ADVANCED SINGLE FUNCTION MIL-STD-1553 BUS INTERFACE (ASF)

Reference Manual

TABLE OF CONTENTS

System Operation

INTRODUCTION

The Advanced Single Function (ASF) board provides an intelligent, low-cost interface between the MIL-STD-1553 bus and a host computer system. The ASF-V is a single function, MIL-STD-1553 board for VMEbus systems, and the ASF-PC is a single function, MIL-STD-1553 board for PC ISA systems. The ASF may operate in one of three possible MIL-STD-1553 modes: Bus Controller (BC), a single Remote Terminal (RT), or a Bus Monitor (BM). The mode of operation can be controlled either by the host computer software or by hardware switches on the board.

The purpose of this reference manual is to provide information on the use of the ASF board. This manual is divided into four sections:

Introduction describes the features of the ASF.

ASF-PC Installation describes the procedures necessary to configure your ASF-PC for a PC system.

ASF-V Installation describes the procedures necessary to configure your ASF-V for a VMEbus system.

System Operation describes the programming sequence for all modes of operation.

Feature List

The following is a list of the ASF features:

- Memory mapped operation
- Comprehensive MIL-STD-1553 dual-redundant Bus Controller, Remote Terminal, and Bus Monitor modes
- Programmable interrupts and interrupt history list for all modes of operation
- Bus Controller Mode:
  Command block list control structure
  Complete control of each message including:
  - Redundant bus selection
  - Programmable intermessage delay
  - Programmable automatic message retry capability
- Remote Terminal Mode:
  RT address selectable by software or DIP switches
  Efficient memory utilization -- user allocation of all buffers
  Table-based configuration
  - Multiple buffers available for subaddress and mode codes
  - Interrupt on specific subaddress or mode code
  - Programmable *no response* for an illegal subaddress or mode code
- Bus Monitor Mode:
  Performs sequential bus monitoring
  Monitors any combination of selected RT addresses
  Supports operation in either circular-buffer or double-buffered modes

- Host Interface:
  Memory-mapped operation and control
  Switch selectable base address
  Switch selectable interrupt level
  PC ISA Interface:
    - 8 bit and 16 bit data access
    - Extended and conventional addressing support
    - I/O Register controls board operation
    - 3/4 size PC board

  VMEbus Interface:
    - Double Height 6U x 160mm module.
    - P1 and P2 (optional for extended addressing) connection to the VMEbus backplane
    - D16 data access
    - A24/A32 addressing support
    - D08 interrupt vector
    - BLT block tranfers

- Error Detection/Notification

System Interface

The ASF board is programmed by the host computer using a block of control registers and dual port random access memory (RAM). The RAM is initialized to contain the control block lists and associated data buffers used to define the desired operation.

ASF-V Interface
The ASF-V appears on the VMEbus as a 64K block of 16-bit words. This memory region consists of control registers and RAM that is accessible to the host computer at all times. The base address, interrupt level, and interrupt vector are selected using switches on the board.

ASF-PC Interface
There are two ASF-PC installation configurations - one with the ASF-PC in extended memory, and the second with the ASF-PC in the upper memory space of conventional memory. The board may be placed on any 64K x 16-bit word boundary in extended memory space or any available 32K x 16 bit word boundary in conventional memory space. Note that when the board is installed in conventional memory space, only a 32K block of 16 bit words of control registers and RAM is accessable at one time. Access to the second 32K block of memory is possible through a bank select mechanism that is controlled by an on-board I/O Register. See section 4, System Operation, for a description of the ASF-PC I/O Register. This register manages the functionality of the ASF-PC. The base address, interrupt level, and I/O Register address are selected using switches on the board.

MIL-STD-1553 Interface

The ASF may operate as a Bus Controller, Remote Terminal, or as a Bus Monitor on the MIL-STD-1553 bus. The mode can be selected by the host computer using the control registers or by switches located on the board. The RT address is also selectable by software or switches.

Bus Controller Mode (BC)
The bus controller architecture is based upon a linked list of Command Block structures and internal registers. Each message that is to be transmitted over the bus has an associated eight word Command Block that is defined in RAM by the application software. A Command Block contains Control and Command Words to initiate a message transfer. It also contains locations for the resulting status response word(s), a pointer to a data buffer, and pointers to previous and subsequent Command Blocks which together form a bus controller program. The bit-programmable Control Word in each

Command Block enables specific features or defines parameters to be used in each message transfer. These features include program termination, auto retry when an error occurs, interrupt enable, message transfer type, or block skip. A delay interval may also be specified between the start of this message and the next.

The host computer executes the bus controller program by setting an initial pointer to a Command Block data structure in the Current Command Block Register and then starting the board. Multiple bus controller programs can be stored in RAM and execution may be switched rapidly from one program to another. A bus controller program may be started and stopped by the host application software at any time.

Further details are found in section 7.0 of United Technologies Microelectronics Center, UT1553B BCRTM Data Sheet, August 1992.

Remote Terminal Mode (RT)

The remote terminal architecture is composed of an RT Descriptor Space and internal registers. The Descriptor Space contains programmable subaddress parameters that direct message transfers. The Descriptor Space contains an entry for each of the possible subaddresses and mode codes. These descriptor entries are located in a contiguous memory region 320 words long. Each entry consists of four words: 1) a Control Word, 2) a Message Status List Pointer, 3) a Data List Pointer, and 4) a word reserved for future use.

