bc330VME
Time Code Processor

User’s Guide
December, 1996

CHAPTER ONE
INTRODUCTION

1.0 GENERAL

The bc330VME Time Code Processor Operational and Technical Manual provides the following information:

- General Introduction.
- Installation and Setup Details.
- Operation and Software Interface Details.
- I/O Signal Information.
- Theory of Operation.
- Programming Examples.
- Drawing Set.

1.1 KEY FEATURES

The salient features of the bc330VME Time Code Processor include:

- Decodes commonly used time code formats: IRIG A, IRIG B, 2137, MILA, NASA36.
- Continues to provide time during loss of input time code.
- Provides microsecond resolution.
- Allows time capture via an external event trigger input.
- Provides programmable propagation delay compensation.
- Functions as an A16:D08(O) slave with flexible interrupt capabilities.
- Its 4K byte block can be located on any 4K byte boundary in the VMEbus short address space.
- Provides both front panel and P2 (with -6UD option) I/O connections.
- Accommodates 3U or 6U (with optional front panel) racks.
- Drives Datum’s optional LED display modules with the decoded time.

1.2 bc330VME OVERVIEW

The bc330VME is a single height (3U) VMEbus module designed to decode serial time code signals and provide additional capabilities not normally found on a single board time code reader. The module consists of a 3U printed circuit board designed to connect to a Datum extension module which converts the bc330VME into a 6U module and provides additional functionality such as the -6UD LED display extension shown in Figure 1-1. (Consult the factory for the availability of other extension modules.)

The operation of the bc330VME is controlled by registers written and read by the host via VMEbus A16:D8(O) data transfers. These registers are used for:

- Defining the time code translation modes of operation.
- Activating time capture operations.
- Holding the captured time and status.
- Defining on-board interrupt priority levels and vectors.
- Defining the heartbeat interval rate.
- Defining the propagation delay compensation value.

The principal performance characteristics of the bc330VME are listed in Table 1-1 (located on the following page).

Table 1-1

bc330VME Performance Specifications

Item	Description
Time Code Reader	
Time Code Formats	IRIG A, IRIG B, 2137, NASA 36, MILA
Carrier Range	IRIG A, 10 kHz ± 2%
	IRIG B, 1 kHz ± 2%
	2137 1 kHz ± 2%
	NASA 36 1 kHz ± 2%
	MILA 1 kHz ± 2%
Flywheel Accuracy	<3.6 ms per hour
Modulation Ratio	3:1 to 6:1
Input Amplitude	.5V to 5 Vpp
Input Impedance	5K Ohms (AC coupled)
VMEbus Interface	
Standardization	IEEE 1014-1987 Revision C.1 of the VMEbus Spec
Address Space	A16, AM Codes $29 and $2D 4K contiguous bytes
Data Transfer	D08(O)
Interrupter	D08(O), I(1-7), ROAK
Power	+5VDC @600mA
TTL/CMOS Input Signals	
Event Capture	TTL/CMOS, Positive or Negative Edge Triggered, 50ns min width, 500 µs min period
TTL/CMOS Output Signals	
1 Pulse Per Second	TTL/CMOS, Positive edge on time
Heartbeat Pulse	TTL/CMOS, Positive edge on time
Operating and Storage Environments	
Temperature	
Operating	0° to 70° C
Non- Operating	-50° to 125° C
Relative Humidity	
Operating	10% to 80% (non-condensing)
Non-Operating	5% to 95%
Altitude	
Operating	1,000 ft below sea level to 60,000 ft above sea level

Non- Operating 1,000 ft below sea level to 20,000 ft above sea level

1.3 TIME CODE FORMATS

The widespread use of coded timing signals to assist in the correlation of intercept and test data began in the early 1950's. These signals can be decoded in real time to indicate the current Time of Day (TOD) or recorded along with intercept/test data on magnetic tape recorders for post processing and time correlation.

Hundreds of time code formats were developed: one for each agency involved. During the early 1960's the InterRange Instrumentation Group (IRIG) promoted a series of "standard" time code formats now loosely referred to as IRIG Time Codes. The bc330VME decodes two of these formats: IRIG A and IRIG B.

More complete details on these and other time code formats is available free of charge, on request, from Datum Inc, in the form of the Datum Inc Handbook of Time Code Formats. Figure 1-2 illustrates a frame of IRIG A, B, or G time code.

