UNITED STATES DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY

DTI–11B
Digital Telemetry Interface
User's Guide

PRELIMINARY

Lawrence M. Baker
Tim MacDonald
U.S. Geological Survey

Robert C. Goff
Science Horizons, Inc.

Jon Berger
U.C. San Diego

OPEN-FILE REPORT 88-414

Menlo Park, California

1988

Preliminary Edition, May 1988

Open File Report No. 88-414

Acknowledgements

Funding for this work was provided, in part, by USGS grant number 14-08-0001-21893 under the National Earthquake Hazards Reduction Program.

TABLE OF CONTENTS

Figures

Tables

PREFACE

The DTI–11B Digital Telemetry Interface design was initiated as part of a digital seismic network experiment at Anza, California — a cooperative project of the USGS and the Institute of Geophysics and Planetary Physics at U.C. San Diego. It was intended to replace an earlier device, called the Telemetry Interface Unit, or TIU, which connects the seismic data stream to a computer system for event detection and recording. The purpose of both these interfaces is to offload the processing overhead from the computer system for each data sample as it arrives over a microwave telemetry system.

While the DTI–11B was being developed, the manufacturer of the seismic data acquisition system used at Anza (Refraction Technology, Dallas, Texas) announced a new telemetry option which offered better utilization of the microwave telemetry link and synchronized inter-station data sampling. As a result, the decision was made to abandon further work on the DTI–11B in favor of converting the seismic data acquisition system to use the new telemetry option.

Even though there will be no further development of the DTI–11B, a fully functioning PC-board prototype has been produced, and an RSX–11 device driver has been written and debugged for use on a development PDP–11/73 computer system in the laboratory. This prototype hardware/software system demonstrates the novel features of the device, such as buffer chaining, and buffer swapping and data sample buffer offset calculation using the 1 Hz and 1 KHz output of a radio time-code receiver, respectively. These features are all more fully described in the text.

To provide as much useful documentation to others who may wish to design a similar device, we have attempted to describe as completely as possible the design of the DTI–11B up to the prototype stage. Readers interested in obtaining more information, including copies of the RSX–11 device driver, may contact the author at

U.S. Geological Survey
345 Middlefield Road MS977
Menlo Park, California 94025
(415) 329-5608 or FTS 459-5608

Lawrence M. Baker
May 10, 1988

CHAPTER 1
INTRODUCTION

1.1 GENERAL

The DTI–11B Digital Telemetry Interface User's Guide provides the following information:

- General Introduction
- Programming Details (Software Interface)
- Description of I/O Signals
- Theory of Operation
- Interfacing and Programming Considerations

1.2 SUPPORTING DOCUMENTATION

The following supporting documents are recommended for use with this User's Guide:

1. DTI–11B Print Set
2. DEC *Microcomputer Products Handbook (Order no. EB26078-41/85-09-07-43)*
3. DEC *Supermicrosystems Handbook (Order no. EB27713-41/86-02-08-36)*
4. Refraction Technology Model REF TEK 24 Remote Digitizer User's Manual
5. Refraction Technology Model REF TEK 44B Remote Interface Unit User's Manual
6. Motorola *STARPOINT* MICROWAVE Communications Equipment, Instruction Manual *(Order no. 68P81045E85-B)*
7. Kinemetrics/TrueTime Operating and Service Manual: Model 60-DC WWVB Synchronized Digital Clock
8. Grant Technology Instruction Manual: Models 364–369 Digital I/O Boards

1.3 FUNCTIONAL DESCRIPTION

The DTI–11B Digital Telemetry Interface is a DMA Q-bus device designed to store remotely sensed seismic data into the memory of a Digital Equipment Corp. (DEC) MicroPDP–11 or MicroVAX computer system.

Figure 1-1 is a block diagram of a typical seismic data acquisition system incorporating a DTI–11B.

Two horizontal components and one vertical component of ground motion are sampled and digitized using Refraction Technology model REF TEK 24 Remote Digitizers. Each 16-bit data sample is combined with additional status bits and two component-identification bits to form a 26-bit data word.

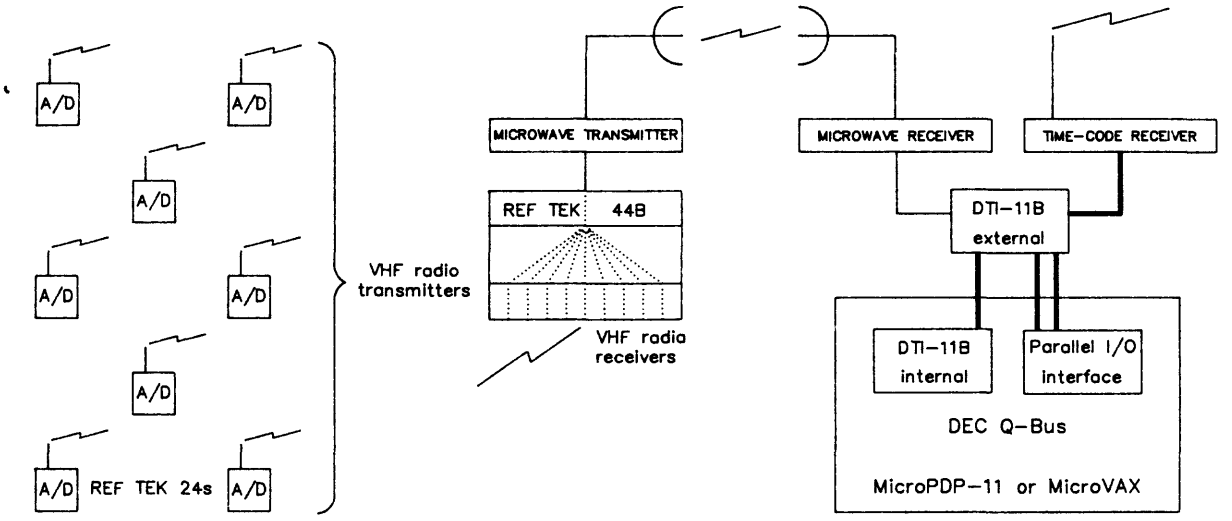


Figure 1-1. Block Diagram of a Typical Seismic Data Acquisition System

Up to 16 Remote Digitizers simultaneously transmit their data words to a REF TEK 44B Remote Interface Unit using line-of-sight VHF radio. The Remote Interface Unit adds parity bits and four station-identification bits to form a 32-bit data word, which is transmitted over a Motorola *STARPOINT* microwave link to a MicroPDP–11 or MicroVAX computer system for reception and analysis. As each 32-bit data word arrives, the DTI–11B deposits the 16-bit data sample into a fixed length buffer (determined by configuration jumpers) in Q-bus memory.

Each data buffer is organized as a matrix, with a row of sample time slots for each of the three components of motion at each field station. Figure 1-2 illustrates the data buffer layout for a matrix of t time slots for n field stations. Using the station and component codes that arrive with each data sample, the DTI–11B transfers each data sample directly to the row corresponding to the originating station and component, and the column corresponding to the elapsed time since the start of the current data buffer. The elapsed time is derived from the 1 Hz and 1 KHz outputs of a time-code receiver (*e.g.*, WWVB or GOES) as a reference.

The computer software loads the starting address of each data buffer into the Base Address and Base Address Extension registers of the DTI–11B, and enables data sampling using the Control and Status register. Since the length of each buffer is fixed, there is no word count register. The DTI–11B interrupts the computer and begins to fill the next buffer, if requested, when the time allocated to fill a buffer has elapsed — called a *buffer swap* interrupt.

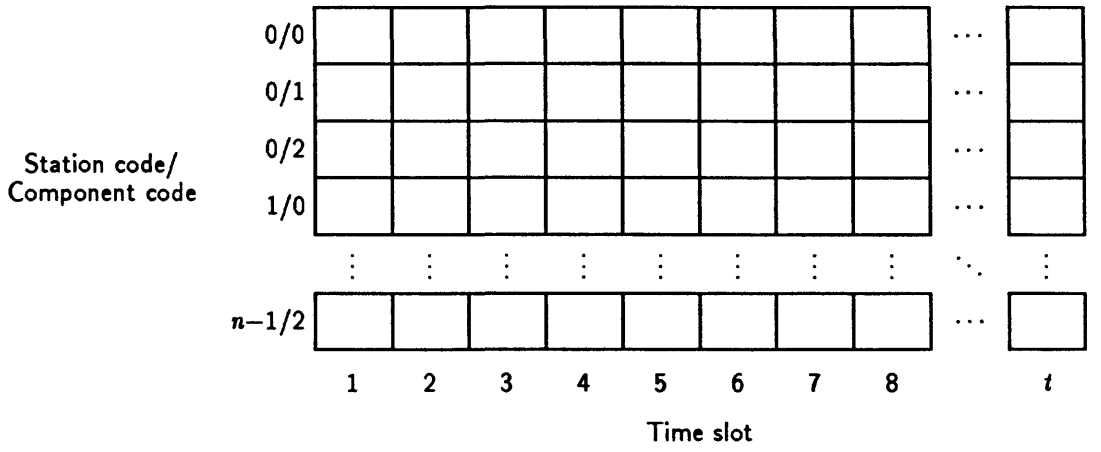


Figure 1-2. Data Buffer (t time slots, n stations)

1.4 SPECIFICATIONS

The DTI–11B is a 22-bit DMA device with four registers on the I/O page. These registers are:

Address	Register Name
$7XXXX0_8$	Base address register (BAR)
$7XXXX2_8$	Base address extension and high station register (BHR)
$7XXXX4_8$	Control and status register (CSR)
$7XXXX6_8$	Buffer size register (BSR)

An unlimited number of DTI–11Bs can be used in a system if the addresses and vector selected for each device do not conflict with the addresses and vectors of other devices.