The Control Word contains programmable bits to enable the associated subaddress or mode code for both normal and broadcast operation, enable an interrupt if a message to this subaddress or mode code is received, and define the number of messages to be received into or transmitted from the data list. The Message Status List Pointer defines a buffer that stores a single status word for each message transaction which occurred to the subaddress or mode code. If multiple transactions occurred, then multiple entries will be present in this buffer. The Data List Pointer points to a buffer that stores transmit or receive data words. The ASF board is programmed to respond to a particular remote terminal address by setting the address in an internal register or on switches resident on the board.

Further details are found in section 6.0 of United Technologies Microelectronics Center, UT1553B BCRTM Data Sheet, August 1992.

Bus Monitor Mode (BM)

The bus monitor architecture is similar to that of the bus controller in that it is based upon a linked list of Command Block structures and internal registers. Each message that is received is placed in a Command Block which was previously defined in RAM by the application software. The seven word Command Block contains a Control and Status Word, locations for two Command and Status Words, a Data List Pointer, and a pointer to the subsequent command block. The Control and Status Word contains a programmable bit that enables an interrupt when the associated block is used, a time tag, and status bits which are set by the board. The number of blocks in the Command Block list is limited only by the amount of available memory.

Two types of sequential monitoring are possible. A circular buffer can be implemented by pointing the last block back to the first. The monitor will continue to run until stopped by the application software. Double-buffering is achieved by enabling the interrupt bit in the Control and Status Word of the middle and last Command Blocks in a list of blocks defined by the user. When the interrupt occurs, the application software processes the buffer with new data while monitoring continues using the other buffer.

Further details are found in section 8.0 of United Technologies Microelectronics Center, UT1553B BCRTM Data Sheet, August 1992.

Interrupt Handling and Logging

The interrupt structure consists of control registers that enable interrupt generation; control bits in the BC, RT, and BM data structures; and an Interrupt Log List that sequentially stores interrupt events in memory. The ASF exception handling scheme provides a high degree of flexibility by: 1) defining the events that cause an interrupt and 2) selecting the amount of interrupt history retained.

The ASF uses the host defined Interrupt Log List to store multiple interrupt occurrences in chronological order prior to host processing. Each log entry contains an Interrupt Status Word, a pointer to a Command Block or Descriptor Word associated with this interrupt, and a link to the next list entry. The application software defines the Interrupt Log List structure in RAM and then loads a pointer to the first block in an internal register.

Further details are found in section 9.0 of United Technologies Microelectronics Center, UT1553B BCRTM Data Sheet, August 1992.

SYSTEM OPERATION

The ASF is a board capable of emulating one of the following at a time:

- MIL-STD-1553 Bus Controller
- MIL-STD-1553 Remote Terminal
- MIL-STD-1553 Bus Monitor

The control of the ASF is accomplished using internal registers and memory located on the board. The registers are used to define the type of operation that is desired and to store pointers to data structures located in memory.

Memory Map

The memory map of the ASF is described in Table 4-1. The first 18 word locations are used to access the internal registers. The next 11 word locations are reserved for future use. The next three locations are the ASF-PC Variable Voltage DAC Register, the Host Control Register, and the Host Status Register. The remaining memory is available to the user for defining required data structures and buffers.

Word Offset	Byte Offset	Description
0000h-0011h	00000h-00022h	Internal Registers
0012h-001Ch	00024h-00038h	Reserved for Future Use -- DO NOT ACCESS
001Dh	0003Ah	DAC Register (ASF-PC Only)
001Eh	0003Ch	Host Control Register
001Fh	0003Eh	Host Status Register
0020h-FFFFh	00040h-1FFFFh	Usable Memory

Table 4-1. ASF Memory Map

The ASF contains almost 64K by 16 bit words of usable RAM. The starting word offset of this RAM is 0020 (hex) and continues to FFFF (hex). The ASF-V RAM can be accessed as words (16 bits) only and any attempt to access RAM as bytes or long words will result in a bus timeout. The ASF-PC RAM can be accessed by words or bytes.

When using Command Block Tables and Data List Tables, the ASF accesses the RAM using word addressing in contrast to the byte addressing used by the host computer. Because of this difference, all pointer values written into these tables must be word offsets. ASF pointers are referenced as words offsets throughout this manual.

Internal Registers

Internal registers enable the host computer to control the actions of the ASF. Further details are found in section 3.0 of United Technologies Microelectronics Center, UT1553B BCRTM Data Sheet, August 1992.

Host Registers

Host registers are used to manage ASF features that are not associated with MIL-STD-1553 operations. These features include VMEbus or PC interrupt control, reset, board status, LEDs, and variable output voltage. Only two host registers are available on the ASF-V - the Host Control and Status registers. These registers are available on the ASF-PC as well as the Variable Voltage DAC register and the I/O register.

Host Control Register
The Host Control Register is a bit programmabale register located at word offset 1E (hex) that is used for system control. The ASF-V has four programmable bits and the ASF-PC has six. Table 4-2 contains a list of the bit definitions.