Figure 1-1
bc330VME VMEbus Time Code Processor

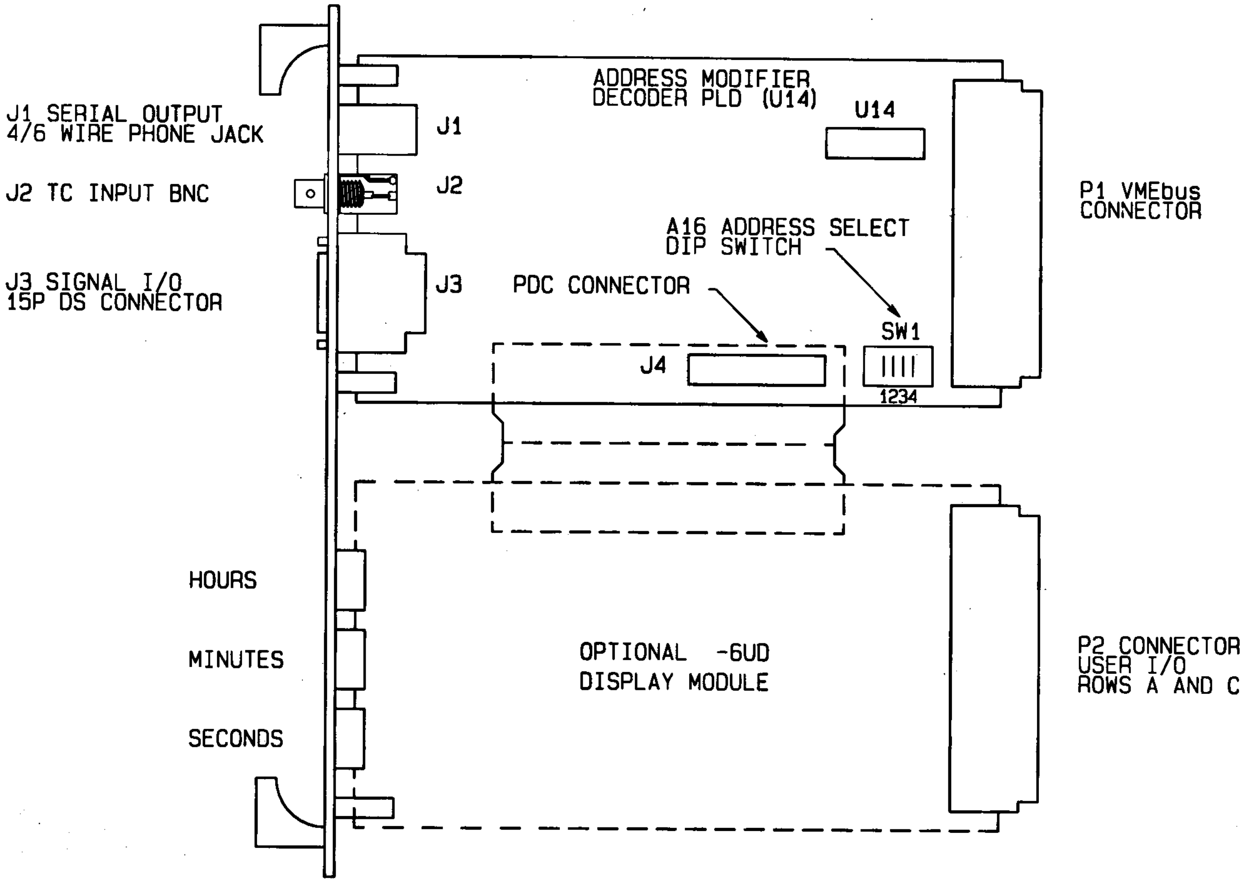


Figure 1-2
IRIG Time Code Frame

IRIG B Time Code Frame

Symbols:

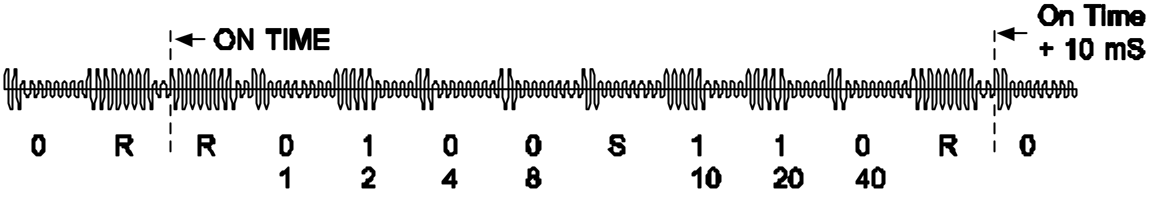
Reference Mark: 8 HC, 2 LC

Logical '1': 5 HC, 2 LC

Logical '0' or Space: 2 HC, 8 LC

10 Cycles or 10 milliseconds for IRIG B or IEEE1344

On Time and First Sub Frame:



A double reference mark denotes the start of the frame, 'On Time'. Encoded BCD data is the time at On Time. The encoded data is this first subframe is 32 Seconds.

Frame:

IRIG B:

OT
R R SS R MM R HH R DD R D _ R C C R C C R C C R B B R B B R
Seconds
Minutes
Hours
Days
Control Characters
Control Characters
Control Characters
17 BIT BINARY
TOD

This Page Intentionally Left Blank.

CHAPTER TWO

INSTALLATION AND SETUP

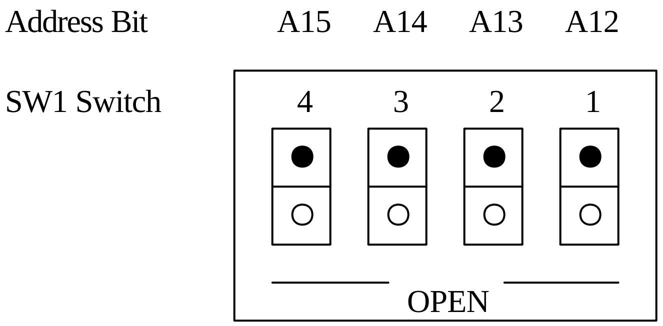
2.0 GENERAL

The bc330VME is a single height (3U) VMEbus board designed to be installed in a standard VMEbus subrack. The bc30VME can be equipped with either a 3U or (optional) 6U front panel. This section details the steps required to install the module in the subrack.

2.1 BASE ADDRESS SELECTION

Before installing the module in the subrack, the address DIP switch (SW1) must be set. The location of SW1 is shown on Figure 1-1. The bc330VME occupies the 4K bytes in the VMEbus short address space and can be freely located on any 4K byte boundary. The 4 DIP switch positions of SW1 correspond to address bits A15-A12 as shown in Figure 2-1 and determine the base address for the module. The address is defined as the address selected by the SW1 DIP switch when A11-A1 are 0.

Figure 2-1
DIP Switch SW1



To select a base address, set each of the 4 DIP switches to the ON (same as CLOSED) or OFF (same as OPEN) position. Setting a DIP switch to the ON position selects a logical 0 for that address bit, and the OFF position selects a logical 1.

The bc330VME responds to address modifiers $2D (Short Supervisory Access) and $29 (Short Non-Privileged Access) as decoded by the U14 address modifier decoder PLD shown in Figure 1-1. This decoder PLD can be modified by the factory to decode different and/or additional address modifiers. Consult the factory for custom address modifier decoding.

2.2 INSTALLATION PROCEDURE

Once the base address has been selected, the bc330VME is ready to be installed in the VMEbus subrack. Install the bc330VME as follows:

- Remove the IACKIN*/IACKOUT* backplane jumper for the bc330VME slot. This step should be done even if you will not be using interrupts for the bc330VME.
- Verify that the power to the subrack is turned off before inserting the bc330VME module into the subrack.
- Insert the bc330VME into the subrack slot and secure the board in this slot by tightening the two front panel screws.

CHAPTER THREE

OPERATION AND SOFTWARE INTERFACE

3.0 GENERAL

The bc330VME occupies 4K bytes in the VMEbus short address space (2K D08 [O] memory locations). Refer to Section 2.1 for details on base address selection. All data transfers between the VMEbus and the bc330VME are D08(O) type (single odd byte transfers). This chapter describes the bc330VME registers and their use.