Two interrupt vectors are associated with the DTI–11B: interrupt A is used to signal an error condition and interrupt B is the buffer swap interrupt. Vector B is located at the vector base address and vector A is located at the vector base address plus 4_8.

The DTI–11B requests an interrupt using priority level BR5.

1.5 PHYSICAL DESCRIPTION

The DTI–11B consists of one quad-height Q-bus module (compatible with both C-D and Q22 backplanes) containing the address computation and Q-bus interface logic, and one external 5"×7" circuit board containing the serial framing and illegal data rejection logic. In addition, a parallel I/O interface board (*e.g.*, a dual-height Grant Technology GT-368) may optionally be used to latch the BCD output of the time-code receiver at each buffer swap for time-stamping.

The DTI–11B connects to the microwave receiver via a coaxial cable and the time-code receiver via two coaxial cables and an optional 50-conductor ribbon cable. The optional parallel I/O interface connects to the DTI–11B via two 40-conductor ribbon cables.

1.6 MAINTENANCE

The internally mounted (Q-bus) portion of the DTI–11B may be connected to a DEC DRV11, or equivalent, in place of the external portion to aid in diagnosing hardware failures.

CHAPTER 2
SOFTWARE INTERFACE

2.1 GENERAL

The DTI–11B uses four programmable registers located on the I/O page. (See Section 1.3 for their respective addresses.) In addition to the registers contained on the DTI–11B, a parallel I/O interface with four additional registers is required if the time-code receiver will be used for time-stamping the data buffers. (The DTI–11B provides an external TTL-compatible buffer swap signal for latching the current clock value.) However, the presence or absence of a parallel I/O interface has no effect on the operation of the DTI–11B.

The DTI–11B is an unusual DMA device in that it has no word count register to determine the size of each data transfer. Instead of placing successive data samples in sequential locations in memory, the DTI–11B selects a location for each data sample according to the station and channel number affixed to each 32-bit data word, and the elapsed time from the start of the buffer currently being filled. A buffer is "full" when the time interval allocated for a buffer has elapsed (fixed by the hardware) — usually synchronized to the half-second. Even if the data stream is interrupted, buffer swaps continue to occur, since the buffer full condition is derived from the 1 Hz and 1 KHz signals supplied by the time-code receiver, independent of the arrival of valid data.

The DTI–11B has another unusual feature for minimizing the risk of data lost due to excessive hardware and/or software delays between I/O operations. The device operates from an *internal* set of registers while the current transfer is in progress, and latches the *external* set when the current transfer completes. The software device driver is then able to keep one step ahead of the DTI–11B by using the time during one transfer to preload the device registers for the next transfer. Programming for this special feature is facilitated by status bits in the CSR register that enable the device driver to coordinate its activities with the current state of the DTI–11B. (See Section 5.2.3.)

2.2 CONTROL AND STATUS REGISTER (CSR)

The CSR (Figure 2-1) is a 16-bit *byte-addressable* register, of which 12 bits are used. It is cleared to all "0"s by INIT when the system is powered up. A software reset can be accomplished by writing a word of all "0"s to it. This is one method that an error interrupt handler may use to reinitialize the interface after determining the source of the error (by inspecting the bits in the CSR).

The high-order four bits of the CSR are for error reporting and error interrupt control. Since this register can be read and written a byte at a time, an error interrupt handler

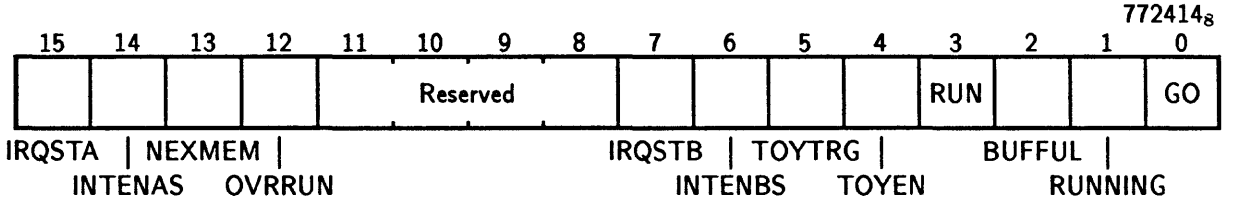


Figure 2-1. Control and Status Register (CSR)

Table 2-1. CSR Register Bit Definitions

Bit	Name	Access[1]	Function
0	GO	R/W	The GO bit is used to start and stop data acquisition gracefully. After writing a "1" to this bit, the interface will wait until the start time for the next buffer (buffer swap) and then set the RUNNING bit to indicate that data acquisition is in progress. If a "0" is written to GO at some later time, the interface will wait until the next buffer swap before resetting RUNNING to gracefully terminate sampling (finishing the buffer which was being filled when GO was cleared).
1	RUNNING	R/W	The RUNNING bit is set whenever the interface is actually acquiring data and storing it into memory. It can be set as a result of setting GO (as discussed above), or by setting RUN (discussed below). If a "0" is written to RUNNING, data acquisition stops immediately. (If GO is set it will start again automatically at the next buffer swap.) Writing a "1" to RUNNING has no effect, so read/modify/write instructions (BIS and BIC on a PDP–11, or BISW and BICW on a VAX) may safely be used to modify the CSR without disturbing a transfer in progress. (See also RUN.)
2	BUFFUL	R/W	The BUFFUL bit is an indication that a buffer (or partial buffer) was filled before the most recent buffer swap. It is simply an indication that the interface was in fact running when the buffer swap occurred. It can be cleared by writing it with "0". However, writing a "1" to this bit has no effect. Since an interrupt can occur even on the first buffer swap (when no data has been acquired yet) this bit can be used by the interrupt handler to determine if a buffer has been filled and is available for further processing.
3	RUN	WO	The RUN bit is used to request an immediate startup of sampling, without waiting for the next buffer swap. Writing a "1" to RUN causes an immediate startup. Writing a "0" to RUN has no effect. (The DTI–11B should be idle when RUN is set — any transfer in progress is abruptly terminated, and the buffer swap request bit (IRQSTB) is not set.) RUN always reads as a "0", so read/modify/write instructions may safely be used to modify the CSR without disturbing a transfer in progress. (See also RUNNING.)
4	TOYENS	R/W	The TOYENS bit is used to enable and disable the time-of-year (TOY) clock trigger circuit. When set, the trigger circuit is enabled and will generate a TTL-compatible external pulse that can be used to latch the TOY clock value for the time of day at each buffer swap. When cleared, the circuit is disabled.
5	TOYTRG	R/W	The TOYTRG bit is used to force the TOY clock latch signal. If, for example,

[1]RO: read-only, R/W: read/write, WO: write-only.

Table 2-1 CSR Register: Bit Definitions (con't)

Bit	Name	Access[1]	Function
			the TOY clock value is latched on the rising edge of the trigger signal, the time can be latched manually by writing a "0", followed by a "1" to TOYTRG (provided TOYENS is set). The hardware on the interface can only set this bit automatically at buffer swap time. To get an accurate time latched, it must have been previously cleared (presumably when the registers were set up for the data transfer).
6	INTENBS	R/W	The INTENBS bit enables or disables the buffer swap interrupt (interrupt B). When set, interrupts will be issued at each buffer swap (including the first one after GO is set) while the interface is running. When reset, interrupt B is disabled. (This does not affect the operation of the IRQSTB bit, however.)
7	IRQSTB	R/W	The IRQSTB bit is the buffer swap interrupt request (interrupt B). It is set to "1" whenever a buffer swap occurs while the interface is running (including the first one after GO is set). Its operation is independent of the state of the INTENBS bit. (This is for software which does not wish to enable interrupts, but wants to poll IRQSTB instead.) If interrupts are enabled, this bit must be reset (by writing a "0" to it) prior to leaving the interrupt handler, or another interrupt will be issued immediately. A buffer swap interrupt can be issued manually by writing a "1" to this bit (useful for debugging the interrupt handler).
8–11			Reserved.
12	OVRRUN	RO	When set, the OVRRUN bit indicates that an overrun condition has been detected by the interface. This occurs when a new data sample is framed in the shift register before the DMA transfer of the previously received sample has completed. (The most likely cause of this error is another, perhaps higher priority, DMA device failing to relinquish the bus in time for the transfer to complete before the next data sample arrives. The processor will always relinquish the bus in a timely manner to DMA bus masters.) OVRRUN is cleared by writing a "0" to IRQSTA.
13	NEXMEM	RO	When set, the NEXMEM bit indicates that a DMA reference was made to a nonexistent memory location. This is commonly called a bus timeout, and is detected when a DMA transfer is attempted, but the slave device (memory) does not respond (by raising RRPLY) within 10 microseconds. NEXMEM is cleared by writing a "0" to IRQSTA.
14	INTENAS	R/W	The INTENAS bit enables or disables error interrupts (interrupt A). When set, an interrupt will be issued when the IRQSTA combined data transfer error bit is set. When reset, interrupt A is disabled. (This does not affect the operation of the IRQSTA bit, however.)
15	IRQSTA	R/W	The IRQSTA bit is the combined data transfer error interrupt request (interrupt A). It is the logical *OR* of OVRRUN and NEXMEM, and is set at the same time that either one is set. Writing a "0" to IRQSTA will clear both OVRRUN and NEXMEM, thus resetting IRQSTA. Writing a "1" to this bit has no effect, so it is not possible to force an error interrupt manually. As with IRQSTB, this bit will be set even if INTENAS is clear, so polling for error conditions is possible. The interface will refuse to set GO or RUNNING whenever IRQSTA is set.