Bit	Name	Access	Description
0	MRST	R/W	ASF Master Reset Signal
1	SSYSF	R/W	Subsystem Fail Flag
2	INTENB	R/W	Enable Interrupts
3	INTCLR	WO	Clear Pending Interrupts
3	INTPND	RO	Interrupt Pending Indicator
6	LED0	R/W	ON = Lit (ASF-PC Only)
7	LED1	R/W	ON = Lit (ASF-PC Only)

Table 4-2. Host Control Register

MRST - Setting this bit will reset all internal state machines, encoders, decoders, and registers. It is important to note that the MRST signal is active upon power-up. In order for the ASF to function properly this bit must be reset.

SSYSF - Setting this bit will generate an interrupt mask with OXFC in the interrupt handler. It will also set the Subsystem Fail Indicator in the Status Register and in the RT status response word.

INTENB - When this bit is set, interrupts generated by the ASF will be passsed to the host.

INTCLR - Writing a one to this bit will clear pending interrupts. For the ASF-V, this bit should be cleared once at initialization and never again unless polling is used. For the ASF-PC, this bit should be cleared each time in the interrupt handler to allow the generation of another interrupt to the host.

INTPND - When this bit is read, and it is set, an interrupt request is pending. If the host computer does not have an interrupt service routine installed, this bit may be polled to indicate an interrupt request. If polling is used, INTENB must be cleared.

LED0 - LED0 on the ASF-PC may be lit by setting this bit.

LED1 - LED1 on the ASF-PC may be lit by setting this bit.

Host Status Register

The Host Status Register is a two bit register located at word offset 1F (hex) that is used for system error status. Table 4-3 contains a list of the bit definitions for this read only register.

Bit	Name	Access	Description
0	LOCKUP	RO	Hardware Failure
1	BCRTMF	RO	Internal Module Failure

Table 4-3. Host Status Register

LOCKUP - If this bit is set, a hardware failure in a component other than the BCRTM has occurred.

BCRTMF - If this bit is set, a hardware failure in the BCRTM has occurred or the MRST bit is set.

ASF-PC Variable Voltage DAC Register

The Variable Voltage DAC Register is an eight bit register located on the ASF-PC at word offset 1D (hex). It controls the amplitude of the signal that is transmitted. This value must be initialized by the user prior to transmitting upon the MIL-STD-1553 bus; the nominal output level is obtained by writing an F0 (hex) into this register. Table 4-4 illustrates some examples of DAC data values versus MIL-STD-1553 bus amplitude (peak to peak).

DAC Value	1553 Bus Voltage (peak-peak)
F0	21.5
E0	20
D0	19
C0	17.5
B0	15.5
A0	14
90	12.5
80	10.5
70	9
60	7.5
50	6.5
40	4.5
30	2.9
20	1.1
10	0.4
0	0

Table 4-4. ASF-PC Variable Voltage DAC Register

The bit weighting that is programmed into the DAC is approximately 50 millivolts/bit.

ASF-PC I/O Register

The I/O Register is an eight bit register found only on the ASF-PC. Only the three least significant bits are used. Table 4-5 gives the individual bit definitions of the I/O Register.

Data Bit	Function	Bit="1"	Bit="0"
0	Board Enable	Enable	Disable*
1	Select 8/16 bit data transfer and 32K or 64K memory size	8 Bit*/32KW	16 Bit*/64KW
2	Bank Select	Upper Bank	Lower Bank*

*default configuration at power-up

Table 4-5. ASF-PC I/O Register

The Board Enable function enables or disables ASF-PC from responding to PC system data accesses. This can be useful if multiple ASF-PC boards are installed within conventional memory space. The I/O Enable LED will turn on when the board becomes enabled.

The 8/16 bit data transfer function specifies 8 or 16 bit data transfers and the amount of available ASF RAM. If the ASF-PC is installed in conventional memory space, then 8 bit data transfers MUST be selected - errors may occur if 16 bit transfers are selected.

The Bank Select function is only used when 8 bit data transfers are selected, and it is used to switch between two banks of 32K x 16 bit memory. The Control Registers are only present in Bank 0.

ASF-PC Light Emitting Diodes (LEDs)

The ASF-PC has four LEDs mounted on its face plate which are illustrated in Figure 4-1.

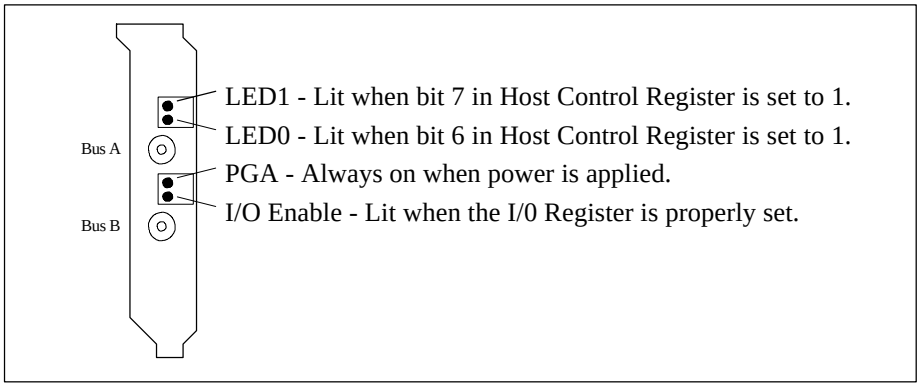

Figure 4-1. ASF-PC LED Description

Bus Controller Programming Sequence

For BC operation, the user must initialize the Command Block structure, Current Command Block Register (#2), Control Register (#0), and transmit data buffers. The user may enable BC specific interrupts by setting control bits found in the Polling Compare (#3), High-Priority Interrupt Enable (#7), and Standard Interrupt Enable Registers (#9). The BC mode is selected by setting bit 10 of the Control Register to a logical 1.