3.1 REGISTERS

This section describes the registers used on the bc330VME for controlling its operation and transferring time data. Section 3.2 details the use of these registers. The Control Register memory map for the bc330VME is listed in Table 3-1. The memory map for the bc330VME registers is listed in Table 3-2. The first column of these tables shows the offset from the base address of each register. The value of each register following a VMEbus SYSRESET* is shown where '- -' indicates that the register contents is undefined. A label for each register is listed as is a brief description of the register's function.

3.1.1 CONTROL REGISTERS AND WARM START REGISTER

The Control Registers govern the operation of the bc330VME. A write of any non-zero value to the WARM START Register (offset 7FD) causes the bc330VME to perform a warm start reset. During a warm start reset, the bc330VME reads the Control Registers to determine its mode of operation. To control the operation of the bc330VME, first write the appropriate values to the control registers, then write any non-zero value to the WARM START Register. When the bc330VME is ready to accept new Control Register values, it will set the WARM START Register contents to zero. The Control Registers are described below. All values are in base hexadecimal.

TCSEL: Input Time Code Selection (offset 001)

00 = Automatically detects an IRIG B, IRIG A, 2137, or NASA 36 time code signal.
42 = Decodes IRIG B only. Automatic code detection is disabled.
41 = Decodes IRIG A only. Automatic code detection is disabled.
43 = Decodes 2137 only.
4D = Decodes MILA only.
4E = Decodes NASA 36 only.
52 = Begins flywheeling immediately.

When ICSEL is set to 52, the start time will be the time contained in REQTIME (Requested Time Data Block) when a WARM START is performed. The user should load REQTIME with the desired start time (see Section 3.1.6) and FERR0/1 with frequency adjust data if necessary (see Section 3.1.5) prior to issuing the WARM START command.

Table 3-1
bc330VME Control Register Memory Map

OFFSET (HEX)	RESET VALUE	LABEL	DESCRIPTION
001	00	TCSEL	Time Code Format Select
003	00	TVINTEN	Time Valid Int Enable
005	00	EVENT	External Event Control
007	00	MODE	Mode Control
009	00	PROPDEL0	Propagation Delay (MSB)
00B	00	PROPDEL1	Propagation Delay (LSB)
021	00	HBCTRL	Heartbeat Control
023	00	HBRATE0	Heartbeat Rate (MSB)
025	00	HBRATE1	Heartbeat Rate (LSB)
7FD	00	WARMSTRT	Warm Start Command

TVINTEN:Time Valid Interrupt Enable (offset 003)

00 = Disable time valid interrupt (interrupt source 3).
01 = Enable interrupt on Time Request time valid.
02 = Enable interrupt on External Event Capture time valid.
03 = Enable interrupt on either Bus Time Request or External Event Capture time valid.

EVENT: External Event Control (offset 005)

00 = Disable External Event Capture.
01 = Enable External Event Capture on the rising edge of the External Event Capture Input.
02 = Enable External Event Capture on the failing edge of the External Event Capture Input.
03 = Enable External Event Capture on both the rising and falling edges of the External Event Capture Input.

MODE: Mode Control (offset 007)

00 = Read time code mode.

PROPDEL0 and PROPDEL1: Propagation Delay (offset 009 and 00B)

The propagation delay compensation function can offset the effects of long cables between the time code source and the bc330VME. These two registers are combined to produce a signed 16-bit quantity. The range of values supported is -2048 to +2047. Each unit represents a delay of 0.5 µsec. For example, a value of 1000 would effect a delay of 500 µsec. Positive values advance the 1 PPS

(Pulse Per Second) and heartbeat pulses relative to the time code; negative values retard the 1 PPS and heartbeat pulses.

HBCTRL: Heartbeat Control (0ffset 021)
00 = Disable heartbeat pulses (output remains high).
01 = Enable heartbeat pulses at the rate defined by HBRATE0 and HBRATE1. The heartbeat feature is not available for IRIG A.

HBRATE0 and HBRATE1: Heartbeat Rate (offset 023 and 025)

The rate at which the heartbeat pulses are generated is controlled by these registers which are combined to produce a 16-bit unsigned value. The rate of heartbeat pulses is in pulses per second. For example, a value of 1000 would generate 1000 pulses per second. Heartbeat rates are supported from 1 to 2000 pulses per second. Rates above 2000 are not recommended. The heartbeat pulses are synchronized to the time code. The bc330VME must be decoding time or flywheeling before the heartbeat rate is programmed.

3.1.2 TIME REQUEST REGISTER

The Time Request Register is used to request time from the bc330VME over the VMEbus. A read or write to this register causes the bc330VME to freeze the time and transfer it to the Requested Time Data Block so that the time can be read over the VMEbus.

3.1.3 STATUS BYTE

The bc330VME provides a status byte containing the status of the time code decoder. The status byte is organized into two 4-bit fields. The lower order field (bits 0-3) designates the last known time code format detected as follows:

0	=	Warm Start complete, time code not yet found.
1	=	IRIG B
2	=	IRIG A
3	=	MILA
4	=	2137
5	=	NASA 36

The upper field (bits 4-7) denotes the time code tracking status as follows:

0	=	currently decoding time
1	=	flywheeling

The term ‘flywheeling’ means that the input time code has been lost or is unusable, but that the bc330VME is still providing time, heartbeats, etc., as if the time code was still present.

3.1.4 TIME VALID FLAG REGISTER

The Time Valid Flag Register (offset 41F) is used to indicate when the time data blocks contain valid time data.

TVFLAG: Time Valid Flag Register (offset 41F)

Bit0	=	1 when Time Request time is valid.
Bit1	=	1 when External Event Capture time is valid.
Bits 2-7	=	Not Used.