[1]RO: read-only, R/W: read/write, WO: write-only.

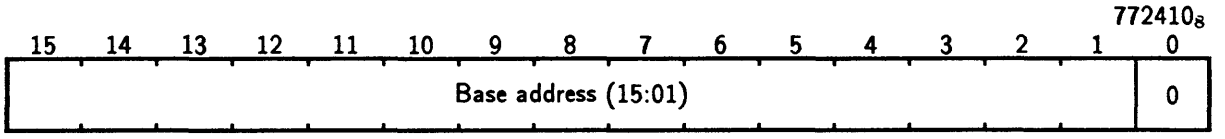


Figure 2-2. Base Address Register (BAR)

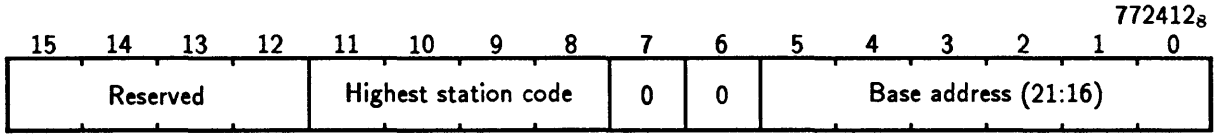


Figure 2-3. Base Address Extension and High Station Register (BHR)

need only concern itself with the MSB of the CSR, as the remaining bits of this byte are unused. See Table 2-1 for a description of each bit in the CSR and its function.

If the DTI–11B is set up for a base address of 772410_8, the CSR address is 772414_8.

2.3 BASE ADDRESS REGISTER (BAR)

The BAR (Figure 2-2) is a *byte-addressable*, *read/write* register that contains the low-order 16 bits of the 22-bit buffer address where the incoming data is to be stored. The low order bit is forced to be "0" by the hardware, since all data transfers must be to even memory addresses.

If the DTI–11B is set up for a base address of 772410_8, the BAR address is also 772410_8.

2.4 BASE ADDRESS EXTENSION AND HIGH STATION REGISTER (BHR)

The BHR (Figure 2-3) is a *byte-addressable* register. The LSB of the BHR is a *read/write* register containing the high-order 6 bits of the 22-bit Q-bus buffer address. The two high-order bits are not used and are always read as "0"s.

The MSB of the BHR is a *read-only* register which contains the highest station code number in the low-order 4 bits. These 4 bits determine the highest station from which the interface will accept data and store it into memory. They are set with jumpers on the external board prior to installation. The high-order 4 bits are left floating and should be considered *undefined*.

A word write to the BHR is allowed — the MSB will simply be ignored. If the DTI–11B is set up for a base address of 772410_8, the BHR address is 772412_8.

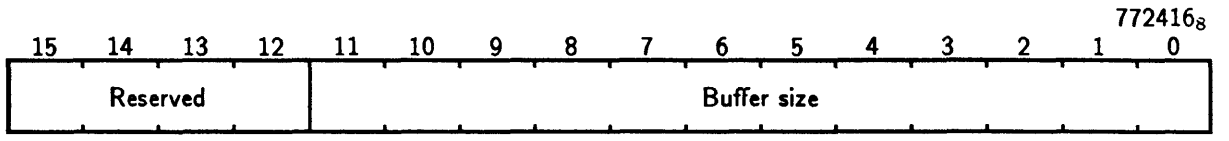


Figure 2-4. Buffer Size Register (BSR)

2.5 BUFFER SIZE REGISTER (BSR)

The BSR (Figure 2-4) is a *read-only* register whose low-order 12 bits give the size (in bytes) of the buffer which will be filled for each channel of data being acquired. The high-order 4 bits are left floating and should be considered *undefined*.

Note that this is not a true indication of the amount of memory that must be *allocated* for each channel buffer, it is only the actual number of bytes which will be filled. The channel buffers are always spaced a power of 2 apart in memory (regardless of the base address chosen). By using the value in the BSR and that found in the MSB of the BHR, a program can find out all it needs to know about how the interface is currently strapped so that buffers of the appropriate size can be allocated. (See Section 5.2.2.)

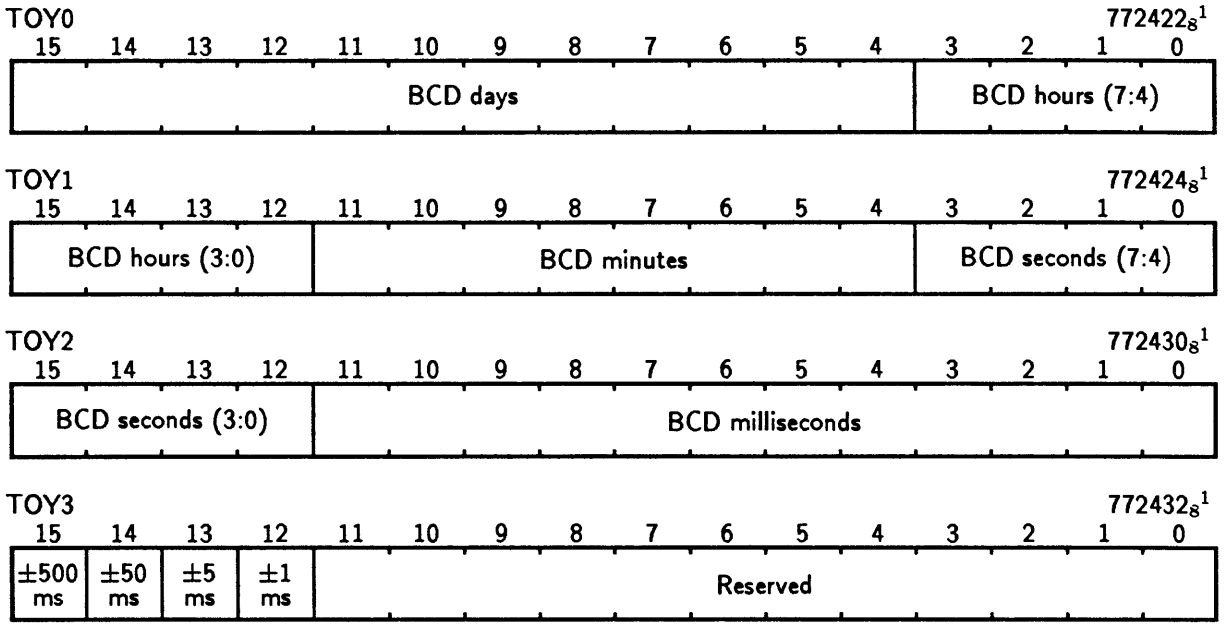
If the DTI–11B is set up for a base address of 772410_8, the BSR address is 772416_8.

2.6 TIME-OF-YEAR CLOCK INTERFACE REGISTERS (TOY0–TOY3)

If the time-code receiver will be used for time-stamping the data buffers, four 16-bit *read-only* registers (Figure 2-5) are required to latch the 12 BCD time digits and the four accuracy bits provided by the time-of-year clock when a buffer swap occurs. (The DTI–11B provides the TOYLAT signal on an external connector, which is used to strobe the BCD time-code into the data buffers on the parallel I/O interface.)

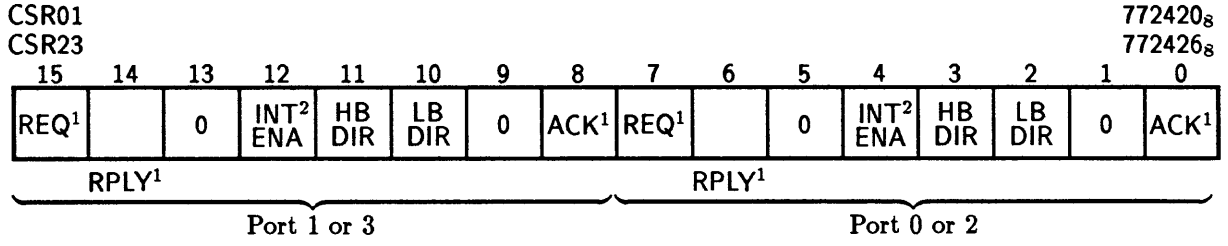
The base address for TOY0–TOY3 is determined by the address selected on the particular parallel I/O interface used. It is preferable to locate them immediately after the last DTI–11B register (BSR). If a GT-368 is used, the recommended base address is 772420_8. In that case, TOY0–TOY3 will be located at 772422_8, 772424_8, 772430_8, and 772432_8, respectively.

The GT-368 uses four bits in each of two CSR registers (Figure 2-6) to configure each byte of the four bidirectional 16-bit parallel ports for input or output (CSR01 for PORT0 and PORT1, CSR23 for PORT2 and PORT3). HB DIR and LB DIR control the transfer direction ("0" for input, "1" for output) for each high byte and low byte, respectively. At power-up, all four ports are initialized for 16-bit input. Therefore, it is not necessary to modify the CSRs. (In fact, they must *not* be modified for the time-of-year clock interface to work properly.)



[1]GT-368 PORT0–PORT3 addresses.

Figure 2-5. Time-of-Year Clock Interface Registers (TOY0–TOY3)



[1]Not used.
[2]Model GT-364 only (not used).