The following example sets up a BC to transmit and receive a command. It should be used as a guideline for basic operation. Note that not all of the optional features are programmed in this example. Please refer to section 7.0 of United Technologies Microelectronics Center, UT1553B BCRTM Data Sheet, August 1992, for description of the BC data structures and registers.

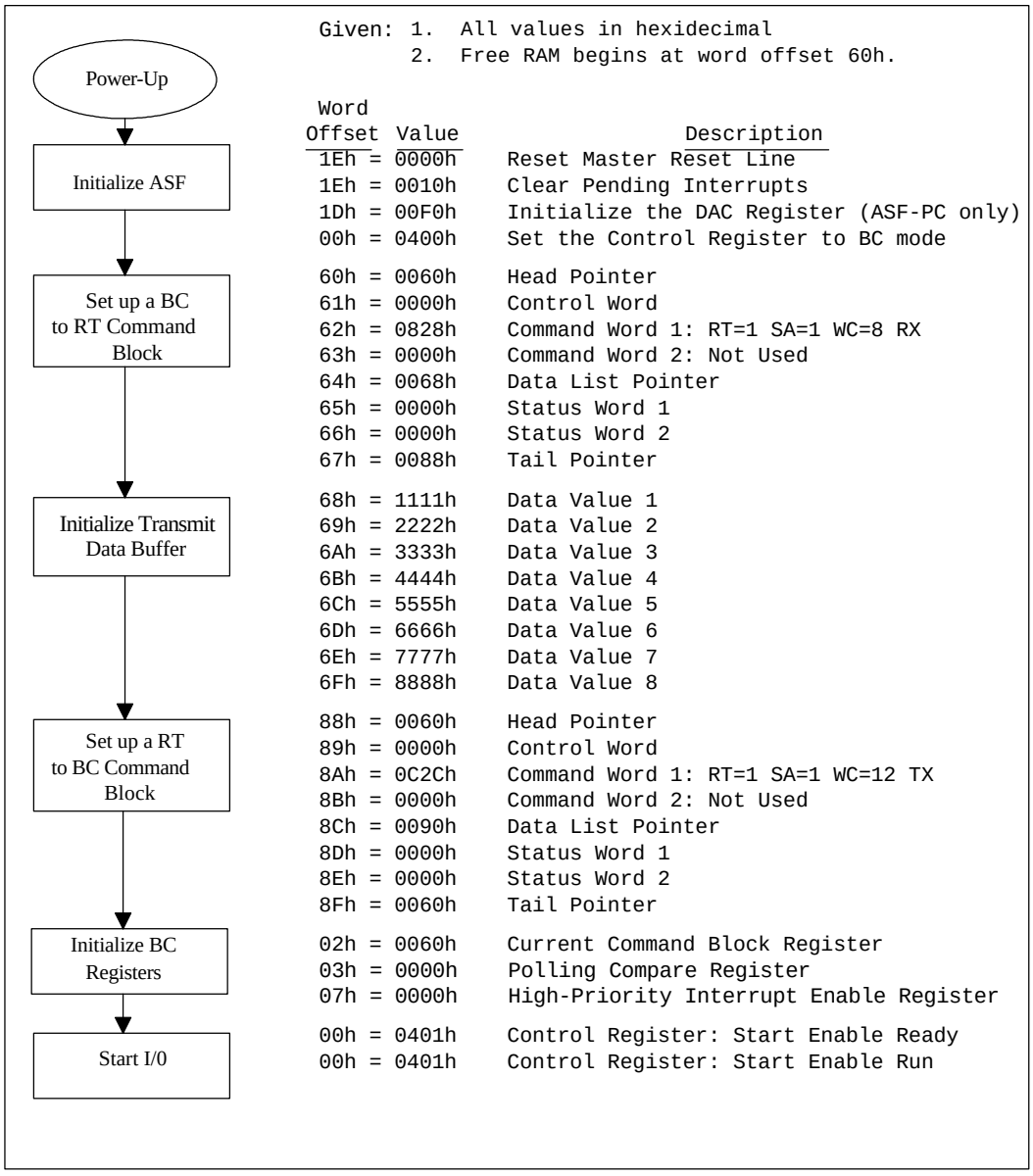

Figure 4-2. Bus Controller Initialization Sequence

After Start Enable has been set in the Control Register, the BC will transmit eight data words to RT 1 and receive 12 data words from RT 1. The status response words transmitted by the RT will be automatically written to the Command Blocks at word offset locations 65 (hex) and 8D (hex).

Remote Terminal Programming Sequence

For RT operation, the user must initialize the Descriptor Space, Control Register (#0), RT Descriptor Space Address Register (#2), and transmit subaddress data buffers. The user may enable RT specific interrupts by setting control bits found in the High-Priority Interrupt Enable and Standard Interrupt Enable Registers (#7 and #9). The RT mode is selected by resetting bit 10 of the Control Register to a logical 0.