Table 3-2
bc330VME Register Memory Map

OFFSET (HEX)	RESET VALUE	LABEL	DESCRIPTION
401-417	- -	REQTIME	Requested Time Data Block
421-437	- -	EVTIME	Ext Event Time Data Block
419	- -	STATUS	Status Byte
41B	- -	FERR0	Frequency Error Byte 0 (MSB)
41D	- -	FERR1	Frequency Error Byte 1 (LSB)
41F	00	TVFLAG	Time Valid Flag
7FF	- -	INT3ACK	Int Source 3 Acknowledge
C01	- -	TIMEREQ	Time Request
801	00	INTCR0	INT Control Register 0
803	00	INTCR1	INT Control Register 1
805	00	INTCR2	INT Control Register 2
807	00	INTCR3	INT Control Register 3
809	1F	INTV0	INT Vector Register 0
80B	1F	INTV1	INT Vector Register 1
80D	1F	INTV2	INT Vector Register 2
80F	1F	INTV3	INT Vector Register 3
E01	- -	PINTCLR0	Pending Interrupt Clear 0
E03	- -	PINTCLR1	Pending Interrupt Clear 1
E05	- -	PINTCLR2	Pending Interrupt Clear 2
E07	- -	PINTCLR3	Pending Interrupt Clear 3

Once the user detects that a valid time is available, the time data block can be read. Once the time data is read, the user must clear the appropriate bit in the Time Valid Flag Register. The bc330VME will only set these bits (except during a WARM START); the user must clear them. The user should clear bit 0 of the TVFLAG register before performing a time request.

Table 3-3
Time Data Block Format

OFFSET (HEX) REQ/EVENT	DATA
410/421	Days Hundreds (± for MILA)
403/423	Days Tens
405/425	Days Units
407/427	Hours Tens
409/429	Hours Units
40B/42B	Minutes Tens
40D/42D	Minutes Units
40F/42F	Seconds Tens
411/431	Seconds Units
413/433	Subsecond Count Byte 0 (MSB)
415/435	Subsecond Count Byte 1
417/437	Subsecond Count Byte 2 (LSB)

3.1.5 FREQUENCY ERROR DATA

The Frequency Error Data is used in conjunction with the subsecond count bytes to determine an accurate subsecond time. Section 3.3 describes the meaning and use of the Frequency Error data. Both the Time Request and External Event Capture times use the same Frequency Error data.

FERR0 and FERR1 must be loaded by the user to set the flywheel rate when the TCSEL byte is set to 52 (FERR0/1 are set by the bc330VME for all other values of TCSEL.) A value of 33920 decimal causes the bc330VME crystal frequency to be used directly. A value of 33921 lowers the output rate by 5 parts in 10E7. A value of 33919 increases the output rate by 5 parts in 10E7. The bc330VME firmware does not initialize FERR0/1 at power up; these memory locations will contain undetermined values.

3.1.6 TIME REQUEST/EXTERNAL EVENT TIME DATA BLOCKS

Two separate blocks of data are used to hold time data. One block holds the time for a Time Request (from the VMEbus) and one block holds the time for an External Event capture. The format of these time data blocks is shown in Table 3-3. The time digits for days hundreds through seconds units contain the values 00 - 09. The subsecond count bytes are described in Section 3.3. The requested time data block is loaded with zeros on power up.

For the MILA countdown time code format the days hundreds digit (offset 401/421) holds the sign bit (MILA does not contain a days hundreds digit). The values for the sign bit are:

00	+	(plus time).
01	-	(minus time).

3.1.7 INTERRUPT CONTROL, VECTOR AND PINTCLR REGISTERS

The bc330VME supports four independent interrupt sources (interrupt source 0-3). Associated with each interrupt source are three registers: one Interrupt Control Register, one Interrupt Vector Register, and one Pending Interrupt Clear Register.

3.1.7.1 INTERRUPT CONTROL REGISTERS

The Interrupt Control Registers govern the operation of the VMEbus interrupts. There is one control register for each interrupt source (i.e. INTCR0 controls interrupt source 0, INTCR 1 controls interrupt source 1, etc.). The Interrupt Control Register format is shown below.

BIT	7	6	5	4	3	2	1	0
FUNCTION	FLAG	FAC	X/IN	IRE	IRAC	L2	L1	L0

L2, L1, L0: Interrupt Level

The three interrupt level bits determine the level at which an interrupt will be generated:

L2	L1	L0	RQ LEVEL
0	0	0	DISABLED
0	0	1	IRQ1
0	1	0	IRQ2
0	1	1	IRQ3
1	0	0	IRQ4
1	0	1	IRQ5
1	1	0	IRQ6
1	1	1	IRQ7

IRAC: Interrupt Auto Clear

If the IRAC is set, IRE (Bit 4) is cleared during an interrupt acknowledge cycle responding to this request. This action of clearing IRE disables the interrupt request. To re-enable the interrupt associated with this register, IRE must be set again by writing to the control register.

IRE: Interrupt Enable

This bit must be set to 1 to enable the bus interrupt request associated with the control register.

X/IN: External/Internal Vector

This bit must be cleared to 0 in all cases.

FAC: Flag Auto Clear

If FAC is set to 1, the FLAG bit is automatically cleared during an interrupt acknowledge cycle.

FLAG: Flag Bit

This bit is a flag bit that can be used for processor-to-processor communication and resource allocation. The FLAG bit has no affect on the operation of the interrupts.

3.1.7.2 INTERRUPT VECTOR REGISTERS

Each of the four interrupt sources has associated with it an interrupt vector register. Interrupt source 0 uses INTV0, interrupt source 1 uses INTV1, etc. The 8 bit interrupt vector is supplied to the VMEbus during an interrupt acknowledge cycle. The four Interrupt Vector Registers are set to 0F at reset which corresponds to the MC68000 vector for an uninitialized interrupt vector.

3.1.7.3 INTERRUPT SOURCE 3 ACKNOWLEDGE REGISTER

Interrupt source 3 requires an acknowledge operation following the interrupt. Before another interrupt from source 3 can take place, the Interrupt Source 3 Acknowledge Register must be read. The contents of this register are meaningless. Interrupt sources 0-2 do not have corresponding interrupt acknowledge registers.

3.2 FUNCTIONAL DESCRIPTION

Section 3.1 provided an overview of the bc330VME registers and their function. This section provides a description of how these registers are used to achieve the desired functions.

3.2.1 CONTROLLING THE OPERATING MODE OF THE bc330VME

To control the operating mode of the bc330VME, first write the appropriate data to the Control Registers. Then, to cause the bc330VME to take action on the Control Register data, write any non-zero value to the WARM START Register. This action will cause the bc330VME to perform a warm reset and to act on the data in the Control Registers. When the bc330VME is ready to accept another

WARM START command, it will clear the contents of the WARM START Register to 0. Do not initiate a warm reset unless the WARM START Register is 0.

