPU IN THE WAIT STATE
DISPLY
WRITE THE DEVICE ADD. AND STATUS TO THE DISPLAY
HALT1
SET THE XMIT AND RCV. DEVICES TO IDLE
HALT THE CPU

2-Wire non-ZBID



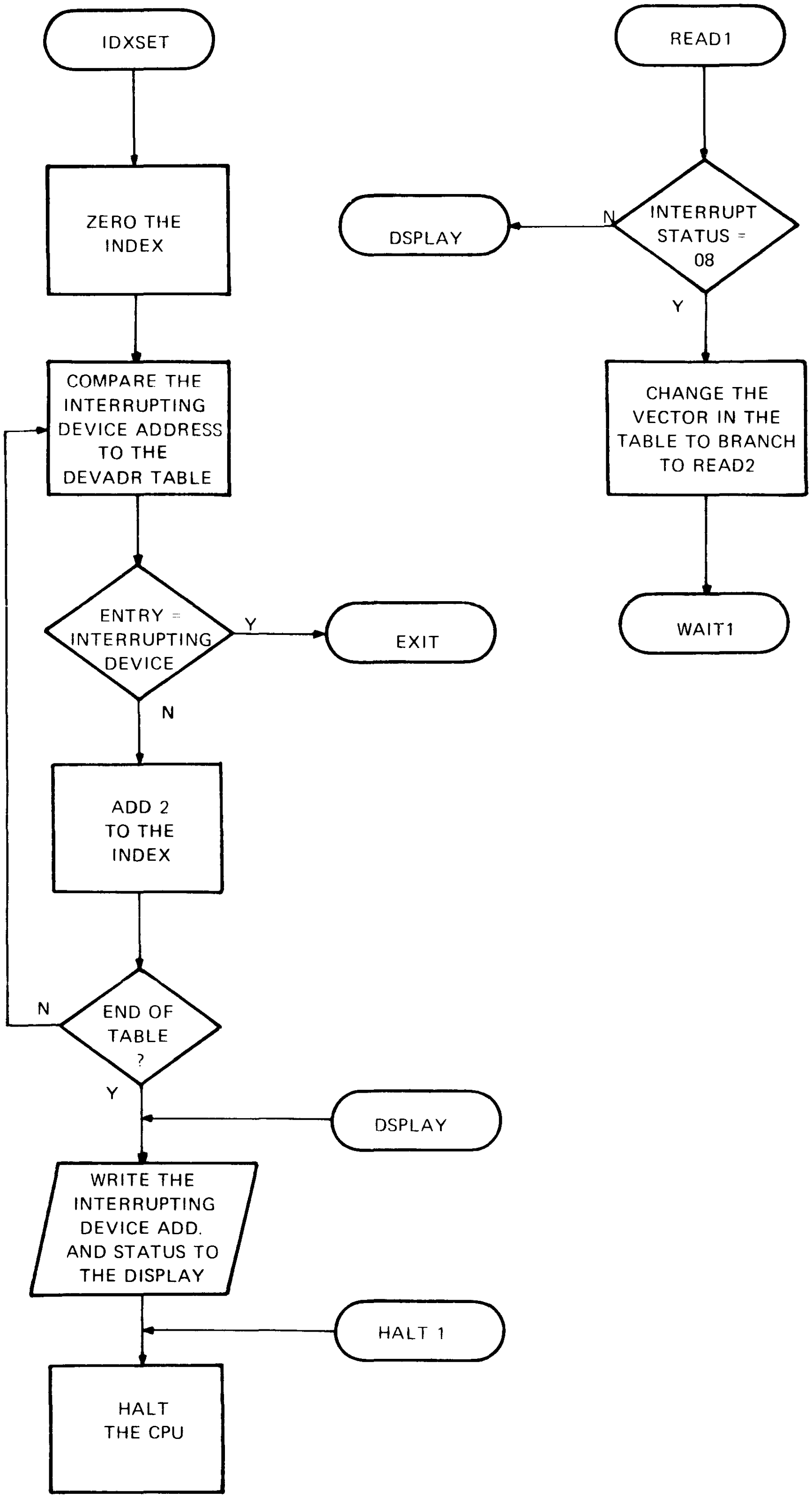
2-Wire — non-ZBID

APPENDIX C (Continued)

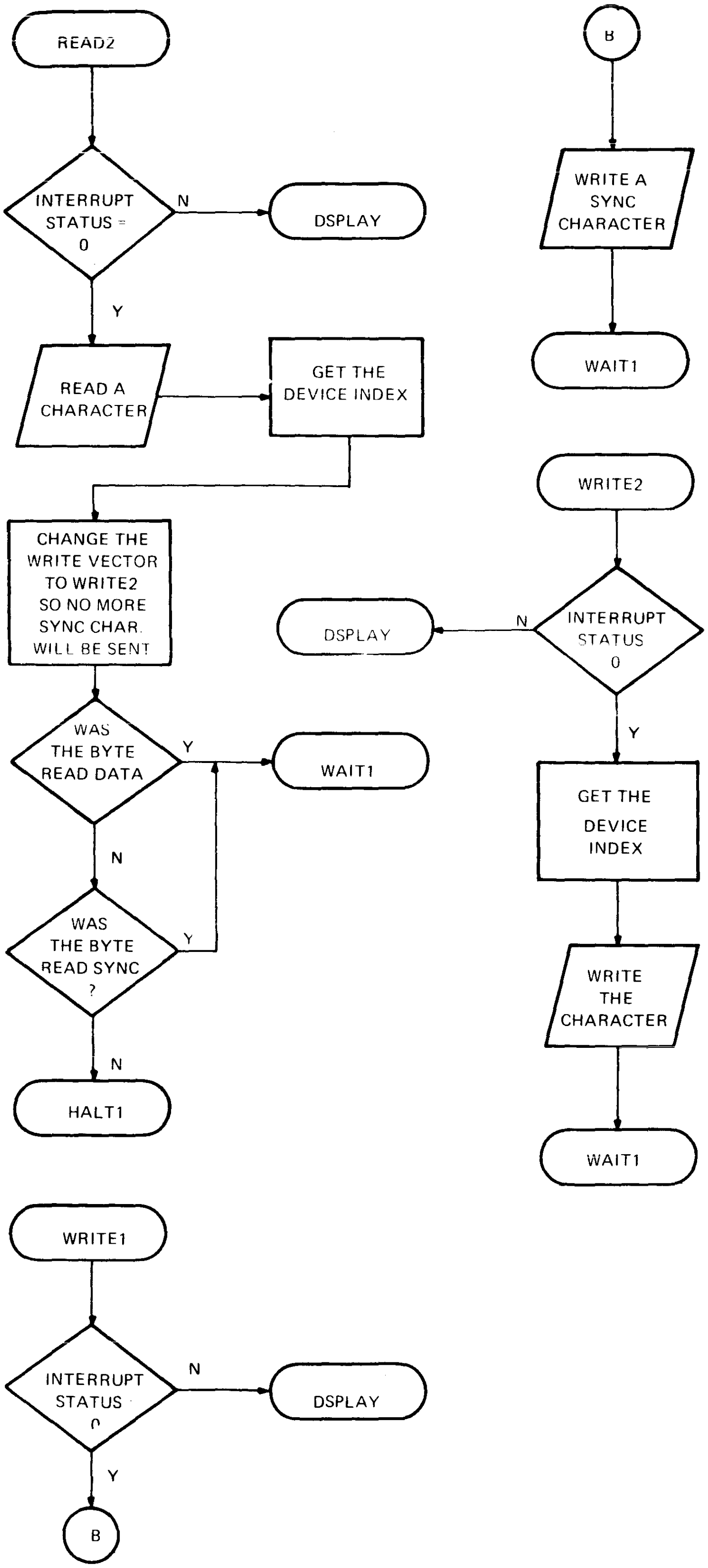

START
LOAD A PSW TO UTILIZE THE USER REGISTER SET
IDLE ALL DEVICES
ZERO OUT THE DISPLAY
CLEAR THE INDEX FOR THE INTERNAL VECTOR TABLE
THE ROUTINE ASSUMES THAT THE DEVICE ADD TABLE IS ARRANGED WITH RCV XMIT ADDRESS PAIRS
STORE THE READ1 ROUTINE ADDRESS IN THE VECTOR TABLE
GET A RCV. ADDRESS FROM THE DEVICE ADD. TABLE
GIVE THE COMMON COMMAND AND SET THE SYNC REGISTER
GIVE THE ENABLE SYNC SEARCH COMMAND
A
A
ADD 2 TO THE INDEX
STORE THE WRITE1 ROUTINE ADDRESS IN THE VECTOR TABLE
GET A XMIT ADDRESS FROM THE DEVICE ADD. TABLE
GIVE THE RQ2S CMD. TO THE XMIT ADDRESS
ADD 2 TO THE INDEX
END OF TABLE ?
N
Y
WAIT 1
ENABLE EXTERNAL INTERRUPTS AND PUT THE CPU IN THE WAIT STATE

2 Lines 4-Wire – non-ZBID

APPENDIX C (Continued)

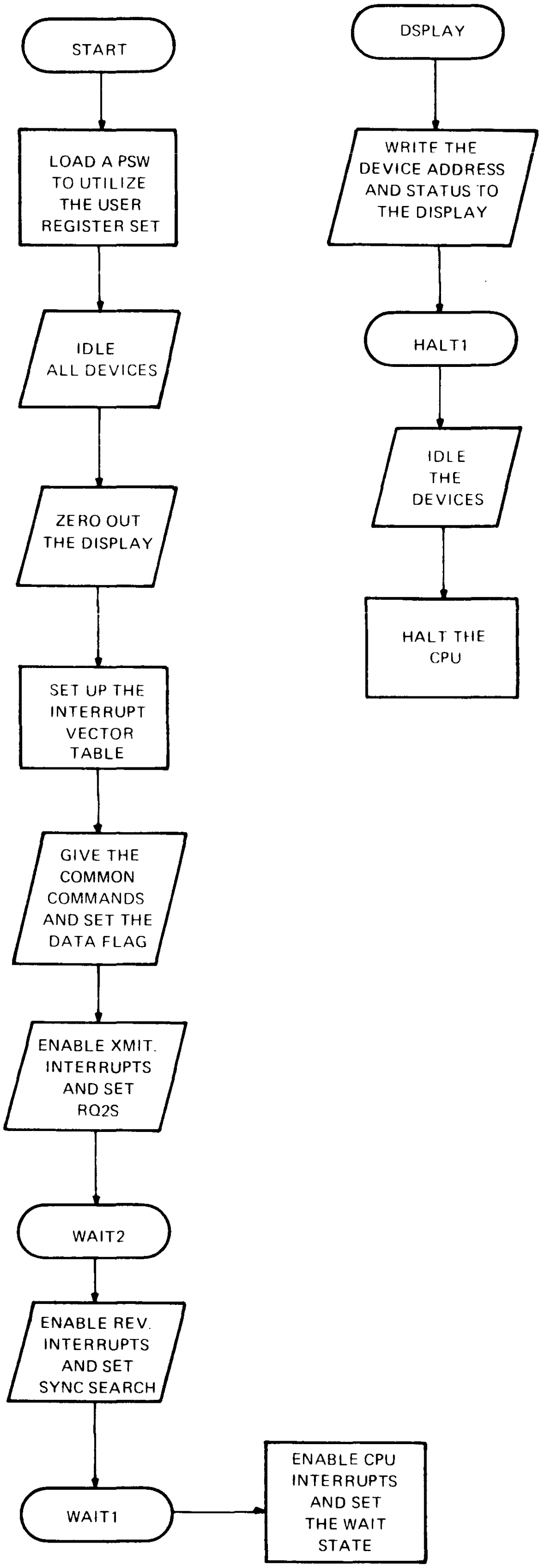

IDXSET
ZERO THE INDEX
COMPARE THE INTERRUPTING DEVICE ADDRESS TO THE DEVADR TABLE
ENTRY = INTERRUPTING DEVICE
Y
EXIT
N
ADD 2 TO THE INDEX
END OF TABLE ?
N
Y
DSPLAY
WRITE THE INTERRUPTING DEVICE ADD. AND STATUS TO THE DISPLAY
HALT 1
HALT THE CPU
READ1
INTERRUPT STATUS = 08
N
DSPLAY
Y
CHANGE THE VECTOR IN THE TABLE TO BRANCH TO READ2
WAIT1