Figure 2-6. GT-368 Control and Status Registers (CSR01 and CSR23)

CHAPTER 3
I/O SIGNALS

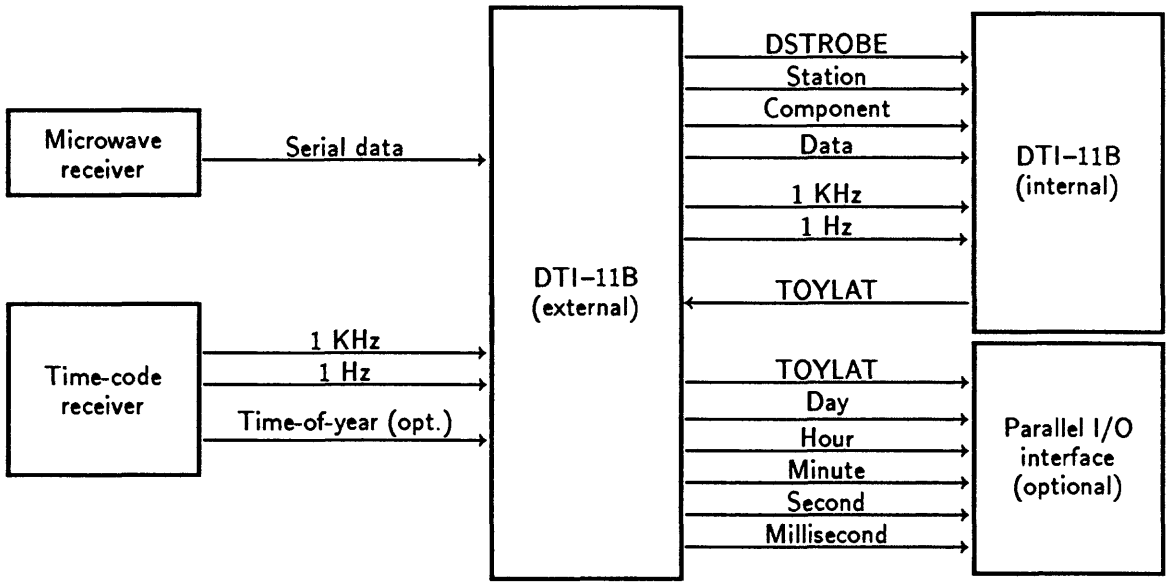


Figure 3-1. Block Diagram of External Board Connections

Figure 3-1 is a block diagram of the I/O signals and additional circuit boards required to assemble a fully functioning data acquisition system. (Suppliers for the additional components are given in the Appendix.)

The external portion of the DTI-11B connects to the microwave receiver using a coaxial cable, to the time-code receiver using two coaxial cables and an optional 50-conductor ribbon cable, and to the optional parallel I/O interface using two 40-conductor ribbon cables. The internal (Q-bus) portion of the DTI-11B connects to the external portion of the DTI-11B using a 40-conductor ribbon cable.

Table 3-1 contains a description of all input signals to the external portion of the DTI-11B. Table 3-2 contains a description of all output signals from the external portion of the DTI-11B. Table 3-3 contains a description of the signals carried between the internal and external portions of the DTI-11B. All signal levels are TTL-compatible (0 to 5 volts), unless otherwise noted.

Table 3-4 contains the signal pin assignments for the 50-pin ribbon cable from the time-code receiver. (Note: Significant bits only of the BCD time-code are provided.) Three signals provided on this cable are not used by the DTI-11B and are left floating: IRIG B, 1 Hz, and 1 KHz. (The 1 Hz and 1 KHz signals required by the DTI-11B are provided

Table 3-1. DTI–11B Input Signals

Signal	Function
From the microwave receiver:	
Serial data	Serial data stream (±500 mV max, DC to 500 KHz).
From the time-code receiver:	
1 KHz	1 KHz reference signal, rising edge on time.
1 Hz	1 Hz reference signal, rising edge on time.
Time-of-year	BCD time-of-year, significant bits only (optional).

Table 3-2. DTI–11B Output Signals

Signal	Function
To the optional parallel I/O interface:	
TOYLAT	Buffer swap signal used to latch the time-of-year supplied by the TOY clock. This allows the interrupt handler to read the correct buffer-swap time from the parallel I/O interface regardless of the latencies introduced by the computer hardware, the operating system, and other system activity.
Day	3-digit BCD day of year (0–366).
Hour	2-digit BCD hour of day (0–23).
Minute	2-digit BCD minute of hour (0–59).
Second	2-digit BCD second of minute (0–59).
Milliseconds	3-digit BCD milliseconds of second (0–999).

Table 3-3. DTI–11B Inter-Board Signals

Signal	Function
To the internal portion of the DTI–11B:	
DSTROBE	New data ready signal.
Station	4-bit station code (0–15).
Component	2-bit component code (0–2).
Data	16-bit data sample.
1 KHz	1 KHz reference signal, rising edge on time.
1 Hz	1 Hz reference signal, rising edge on time.
From the internal portion of the DTI–11B:	
TOYLAT	Buffer swap signal (see Table 3-2).

Table 3-4. Time-of-Year Clock Input Connector Pin Assignments

Pin	Signal	Pin	Signal
1	Ground	26	1's of 10's of minutes
2	IRIG B[1]	27	8's of units of minutes
3	2's of 100's of days	28	4's of units of minutes
4	1's of 100's of days	29	2's of units of minutes
5	8's of 10's of days	30	1's of units of minutes
6	4's of 10's of days	31	4's of 10's of seconds
7	2's of 10's of days	32	2's of 10's of seconds
8	1's of 10's of days	33	1's of 10's of seconds
9	1 KHz[1]	34	8's of units of seconds
10	8's of units of days	35	4's of units of seconds
11	4's of units of days	36	2's of units of seconds
12	2's of units of days	37	1's of units of seconds
13	1's of units of days	38	8's of 100's of milliseconds
14	±5 ms accuracy	39	4's of 100's of milliseconds
15	±50 ms accuracy	40	2's of 100's of milliseconds
16	1 Hz[1]	41	1's of 100's of milliseconds
17	±500 ms accuracy	42	8's of 10's of milliseconds
18	2's of 10's of hours	43	4's of 10's of milliseconds
19	1's of 10's of hours	44	2's of 10's of milliseconds
20	8's of units of hours	45	1's of 10's of milliseconds
21	4's of units of hours	46	8's of units of milliseconds
22	2's of units of hours	47	4's of units of milliseconds
23	1's of units of hours	48	2's of units of milliseconds
24	4's of 10's of minutes	49	1's of units of milliseconds
25	2's of 10's of minutes	50	±1.0 ms accuracy

[1]Not connected (floating).

on two separate coaxial cables.) This cable is required only if the BCD output of the time-code receiver will be used at each buffer swap for time-stamping. All signal levels are TTL-compatible (0 to 5 volts).

Tables 3-5 and 3-6 contain the signal pin assignments for the two 40-pin ribbon cables to the optional parallel I/O interface (*e.g.*, Grant Technology GT-368). The high order bits of the BCD time-of-year not provided by the time-code receiver are connected to ground (logic level "0"). The ACKNOWLEDGE signals from the GT-368 are not used and are left floating. All signal levels are TTL-compatible (0 to 5 volts).

Table 3-5. Parallel I/O Interface Output Connector A Pin Assignments

Pin	DTI–11B Signal	GT-368 Signal
1	Ground	Port 0 Bit 15 (MSB)
2	8's of 10's of days	Port 0 Bit 11
3	Ground	Port 0 Bit 3
4	8's of units of days	Port 0 Bit 7
5	1's of units of days	Port 0 Bit 4
6	1's of 10's of hours	Port 0 Bit 0 (LSB)
7	1's of 10's of days	Port 0 Bit 8
8	1's of 100's of days	Port 0 Bit 12
9	Ground	Port 0 Bit 14
10	4's of 10's of days	Port 0 Bit 10
11	Ground	Port 0 Bit 2
12	4's of units of days	Port 0 Bit 6
13	2's of units of days	Port 0 Bit 5
14	2's of 10's of hours	Port 0 Bit 1
15	2's of 10's of days	Port 0 Bit 9
16	2's of 100's of days	Port 0 Bit 13
17	8's of units of hours	Port 1 Bit 15 (MSB)
18	Ground	Port 1 Bit 11
19	Ground	Port 1 Bit 3
20	8's of units of minutes	Port 1 Bit 7
21	1's of units of minutes	Port 1 Bit 4
22	1's of 10's of seconds	Port 1 Bit 0 (LSB)
23	1's of 10's of minutes	Port 1 Bit 8
24	1's of units of hours	Port 1 Bit 12
25	4's of units of hours	Port 1 Bit 14
26	4's of 10's of minutes	Port 1 Bit 10
27	4's of 10's of seconds	Port 1 Bit 2
28	4's of units of minutes	Port 1 Bit 6
29	2's of units of minutes	Port 1 Bit 5
30	2's of 10's of seconds	Port 1 Bit 1
31	2's of 10's of minutes	Port 1 Bit 9
32	2's of units of hours	Port 1 Bit 13
33	TOYLAT	Port 1 REQUEST
34	TOYLAT	Port 0 REQUEST
35	N/C[1]	Port 0 ACKNOWLEDGE
36	Ground	Port 0 REPLY
37	Ground	Port 1 REPLY
38	N/C[1]	Port 1 ACKNOWLEDGE
39	Ground	Ground
40	Ground	Ground

[1]Not connected (floating).