The following example sets up RT one to transmit and receive a command. It should be used as a guideline for basic operation. Note that not all of the optional features are programmed in this example. Please refer to section 6.0 of United Technologies Microelectronics Center, UT1553B BCRTM Data Sheet, August 1992, for description of the RT data structures and registers.

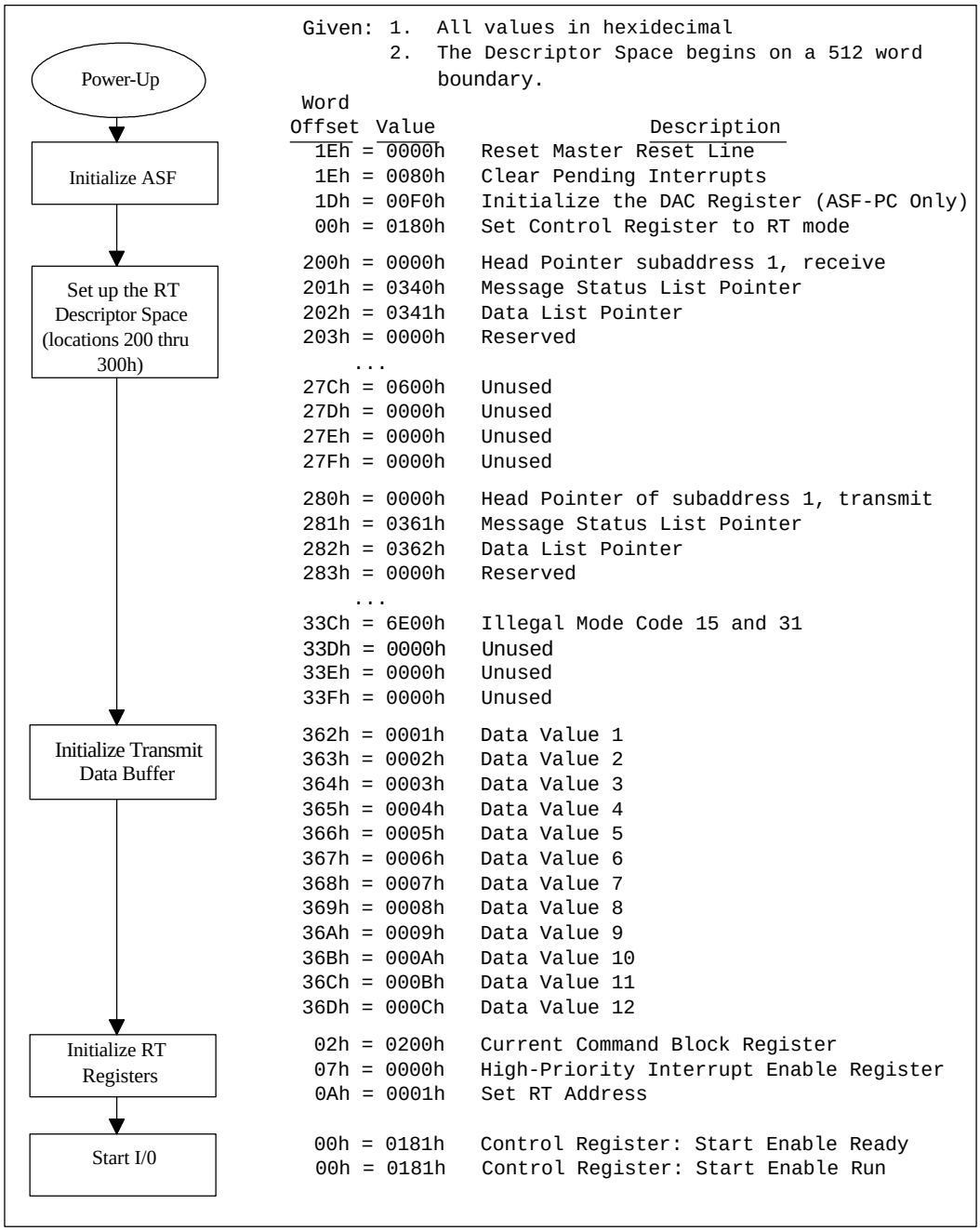

Figure 4-3. RT Initialization Sequence

After Start Enable has been set in the Control Register, RT one will response to transmit and receive commands on subaddress one. The status response words are automatically transmitted.

RT Multiple Message Handling Flow Chart

This section discusses the recovery of individual data packets from a receive subaddress using message indexing. The BCRTM stores a message status word for each attempted data transaction. The message status word reflects the number of data words received along with message validity and time-tag data.

To recover multiple data packets from a subaddress data buffer, the host is required to extract data from the buffer and match it with the message status word generated as a result of the message. For more information on the message status word and data buffer architecture refer to the BCRTM data sheet. The data recovery routine was designed for execution after the index field of the subaddress control word has decremented to zero. Figure 4-4 illustrates this sequence. Assume that the host has stored the initial condition of the message status list pointer [MESS(0)]. MESS(0) points to the top of the message buffer (MSG).