The bc330VME will take 140 µsecs to load the Time Request Data Block with the time in response to a Time Request. The time, however, is frozen at the instant that the Time Request Register is accessed.

3.2.2 EXTERNAL EVENT CAPTURE

The bc330VME has the ability to capture time and generate an interrupt in response to an external event. Use the External Event Control Register (offset 005) in the control block to enable the external event function and to select the active edge for capture. When the bc330VME detects the appropriate edge of the event signal it will freeze the time and load the Event Capture Time Data Block (offset 421-437) with the time. Following the time data transfer, the bc330VME will set bit 1 of the Time Valid Flag Register (offset 41F), and if the Time Valid Interrupt Enable byte in the control block is set to 02 or 03 the bc330VME will generate a VMEbus interrupt to take place. The Event Capture Time Data Block can then be read. After reading the time, bit 1 of the Time Valid register must be cleared. The bc330VME will not overwrite the Event Capture Time Block unless it detects that bit 1 of the Time Valid Flag Register has been cleared. This prevents the Event Time data from being destroyed by a subsequent event before the user has a chance to read it. The user should be certain that bit 1 of the Time Valid Flag is cleared before starting External Event Captures.

The bc330VME will take 140 µsecs to load the Time Request Data Block with the time in response to a Time Request. The time, however, is frozen at the instant that the Time Request Register is accessed.

3.2.3 RATE PROGRAMMABLE HEARTBEAT PULSES

It is often useful to generate a periodic pulse (heartbeat) which is synchronized to the time code signal. The bc330VME has the ability to generate this heartbeat. The control register HBCTRL enables the heartbeat, and the registers HBRATE0 and HBRATE1 determine the rate of the pulses (in pulses per second). Following a change to the heartbeat control and rate registers, the user must perform a bc330VME warm start as described above. The bc330VME must be decoding time or flywheeling (see Status Byte) before the Heartbeat Pulses are started or else synchronization of the pulses in the time code will not occur. Heartbeat rates are supported from 1 to 2000 pulses per second. Heartbeat rates above 2000 are not recommended. The heartbeat pulses are available on the front panel I/O connector. The heartbeat can generate VMEbus interrupts (interrupt source 2) as well. The rising edge of the heartbeat occurs on time. The heartbeat feature is not available for the IRIG A time code format.

3.2.4 PROPAGATION DELAY COMPENSATION

When the time code source is located an appreciable distance from the bc330VME, a significant propagation delay will be introduced. The bc330VME incorporates a propagation delay compensation feature which removes the effects of this delay. To utilize this feature, simply load the desired value into the PROPDEL0 and PROPDEL1 registers and then perform a bc330VME warm start.

The PROPDEL0 (MSB) and PROPDEL1 (LSB) registers are loaded with a 16-bit signed value between -2048 and +2047. Each unit represents 0.5 µsec. Positive values advance the 1 PPS and heartbeat relative to the time code; negative values retard the 1 PPS and heartbeat. For example, if the cable between the time code source and the bc330VME introduces a delay of 50 µsec, and you want to make the system act as if the time code source is sitting right next to the bc330VME, then load the value +100 into the propagation delay registers.

3.3 SUBSECOND COUNT AND FREQUENCY ERROR

NOTE: The notation 2E6 means 2 x 10^6 or 2,000,000.

The subsecond count in conjunction with the frequency error is used to derive an accurate subsecond time. The subsecond count is an unsigned 24-bit binary number, and the Frequency Error is an unsigned 16-bit binary number. The subsecond count multiplied by 0.5 µsec will provide the nominal subsecond time. Therefore, the subsecond count varies nominally between 0 and 2E6 - 1 (1,999,999) each second. However, due to the difference between the bc330VME's crystal clock frequency and the frequency of the time code, the subsecond count can vary between 0 and some number greater than or less than 2E6-1. This variation in subsecond counts in a 1 second interval gives rise to the frequency error bytes.

If the bc330VME clock is exactly on frequency and the input time code time base is also perfect, then the frequency error is 0. When the frequency error is 0, the frequency error bytes contain the value 33920 decimal (8480 hex). If the frequency error bytes contain the value 34000 decimal (33920 + 80), for example, then the bc330VME time base is fast by 80 parts in 2E6, and the subsecond count would need to be scaled down by the ratio of 2E6 / (2E6 + 80). Likewise, if the frequency error bytes contain the value 33840 decimal (33920 - 80) then the bc330VME time base is slow by 80 parts in 2E6, and the subsecond count would need to be scaled up by the ratio of 2E6 / (2E6 - 80). In general, the subsecond count needs to be scaled by the following factor:

2E6 / (2E6 + (Frequency Error - 33920))

When scaled by the above factor, the subsecond count will always vary between 0 and 2E6-1, giving an accurate number of 0.5 µsec counts. This methodology allows the user to completely remove the effects of the bc330VME crystal clock frequency offset and aging.

The user may also simply divide the subsecond count by the factor (2e6 + (Frequency Error - 33920)) to produce a real number which varies between 0.0000000 and 0.9999995.

The programming examples in Chapter 6 provide some useful data structures and algorithms for working with the subsecond count and frequency error values.

3.4 VMEbus INTERRUPTS

The bc330VME provides four independent VMEbus interrupt sources as shown in Table 3-4. As described above, associated with each interrupt source are three registers: Interrupt Control Register, Vector Register, Pending Interrupt Clear Register. Each interrupt source can generate an interrupt on any one of the seven interrupt request levels (IRQL) on the VMEbus. All four sources could also use the same IRQL. Additionally, each interrupt source can have its own unique interrupt vector. The following sections describe the operation of each interrupt source.

Table 3-4
Interrupt Sources

Interrupt Sources	
INT SOURCE	**FUNCTION**
0	1 Pulse Per Second
1	Future Implementation
2	Heartbeat Pulses
3	Ext Event/Time Request Time Valid

3.4.1 INTERRUPT SOURCE 0 (1 Pulse Per Second)

The bc330VME generates a 1 Pulse Per Second (1PPS) signal. This signal generates a rising edge which occurs at the one time mark of the time code signal and is available on the J3 I/O connector. This 1 PPS signal can also be used to generate a VMEbus interrupt. Since this signal is always active, there will probably be an interrupt pending associated with it. Therefore, when the Interrupt Control Register is set to enable interrupts for this source, an interrupt would be immediately generated. To clear this pending interrupt, simply read or write the PINTCLR0 register before enabling interrupt source 0 with INTCR0.