Table 3-6. Parallel I/O Interface Output Connector B Pin Assignments

Pin	DTI–11B Signal	GT-368 Signal
1	8's of units of seconds	Port 2 Bit 15 (MSB)
2	8's of 100's of milliseconds	Port 2 Bit 11
3	8's of units of milliseconds	Port 2 Bit 3
4	8's of 10's of milliseconds	Port 2 Bit 7
5	1's of 10's of milliseconds	Port 2 Bit 4
6	1's of units of milliseconds	Port 2 Bit 0 (LSB)
7	1's of 100's of milliseconds	Port 2 Bit 8
8	1's of units of seconds	Port 2 Bit 12
9	4's of units of seconds	Port 2 Bit 14
10	4's of 100's of milliseconds	Port 2 Bit 10
11	4's of units of milliseconds	Port 2 Bit 2
12	4's of 10's of milliseconds	Port 2 Bit 6
13	2's of 10's of milliseconds	Port 2 Bit 5
14	2's of units of milliseconds	Port 2 Bit 1
15	2's of 100's of milliseconds	Port 2 Bit 9
16	2's of units of seconds	Port 2 Bit 13
17	±500 ms accuracy	Port 3 Bit 15 (MSB)
18	Ground	Port 3 Bit 11
19	Ground	Port 3 Bit 3
20	Ground	Port 3 Bit 7
21	Ground	Port 3 Bit 4
22	Ground	Port 3 Bit 0 (LSB)
23	Ground	Port 3 Bit 8
24	±1.0 ms accuracy	Port 3 Bit 12
25	±50 ms accuracy	Port 3 Bit 14
26	Ground	Port 3 Bit 10
27	Ground	Port 3 Bit 2
28	Ground	Port 3 Bit 6
29	Ground	Port 3 Bit 5
30	Ground	Port 3 Bit 1
31	Ground	Port 3 Bit 9
32	±5 ms accuracy	Port 3 Bit 13
33	Ground	Port 3 REPLY
34	N/C[1]	Port 3 ACKNOWLEDGE
35	TOYLAT	Port 3 REQUEST
36	N/C[1]	Port 2 ACKNOWLEDGE
37	TOYLAT	Port 2 REQUEST
38	Ground	Port 2 REPLY
39	Ground	Ground
40	Ground	Ground

[1]Not connected (floating).

CHAPTER 4
THEORY OF OPERATION

4.1 GENERAL

Figure 4-1 illustrates the DTI-11B Digital Telemetry Interface signal flow. The DTI-11B receives incoming data from a microwave telemetry link, performs a serial-to-parallel conversion of the individual data samples, and stores the data, demultiplexed by channel, in buffers in Q-bus memory. It works with buffers of fixed length (the choice of buffer length is jumper programmable) which represent fixed intervals in *absolute time.* The starting time for each buffer is aligned to an integral fraction of a second by synchronizing the buffer swap logic to the 1 Hz signal from an external time-of-year (TOY) clock.

The interface is equipped with all of the hardware necessary to perform the DMA transfer of incoming data to memory. Logic is provided for the calculation of physical memory addresses for the storage of each data sample. It is also equipped with interrupt logic which can be activated for error reporting and to announce the completion of each buffer load. It incorporates a TOY clock trigger circuit which will prompt an external parallel I/O interface to record the exact time of day at which each buffer swap occurs.

In the discussion which follows, individual sections of the interface are discussed by name. These names appear (underlined twice) above the appropriate section of the schematic diagram. The board is designed around either of two equivalent logic families: the 74LS series of *low-power Schottky TTL*, or the newer 74HC series of pin-compatible *high-speed CMOS* parts. When mentioning parts on the board by number, the notation 74XXnnn will be used, where nnn identifies the part.

4.2 SERIAL DISCRIMINATOR

The incoming telemetry stream comes onto the interface via a chassis-mounted BNC connector. It is amplified by an LF-357A IC operational amplifier. The amplified signal is AC coupled to the inputs of two LM-339 voltage comparators. These devices detect the positive and negative voltage swings in the signal. Their output is fed to a summing junction and drives a third LM-339, which serves as a level shifter. The output of this third stage is a bit stream which swings between 0 and 5 volts (TTL-compatible levels).

4.3 SHIFT REGISTER

The serial bit stream from the discriminator is fed to the input of a 32-stage shift register which is fabricated from four 74XX164 8-bit devices. The shift registers are clocked by the X1CLK signal produced by dividing a 6.144 MHz oscillator signal by 16. The oscillator

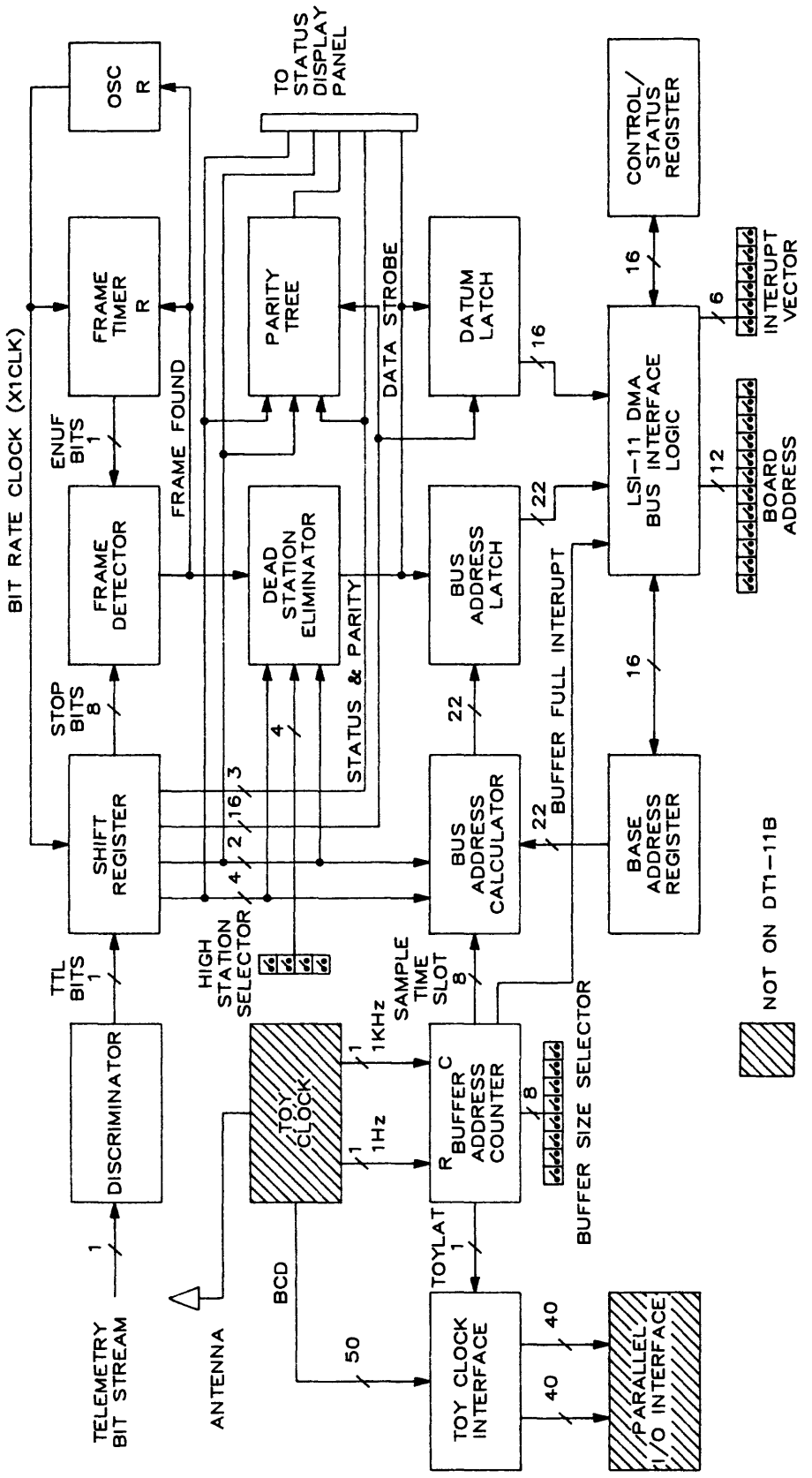


Figure 4-1. DTI-11B Telemetry Interface Block Diagram

is composed of a PI circuit, which uses two stages of a CD4001A CMOS NOR gate. The division by 16 is performed by a 74XX191 4-bit binary counter.

Three conditions must be met for the circuit to recognize that a new sample has been framed in the shift register:

1. The 6 most recently received bits must all be "1"s
2. 30 bit times must have elapsed since the last frame
3. The serial input must return to the low state

The first condition is met when six of the inputs to a 74XX30 8-input NAND gate are high. The second condition is enforced by the FRAME TIMER circuit, which is composed of two 74XX191 binary counters. These counters are configured to count in the downward direction from a starting count of 30. The count is reloaded and the counters restarted each time a frame is detected. When both counters count out (reach 0) a pulse sets a set/reset flip-flop formed from two 74XX02 NOR gates. This causes a signal called ENUF BITS to go true, thus allowing the next sample to be framed. The third condition is insured by applying the inverted serial bit-stream signal to the remaining input of the 74XX30 NAND gate.

When all three conditions are met, a new frame of data is present (for ~2 microseconds) on the parallel outputs of the shift register. This triggers a 74XX123 one shot, which generates a pulse (DSYNC) of approximately 200 nanoseconds duration. This pulse resets the set/reset flip-flop (thus negating ENUF BITS) and restarts the FRAME TIMER. It also reloads the counter used to generate the X1CLK signal. This synchronizes the X1CLK with the serial bit stream (by detecting the end of the last stop bit). A DATA STROBE HOLDOFF circuit (74XX00) will stretch this pulse as much as an additional 200 nanoseconds if it arrives any time just before the 1 KHz signal from the TOY clock rises. (This is done because much of the other circuitry involved in processing the data will change state on the rising edge of this clock pulse.)