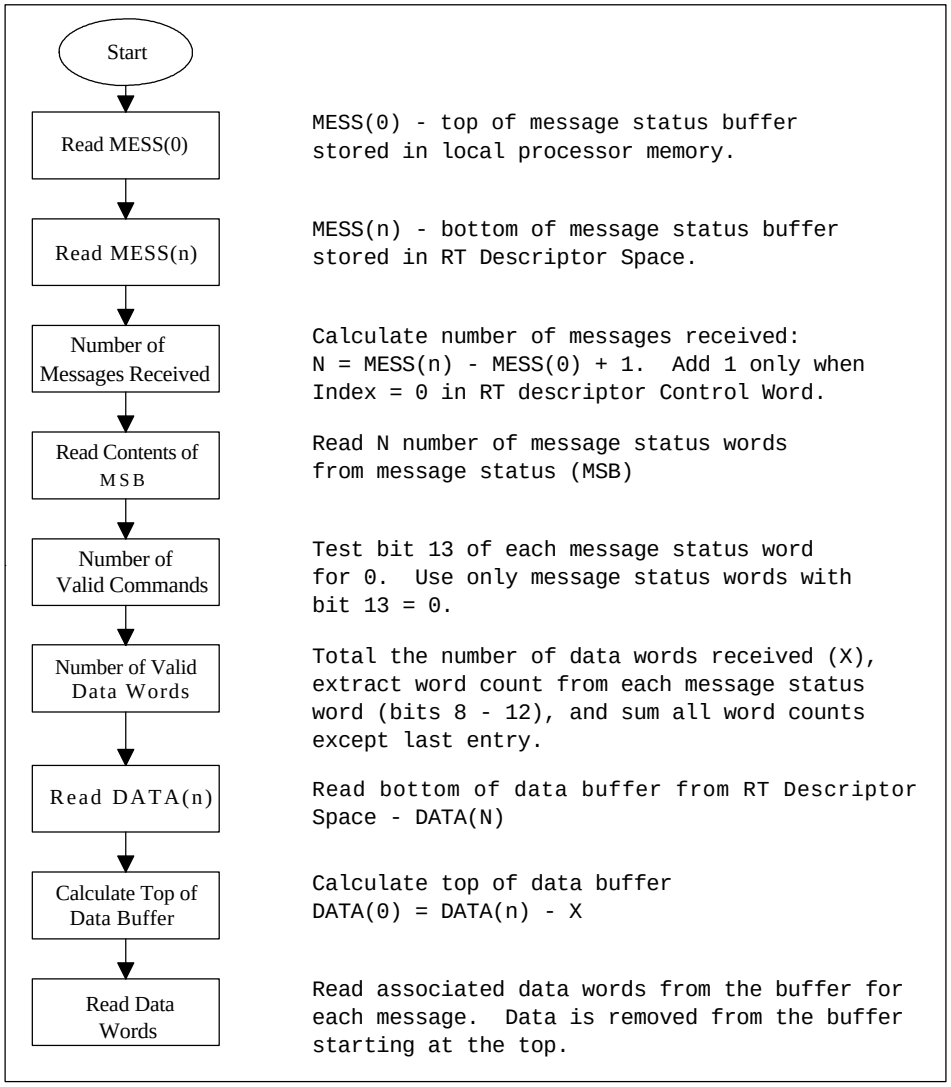

Figure 4-4. RT Message Handling Flow Chart

Example:

INDEX = 6	Control Word index field, buffer 7 messages (0 through 6)
MESS(0) = 100 (hex)	Stored in local memory
MESS(n) = 106 (hex)	Read from RT Descriptor Space
N = MESS(n) - MESS(0) + 1 = 7	Calculate the number of messages received
X = 15	Sum of word count fields (0, 2, 3, 5)

DATA(n) = 20F (hex) Read from RT Descriptor Space

Calculate: DATA(0) = DATA(n) - X = 20F - F = 200 (hex)

DATA(0)	data word 0
	data word 1
	data word 2
	data word 3
	data word 0
	data word 0
DATA(n)	data word 0
	data word 1
	data word 2
	data word 3
	data word 4
	data word 5
	data word 6
	data word 7
	data word 8
	next data word

MESS(0)	xx00 0100 xxxx xxxx
	xx1x xxxx xxxx xxxx (error)
	xx00 0001 xxxx xxxx
	xx00 0001 xxxx xxxx
	xx1x xxxx xxxx xxxx (error)
	xx00 1001 xxxx xxxx
MESS(n)	xx1x xxxx xxxx xxxx (error)

Note: Subaddress is setup to buffer seven messages (index = 6). Messages 2, 5, and 7 resulted in message errors. The final value of index field is zero.

Message #1, receive 4 data words, MESS(0)
Message #3, receive 1 data words, MESS(2)
Message #4, receive 1 data words, MESS(3)
Message #5, receive 9 data words, MESS(5)

BCRTM RT Transmission Error Message Recovery

Re-synchronization of the remote terminal is required when the bus controller observes a message error, and the remote terminal has invoked transmit message buffering (i.e., Control Word Index field not equal to zero). For this scenario, the remote terminal is instructed to transmit "N" data words, The remote terminal responds by transmitting a status word followed by "N" data words. The subaddress data pointer, message status pointer, and control word are updated after the last data word DMA is performed. Since the transaction results in a message error, the bus controller can attempt to retry the same transmit command. However, the remote terminal descriptor was updated and no longer points to the data from the failed command. To re-transmit the failed data packet, set the remote terminal descriptor to transmit the failed message.