3.4.2 INTERRUPT SOURCE 1 (Future Implementation)

3.4.3 INTERRUPT SOUCE 2 (Heartbeat Pulses)

The rate programmable heartbeat pulses can be used to generate VMEbus interrupts. Use INTCR2, INTV2, and PINTCLR2 to control this interrupt source. The interrupt is generated on the rising edge of the heartbeat pulse.

3.4.4 INTERRUPT SOURCE 3 (Time Valid)

Interrupt Source 3 is used to inform the user that a valid time (External Event or Time Request) has been loaded into the Time Data Block without having to poll the Time Valid Flag Register. The Time Valid Interrupt Control Register (offset 003) is used (along with INTCR3, INTV3, and PINTCLR3) to enable this interrupt. The user should clear any source 3 pending interrupts before enabling interrupts with INTCR3 by accessing the PINTCLR3 Register. Also, be sure to read the INT3ACK register after servicing the interrupt or a subsequent interrupt cannot take place.

CHAPTER FOUR

INPUT/OUTPUT CONNECTORS

4.0 GENERAL

All I/O signals (except the PDC signals) are available on the front panel connectors. Time code input is available on the front panel BNC and 15 pin DS connector. The PDC signals are available on a 20 pin rectangular connector located near the bottom edge of the module (3U version only). The location of all connectors is shown in Figure 1-1.

4.1 J1 SERIAL DIFFERENTIAL TTL OUTPUT

A serial output (Differential TTL) is available on the front panel phone jack connector labeled “J1.” Either a four wire or six wire phone jack can be used with J1. The pinouts for J1 are listed in Table 4-1. Figure 4-1 shows the orientation of the J1 pinouts.

The Tx(-) signal will drive most RS-232 interfaces.

4.2 TIME CODE INPUT

The time code input is available on the front panel BNC labeled “J2” and is connected in parallel with the other connectors which carry this signal.

4.3 J3 SIGNAL I/O CONNECTOR

All I/O signals (except the PDC signals and serial output) are connected to the front panel 15 pin DS connector labeled “J3” and are connected in parallel with the other connectors which carry these signals. The pinout assignments for the J3 connector are shown in Table 4-2.

4.4 J4 PDC CONNECTOR

The J4 pin connector carries the PDC signals which can be used to drive Datum Time Display modules. The pin assignments for the J4 connector are shown in Table 4-3.

4.5 PDC SIGNAL DESCRIPTION

The signals carried on the PDC (Peripheral Data Connector) are generally used to drive other Datum products which require decoded time such as the PC26V display module. These signals can, however, be used by the user for a variety of applications.

Decoded time is transmitted over the PDC once per time code frame using the PDC ENABLE* and D0-D7 lines in a byte serial fashion (i.e. a burst of 9 bytes are sent) just after the on time mark. The D0-D7 encoding is shown in Table 4-4. The lower nibble contains the BCD encoded time digit and the upper nibble determines which digit has been transmitted. The D0-D3 data lines are valid at the rising edge of PDC ENABLE*. The D4-D7 data lines are valid anytime PDC ENABLE* is low.

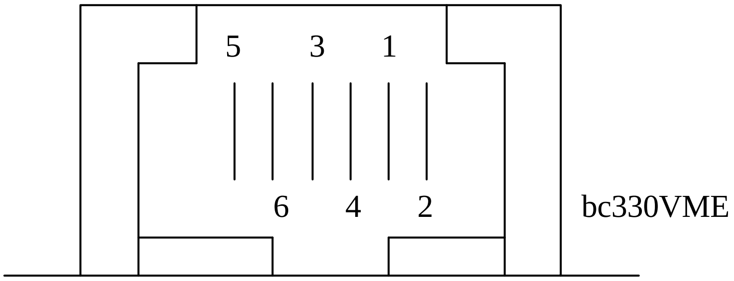
EACH CYCLE is a TTL representation of the time code carrier (high during positive half cycles, low during negative half cycles). 1 PPS is a TTL signal whose rising edge occurs once per second on time.

Table 4-1

J1 Serial Output Pinouts

J1	SIGNAL DESCRIPTION
1	Tx(-)
2	Not Used
3	Tx(+)
4	Ground
5	Not Used
6	Ground

Figure 4-1
J1 Phone Jack Pinout Orientation



Front View

Table 4-2

J3 I/O Connector Pinouts

J3	Signal Description
1	Time Code Input
2	Ground
3	Auxiliary Analog Input 1
4	Ground
5	Auxiliary Analog Input 2
6	Ground
7	External Event Input
8	(Future Implementation)
9	Heartbeat Pulse Output
10	(Future Implementation)
11	1 PPS Output
12	Ground
13	-6U Option I/O
14	-6U Option I/O
15	Not Used

Table 4-3

J4 PDC Connector Pinouts	
J4	**Signal Description**
1	Ground
2	PDC ENABLE*
3	D0
4	D1
5	D2
6	D3
7	D4
8	D5
9	D6
10	D7
11	EACH CYCLE
12	Not Used
13	1 PPS
14-17	Not Used
18	Ground
19	+5 VDC
20	+5 VDC

Table 4-4

PDC Data Format				
D7	**D6**	**D5**	**D4**	**D3 - D0 (BCD)**
0	0	0	0	Days Hundreds
0	0	0	1	Days Tens
0	0	1	0	Days Units
0	0	1	1	Hours Tens
0	1	0	0	Hours Units
0	1	0	1	Minutes Tens
0	1	1	0	Minutes Units
0	1	1	1	Seconds Tens
1	0	0	0	Seconds Units

CHAPTER FIVE

THEORY OF OPERATION

5.0 GENERAL

This section describes the Theory of Operation for the bc330VME Time Code Processor Module. Reference Schematic Diagram 11370.

5.1 TIME CODE READER

The heart of the bc330VME utilizes Datum's reduced chip set decoder circuitry. Details on the operation of this circuitry are proprietary.

5.2 VMEbus INTERFACE

The VMEbus interface consists of the usual assortment of bus transceivers, buffers, and decoders commonly found on any microprocessor based system. Support for the interface is provided by the VME 2000 Slave Module Interface device from PLX Technology and the MC68153 Bus Interrupter Module from Motorola.