4.4 SAMPLE TIME SLOT COMPUTATION AND BUFFER SWAP LOGIC

The trailing edge of the 1 KHz signal from the TOY clock is used to drive the BUFFER ADDRESS COUNTER circuit, composed of three 74XX191 4-bit up/down counters. The low order two bits of the counter are ignored, so there are 250 sample time slots per second. When the contents of the BUFFER ADDRESS COUNTER reaches a specified (switch settable) value (determined by three 74XX85 4-bit comparators), a BUFFER SWAP INTERRUPT is generated. The BUFFER SWAP INTERRUPT resets the BUFFER ADDRESS COUNTER, thus starting a new sequence of relative buffer addresses.

A BUFFER SWAP INTERRUPT also occurs on trailing edge of the 1 Hz signal from the TOY clock, which forces the BUFFER ADDRESS COUNTER to synchronize buffer time slots to the second. Thus, to obtain buffers of equal length, the limit for the BUFFER ADDRESS COUNTER should be set to 50, 125, or 250, depending on the channel spacing selected to form the data sample memory address (see next section).

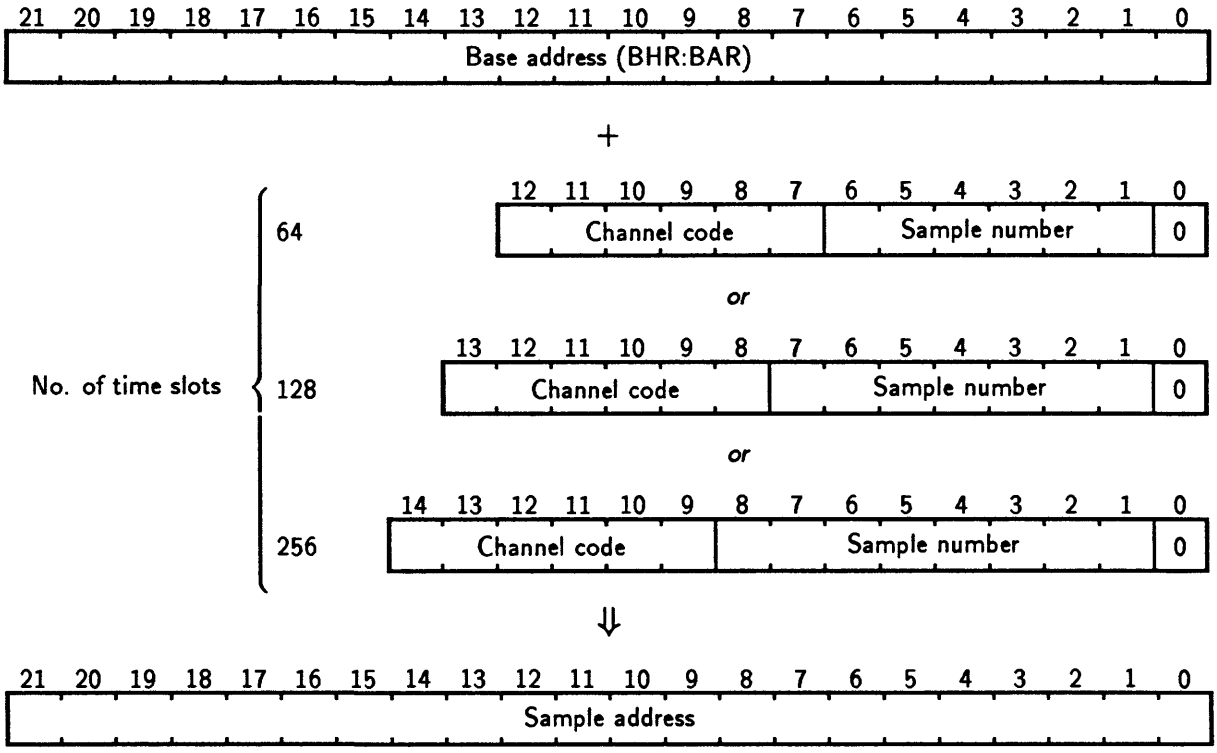


Figure 4-2. Data Sample Address Construction

4.5 MEMORY ADDRESS COMPUTATION

Physical memory addresses, at which data are to be stored, are computed from four sources of information:

1. The 4 station code bits supplied by the relay station (REF TEK 44B Remote Interface Unit)
2. The 2 component bits sent by the remote station (REF TEK 24 Remote Digitizer)
3. From 6 to 8 bits from the buffer counter
4. The 22 base address bits supplied from the BHR:BAR

These data are combined to produce a 22-bit physical memory address (Figure 74XX374 tri-state latches) by the trailing edge of the data strobe (DSTROBE).

First, the 4-bit station code and the 2-bit component code are combined by the NO MISSING CODES circuit to form a 6-bit channel code. This circuit is constructed with two 74XX283 4-bit binary adders. It performs the following calculation:

$$CH(5:0) = 3 \times SC(3:0) + CC(1:0)$$

Where:
CH is the 6-bit channel code
SC is the 4-bit station code
CC is the 2-bit component code

This result is combined with the low-order relative address bits produced by the BUFFER ADDRESS COUNTER circuit. There are three jumper selectable choices for how this combination is done. Either 6, 7, or 8 bits from the counter are used (depending on the desired size of each channel buffer). (This implies that the channel spacing must be either 64, 128, or 256 samples, which correspond to buffer swap intervals of 1/5, 1/2, and 1 second, respectively.) The 6 channel bits start just above the last counter bit to form the 13-, 14-, or 15-bit relative address. An implied "0" least significant bit ensures that data are stored at word addresses on the Q-bus.

This relative buffer address is fed to the "B" inputs of a 32-bit binary adder (BASE ADDRESS ADDER) formed from four 74XX283s. The "A" inputs are driven with the contents of the SECONDARY BASE ADDRESS REGISTER.

The base address for a buffer is first written by the program using the interface into the PRIMARY BASE ADDRESS REGISTER. The mechanism for reading and writing the various registers on the interface board will be discussed later. Each time a buffer swap occurs, the contents of this register are copied to the SECONDARY BASE ADDRESS REGISTER. This allows a program to write the address which will be used for the next buffer into the primary register any time while the current buffer is being filled, thus removing the requirement for an interrupt handler to supply a new base address in the narrow time window between the acknowledgement of the buffer swap interrupt and the arrival of the next data sample.

4.6 DATA ACCEPTANCE AND VALIDATION

Each time a new datum is framed, it is checked to see if it should be stored in memory by the interface. This checking consists of determining first if the station code falls within the range of stations for which data is currently being acquired. The highest acceptable station code is jumper programmable and is compared with the incoming station codes by a 74XX85 4-bit magnitude comparator which forms part of the DEAD STATION ELIMINATOR circuit. The other portion of this circuit checks that no datum is allowed if it has a component code of 3 ("11" binary). This occurs as a result of telemetry errors or because a receiver card for a lower-numbered station has been unplugged at the relay station. If the station code and component code bits check, the datum is allowed to pass through the system by passing the DSYNC framing pulse on as DSTROBE. As discussed above, this is the signal used to latch the memory address into the DMA ADDRESS LATCH. It also is used to latch the 16-bit data sample into the DMA DATA LATCH. The trailing (rising) edge of the pulse is used to allow the address computation circuitry time to settle.

4.7 DATA TRANSFER LOGIC

The data strobe pulse (DSTROBE) which latched in the new datum is also used to initiate a DMA transfer on the Q-bus. Whenever the RUNNING signal is true (see Table 2-1), the trailing edge of DSTROBE will set the DMA REQUEST FLIP-FLOP. This sends DMAREQ to the DC010 DMA protocol chip. This integrated circuit contains nearly all of the logic required to carry out DMA transactions on the bus. The timing delays for the generation

of the various bus signals are handled within this chip using binary counter logic. This logic is clocked by an external ~8 MHz free-running oscillator which is made from a 74XX132 Schmidt trigger NAND gate. As soon as bus mastership has been obtained by the DC010, the DMA REQUEST FLIP-FLOP is reset (only one DATO bus cycle is done and then bus mastership is relinquished).

At the start of the DATO cycle (the address phase) the DMA ADDRESS LATCH tri-state outputs are enabled, thus placing the memory address on the bus. At the same time the NON-EXISTENT MEMORY BUS TIMEOUT logic is fired. This consists of a 74XX123 one-shot set for a pulse width of approximately 10 microseconds. When the trailing (rising) edge of this pulse arrives, it clocks a 74XX74 "D" flip-flop. If at this time the slave device has not responded by raising RRPLY, a "1" will be clocked in which will set the error signal NEXMEM true. If RRPLY arrives in time, it will reset (shorten) the pulse from the 74XX123 and set the "D" input to the flip-flop low, thus averting the assertion of NEXMEM.

Overrun detection is accomplished by noticing if a DMA cycle has been requested or is in progress. If a new datum is framed (another DSTROBE pulse arrives) and the previous DMA transaction has not completed, the OVERRUN DETECTION circuit will assert OVRRUN. This flip-flop can only be reset by software (presumably in an error-interrupt handler).

If either NEXMEM or OVRRUN go true, IRQSTA will be true. If the error interrupt is enabled (INTENAS true), an interrupt is issued. Whether or not the error interrupt is issued, the interface is shut down and data acquisition ceases immediately. This is done by resetting both the GO and the RUNNING flip-flops.

4.8 INTERRUPT GENERATION

The interface is capable of generating two types of interrupts. As discussed above, one (interrupt A) is associated with either an overrun error or a nonexistent memory error. The other interrupt (interrupt B) happens at each buffer swap. The generation of the interrupt A request was already discussed.

The BUFFER SWAP INTERRUPT flip-flop is responsible for generating interrupt B. It is set at buffer swap time whenever the interface is already running (acquiring data) or when it is about to start running (GO is true). If INTENBS is true, an interrupt will be generated with the appropriate vector being placed on the bus during the interrupt acknowledge phase.