2 Lines – 4-Wire non-ZBID

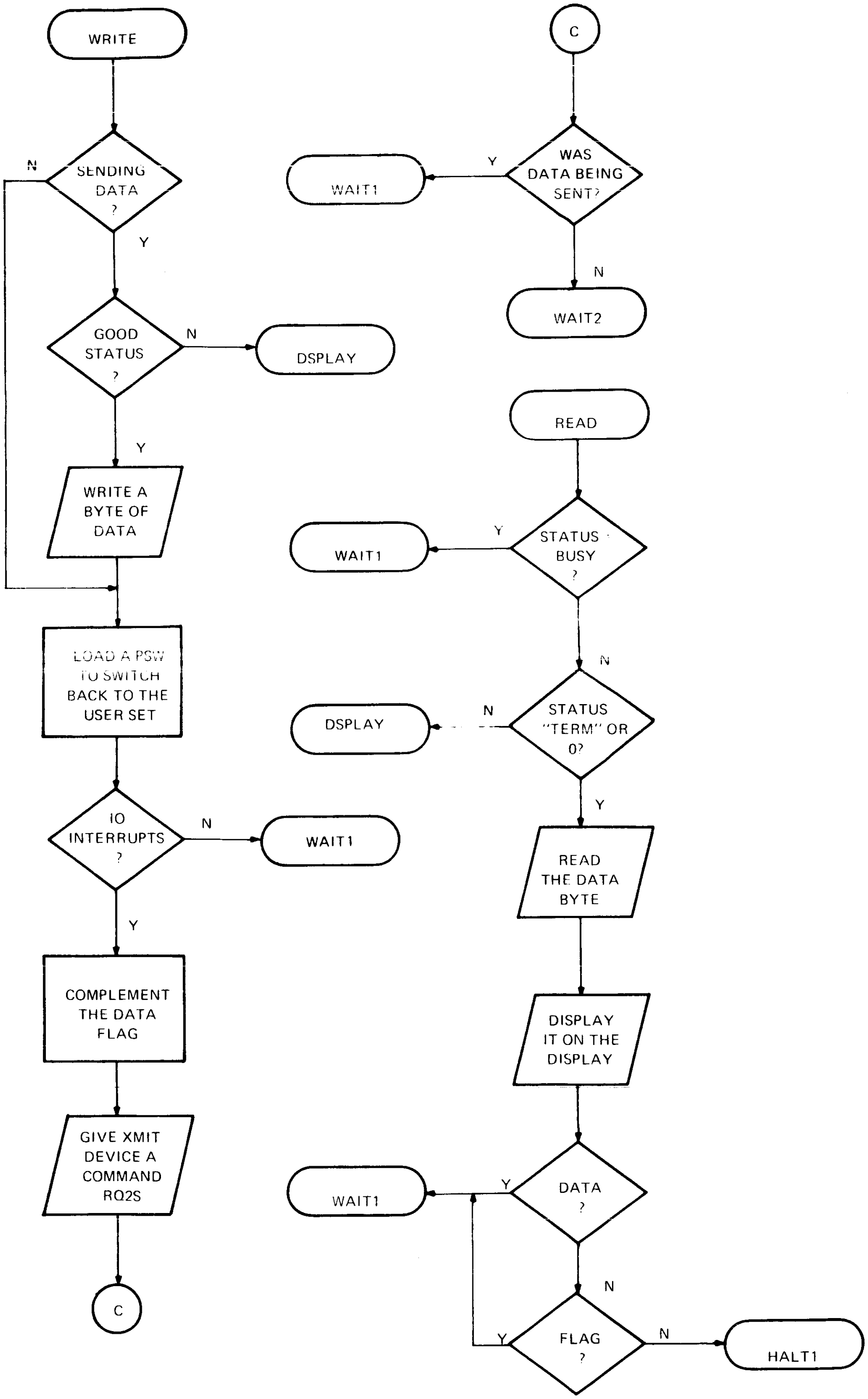


2 Lines – 4-Wire – non-ZBID

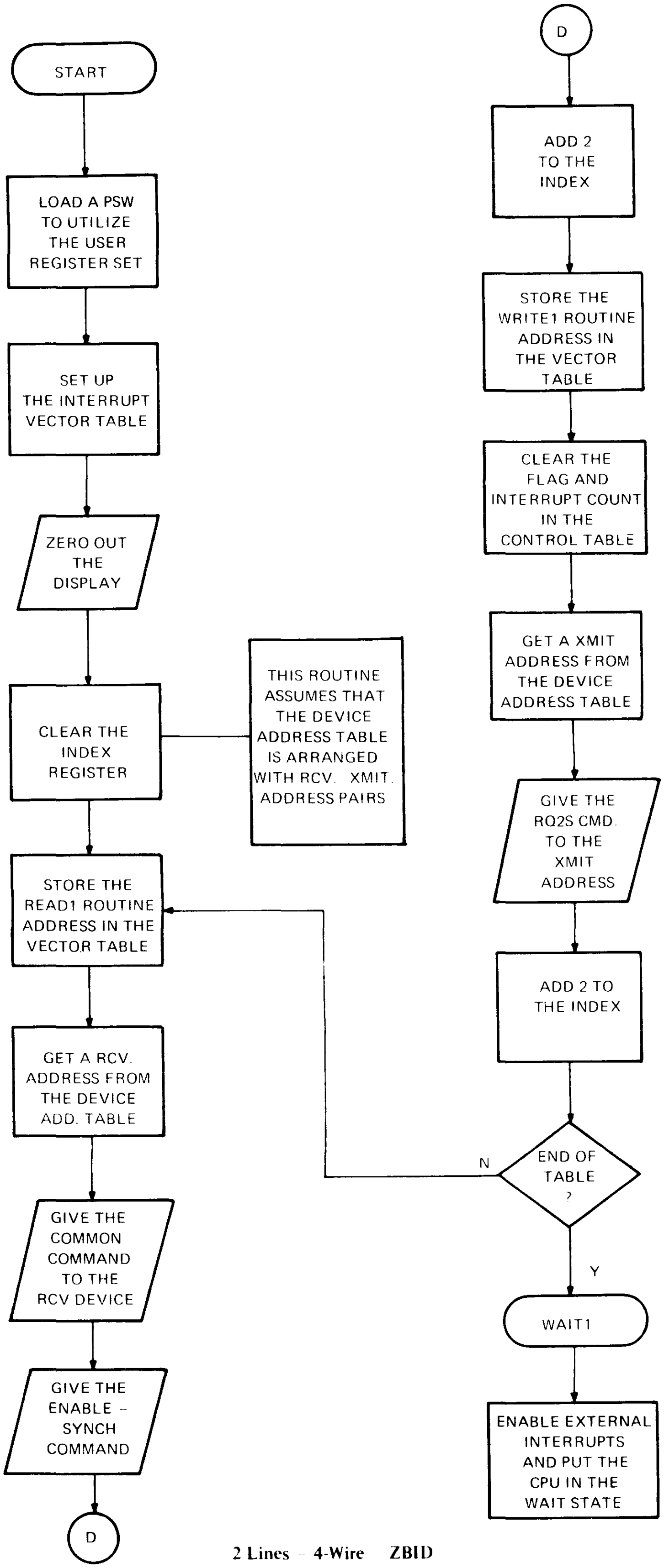
APPENDIX C (Continued)


START
LOAD A PSW TO UTILIZE THE USER REGISTER SET
IDLE ALL DEVICES
ZERO OUT THE DISPLAY
SET UP THE INTERRUPT VECTOR TABLE
GIVE THE COMMON COMMANDS AND SET THE DATA FLAG
ENABLE XMIT. INTERRUPTS AND SET RQ2S
WAIT2
ENABLE REV. INTERRUPTS AND SET SYNC SEARCH
WAIT1
ENABLE CPU INTERRUPTS AND SET THE WAIT STATE
DSPLAY
WRITE THE DEVICE ADDRESS AND STATUS TO THE DISPLAY
HALT1
IDLE THE DEVICES
HALT THE CPU