To re-synchronize the remote terminal, the host processor must modify, on command, the transmit subaddress descriptor. Utilize the Synchronize with Data mode code to instruct the remote terminal host to begin modification of the transmit subaddress descriptor for retry. The data word associated with the mode code designates which subaddress to modify along with the word count of the failed message. Use the failed word count to calculate the retry data list pointer. As part of the descriptor re-synchronization, the message status list pointer is decremented by one to update the failed message status word. The host increments the Control Word Index field by one to maintain the proper buffer size.

To begin the re-synchronization process, the bus controller transmits a Synchronize with Data mode code. Upon reception of this command word, the remote terminal generates an interrupt to signal the

host. The host reads the data word associated with the mode code and determines which subaddress descriptor to service. The status list pointer, and control word. Figure 4-5 and 4-6 shows a transmit message sequence and recovery after a message error.

```
MESS(0) = 100 (hex)    - message status list point initial condition
MESS(2) = 102 (hex)    - message status list pointer after two transmit commands
DATA(0) = 200 (hex)    - data list pointer initial condition
DATA(7) = 207 (hex)    - data list pointer after two transmit commands


                    15                  0
MESS(0) 100 (hex)  | xxxx xxxx xxxx xxxx |  ME = 0, WC = 4
MESS(1) 101 (hex)  | xxxx xxxx xxxx xxxx |  ME = 1, WC = 3
MESS(2) 102 (hex)  | xxxx xxxx xxxx xxxx |
```

Figure 4-5. Transmit Message Sequence

```
                    15                  0
DATA(0) 200 (hex)  | xxxx xxxx xxxx xxxx |
DATA(1) 201 (hex)  | xxxx xxxx xxxx xxxx |
DATA(2) 202 (hex)  | xxxx xxxx xxxx xxxx |   First data packet
DATA(3) 203 (hex)  | xxxx xxxx xxxx xxxx |
DATA(4) 204 (hex)  | xxxx xxxx xxxx xxxx |
DATA(5) 205 (hex)  | xxxx xxxx xxxx xxxx |   Second data pack
DATA(6) 206 (hex)  | xxxx xxxx xxxx xxxx |   (message error occurs
DATA(7) 207 (hex)  | xxxx xxxx xxxx xxxx |    in transmission)
```

Figure 4-6. Recovery Sequence

To begin the re-synchronization process, the bus controller instructs the host to reset the data list pointer (DATA(n)), message status list pointer (MESS(n)), and control word. The host resets the data list pointer by subtracting the failed transmission word count from the present data list pointer value. Word count information along with a subaddress identifier is contained in the data word associated with the Synchronize with Data mode code. Organization of information contained in the data word is left up to the system designer. The host completes the remote terminal reset by decrementing the message status list by one and incrementing the control word index field by one. For the above example the data list pointer is decremented to 204 (hex), message status list pointer to 101 (hex), and the Control Word Index is incremented by one.

Bus Monitor Programming Sequence

For monitor operation, the user must initialize the Command Block structure, Current Command Block Register (#2), Control Register (#0), Monitor Control Register (#14), and Monitor Selected RT Address Registers (#16 and #15). The user may enable monitor specific interrupts by setting control bits found in the High-Priority Interrupt Enable and Standard Interrupt Enable Registers. The monitor mode is selected by setting bit 10 of the Control Register to a logical 0 and bit 15 of the Bus Monitor Control Register to a logical 1.

The following example sets up the bus monitor to capture data transmitted to and from remote terminal one. It should be used as a guideline for basic operation. Note that not all of the optional features are programmed in this example. Please refer to section 8.0 of United Technologies

Microelectronics Center, UT1553B BCRTM Data Sheet, August 1992, for description of the monitor data structures and registers.