The interrupt enable flip-flops for both interrupt A and interrupt B reside inside of the DC003 interrupt protocol chip. This chip, when teamed with the DC005 bus tranceiver chips, contains most of the logic required to issue two different priority interrupts and place the appropriate interrupt vectors on the bus. The base interrupt vector is strapped (switch selectable) on several of the pins on the four DC005 bus tranceivers. The circuit is arranged so that the second vector points to an address which is four higher than the base vector. Vector B is the base vector and vector A is the base vector plus 4_8. Bits 0 and 1 of these vectors are constrained to be "0", and bits 3 through 8 can be selected.

All higher-order bits are also "0", thus the vector base points to a location in the range 000_8–770_8 in memory.

4.9 ON-BOARD REGISTERS

The PRIMARY BASE ADDRESS REGISTER (BAR) and many of the bits in the CONTROL/STATUS REGISTER (CSR) have already been discussed. The two other registers available are the BASE ADDRESS PLUS HIGH STATION REGISTER (BHR) and the BUFFER SIZE REGISTER (BSR). Much of the mechanism for reading and writing to these registers is provided by the DC004 bus protocol chip.

The DC005 bus transceivers also provide a BOARD ADDRESS MATCH circuit, whereby incoming bus addresses are compared with the base register address for the board which is strapped (switch selectable) on several pins of the DC005s. When an address match is made, the BOARD ADDRESS MATCHsignal will remain high (it is an open collector signal which can be wire-*OR*ed from several DC005s). This enables the DC004 register select logic (which samples the other bus control signals) to determine which register is being selected and whether a DATI or DATO (read or write) cycle is to be performed.

Some external REGISTER SELECT LOGIC is required to complete the generation of the various signals which will clock latches or enable buffers. This consists of some 74XX27 triple-input NOR gates and some 74XX02 two-input NOR gates and 74XX04 inverters. The DALnn lines, which are the internal data/address lines on the board, form a tri-state bus which is connected to the Q-bus interface. During a DATI cycle (register read) the appropriate tri-state buffer logic (74XX244 or 74XX374) is simply enabled. During DATO cycles (register writes) the signals produced are suitable for clocking "D" flip-flops or transparent latches.

4.10 Q-BUS INTERFACE

Much of the Q-bus interface logic has already been discussed. The DC004 bus protocol chip provides and interprets bus control signals for register reads and writes. The DC005 bus tranceivers buffer the low-order 16 data/address lines and provide address and vector selection. To determine whether the bus drivers or the bus receivers in the DC005s are enabled, a small amount of external logic (two 74XX132 NAND gates, a 74XX27 triple-input NOR gate, and a 74XX04 inverter) is used to drive the XMIT and REC lines.

The remainder of the bus interface consists of an 8641 bus transceiver and a couple of 8881 bus drivers to drive the high-order address lines during DMA bus transactions.

4.11 EXTERNAL NETWORK STATUS DISPLAY

To accommodate the monitoring of various signals associated with the state of health of the remote stations or of the telemetry link, some additional logic on the interface is used. This allows the connection of an external status display unit to the interface via a connector at the rear of the external board.

The most involved logic monitors the microwave link parity errors. The MPERR signal results when the 24-bit plus parity check performed by the PARITY TREE fails. This

logic uses three 74XX180 parity generator/checkers. This signal, along with the others needed by the status display, is buffered and brought to the connector.

The remaining logic required to produce a meaningful display of the station battery-low bits and the uplink parity-error bits received with each data sample resides in the external display unit itself. The signals needed to drive this logic are buffered by the DISPLAY PANEL BUFFERS and brought to the connector.

4.12 TIME-OF-YEAR CLOCK INTERFACE

The interface includes a simple circuit that routes the BCD digits supplied by the time-code receiver to the connector for the optional Q-bus parallel I/O interface (required only if the software uses the BCD digits from the time-code receiver to time stamp the data buffers).

Also supplied to the same connector is the TOYLAT signal (set by either a buffer swap request, or by setting TOYTRG in the CSR), which may be used by the parallel I/O interface to latch the time from the time-code receiver, either at the same time a buffer swap occurs, or at the request of the device handler. The TOYLAT signal is enabled by setting TOYENS in the CSR.

CHAPTER 5
INTERFACING AND PROGRAMMING CONSIDERATIONS

5.1 TIME-OF-YEAR CLOCK SELECTION

In order to function, the DTI–11B requires a 1 Hz and a 1 KHz signal source. These can be generated by a time-code receiver (*e.g.,* using either the WWVB or GOES radio broadcast time signal), or by a locally operated master clock. If the same clock is to be used for time-stamping the buffer swaps, it must provide a TTL-compatible source of BCD digits for the day, hour, minute, second, and millisecond. These are then read by the interrupt handler through a parallel I/O interface, which must allow for external latching by the DTI–11B TOYLAT signal. See Chapter 3, *I/O Signals*, for a complete description of these signals.

5.2 PROGRAMMING CONSIDERATIONS

This section contains general information for programming the DTI–11B. Examples are given for use in both RSX–11 and VMS device drivers.

5.2.1 Buffer Alignment

The DTI–11B performs word-aligned, 16-bit transfers to Q-bus memory. The low-order bit of the Base Address Register (BAR) is forced to be "0" to ensure the proper alignment of the data buffer. Thus, it is up to the device handler to reject any transfer requests to misaligned buffers (odd base address).

5.2.2 Buffer Allocation

The memory allocated for each buffer must include enough padding to round the number of sample time slots up to the next higher power of 2 (see Section 4.5). The information needed to perform this calculation is provided in the Base Address Extension and High Station Register (BHR) and the Buffer Size Register (BSR), as follows:

1. Read the contents of the BSR. This gives the number of bytes filled for each channel between buffer swaps.
2. Round up to the next higher power of 2. This yields the spacing between channel buffers in memory, or the size of one channel buffer.
3. Read the MSB of the BHR. This gives the highest station code for which data will be accepted by the interface (numbered from 0).
4. Add 1 for the number of active stations for which channel buffers must be allocated.
5. Allocate three channel buffers for each active station.

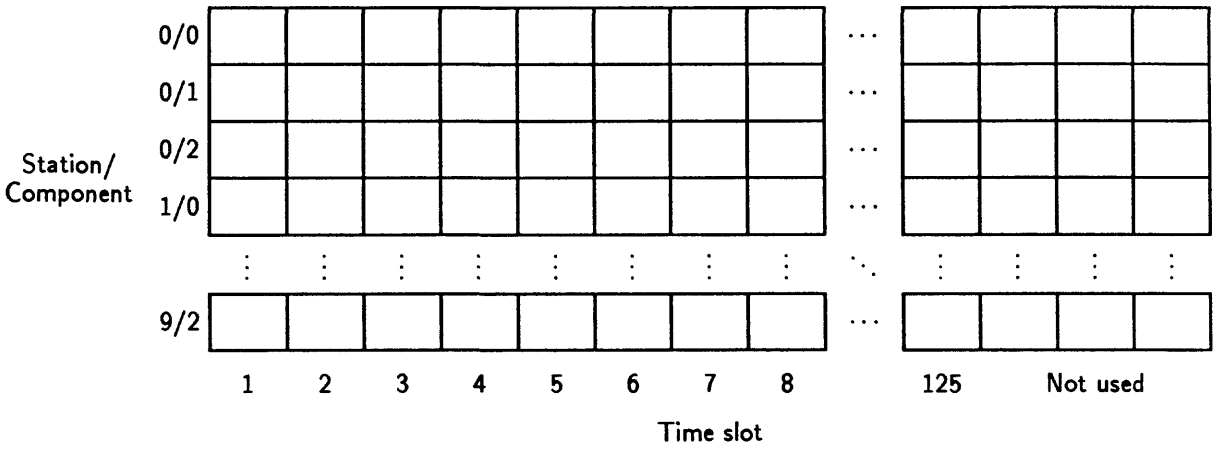


Figure 5-1. Data Buffer Organization (125 time slots, 10 stations)

For example, assume the DTI–11B is configured as shown in Figure 5-1. In that case, the BSR will read 250_{10} (125 samples × 2 bytes per sample) and the MSB of the BHR will read 9_{10}. The program must then allocate 30 channel buffers each spaced 256_{10} bytes apart in memory, for a total buffer size of 7680_{10} bytes. The first channel (station 0, component 0) will begin exactly at the base address of the buffer. The second channel (station 0, component 1) will begin at the base address plus 256_{10} bytes, and so on. The last 6 bytes of each channel buffer will never be written by the interface.

Figure 5-2 is an excerpt from an RSX–11 device driver showing the method for calculating the buffer allocation in PDP–11 assembly language. The calculation also includes an additional four word prefix for the buffer time stamp. The device registers are referred to by their offsets from the CSR, which are prefixed by the device mnemonic, DT, by convention. The results are stored in preallocated locations in a data structure called a Unit Control Block (UCB). They are referred to by their offsets from the start of the UCB, which are of the form U.name, by convention.

Figure 5-3 is an excerpt from a VMS device driver showing the method for calculating the buffer allocation in VAX assembly language. The device registers are referred to by their offsets from the CSR, which are prefixed by the device mnemonic, DT_, by convention. The results are stored in preallocated locations in the UCB. They are referred to by their offsets from the start of the UCB, which are of the form UCB$s_name, by convention, where s is replaced by either "L", "W", or "B" for a longword, word, or byte field, respectively. (Note: Instructions must access I/O page operands using word or byte context — instructions with longword operand context, such as the VAX bit-field operations, are not allowed.)