CHAPTER SIX

PROGRAMMING EXAMPLES

6.0 GENERAL

This section provides programming examples to illustrate the operation of the bc330VME. The examples are written in the C programming language. Hexadecimal constants are specified in C by using the prefix ‘Ox.’ The functions read_byte() and write_byte() are used to indicate a direct access to a physical memory location. In most programming environments, read_byte() and write_byte() can be defined as the following macros, assuming that type char is used for D08(0) data and that ‘addr’ can be successfully cast as a pointer to type char.

```
#define read_byte(addr)                 (*(char  *)  (BASE  +  addr) )
#define write_byte(addr, data)          *(char  *)  (BASE   +   addr)  = data
```

A system dependent base address is defined below where ‘YYYY’ indicates a 64k byte page of memory used for A16 data, and ‘S’ indicates the SW1 dip switch setting.

```
#define BASE  0xYYYYS000
```

Memory locations are referred to with the BASE constant plus an offset constant. The offset constants are referred to with a label that is defined in Table 3-1 and Table 3-2.

6.1 SETTING UP THE CONTROL REGISTERS

This example shows how to set up the bc330VME control registers. The bc330VME is set up to automatically detect an IRIG A or IRIG B time code input signal, disable Time Valid Interrupts, disable External Event Capture, and operate in the ‘read time code’ mode.

```
while(read_byte(WARMSTART) ) ;          /* the bc330VME is ready to       */
                                        /* accept setup data when         */
                                        /* WARMSTART = 0                  */
write_byte(TCSEL,0) ;                   /* setup control registers        */
write_byte(TVINTEN, 0);
write_byte(EVENT, 0) ;
write_byte(MODE, 0);
write_byte(HBCTRL, 0) ;
write_byte(PROPDEL0, 0) ;
write_byte(PROPDEL1, 1) ;
write_byte(WARMSTRT, 1) ;               /* initiate bc330VME warm reset   */
```

6.2 SET MAJOR TIME

The major time will be set in REQTIME block when TCSEL is set to 0x03 or 0x52. To load a desired major time to REQTIME block, see the following programming example:

```
#define        read_byte(addr)                  (*(char  *)  (base+addr)  )
#define        write_byte(addr, data)           *(char  *)  (base+addr)=data
#define        time0          0x401             /*  days hundreds                        */
#define        time1          0x403             /*  days tens                  */
#define        time2          0x405             /*  days units                 */
#define        time3          0x407             /*  hours tens                 */
#define        time4          0x409             /*  hours units                */
#define        time5          0z40b             /*  minutes tens               */
#define        time6          0x40d             /*  minutes units              */
#define        time7          0x40f             /*  seconds tens               */
#define        time8          0x411             /*  seconds units              */

#define        WARMSTART      0x07fd            /*  warmstart                  */

/*  Set major time  */

write_byte(TCSEL,   0x03);                 /*  select external 1pps        */
write_byte(time0,  1) ;                         /*  setime days hundreds =1    */
write_byte(time1,  0) ;                         /*  setime days tens =0        */
write_byte(time2,  0) ;                         /*  setime days units =0       */
write_byte(time3,  2) ;                         /*  setime hours tens =2       */
write_byte(time4, 3) ;                          /*  setime hours units =3      */
write_byte(time5, 4) ;                          /*  setime minutes tens =4     */
write_byte(time6, 7) ;                          /*  setime minutes units =7    */
write_byte(time7, 5) ;                          /*  setime seconds tens =5     */
write_byte(time8, 9) ;                          /*  setime seconds units =9              */
write_byte(TCSEL,   0x03) ;                     /*  select external 1pps       */

write_byte(WARMSTART, 0x01) ;                   /*  warmstart command          */
while(read_byte(WARMSTART) {} ;                 /*  wait until the command is done   */
```

6.3 CAPTURING AND READING TIME

This example shows how to capture time across the VMEbus using the REQTIME register. The bc330VME is polled to determine when the time data is valid. A useful data structure is defined for (8 bit) bc330VME time data block. The union 'btol' converts four (8 bits) bytes to one (32 bit) unsigned long integer. The structure 'time_but' provides storage for the entire time data block and the frequency error bytes. The structure provides storage for the 9 major time (days - seconds) bytes in the **major_time** char array. The three subsecond bytes are written to the **time.subsec** char array. The two frequency error bytes are written to the **time.freqerr** char array. The subsecond count and frequency error can then be read out as two unsigned long integers.

```
Union btol  {
      char ba [4] ;
      unsigned long li;
} ;
struct time_buf  {
      char major_time[9] ;
      union botl subsec ;
      union btol freqerr;
}           time = {0} ;

write_byte(TVFLAG, 0) ;                                  /*  clear item valid flag              */
write_byte(TIMEREQ, 0) ;                           /*  request time                     */
/*  wait for time valid (TVFLAG bit 0 = 1)  */
while( ! (read_byte(TVFLAG)  &  0x01) ) ;
/*  time is now valid and can be read  */
time.major_time[0]  =  read_byte(BASE  +  0x401L)        /*  days hundreds
*/


time.major_time[1]  =  read_byte(BASE  +  0x403L)        /*  days tens                          */
                         .
                         .
time.major_time[8]  =  read_byte(BASE  +  0x411L)        /*  secs units                         */

/*  NOTE:  time.subsec.ba[0]  =  0  */
time.subsec.ba[1]  =  read_byte(BASE  +  0x413L)         /*  subsec MSB                         */
time.subsec.ba[2]  =  read_byte(BASE  +  0x415L)         /*  subsec                             */
/*  NOTE:  time.freqerr.ba[0]  =  time.freqerr.ba[1]  =  0  */
time.freqerr.ba[2]  =  read_byte(BASE  =  0x41bL)        /*  freq err MSB                       */
time.freqerr.ba[3]  =  read_byte(BASE  =  0x41dL)        /*  freq err LSB                       */
/*  convert subsecond count and freq error to real number   */
/*  between  0.0000000  and  0.9999995  (SEE SECTION 3.3)  /*
temp  =  2000000L  +  (time.freqerr.li  -  33920) ;
subseconds_real  =  (float)  time.subsec.li  /  (float)  temp;
```