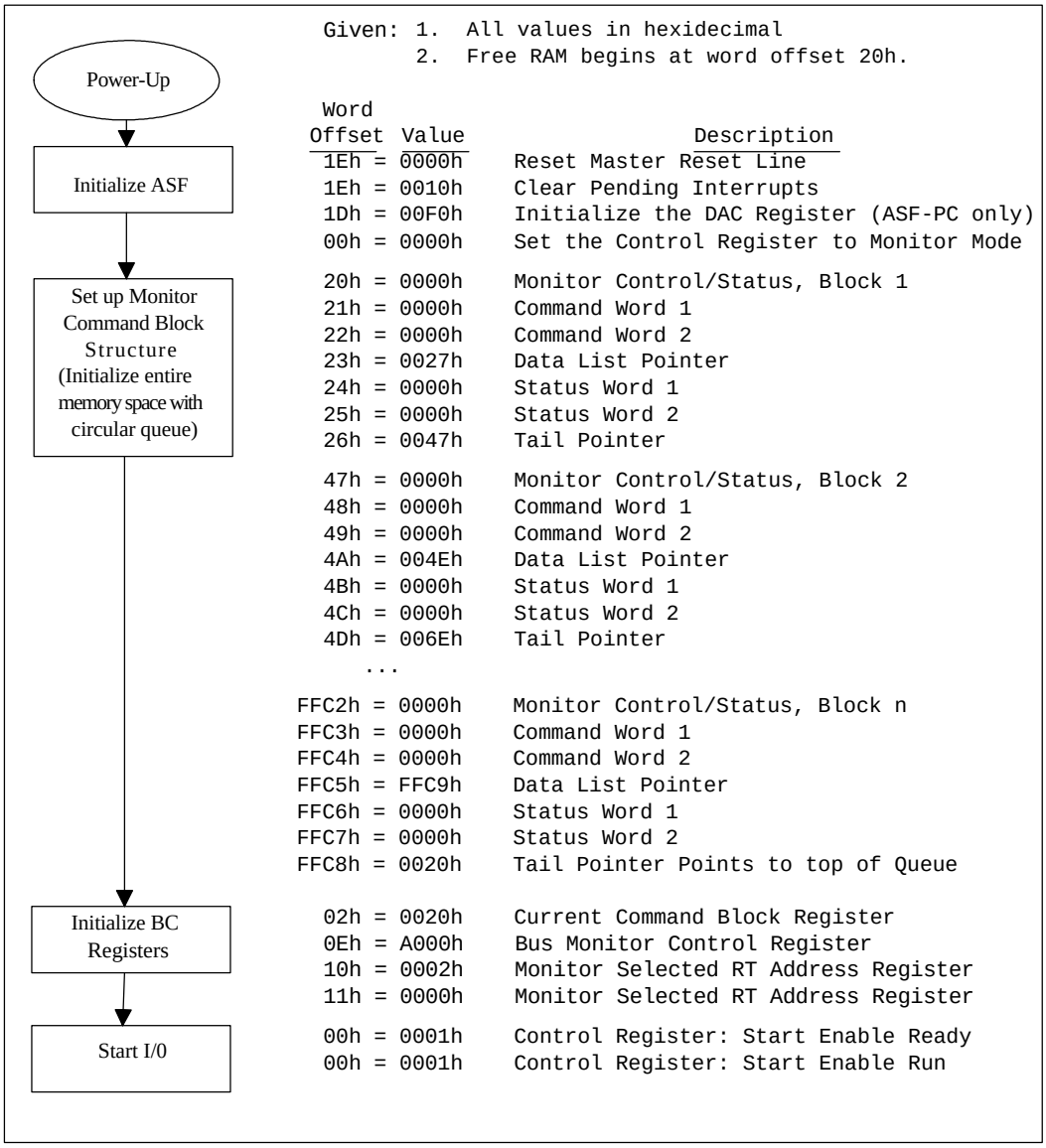

Figure 4-7. Bus Monitor Initialization Sequence

After Start Enable has been set in the Control Register, the Monitor will be enabled to receive commands.

Interrupt Handling Programming Sequence

The interrupt handling scheme uses the Interrupt Log List Pointer Register (#6) and an Interrupt Log List. Interrupts may be used for all modes of operation to log standard interrupt events.

The following example sets up an Interrupt Log List to record up to 8 standard interrupts. It should be used as a guideline for basic operation. Please refer to section 9.0 of United Technologies

Microelectronics Center, UT1553B BCRTM Data Sheet, August 1992, for description of the interrupt data structures and registers.

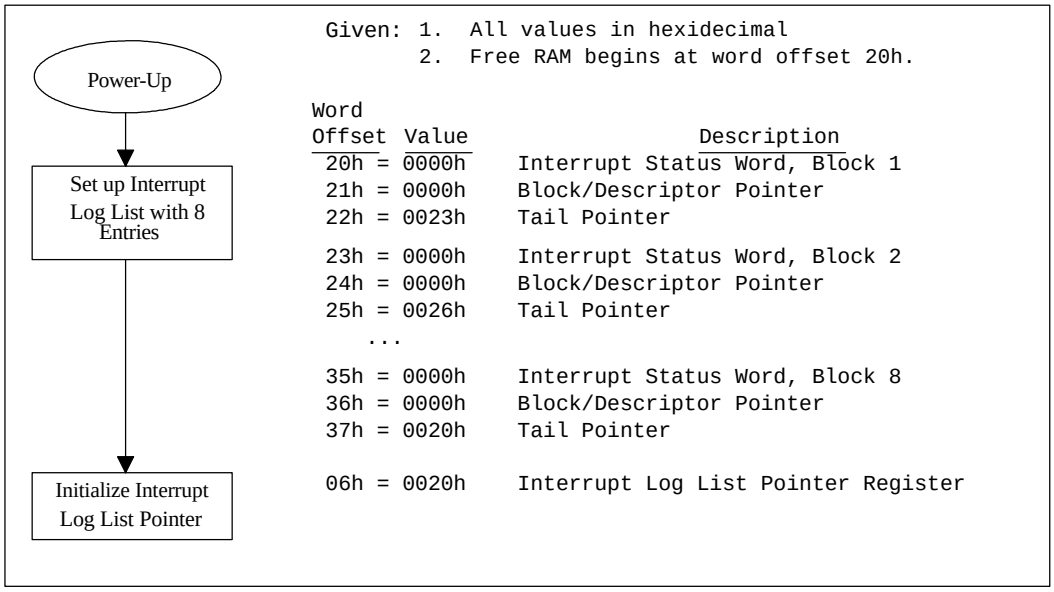

Figure 4-8. Interrupt Initialization Sequence