2-Wire ZBID

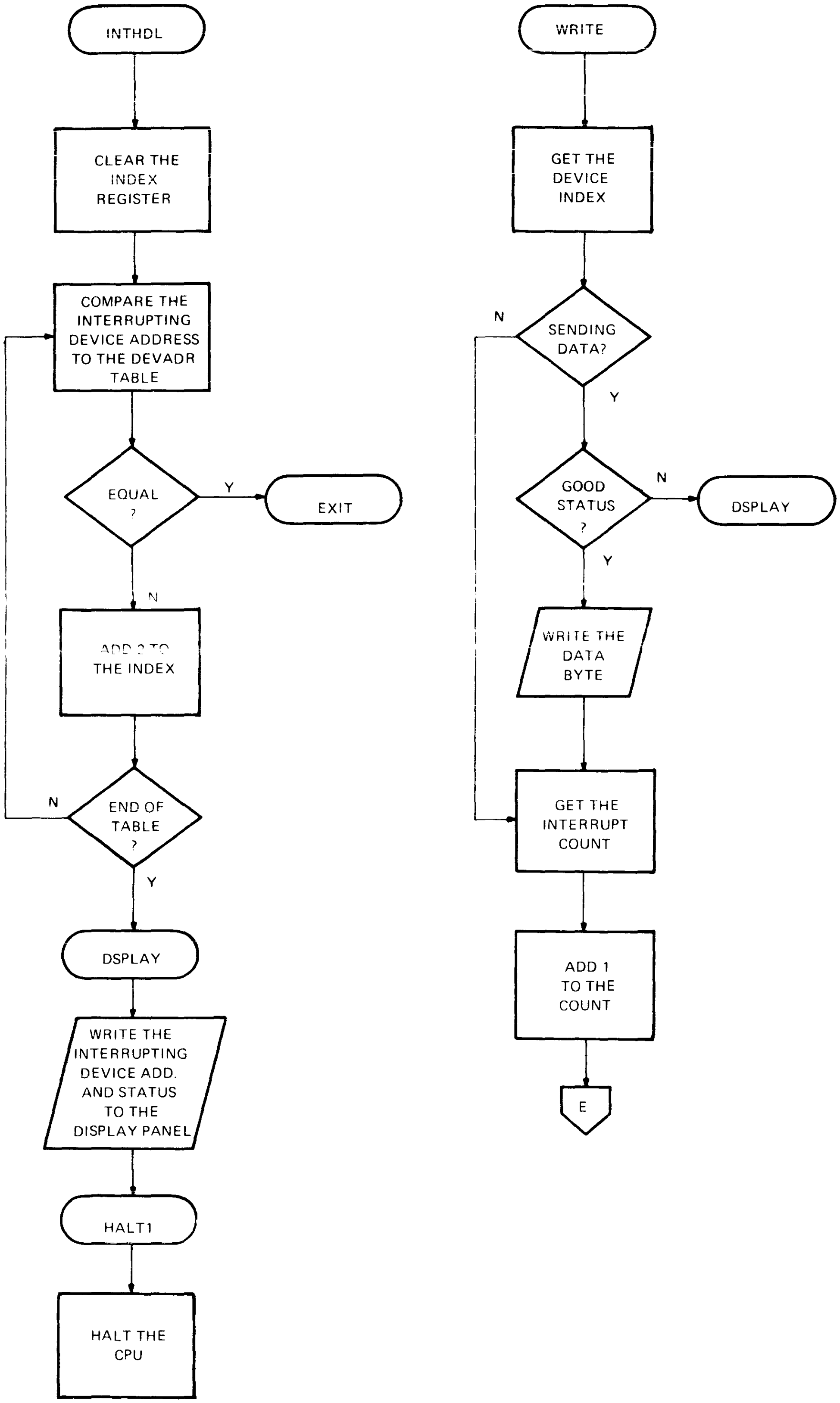

WRITE
SENDING DATA ?
N
Y
GOOD STATUS ?
N
DSPLAY
Y
WRITE A BYTE OF DATA
LOAD A PSW TO SWITCH BACK TO THE USER SET
IO INTERRUPTS ?
N
WAIT1
Y
COMPLEMENT THE DATA FLAG
GIVE XMIT DEVICE A COMMAND RQ2S
C
C
WAS DATA BEING SENT?
Y
WAIT1
N
WAIT2
READ
STATUS BUSY ?
Y
WAIT1
N
STATUS "TERM" OR 0?
N
DSPLAY
Y
READ THE DATA BYTE
DISPLAY IT ON THE DISPLAY
DATA ?
Y
WAIT1
N
FLAG ?
Y
N
HALT1

2-Wire ZBID


START
LOAD A PSW TO UTILIZE THE USER REGISTER SET
SET UP THE INTERRUPT VECTOR TABLE
ZERO OUT THE DISPLAY
CLEAR THE INDEX REGISTER
THIS ROUTINE ASSUMES THAT THE DEVICE ADDRESS TABLE IS ARRANGED WITH RCV. XMIT. ADDRESS PAIRS
STORE THE READ1 ROUTINE ADDRESS IN THE VECTOR TABLE
GET A RCV. ADDRESS FROM THE DEVICE ADD. TABLE
GIVE THE COMMON COMMAND TO THE RCV DEVICE
GIVE THE ENABLE - SYNCH COMMAND
D
D
ADD 2 TO THE INDEX
STORE THE WRITE1 ROUTINE ADDRESS IN THE VECTOR TABLE
CLEAR THE FLAG AND INTERRUPT COUNT IN THE CONTROL TABLE
GET A XMIT ADDRESS FROM THE DEVICE ADDRESS TABLE
GIVE THE RQ2S CMD. TO THE XMIT ADDRESS
ADD 2 TO THE INDEX
END OF TABLE ?
N
Y
WAIT1
ENABLE EXTERNAL INTERRUPTS AND PUT THE CPU IN THE WAIT STATE

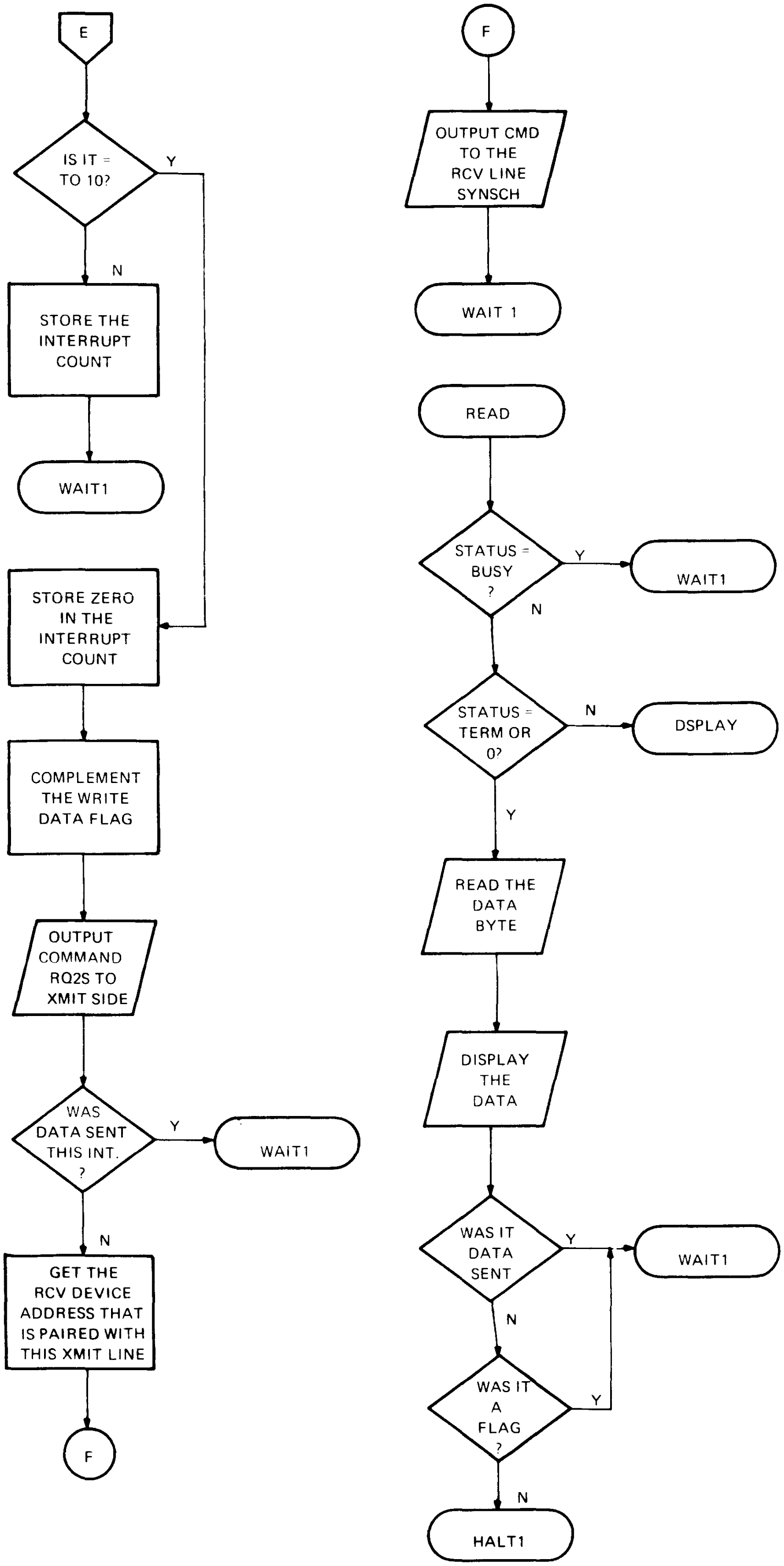
2 Lines – 4-Wire ZBID

APPENDIX C (Continued)



2 Lines - 4-Wire - ZBID

APPENDIX C (Continued)


E
IS IT = TO 10?
Y
N
STORE THE INTERRUPT COUNT
WAIT1
STORE ZERO IN THE INTERRUPT COUNT
COMPLEMENT THE WRITE DATA FLAG
OUTPUT COMMAND RQ2S TO XMIT SIDE
WAS DATA SENT THIS INT. ?
Y
WAIT1
N
GET THE RCV DEVICE ADDRESS THAT IS PAIRED WITH THIS XMIT LINE
F
F
OUTPUT CMD TO THE RCV LINE SYNSCH
WAIT 1
READ
STATUS = BUSY ?
Y
WAIT1
N
STATUS = TERM OR 0?
N
DSPLAY
Y
READ THE DATA BYTE
DISPLAY THE DATA
WAS IT DATA SENT
Y
WAIT1
N
WAS IT A FLAG ?
Y
N
HALT1