5.2.3 I/O Request Look-Ahead

In order to fully utilize the capabilities of the DTI–11B to sustain continuous transfers, an I/O request look-ahead capability is necessary so that the device driver can preload the external base address register for the next transfer, or request a graceful halt at the end

```
 1 ;
 2 ; Inputs:  R2 = CSR address
 3 ;          R5 = UCB address
 4 ;
 5 ; Outputs: U.NDAT(R5) = Number of bytes per sample per channel
 6 ;          U.NPAD(R5) = Number of padding words at the end of each channel
 7 ;          U.HSTA(R5) = Highest station number (from 0)
 8 ;          U.BSIZ(R5) = Total buffer size for data and time stamp (in bytes)
 9 ;
10
11 HISTAT  = 177760                    ; Mask for high station register
12 BUFSIZ  = 177400                    ; Mask for buffer size register
13 N$$TOY  =      4                    ; Four words of time stamp
14
15         MOV     DTBSR(R2),R3        ; Number of bytes per sample per channel
16         BIC     #BUFSIZ,R3          ; Mask out lines left floating
17         ASR     R3                  ; Convert to word count
18         MOVB    R3,U.NDAT(R5)       ; Save for reference later
19
20 ; Put next higher power of 2 in R0
21
22         MOV     #1,R0               ; Start with 2**0
23 10$:    CMP     R0,R3               ; 2**n GE sample size?
24         BHIS    20$                 ; If HIS, yes
25         ASL     R0                  ; n = n+1
26         BCC     10$                 ; Try next power of 2
27         JMP     DTFHE               ; Should never get here --
28                                     ;    fatal hardware error
29
30 20$:    MOV     R0,R4               ; Calculate the number of padding
31         SUB     R3,R4               ;    words at the end of each channel
32         MOVB    R4,U.NPAD(R5)       ;    and save for reference later
33
34         MOVB    DTBHR+1(R2),R3      ; Highest station number (from 0)
35         BIC     #HISTAT,R3          ; Mask out unused bits
36         MOVB    R3,U.HSTA(R5)       ; Save for reference later
37
38         INC     R3                  ; Number of stations
39         MUL     #3,R3               ; Number of channels
40         MUL     R0,R3               ; Buffer size, in words
41         ADD     #N$$TOY,R3          ; Add four words for time stamp
42         ASL     R3                  ; Convert to byte count
43         MOV     R3,U.BSIZ(R5)       ; Save for reference later
```

Figure 5-2. PDP-11 Buffer Allocation Computation (RSX-11 version).

of the current transfer. However, the driver must be careful not to reload the external registers until they have been copied to the internal set, which will occur at the next buffer swap.

5.2.4 Interrupts

The device driver must be prepared for an interrupt (IRQSTB) when sampling begins for the first buffer (for synchronized time-stamping), between each buffer (a buffer swap),

```
 1 ;
 2 ; Inputs:  R4 = CSR address
 3 ;          R5 = UCB address
 4 ;
 5 ; Outputs: UCB$B_NDAT(R5) = Number of bytes per sample per channel
 6 ;          UCB$B_NPAD(R5) = Number of padding words at the end of each channel
 7 ;          UCB$B_HSTA(R5) = Highest station number (from 0)
 8 ;          UCB$W_BSIZ(R5) = Total buffer size for data and time stamp (in bytes)
 9 ;
10
11 HISTAT  = ^C< ^XF>                    ; Mask for high station register
12 BUFSIZ  = ^C<^XFF>                    ; Mask for buffer size register
13 N$$TOY  =       4                     ; Four words of time stamp
14
15         MOVZWL  DT_BSR(R4),R3         ; Number of bytes per sample per channel
16         BICL    #BUFSIZ,R3            ; Mask out lines left floating
17         ASHL    #-1,R3,R3             ; Convert to word count
18         MOVB    R3,UCB$B_NDAT(R5)     ; Save for reference later
19
20 ; Put next higher power of 2 in R0
21
22         MOVL    #1,R0                 ; Start with 2**0
23 10$:    CMPL    R0,R3                 ; 2**n GE sample size?
24         BGEQU   20$                   ; If GEQU, yes
25         ASHL    #1,R0,R0              ; n = n+1
26         BCC     10$                   ; Try next power of 2
27         BRW     DTFHE                 ; Should never get here --
28                                       ;    fatal hardware error
29
30 20$:    MOVL    R0,R2                 ; Calculate the number of padding
31         SUBL    R3,R2                 ;    words at the end of each channel
32         MOVB    R2,UCB$B_NPAD(R5)     ;    and save for reference later
33
34         MOVZBL  DT_BHR+1(R4),R3       ; Highest station number (from 0)
35         BICL    #HISTAT,R3            ; Mask out unused bits
36         MOVB    R3,UCB$B_HSTA(R5)     ; Save for reference later
37
38         INCL    R3                    ; Number of stations
39         MULL    #3,R3                 ; Number of channels
40         MULL    R0,R3                 ; Buffer size, in words
41         ADDL    #N$$TOY,R3            ; Add four words for time stamp
42         ASHL    #1,R3,R3              ; Convert to byte count
43         MOVW    R3,UCB$W_BSIZ(R5)     ; Save for reference later
```

Figure 5-3. VAX Buffer Allocation Computation (VMS version).

and at the end of the last buffer. BUFFUL in the CSR (see Table 2-1) is used to distinguish interrupts that signal the completion of a data transfer (BUFFUL is set) from those that signal the synchronized start of the first data transfer in a sequence (BUFFUL is clear).

An error interrupt (IRQSTA) can occur any time during a data transfer, which the error-interrupt handler must be prepared to deal with to prevent erroneous initiation of further sampling at the next buffer swap.

```
1 ;
2 ; 1.  Clear the GO bit immediately until we can check if another transfer can be
3 ;     started (this will have no effect on the current transfer already in pro-
4 ;     gress).  (See also DTSIO.)
5 ; 2.  If this is a buffer swap interrupt (with successful data transfer), save
6 ;     the current CSR contents.
7 ; 3.  If this is a buffer swap or error interrupt propogated from $DTERR, trans-
8 ;     fer the I/O packet at the head of the XFERQueue to the tail of the
9 ;     DONEQueue.
10 ; 4.  Time stamp the I/O packet at the head of the XFERQueue (which is now being
11 ;     filled).  If this was not a packet full interrupt (BUFFUL is clear), set
12 ;     the missing buffers flag in the new packet.
13 ; 5.  Load the device registers and set the GO bit for a subsequent transfer
14 ;     from the second I/O packet in the XFERQueue.
15 ; 6.  If the DONEQueue was previously empty, and has something in it now, call
16 ;     $FORK.  The fork routine completes any packets it finds in the DONEQueue
17 ;     before branching back to get more work to do.
18 ;
19
20          :
21          MOV     (R4),R3                 ;;; Get CSR contents
22
23          BIC     #IRQSTB!TOYEN!TOYTRG!GO,(R4) ;;; Reset interrupt req and GO bits
24
25          MOV     (R0),R1                 ;;; Get address of current I/O packet
26          TSTB    I.STAT(R1)              ;;; Transfer already completed w/error?
27          BMI     10$                     ;;; If MI, yes -- DTERR got us here
28          BIT     #BUFFUL,R3              ;;; Data transfer completed?
29          BEQ     20$                     ;;; If EQ, no -- must be sync interrupt
30          MOV     R3,I.CSR(R1)            ;;; Save copy of CSR contents for IOSB
31
32 10$:      :                              ;;; Remove entry at head of XFERQueue
33
34 ; If another packet is now being filled, write the time stamp into the internal
35 ; I/O packet save area.
36
37 20$:     MOV     (R0),R1                 ;;; Get entry at head of XFERQueue
38          BNE     25$                     ;;; If NE, more packets to fill
39          CLR     (R4)                    ;;; Clear interrupt enable bits
40          BR      30$                     ;;;
41
42 25$:      :
43          MOV     R4,R2                   ;;; $GTTOY wants CSR address in R2
44          CALL    $GTTOY                  ;;; Fill in time stamp
45          MOV     @(R0),R1                ;;; Time to load next set of registers?
46          BEQ     30$                     ;;; If EQ, nothing there
47          CALL    DTSIO                   ;;; Start next I/O transfer
48
49 30$:      :                              ;;; Fork already pending/executing?
50          BNE     40$                     ;;; If NE, yes -- return
51          TST     (R4)                    ;;; Anything in DONEQueue?
52          BNE     50$                     ;;; If NE, yes -- go ahead and fork
53 40$:     RETURN                          ;;; Otherwise just dismiss interrupt
54 50$:     CALL    $FORK                   ;;; Fork
```

Figure 5-4. PDP–11 Buffer Swap Interrupt (IRQSTB) Handler (RSX–11 version).

Figure 5-4 is an excerpt from an RSX–11 device driver showing the method for determining if the buffer swap interrupt is due to the synchronized start of sampling, and whether to preload the (external) device registers for a subsequent transfer or schedule a graceful halt at the end of the current buffer.

APPENDIX A
LIST OF SUPPLIERS

The following vendors supply the additional hardware required to assemble a fully functioning data acquisition system.

Field Instrumentation	Refraction Technology, Inc. 2526 Mañana Dr. Suite 106 Dallas, TX 75220
Microwave Telemetry	Motorola, Inc. Communications Group Component Products Sales & Service 2553 N. Edgington St. Schaumburg, IL 60131
Reference Time Clock	Kinemetrics, Inc. True Time Division 3243 Santa Rosa Ave. Santa Rosa, CA 95401
Parallel I/O Interface	Computer Products, Inc. Grant Technology Division 11 Summer St. Chelmsford, MA 01824