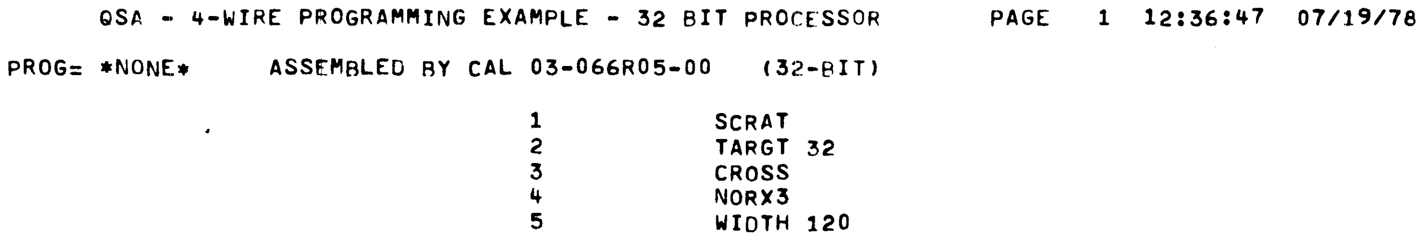
2 Lines – 4-Wire – ZBID

```
      QSA - 2-WIRE PROGRAMMING EXAMPLE - 32 BIT PROCESSOR       PAGE   1  12:20:27  07/19/78
PROG= *NONE*     ASSEMBLED BY CAL 03-066R05-00   (32-BIT)
                                        1             SCRAT
                                        2             CROSS
                                        3             WIDTH 120
                                        4             TARGT 32
                                        5             NORX3
```

```
QSA - 2-WIRE PROGRAMMING EXAMPLE - 32 BIT PROCESSOR       PAGE   2  12:20:27  07/19/78

NON ZBID

                                 *
                                 * ************************************************************
                                 * THIS PROGRAM ILLUSTRATES A DATA TRANSFER BETWEEN TWO LINES ON  *
                                 * THE QSA UNDER LOCAL LOOPBACK.  2 - WIRE OPERATION IS ASSUMED.
                                 * A DATA BYTE IS TRANSMITTED CONTINUOUSLY, AND RECEIVED AND      *
                                 * COMPARED ON THE RECEIVE END. THE BYTE IS DISPLAYED ON THE      *
                                 * DISPLAY PANEL. IF A STATUS OR DEVICE NUMBER ERROR IS DETECTED  *
                                 * THE PROCESSOR HALTS WITH THE DEVICE NUMBER AND THE STATUS      *
                                 * DISPLAYED ON THE PANEL. IF THE ERROR OCCURS ON THE DATA, ONLY  *
                                 * THE DATA IS DISPLAYED. PROGRAM IS EXECUTED UNDER INTERRUPT.    *
                                 * ************************************************************
                                 *                                                                *
          0000 0002              IDEV     EQU   2                      INTERRUPTING DEVICE ADDRESS
          0000 0003              ISTA     EQU   3                      INTERRUPTING DEVICE STATUS
          0000 000B              WORK1    EQU   11                     WORK REGISTER
          0000 000C              DISP     EQU   12                     DISPLAY ADDRESS REGISTER
          0000 000D              WORK     EQU   13                     WORK REGISTER
          0000 000E              RDEV     EQU   14                     RECV DEVICE ADDRESS REG
          0000 000F              XDEV     EQU   15                     XMIT DEVICE ADDRESS REG
          0000 0008              BUSY     EQU   8                      BUSY STATUS
                                 *
000000I   F8D0 0000 00F0                  LI    WORK,Y'00F0'           SET REG. SET F PSW
000006I   95BD                            EPSR  WORK1,WORK             CHANGE THE PSW
000008I   48E0 80C8 =0000D4I              LH    RDEV,RADR              GET RECV DEVICE ADDRESS
00000CI   DEE0 80CC =0000DCI              OC    RDEV,IDLE              IDLE THE RECEIVE DEVICE
000010I   E6D0 8080 =000094I              LA    WORK,READ              LOAD THE READ INTERRUPT ADDRESS
000014I   40DE 4E00 00D0                  STH   WORK,X'D0'(RDEV,RDEV) PUT THE ADD. IN THE VECTOR TABLE
00001AI   48F0 80B8 =0000D6I              LH    XDEV,XADR              GET XMIT DEVICE ADDRESS
00001EI   DEF0 80BA =0000DCI              OC    XDEV,IDLE              IDLE THE TRANSMIT DEVICE
000022I   E6D0 804E =000074I              LA    WORK,WRITE             LOAD THE WRITE INTERRUPT ADDRESS
000026I   40DF 4F00 00D0                  STH   WORK,X'D0'(XDEV,XDEV) PUT THE ADD. IN THE VECTOR TABLE
00002CI   24C1                            LIS   DISP,1                 DISPLAY PANEL ADDRESS
00002EI   07DD                            XR    WORK,WORK
000030I   DEC0 80A4 =0000D8I              OC    DISP,INCR              SET DISPLAY TO INCREMENT MODE
000034I   9ACD                            WDR   DISP,WORK              CLEAR DISPLAY
000036I   9ACD                            WDR   DISP,WORK
000038I   9ACD                            WDR   DISP,WORK
00003AI   9ACD                            WDR   DISP,WORK
00003CI   DEC0 8099 =0000D9I              OC    DISP,NORM              SET DISPLAY BACK TO NORMAL MODE
000040I   C8D0 00FF                       LHI   WORK,X'FF'             SET SYNC FLAG
000044I   40D0 8088 =0000D0I              STH   WORK,SYNFLG
000048I   DEE0 8091 =0000DDI              OC    RDEV,COMMON            SET LINE MODE
00004CI   DEF0 808D =0000DDI              OC    XDEV,COMMON
                                 *
                                 *  NOTE :
                                 *
                                 *     IF THE NEXT TWO OUTPUT COMMANDS ARE BEING ISSUED TO
                                 *     TURN A 2-WIRE LINE AROUND (FROM WRITE TO READ OR READ TO
                                 *     WRITE) THE COMMANDS MUST BE ISSUED TWICE.  FIRST WITH
                                 *     INTERRUPTS DISARMED, AND THEN RE-ISSUED WITH THE DESIRED
                                 *     INTERRUPT CONDITION, UNLESS THE DISARM CONDITION IS DESIRED.
                                 *
000050I   DEE0 8086 =0000DAI              OC    RDEV,ENBSCH            RECV ENABLE, SYNCSH
```

```
          QSA - 2-WIRE PROGRAMMING EXAMPLE - 32 BIT PROCESSOR       PAGE   3  12:20:27  07/19/78

          NON ZBID

000054I  DAE0 8086 =0000DEI       60              WD     RDEV,SYNC          SET SYNC MATCH REGISTER
000058I  DEF0 807F =0000DBI       61              OC     XDEV,ENBRQS        XMIT ENABLE, RQ2S
00005CI  C200 8060 =0000C0I       62  WAIT1       LPSW   WAIT
                                  63  *
000060I  DEC0 8074 =0000D8I       64  DISPLY      OC     DISP,INCR          NEITHER -
000064I  9AC3                     65              WDR    DISP,ISTA          DISPLAY STATUS
000066I  9AC2                     66              WDR    DISP,IDEV          DISPLAY DEVICE ADDRESS
000068I  DEE0 8070 =0000DCI       67  HALT1       OC     RDEV,IDLE          IDLE LINES
00006CI  DEF0 806C =0000DCI       68              OC     XDEV,IDLE
000070I  C200 8054 =0000C8I       69              LPSW   HALT               HALT PROCESSOR
                                  70  *
         0000 0074I               71  WRITE       EQU    *
000074I  24C1                     72              LIS    DISP,1             LOAD THE DISPLAY ADDRESS
000076I  0833                     73              LR     ISTA,ISTA          SET THE CONDITION CODE
000078I  4230 FFE4 =000060I       74              BNZ    DISPLY             EXIT IF STATUS NOT ZERO
00007CI  48D0 8050 =0000D0I       75              LH     WORK,SYNFLG        SYNC FLAG SET?
000080I  4330 8008 =00008CI       76              BZ     WRTDAT             NO - SEND DATA
000084I  DA20 8056 =0000DEI       77              WD     IDEV,SYNC          YES - SEND SYNC
000088I  4300 FFD0 =00005CI       78              B      WAIT1
00008CI  DA20 804F =0000DFI       79  WRTDAT      WD     IDEV,DATA          SEND DATA
000090I  4300 FFC8 =00005CI       80              B      WAIT1
                                  81  *
000094I  C530 0008                82  READ        CLHI   ISTA,BUSY          READ ROUTINE
000098I  4330 FFC0 =00005CI       83              BE     WAIT1              WAIT IF CARRIER INTERRUPT
00009CI  24C1                     84              LIS    DISP,1             LOAD THE DISPLAY PANEL ADDRESS
00009EI  0833                     85              LR     ISTA,ISTA
0000A0I  4230 FFBC =000060I       86              BNZ    DISPLY             EXIT IF STATUS NOT ZERO
0000A4I  9B2D                     87              RDR    IDEV,WORK          READ THE DATA BYTE
0000A6I  4030 8026 =0000D0I       88              STH    ISTA,SYNFLG        RESET SYNC FLAG
0000AAI  9ACD                     89              WDR    DISP,WORK          DISPLAY DATA READ
0000ACI  D4D0 802F =0000DFI       90              CLB    WORK,DATA          IS IT DATA SENT?
0000B0I  4330 FFA8 =00005CI       91              BE     WAIT1
0000B4I  D4D0 8026 =0000DEI       92              CLB    WORK,SYNC          NO - IS IT SYNC?
0000B8I  4230 FFAC =000068I       93              BNE    HALT1              NO - HALT
0000BCI  4300 FF9C =00005CI       94              B      WAIT1
                                  95  *
0000C0I                           96              ALIGN  8
0000C0I  0000 C0F0                97  WAIT        DC     Y'C0F0',A(WAIT1)
0000C4I  0000 005CI
0000C8I  0000 80F0                98  HALT        DC     Y'80F0',A(HALT1)
0000CCI  0000 0068I
0000D0I  0000 0000                99  SYNFLG      DC     0
0000D4I  00B8                    100  RADR        DC     Y'B8'              RECV LINE ADDRESS
0000D6I  00BC                    101  XADR        DC     Y'BC'              XMIT LINE ADDRESS
0000D8I  40                      102  INCR        DB     X'40'              INCREMENT
0000D9I  80                      103  NORM        DB     X'80'              NORMAL
0000DAI  79                      104  ENBSCH      DB     X'79'              ENABLE, LOCAL, SYNSCH
0000DBI  7B                      105  ENBRQS      DB     X'7B'              ENABLE,LOCAL,RESET DATA MODE,RQ2S
0000DCI  F1                      106  IDLE        DB     X'F1'              DISARM, IDLE
0000DDI  30                      107  COMMON      DB     X'30'              8 BITS/CHAR, NO PARITY
0000DEI  16                      108  SYNC        DB     X'16'              SYNC CHARACTER
0000DFI  55                      109  DATA        DB     X'55'              DATA CHARACTER
0000E0I                          110              END
```

```
     QSA - 2-WIRE PROGRAMMING EXAMPLE - 32 BIT PROCESSOR       PAGE   4  12:20:27  07/19/78

     NON ZRID

ASSEMBLED BY CAL 03-066R05-00   (32-BIT)

START OPTIONS:  *NONE*

   NO CAL ERRORS
   NO CAL WARNINGS
    2 PASSES

 ABSTOP    0000 0000
 ADC       0000 0004
 BUSY      0000 0008          26*    82
 COMMON    0000 00DDI         48     49     107*
 DATA      0000 00DFI         79     90     109*
 DISP      0000 000C          22*    38     40     41     42     43     44     45     64     65     66     72     84
                              89
 DISPLY    0000 0060I         64*    74     86
 ENBRQS    0000 00DBI         61     105*
 ENBSCH    0000 00DAI         59     104*
 HALT      0000 00C8I         69     98*
 HALT1     0000 0068I         67*    93     98
 IDEV      0000 0002          19*    66     77     79     87
 IDLE      0000 00DCI         31     35     67     68     106*
 IMPTOP    0000 00E0I
 INCR      0000 00D8I         40     64     102*
 ISTA      0000 0003          20*    65     73     73     82     85     85     88
 LADC      0000 0002
 NORM      0000 00D9I         45     103*
 PURETOP   0000 0000P
 RADR      0000 00D4I         30     100*
 RDEV      0000 000E          24*    30     31     33     33     48     59     60     67
 READ      0000 0094I         32     82*
 SYNC      0000 00DEI         60     77     92     108*
 SYNFLG    0000 00D0I         47     75     88     99*
 WAIT      0000 00C0I         62     97*
 WAIT1     0000 005CI         62*    78     80     83     91     94     97
 WORK      0000 000D          23*    28     29     32     33     36     37     39     39     41     42     43     44
                              46     47     75     87     89     90     92
 WORK1     0000 000B          21*    29
 WRITE     0000 0074I         36     71*
 WRTDAT    0000 008CI         76     79*
 XADR      0000 00D6I         34     101*
 XDEV      0000 000F          25*    34     35     37     37     49     61     68
```



```
          QSA - 4-WIRE PROGRAMMING EXAMPLE - 32 BIT PROCESSOR       PAGE   1  12:36:47  07/19/78
PROG= *NONE*     ASSEMBLED BY CAL 03-066R05-00   (32-BIT)
                                  1          SCRAT
                                  2          TARGT 32
                                  3          CROSS
                                  4          NORX3
                                  5          WIDTH 120
```

```
QSA - 4-WIRE PROGRAMMING EXAMPLE - 32 BIT PROCESSOR      PAGE   2  12:36:47  07/19/78

2 LINES - NON ZBID

                                 *
                                 *
                                 * ****************************************************************
                                 *
                                 * THIS SAMPLE PROGRAM ILLUSTRATES A DATA TRANSFER BETWEEN TWO     *
                                 * 4-WIRE LINES RUNNING UNDER LOCAL LOOPBACK ON
                                 * THE QSA. A VECTOR TABLE IS SET UP TO HANDLE INTERRRUPTS FROM    *
                                 * FOUR DIFFERENT DEVICES THE SAME DATA BYTE IS TRANSMITTED ON     *
                                 * BOTH LINES AND IS DISPLAYED ON THE DISPLAY PANEL. IF A DATA     *
                                 * ERROR IS DETECTED, THE PROGRAM HALTS WITH THE BAD DATA          *
                                 * DISPLAYED. IF A STATUS OR ADDRESS ERROR OCCURS, THE PROGRAM     *
                                 * HALTS WITH THE DEVICE ADDRESS AND STATUS DISPLAYED. PROGRAM     *
                                 * IS UNDER INTERRUPT CONTROL AND STOPS ONLY ON AN ERROR.          *
                                 * ****************************************************************
                                 *
         0000 0002               IDEV     EQU    2
         0000 0003               ISTA     EQU    3
         0000 000A               R1       EQU    10
         0000 000B               WORK1    EQU    11
         0000 000C               DISP     EQU    12
         0000 000D               WORK     EQU    13
         0000 000E               IDX      EQU    14
         0000 000F               RETN     EQU    15
         0000 0008               BUSY     EQU    8
                                 *
000000I  F8D0 0000 00F0                   LI     WORK,Y'00F0'         SET PSW FOR USER SET
000006I  95BD                             EPSR   WORK1,WORK           CHANGE THE PSW.
000008I  07EE                             XR     IDX,IDX              CLEAR THE INDEX REGISTER
00000AI  48DE 80F6 =000104I      SETUP    LH     WORK,DEVADR(IDX)     GET A DEVICE ADDRESS
00000EI  DED0 80FE =000110I               OC     WORK,IDLE            IDLE THE DEVICE
000012I  26E2                             AIS    IDX,2                INCREMENT THE INDEX
000014I  C5E0 0008                        CLHI   IDX,DEVEND-DEVADR    FINISHED ?
000018I  2037                             BNES   SETUP                NO - CONTINUE
00001AI  24C1                             LIS    DISP,1               DISPLAY PANEL ADDRESS
00001CI  07DD                             XR     WORK,WORK
00001EI  DFC0 80EB =00010DI               OC     DISP,INCR            SET DISPLAY TO INCREMENT MODE
000022I  9ACD                             WDR    DISP,WORK            CLEAR DISPLAY
000024I  9ACD                             WDR    DISP,WORK
000026I  9ACD                             WDR    DISP,WORK
000028I  9ACD                             WDR    DISP,WORK
00002AI  DFC0 80DE =00010CI               OC     DISP,NORM            SET DISPLAY BACK TO NORMAL MODE
00002EI  07EE                             XR     IDX,IDX              CLEAR THE INDEX
000030I  48AE 80D0 =000104I      SETVT    LH     R1,DEVADR(IDX)       GET THE RECEIVE DEVICE ADDRESS
000034I  C8D0 0088I                       LHI    WORK,READ1           SET THE VECTOR ADDRESS
000038I  40DA 4A00 00D0                   STH    WORK,X'D0'(R1,R1)    IN THE VECTOR TABLE
00003EI  DEA0 80CF =000111I               OC     R1,COMMON            SET THE COMMON COMMAND
000042I  DEA0 80C8 =00010EI               OC     R1,ENBSCH            RECV ENABLE, SYNSCH
000046I  DAA0 80C8 =000112I               WD     R1,SYNC              SET THE SYNC MATCH REGISTER
00004AI  26E2                             AIS    IDX,2                INCREMENT THE INDEX
00004CI  48AE 80B4 =000104I               LH     R1,DEVADR(IDX)       GET THE TRANSMIT DEVICE ADDRESS
000050I  C8D0 00D0I                       LHI    WORK,WRITE1          SET THE VECTOR ADDRESS
000054I  40DA 4A00 00D0                   STH    WORK,X'D0'(R1,R1)    IN THE VECTOR TABLE
00005AI  DEA0 80B1 =00010FI               OC     R1,ENBRQS            XMIT ENABLE, RQ2S
```

```
QSA - 4-WIRE PROGRAMMING EXAMPLE - 32 BIT PROCESSOR       PAGE   3  12:36:47  07/19/78

2 LINES - NON ZBID

00005EI  26F2                      60          AIS    IDX,2                 INCREMENT THE INDEX
000060I  C5E0 0008                 61          CLHI   IDX,DEVEND-DEVADR     FINISHED ?
000064I  4280 FFC8 =000030I        62          BL     SETVT                 NO
000068I  C200 8084 =0000F0I        63  WAIT1   LPSW   WAIT                  YES
                                   64  *
00006CI  07EE                      65  IDXSET  XR     IDX,IDX
00006EI  452E 8092 =000104I        66  LOOK1   CLH    IDEV,DEVADR(IDX)
000072I  033F                      67          BER    RETN
000074I  26E2                      68          AIS    IDX,2
000076I  C5E0 0008                 69          CLHI   IDX,DEVEND-DEVADR
00007AI  4280 FFF0 =00006EI        70          BL     LOOK1
00007EI  24C1                      71  DSPLAY  LIS    DISP,1
000080I  9AC3                      72          WDR    DISP,ISTA
000082I  9AC2                      73          WDR    DISP,IDEV
000084I  C200 8070 =0000F8I        74  HALT1   LPSW   HALT
                                   75  *
000088I  C530 0008                 76  READ1   CLHI   ISTA,BUSY             CARRIER ON INTERRUPT ?
00008CI  4230 FFEE =00007EI        77          BNE    DSPLAY                NO - ERROR
000090I  C8D0 009EI                78          LHI    WORK,READ2            SET THE NEXT INTERRUPT
000094I  40D2 4200 00D0            79          STH    WORK,X'D0'(IDEV,IDEV) VECTOR IN THE TABLE
00009AI  4300 FFCA =000068I        80          B      WAIT1
                                   81  *
00009EI  0833                      82  READ2   LR     ISTA,ISTA             SET THE CONDITION CODE?-GOOD STATUS
0000A0I  4230 FFDA =00007EI        83          BNZ    DSPLAY                NO - ERROR
0000A4I  9B2D                      84          RDR    IDEV,WORK             READ THE DATA BYTE
0000A6I  24C1                      85          LIS    DISP,1                LOAD THE DISPLAY PANEL ADDRESS
0000A8I  9ACD                      86          WDR    DISP,WORK             WRITE THE DATA TO THE DISPLAY
                                   87  *
0000AAI  41F0 FFBE =00006CI        88          BAL    RETN,IDXSET           GET THE INTERNAL DEVICE INDEX
0000AEI  D4DE 8062 =000114I        89          CLB    WORK,DATA(IDX)        IS IT A DATA BYTE ?
0000B2I  4330 FFB2 =000068I        90          BE     WAIT1                 YES
0000B6I  D4D0 8058 =000112I        91          CLB    WORK,SYNC             NO - IS IT SYNC ?
0000BAI  4230 FFC6 =000084I        92          BNE    HALT1                 NO
0000BEI  C8D0 00DEI                93          LHI    WORK,WRITE2           YES - CHANGE THE WRITE INTERRUPT VECTOR
0000C2I  48AE 8040 =000106I        94          LH     R1,DEVADR+2(IDX)      GET THE ASSOCIATED WRITE DEVICE ADDRE  S
0000C6I  40DA 4A00 00D0            95          STH    WORK,X'D0'(R1,R1)     CHANGE THE VECTOR
0000CCI  4300 FF98 =000068I        96          B      WAIT1
0000D0I  0833                      97  WRITE1  LR     ISTA,ISTA             ZERO STATUS?
0000D2I  4230 FFA8 =00007EI        98          BNZ    DSPLAY                NO - HALT
0000D6I  DA20 8038 =000112I        99          WD     IDEV,SYNC             WRITE SYNC BYTE
0000DAI  4300 FF8A =000068I       100          B      WAIT1
                                  101  *
0000DEI  0833                     102  WRITE2  LR     ISTA,ISTA             ZERO STATUS?
0000E0I  4230 FF9A =00007EI       103          BNZ    DSPLAY                NO - HALT
0000E4I  41F0 FF84 =00006CI       104          BAL    RETN,IDXSET
0000E8I  DA2E 8028 =000114I       105          WD     IDEV,DATA(IDX)        WRITE A DATA BYTE
0000ECI  4300 FF78 =000068I       106          B      WAIT1
0000F0I                           107          ALIGN  4
0000F0I  0000 C0F0                108  WAIT    DC     Y'C0F0',A(WAIT1)
0000F4I  0000 0068I
0000F8I  0000 80F0                109  HALT    DC     Y'80F0',A(HALT1)
0000FCI  0000 0084I
000100I  0000 0000                110  SYNFLG  DC     0
```

```
        QSA - 4-WIRE PROGRAMMING EXAMPLE - 32 BIT PROCESSOR       PAGE   4  12:36:47  07/19/78

        2 LINES - NON ZBID

000104I  00B8                111  DEVADR   DC     X'B8',X'BD',X'BC',X'B9'
000106I  00BD
000108I  00BC
00010AI  00B9
         0000 010CI          112  DEVEND   EQU    *
00010CI  80                  113  NORM     DB     X'80'               DISPLAY NORMAL MODE
00010DI  40                  114  INCR     DB     X'40'               DISPLAY INCREMENT MODE
00010EI  79                  115  ENBSCH   DB     X'79'               ENABLE, LOCAL, SYNSCH
00010FI  7B                  116  ENBRQS   DB     X'7B'               ENABLE,LOCAL,RESET DATA MODE,RQ2S
000110I  F1                  117  IDLE     DB     X'F1'               DISARM, IDLE
000111I  30                  118  COMMON   DB     X'30'               8 BITS/CHAR, NO PARITY
000112I  16                  119  SYNC     DB     X'16'               SYNC CHARACTER
000114I                      120           ALIGN 4
000114I  5500                121  DATA     DC     X'5500'
000116I  5500                122           DC     X'5500'
000118I  AA00                123           DC     X'AA00'
00011AI  AA00                124           DC     X'AA00'
00011CI                      125           END
```

```
    QSA - 4-WIRE PROGRAMMING EXAMPLE - 32 BIT PROCESSOR       PAGE   5  12:36:47  07/19/78

    2 LINES - NON ZBID

ASSEMBLED BY CAL 03-066R05-00    (32-BIT)

START OPTIONS:  *NONE*

   NO CAL ERRORS
   NO CAL WARNINGS
    2 PASSES

 ABSTOP     0000 0000
 ADC        0000 0004
 BUSY       0000 0008        30*    76
 COMMON     0000 0111I       52     118*
 DATA       0000 0114I       89     105    121*
 DEVADR     0000 0104I       35     38     49     56     61     66     69     94     111*
 DEVEND     0000 010CI       38     61     69     112*
 DISP       0000 000C        26*    40     42     43     44     45     46     47     71     72     73     85     86
 DSPLAY     0000 007EI       71*    77     83     98     103
 ENBRQS     0000 010FI       59     116*
 ENBSCH     0000 010EI       53     115*
 HALT       0000 00F8I       74     109*
 HALT1      0000 0084I       74*    92     109
 IDEV       0000 0002        22*    66     73     79     79     84     99     105
 IDLE       0000 0110I       36     117*
 IDX        0000 000E        28*    34     34     35     37     38     48     48     49     55     56     60     61
                             65     65     66     68     69     89     94     105
 IDXSET     0000 006CI       65*    88     104
 IMPTOP     0000 011CI
 INCR       0000 010DI       42     114*
 ISTA       0000 0003        23*    72     76     82     82     97     97     102    102
 LADC       0000 0002
 LOOK1      0000 006EI       66*    70
 NORM       0000 010CI       47     113*
 PURETOP    0000 0000P
 R1         0000 000A        24*    49     51     51     52     53     54     56     58     58     59     94     95
                             95
 READ1      0000 0088I       50     76*
 READ2      0000 009EI       78     82*
 RETN       0000 000F        29*    67     88     104
 SETUP      0000 000AI       35*    39
 SETVT      0000 0030I       49*    62
 SYNC       0000 0112I       54     91     99     119*
 SYNFLG     0000 0100I      110*
 WAIT       0000 00F0I       63     108*
 WAIT1      0000 0068I       63*    80     90     96     100    106    108
 WORK       0000 000D        27*    32     33     35     36     41     41     43     44     45     46     50     51
                             57     58     78     79     84     86     89     91     93     95
 WORK1      0000 000B        25*    33
 WRITE1     0000 00D0I       57     97*
 WRITE2     0000 00DEI       93     102*
```

```
          QSA - 2-WIRE PROGRAMMING EXAMPLE - 32 BIT PROCESSOR       PAGE   1  12:43:46  07/19/78
     PROG= *NONE*     ASSEMBLED BY CAL 03-066R05-00   (32-BIT)
                                  1            SCRAT
                                  2            CROSS
                                  3            WIDTH 120
                                  4            TARGT 32
                                  5            NORX3
```

```
QSA - 2-WIRE PROGRAMMING EXAMPLE - 32 BIT PROCESSOR       PAGE   2  12:43:46  07/19/78

ZBID

                              *
                              * ****************************************************************
                              * THIS PROGRAM ILLUSTRATES A DATA TRANSFER BETWEEN TWO LINES     *
                              * ZBID MODE IS ASSUMED.  10 DATA BYTES ARE
                              * SENT AND FOLLOWED BY 10 FLAGS. THE FLAGS ARE SENT BY HARDWARE. *
                              * THIS PATTERN CONTINUES UNTIL AN ERROR IS DETECTED. IF A STATUS *
                              * OR ADDRESS ERROR IS DETECTED. THE PROCESSOR HALTS WITH THE     *
                              * STATUS AND DEVICE ADDRESS DISPLAYED. IF IT IS A DATA ERROR,    *
                              * ONLY THE DATA BYTE IS DISPLAYED. PROGRAM IS EXECUTED UNDER     *
                              * INTERRUPT.                                                     *
                              * ****************************************************************
                              *
         0000 0002            IDEV     EQU   2
         0000 0003            ISTA     EQU   3
         0000 0008            WORK1    EQU   8
         0000 0009            DISP     EQU   9
         0000 000A            WORK     EQU   10
         0000 000B            RDEV     EQU   11
         0000 000C            XDEV     EQU   12
         0000 000D            COUNT    EQU   13
         0000 000E            R1       EQU   14
         0000 000F            R2       EQU   15
         0000 0008            BUSY     EQU   8                   BUSY
         0000 00BB            TERM     EQU   X'BB'               INVERSE OF TERM STATUS
         0000 007E            ENDFLG   EQU   X'7E'               END FLAG
                              *
000000I  F8A0 0000 00F0                LI    WORK,Y'00F0'        SET REGISTER SET "F" PSW.
000006I  958A                          EPSR  WORK1,WORK          CHANGE THE PSW.
000008I  48B0 80E8 =0000F4I            LH    RDEV,RADR           GET RECV DEVICE ADDRESS
00000CI  DEB0 80FE =0000FEI            OC    RDEV,IDLE
000010I  C8A0 00B6I                    LHI   WORK,READ           SET THE READ INTERRUPT
000014I  40AB 4B00 00D0                STH   WORK,X'D0'(RDEV,RDEV) VECTOR
00001AI  48C0 80D8 =0000F6I            LH    XDEV,XADR           GET XMIT DEVICE ADDRESS
00001EI  DEC0 80DC =0000FEI            OC    XDEV,IDLE
000022I  C8A0 007CI                    LHI   WORK,WRITE          SET THE WRITE INTERRUPT
000026I  40AC 4C00 00D0                STH   WORK,X'D0'(XDEV,XDEV) VECTOR
00002CI  2491                          LIS   DISP,1              DISPLAY PANEL ADDRESS
00002EI  07AA                          XR    WORK,WORK
000030I  DF90 80C4 =0000F8I            OC    DISP,INCR           SET DISPLAY TO INCREMENT MODE
000034I  9A9A                          WDR   DISP,WORK           CLEAR DISPLAY PANEL
000036I  9A9A                          WDR   DISP,WORK
000038I  9A9A                          WDR   DISP,WORK
00003AI  9A9A                          WDR   DISP,WORK
00003CI  DF90 80B9 =0000F9I            OC    DISP,NORM           SET DISPLAY BACK TO NORMAL MODE
000040I  40A0 80AC =0000F0I            STH   WORK,FLAG           RESET FLAG INDICATOR
000044I  24F9                          LIS   R2,9                10 BYTES PER FRAME
000046I  24E1                          LIS   R1,1                INCREMENT
000048I  07DD                          XR    COUNT,COUNT         CLEAR COUNTER
00004AI  DEB0 80B1 =0000FFI            OC    RDEV,COMMON         SET ZBID MODE
00004EI  DEC0 80AD =0000FFI            OC    XDEV,COMMON
000052I  DEC0 80A5 =0000FBI            OC    XDEV,WRMOD          PUT XMIT TO WRITE MODE
000056I  DAC0 80A7 =000101I            WD    XDEV,FILL           ENSURE LINE STARTS UP WITH MARKS
00005AI  DEC0 809E =0000FCI            OC    XDEV,ENBRQS         XMIT LINE IN RQ2S
```

```
        QSA - 2-WIRE PROGRAMMING EXAMPLE - 32 BIT PROCESSOR       PAGE   3  12:43:46  07/19/78

        ZBID

                                    *
                                    *  NOTE :
                                    *
                                    *     IF THE NEXT COMMAND IS BEING ISSUED TO TURN A 2-WIRE LINE
                                    *     AROUND FROM WRITE TO READ, THE COMMAND MUST BE ISSUED TWICE
                                    *     FIRST WITH INTERRUPTS DISARMED,AND THEN REISSUED WITH THE
                                    *     DESIRED INTERRUPT CONDITION, UNLESS THE DISARM CONDITION
                                    *     IS DESIRED.
                                    *
         0000 005EI                 WAIT2    EQU    *
000005EI DEB0 8098 =0000FAI                  OC     RDEV,ENBSCH            RECV LINE IN SYNSCH
000062I  C200 807A =0000E0I         WAIT1    LPSW   WAIT
                                    *
000066I  2491                       DSPLAY   LIS    DISP,1                 LOAD THE DISPLAY PANEL ADDRESS
000068I  DE90 808C =0000F8I                  OC     DISP,INCR
00006CI  9A93                                WDR    DISP,ISTA              DISPLAY STATUS
00006EI  9A92                                WDR    DISP,IDEV              DISPLAY DEVICE ADDRESS
000070I  DEB0 808A =0000FEI         HALT1    OC     RDEV,IDLE              IDLE LINES
000074I  DFC0 8086 =0000FEI                  OC     XDEV,IDLE
000078I  C200 806C =0000E8I                  LPSW   HALT
                                    *
00007CI  48A0 8070 =0000F0I         WRITE    LH     WORK,FLAG              FLAG SET?
000080I  4230 800A =00008EI                  BNZ    CONTIN                 YES - SKIP WRITE DATA
000084I  0833                                LR     ISTA,ISTA              STATUS ZERO?
000086I  4230 FFDC =000066I                  BNZ    DSPLAY                 NO - ERROR
00008AI  DA20 8072 =000100I                  WD     IDEV,DATA              WRITE A DATA BYTE
         0000 008EI                 CONTIN   EQU    *
00008EI  F8A0 0000 00F0                      LI     WORK,Y'00F0'           SET REGISTER SET "F" PSW.
000094I  958A                                EPSR   WORK1,WORK             CHANGE THE PSW.
000096I  C1D0 FFC8 =000062I                  BXLE   COUNT,WAIT1
00009AI  07DD                                XR     COUNT,COUNT            CLEAR COUNTER
00009CI  48A0 8050 =0000F0I                  LH     WORK,FLAG              REVERSE FLAG INDICATOR
0000A0I  C7A0 00FF                           XHI    WORK,X'FF'
0000A4I  40A0 8048 =0000F0I                  STH    WORK,FLAG
0000A8I  DEC0 8051 =0000FDI                  OC     XDEV,ENBRQ             RESET OV
0000ACI  08AA                                LR     WORK,WORK              SET THE CONDITION CODE
                                    *  WHERE FLAGS BEING SENT ON THE LINE
0000AEI  4230 FFB0 =000062I                  BNZ    WAIT1                  NO - DATA WAS ON THE LINE
0000B2I  4300 FFA8 =00005EI                  B      WAIT2                  YES - PUT THE RCV SIDE IN SYNCH.
                                    *
0000B6I  C530 0008                  READ     CLHI   ISTA,BUSY              CARRIER-ON INTERRUPT?
0000BAI  4330 FFA4 =000062I                  BE     WAIT1                  YES - WAIT
0000BEI  C330 00BB                           THI    ISTA,TERM              STATUS ZERO OR TERM?
0000C2I  4230 FFA0 =000066I                  BNZ    DSPLAY                 NO - ERROR
0000C6I  9B2A                                RDR    IDEV,WORK              READ A BYTE
0000C8I  2491                                LIS    DISP,1
0000CAI  9A9A                                WDR    DISP,WORK              DISPLAY THE BYTE
0000CCI  D4A0 8030 =000100I                  CLB    WORK,DATA              IS IT DATA SENT?
0000D0I  4330 FF8E =000062I                  BE     WAIT1                  YES - WAIT
0000D4I  C5A0 007E                           CLHI   WORK,ENDFLG            IS IT THE END FLAG?
0000D8I  4330 FF86 =000062I                  BE     WAIT1                  YES - WAIT
0000DCI  4300 FF90 =000070I                  B      HALT1                  NO - HALT
0000E0I                                      ALIGN 4
```

```
     QSA - 2-WIRE PROGRAMMING EXAMPLE - 32 BIT PROCESSOR       PAGE   4  12:43:46  07/19/78

     ZBID

0000E0I  0000 C0F0             113  WAIT     DC     Y'C0F0',A(WAIT1)
0000E4I  0000 0062I
0000E8I  0000 80F0             114  HALT     DC     Y'80F0',A(HALT1)
0000ECI  0000 0070I
0000F0I  0000 0000             115  FLAG     DC     0
0000F4I  00B8                  116  RADR     DC     X'B8'              RECV LINE ADDRESS
0000F6I  00BC                  117  XADR     DC     X'BC'              XMIT LINE ADDRESS
0000F8I  40                    118  INCR     DB     X'40'              INCREMENT
0000F9I  80                    119  NORM     DB     X'80'              NORMAL
0000FAI  79                    120  ENBSCH   DB     X'79'              ENABLE, LOCAL, SYNSCH
0000FBI  FB                    121  WRMOD    DB     X'FB'              DISARM,LOCAL,READY,WRITE,RESET D.M.
0000FCI  7B                    122  ENBRQS   DB     X'7B'              ENABLE,LOCAL,RESET DATA MODE,RQ2S
0000FDI  6B                    123  ENBRQ    DB     X'6B'              ENABLE, LOCAL, RQ2S
0000FEI  F1                    124  IDLE     DB     X'F1'              DISARM, IDLE
0000FFI  32                    125  COMMON   DB     X'32'              ZBID MODE
000100I  55                    126  DATA     DB     X'55'              DATA BYTE
000101I  FF                    127  FILL     DB     X'FF'              MARK CHARACTER
000102I                        128           END
```

```
    QSA - 2-WIRE PROGRAMMING EXAMPLE - 32 BIT PROCESSOR       PAGE    5  12:43:46  07/19/78

    ZBID

ASSEMBLED BY CAL 03-066R05-00   (32-BIT)

START OPTIONS:  *NONE*

   NO CAL ERRORS
   NO CAL WARNINGS
    2 PASSES

 ABSTOP    0000 0000
 ADC       0000 0004
 BUSY      0000 0008          29*     100
 COMMON    0000 00FFI         55      56      125*
 CONTIN    0000 008EI         82      86*
 COUNT     0000 000D          26*     54      54      89      90      90
 DATA      0000 0100I         85      107     126*
 DISP      0000 0009          22*     43      45      46      47      48      49      50      73      74      75      76      105
                             106
 DSPLAY    0000 0066I         73*     84      103
 ENBRQ     0000 00FDI         94      123*
 ENBRQS    0000 00FCI         59      122*
 ENBSCH    0000 00FAI         70      120*
 ENDFLG    0000 007E          31*     109
 FILL      0000 0101I         58      127*
 FLAG      0000 00F0I         51      81      91      93      115*
 HALT      0000 00F8I         79      114*
 HALT1     0000 0070I         77*     111     114
 IDEV      0000 0002          19*     76      85      104
 IDLE      0000 00FEI         36      40      77      78      124*
 IMPTOP    0000 0102I
 INCR      0000 00F8I         45      74      118*
 ISTA      0000 0003          20*     75      83      83      100     102
 LADC      0000 0002
 NORM      0000 00F9I         50      119*
 PURETOP   0000 0000P
 R1        0000 000E          27*     53
 R2        0000 000F          28*     52
 RADR      0000 00F4I         35      116*
 RDEV      0000 000B          24*     35      36      38      38      55      70      77
 READ      0000 00B6I         37      100*
 TERM      0000 00BB          30*     102
 WAIT      0000 00E0I         71      113*
 WAIT1     0000 0062I         71*     89      97      101     108     110     113
 WAIT2     0000 005EI         69*     98
 WORK      0000 000A          23*     33      34      37      38      41      42      44      44      46      47      48      49
                              51      81      87      88      91      92      93      95      95      104     106     107     109
 WORK1     0000 0008          21*     34      88
 WRITE     0000 007CI         41      81*
 WRMOD     0000 00FBI         57      121*
 XADR      0000 00F6I         39      117*
 XDEV      0000 000C          25*     39      40      42      42      56      57      58      59      78      94
```

```
      QSA - 4-WIRE PROGRAMMING EXAMPLE - 32 BIT PROCESSOR       PAGE   1  12:45:11  07/19/78

PROG= *NONE*     ASSFMBLED BY CAL 03-066R05-00   (32-BIT)

                                      1           SCRAT
                                      2           CROSS
                                      3           WIDTH 120
                                      4           TARGT 32
                                      5           NORX3
                                      6           PROG  QSA - 4-WIRE PROGRAMMING EXAMPLE - 32 BIT PROCESSOR
```

```
                                  * *****************************************************************
                                  * THIS SAMPLE PROGRAM ILLUSTRATES A DATA TRANSFER BETWEEN TWO     *
                                  * 4-WIRE LINES RUNNING IN ZBID MODE UNDER LOCAL
                                  * LOOPBACK. 10 DATA BYTES ARE SENT BY THE PROGRAM, AND THE 10     *
                                  * FLAGS ARE SENT BY THE HARDWARE, DUE TO THE OVERFLOW CONDITION   *
                                  * GENERATED AND CONTROLLED BY THE TRANSMIT SIDE. THIS PATTERN     *
                                  * OF 10 DATA BYTES AND 10 FLAGS IS OUTPUT UNTIL AN ERROR IS       *
                                  * DETECTED, OR PROGRAM EXECUTION IS TERMINATED BY THE USER.       *
                                  * THROUGHOUT THE OPERATION OF THE PROGRAM THE DATA RECEIVED       *
                                  * IS DISPLAYED ON THE DISPLAY PANEL. IF THE PROCESSOR HALTS       *
                                  * ONE OF TWO ERRORS HAVE BEEN DETECTED. IF ONLY ONE DATA BYTE     *
                                  * IS DISPLAYED WHEN THE PROCESSOR HALTS THEN A DATA ERROR WAS     *
                                  * DETECTED, AND THE BYTE DISPLAYED IS THE BYTE IN ERROR. IF TWO   *
                                  * BYTES ARE DISPLAYED, A STATUS ERROR OR AN UNEXPECTED INTERRUPT *
                                  * WAS DETECTED. IN THIS CASE THE LEFTMOST BYTE IS THE DEVICE      *
                                  * ADDRESS, AND THE OTHER BYTE IS THE DEVICE STATUS.               *
                                  *                                                                 *
                                  *                                                                 *
                                  * *****************************************************************
          0000 0002               IDEV      EQU   2
          0000 0003               ISTA      EQU   3
          0000 000B               DISP      EQU   11
          0000 000C               WORK      EQU   12
          0000 000D               IDX       EQU   13
          0000 000E               WORK1     EQU   14
          0000 000F               RETN      EQU   15
                                  *
000000I   FAC0 0000 00F0                    LI    WORK,Y'00F0'         SET REGISTER SET "F" PSW.
000006I   95EC                              EPSR  WORK1,WORK           CHANGE THE PSW.
000008I   07DD                              XR    IDX,IDX              CLEAR THE INDEX
00000AI   48CD 8112 =000120I      SETUP     LH    WORK,DEVADR(IDX)     GET A DEVICE ADDRESS
00000EI   DEC0 812F =000141I                OC    WORK,IDLE            IDLE THE DEVICE
000012I   26D2                              AIS   IDX,2                INCREMENT THE INDEX
000014I   C5D0 0008                         CLHI  IDX,DEVEND-DEVADR    FINISHED ?
000018I   2037                              BNES  SETUP                NO - CONTINUE
00001AI   24B1                              LIS   DISP,1               LOAD THE DISPLAY PANEL ADDRESS
00001CI   07CC                              XR    WORK,WORK            SET ZERO DATA
00001EI   DEB0 8117 =000139I                OC    DISP,INCR            SET THE DISPLAY TO INCREMENTAL MODE
000022I   98BC                              WHR   DISP,WORK            ZERO OUT THE
000024I   98BC                              WHR   DISP,WORK            DISPLAY PANEL
000026I   DEB0 810E =000138I                OC    DISP,NORM            SET THE DISPLAY TO NORMAL MODE
00002AI   07DD                              XR    IDX,IDX              CLEAR THE INDEX REGISTER
          0000 002CI              SETVET    EQU   *
00002CI   48ED 80F0 =000120I                LH    WORK1,DEVADR(IDX)    GET THE RECEIVE DEVICE ADDRESS
000030I   DEE0 8107 =00013BI                OC    WORK1,COMMON         SET UP THE COMMON CONTROL
000034I   DFE0 8102 =00013AI                OC    WORK1,ENBSCH         RCV, ENABLE, SYNC. SEARCH
000038I   E6C0 80AA =0000E6I                LA    WORK,READ            LOAD THE READ INTERRUPT VECTOR
00003CI   40CE 4E00 00D0                    STH   WORK,X'D0'(WORK1,WORK1) STORE IN THE VECTOR TABLE
000042I   26D2                              AIS   IDX,2                INCREMENT THE INDEX
                                  *
000044I   07CC                              XR    WORK,WORK            CLEAR WORK
000046I   40CD 80DE =000128I                STH   WORK,CONTROL(IDX)    CLEAR CONTROL FLAG
00004AI   40CD 80D8 =000126I                STH   WORK,CONTROL-2(IDX) CLEAR THE BYTE COUNT
```

2 LINES - ZBID

```
                                   61  *
                                   62  *
00004EI  48ED 80CE =000120I        63            LH    WORK1,DEVADR(IDX)    GET THE XMIT DEVICE ADDRESS
000052I  DEE0 80E7 =00013DI        64            OC    WORK1,WRMOD          PUT XMIT TO WRITE MODE
000056I  DAE0 80E4 =00013EI        65            WD    WORK1,FILL           ENSURE LINE STARTS UP WITH MARKS
00005AI  DEE0 80DE =00013CI        66            OC    WORK1,ENBRQS         ENABLE XMIT, RQ2S
00005EI  E6C0 8030 =000092I        67            LA    WORK,WRITE           LOAD THE WRITE INTERRUPT VECTOR
000062I  40CE 4E00 00D0            68            STH   WORK,X'D0'(WORK1,WORK1) STORE IN THF VECTOR TABLE
000068I  26D2                      69            AIS   IDX,2                INCREMENT THE INDEX
00006AI  C5D0 0008                 70            CLHI  IDX,DEVEND-DEVADR    FINISHED ?
00006EI  4280 FFBA =00002CI        71            BL    SETVET               YES
000072I  C200 809A =000110I        72  WAIT1     LPSW  WAIT                 NO - WAIT
                                   73  *
000076I  07DD                      74  IDXSET    XR    IDX,IDX              CLEAR THE INDEX
000078I  452D 80A4 =000120I        75  LOOK1     CLH   IDEV,DEVADR(IDX)     TRY TO FIND A MATCH
00007CI  033F                      76            BER   RETN                 MATCH - RETURN
00007EI  26D2                      77            AIS   IDX,2                BUMP THE POINTER
000080I  C5D0 0008                 78            CLHI  IDX,DEVEND-DEVADR    FINISHED ?
000084I  4280 FFF0 =000078I        79            BL    LOOK1                NO
         0000 0088I                80  DSPLAY    EQU   *
000088I  24B1                      81            LIS   DISP,1               LOAD THE DISPLAY PANEL ADDRESS
00008AI  9AB3                      82            WDR   DISP,ISTA            DISPLAY THE STATUS
00008CI  9AB2                      83            WDR   DISP,IDEV            DISPLAY THE DEVICE ADDRESS
00008EI  C200 8086 =000118I        84  HALT1     LPSW  HALT                 HALT THE CPU
                                   85  *
                                   86  *
                                   87  *
         0000 0092I                88  WRITE     EQU   *
000092I  41F0 FFE0 =000076I        89            BAL   RETN,IDXSET          GET THE INTERNAL INDEX
000096I  48CD 808E =000128I        90            LH    WORK,CONTROL(IDX)    IS THE FLAG SET ?
00009AI  4230 800A =0000A8I        91            BNZ   CONTIN               YES - SKIP THE WRITE DATA
00009EI  0833                      92            LR    ISTA,ISTA            NO - IS THE STATUS 0 ?
0000A0I  4230 FFE4 =000088I        93            BNZ   DSPLAY               NO - ERROR
0000A4I  DA20 8097 =00013FI        94            WD    IDEV,DATA
0000A8I  48CD 807A =000126I        95  CONTIN    LH    WORK,CONTROL-2(IDX)  GET THE BYTE COUNT
0000ACI  26C1                      96            AIS   WORK,1               INCREMENT BY 1
0000AEI  C5C0 000A                 97            CLHI  WORK,10              FINISHED ?
0000B2I  4330 8008 =0000BEI        98            BE    CONTIN1              YES - SET UP TO SEND FLAG
0000B6I  40CD 806C =000126I        99            STH   WORK,CONTROL-2(IDX)  NO - SAVE COUNT
0000BAI  4300 FFB4 =000072I       100            B     WAIT1                RETURN
0000BEI  07CC                     101  CONTIN1   XR    WORK,WORK            CLEAR THE COUNT
0000C0I  40CD 8062 =000126I       102            STH   WORK,CONTROL-2(IDX)  STORE IN CONTROL TABLE
0000C4I  48CD 8060 =000128I       103            LH    WORK,CONTROL(IDX)    GET THE FLAG
0000C8I  C7C0 00FF                104            XHI   WORK,X'FF'           CHANGE THE FLAG
0000CCI  40CD 8058 =000128I       105            STH   WORK,CONTROL(IDX)    STORE IT BACK
0000D0I  DE20 806C =000140I       106            OC    IDEV,ENBRQ           RESET THE OVERFLOW BIT
0000D4I  08CC                     107            LR    WORK,WORK            WHERE FLAGS BEING SENT ?
0000D6I  4230 FF98 =000072I       108            BNZ   WAIT1                NO - DATA
0000DAI  48CD 8040 =00011EI       109            LH    WORK,DEVADR-2(IDX)   GET THE ADDRESS OF THE RCV. LINE
0000DEI  DEC0 8058 =00013AI       110            OC    WORK,ENBSCH          PUT THE RCV. SIDE INTO SYNSCH
0000E2I  4300 FF8C =000072I       111            B     WAIT1
                                  112  *
         0000 00E6I               113  READ      EQU   *
```

```
QSA - 4-WIRE PROGRAMMING EXAMPLE - 32 BIT PROCESSOR       PAGE   4  12:45:11  07/19/78

2 LINES - ZBID

0000E6I  24B1                114          LIS   DISP,1
0000E8I  C530 0008           115          CLHI  ISTA,BUSY           CARRIER ON INTERRUPT ?
0000ECI  4330 FF82 =000072I  116          BE    WAIT1               YES - WAIT
0000F0I  C330 00BB           117          THI   ISTA,TERM           NO - IS IT TERMINATE OR ZERO STATUS ?
0000F4I  4230 FF90 =000088I  118          BNZ   DSPLAY              NO - ERROR
0000F8I  9B2C                119          RDR   IDEV,WORK           YES - DISPLAY
0000FAI  9ABC                120          WDR   DISP,WORK           THE DATA READ
0000FCI  D4C0 803F =00013FI  121          CLB   WORK,DATA           WAS DATA RCVED. ?
000100I  4330 FF6E =000072I  122          BE    WAIT1               YES
000104I  C5C0 007E           123          CLHI  WORK,ENDFLG         NO - WAS FLAG RCVED ?
000108I  4330 FF66 =000072I  124          BE    WAIT1
00010CI  4300 FF7E =00008EI  125          B     HALT1
                             126  *
000110I                      127          ALIGN 4
                             128  *
000110I  0000 C0F0           129  WAIT    DC    Y'C0F0',A(WAIT1)
000114I  0000 0072I
000118I  0000 80F0           130  HALT    DC    Y'80F0',A(HALT1)
00011CI  0000 008EI
                             131  *
000120I  00B8                132  DEVADR  DC    X'B8',X'BD',X'BC',X'B9'
000122I  00BD
000124I  00BC
000126I  00B9
         0000 0128I          133  DEVEND  EQU   *
000128I  0000 0000           134  CONTROL DC    0,0,0,0
00012CI  0000 0000
000130I  0000 0000
000134I  0000 0000
000138I  80                  135  NORM    DB    X'80'               DISPLAY NORMAL MODE
000139I  40                  136  INCR    DB    X'40'               DISPLAY INCREMENTAL MODE
00013AI  79                  137  ENBSCH  DB    X'79'               ENABLE,LOCAL,SYNC. SEARCH
00013BI  32                  138  COMMON  DB    X'32'               ZBID MODE
00013CI  7B                  139  ENBRQS  DB    X'7B'               ENABLE,LOCAL,RESET DATA MODE,RQ2S
00013DI  FB                  140  WRMOD   DB    X'FB'               DISARM,LOCAL,READY,WRITE,RESET D.M.
00013EI  FF                  141  FILL    DB    X'FF'               MARK CHARACTER
00013FI  55                  142  DATA    DB    X'55'               DATA BYTE
000140I  6B                  143  ENBRQ   DB    X'6B'               ENABLE,LOCAL,RQ2S
         0000 007E           144  ENDFLG  EQU   X'7E'               END FLAG
000141I  F1                  145  IDLE    DB    X'F1'               IDLE
         0000 0008           146  BUSY    EQU   8                   BUSY STATUS BIT
         0000 00BB           147  TERM    EQU   X'BB'               COMPLEMENT OF TERM STATUS
                             148  *
000142I                      149          END
```

```
    QSA - 4-WIRE PROGRAMMING EXAMPLE - 32 BIT PROCESSOR       PAGE   5  12:45:11  07/19/78

    2 LINES - ZBID

ASSEMBLED BY CAL 03-066R05-00   (32-BIT)

START OPTIONS:  *NONE*

   NO CAL ERRORS
   NO CAL WARNINGS
    2 PASSES

 ABSTOP     0000 0000
 ADC        0000 0004
 BUSY       0000 0008        115    146*
 COMMON     0000 013BI        52    138*
 CONTIN     0000 00A8I        91     95*
 CONTIN1    0000 00BEI        98    101*
 CONTROL    0000 0128I        59     60     90     95     99    102    103    105    134*
 DATA       0000 013FI        94    121    142*
 DEVADR     0000 0120I        38     41     51     63     70     75     78    109    132*
 DEVEND     0000 0128I        41     70     78    133*
 DISP       0000 000B         29*    43     45     46     47     48     81     82     83    114    120
 DSPLAY     0000 0088I        80*    93    118
 ENBRQ      0000 0140I       106    143*
 ENBRQS     0000 013CI        66    139*
 ENBSCH     0000 013AI        53    110    137*
 ENDFLG     0000 007E        123    144*
 FILL       0000 013EI        65    141*
 HALT       0000 0118I        84    130*
 HALT1      0000 008EI        84*   125    130
 IDEV       0000 0002         27*    75     83     94    106    119
 IDLE       0000 0141I        39    145*
 IDX        0000 000D         31*    37     37     38     40     41     49     49     51     56     59     60     63
                              69     70     74     74     75     77     78     90     95     99    102    103    105
                             109
 IDXSET     0000 0076I        74*    89
 IMPTOP     0000 0142I
 INCR       0000 0139I        45    136*
 ISTA       0000 0003         28*    82     92     92    115    117
 LADC       0000 0002
 LOOK1      0000 0078I        75*    79
 NORM       0000 0138I        48    135*
 PURETOP    0000 0000P
 READ       0000 00E6I        54    113*
 RETN       0000 000F         33*    76     89
 SETUP      0000 000AI        38*    42
 SETVET     0000 002CI        50*    71
 TERM       0000 00BB        117    147*
 WAIT       0000 0110I        72    129*
 WAIT1      0000 0072I        72*   100    108    111    116    122    124    129
 WORK       0000 000C         30*    35     36     38     39     44     44     46     47     54     55     58     58
                              59     60     67     68     90     95     96     97     99    101    101    102    103
                             104    105    107    107    109    110    119    120    121    123
 WORK1      0000 000E         32*    36     51     52     53     55     55     63     64     65     66     68     68
 WRITE      0000 0092I        67     88*
 WRMOD      0000 013DI        64    140*
```

CUT ALONG LINE

PUBLICATION COMMENT FORM

Please use this postage-paid form to make any comments, suggestions, criticisms, etc. concerning this publication.

From ______________________ Date ______________________

Title ______________________ Publication Title ______________________

Company ______________________ Publication Number ______________________

Address ______________________

____

____

FOLD FOLD

Check the appropriate item.

□ Error Page No. ________ Drawing No. ______________

□ Addition Page No. ________ Drawing No. ______________

□ Other Page No. ________ Drawing No. ______________

Explanation:

FOLD FOLD

Fold and Staple
No postage necessary if mailed in U.S.A.

STAPLE STAPLE

FOLD FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 22 OCEANPORT, N.J.

POSTAGE WILL BE PAID BY ADDRESSEE

PERKIN-ELMER
Computer Systems Division
2 Crescent Place
Oceanport, NJ 07757

TECH PUBLICATIONS DEPT. MS 322A

FOLD FOLD

STAPLE STAPLE