6.4 EXTERNAL EVENT TRIGGER

The following example shows how to set up the bc330VME to receive External Event Triggers on the rising edge. The event is set up to trigger a VMEbus interrupt. Be sure to read the **INT3ACK** register and write a 0 to the **TVFLAG** register during the interrupt service so that another event can take place.

```
write_byte(EVENT, 0x01) ;                  /*  enable event on rising edge          */
write_byte(TVINTEN, 0x02) ;                        /*  enable int on event time              */
write_byte(WARMSTRT, 1) ;                          /*  initiate bc330VME warm reset          */
write_byte(BIM3, VECTOR) ;                         /*  setup  in vector                      */
dummy = read_byte(INT3ACK)  ;                      /*  ack int source 3                      */
write_byte(PINTCLR3, 0) ;                  /*  clear pending interrupt              */
write_byte(BIMCR3, 0x12) ;                         /*  enable int on IRQ level 2             */
/*  wait for interrupt        */

/*  part of interrupt service routine  */

/*  read event time data block   (offset  421  -  427)  */
dummy = read_byte(INT3ACK) ;                       /*  ack int source 3                      */
write_byte(TVFLAG, 0) ;                            /*  clear time valid register             */

/*  return from interrupt   */
```

6.5 HEARTBEAT PULSE

The following example shows how to set up the bc330VME to generate an interrupt every heartbeat pulse. The heartbeat is set up for a rate of 1,000 pulses per second. The time code format is assumed to be IRIG B. Be sure that the bc330VME is decoding time or flywheeling before starting the heartbeat pulses.

```
while(read_byte(STATUS)   !  =  0x01) ;            /*  wait for decoding                     */
write_byte(HBRATE0, 0x03) ;                        /*  heartbeat rate (MSB)                  */
write_byte(HBRATE1, 0xE8) ;                        /*  heartbeat rate (LSB)                  */
write_byte(HBCTRL, 1) ;                            /*  initiate bc330VME warm reset          */
write_byte(BIMV2,  VECTOR) ;                       /*  set up int vector                     */
write_byte(PINCLR2.0) ;                            /*  clear pending interrupt               */
write_byte(BIMCR2, 0x15) ;                         /*  enable int on IRQ level 5             */

/*  wait for interrupt generate every millisecond   */
```

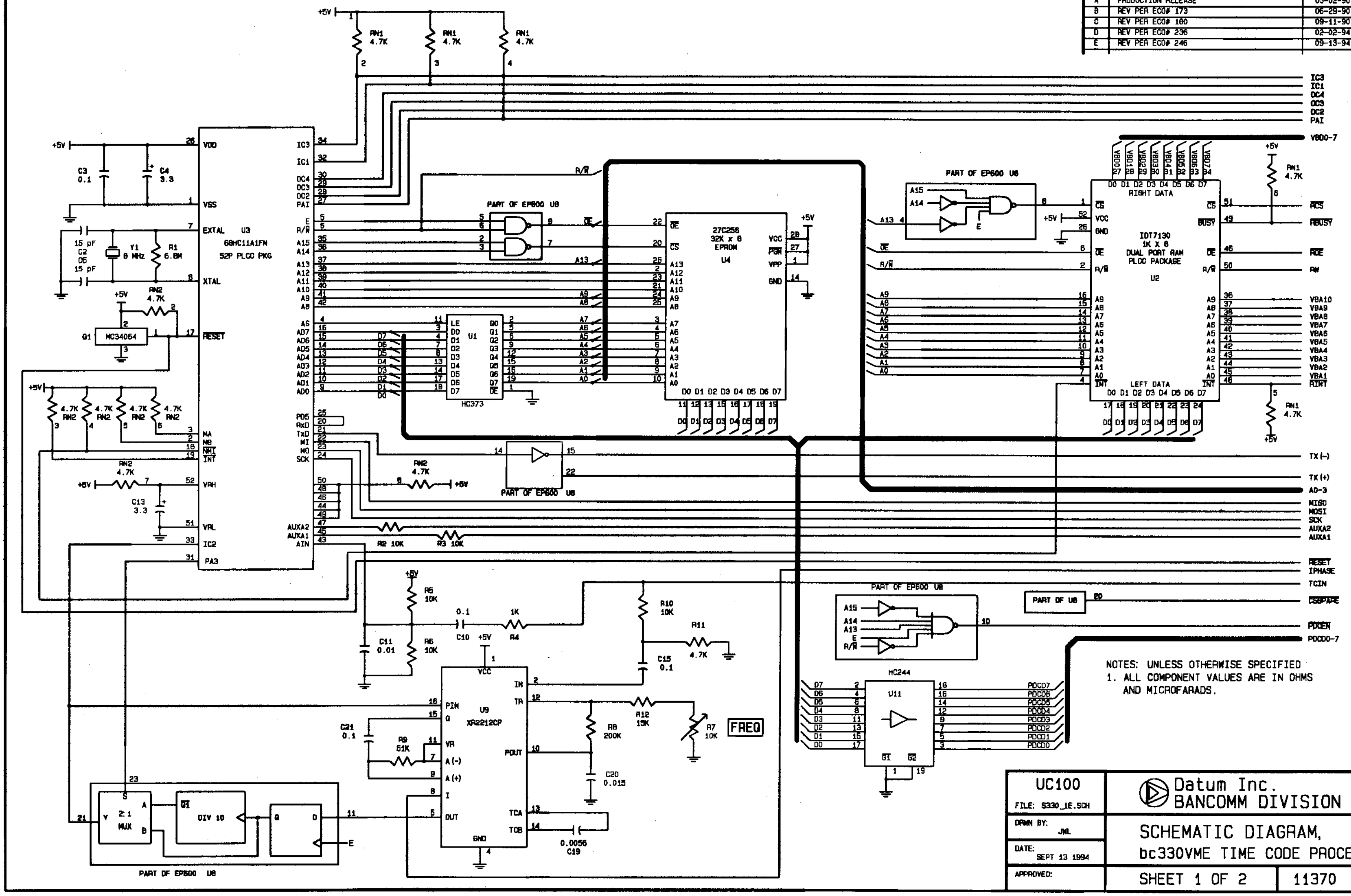
CHAPTER SEVEN

DRAWING SET

7.0 GENERAL

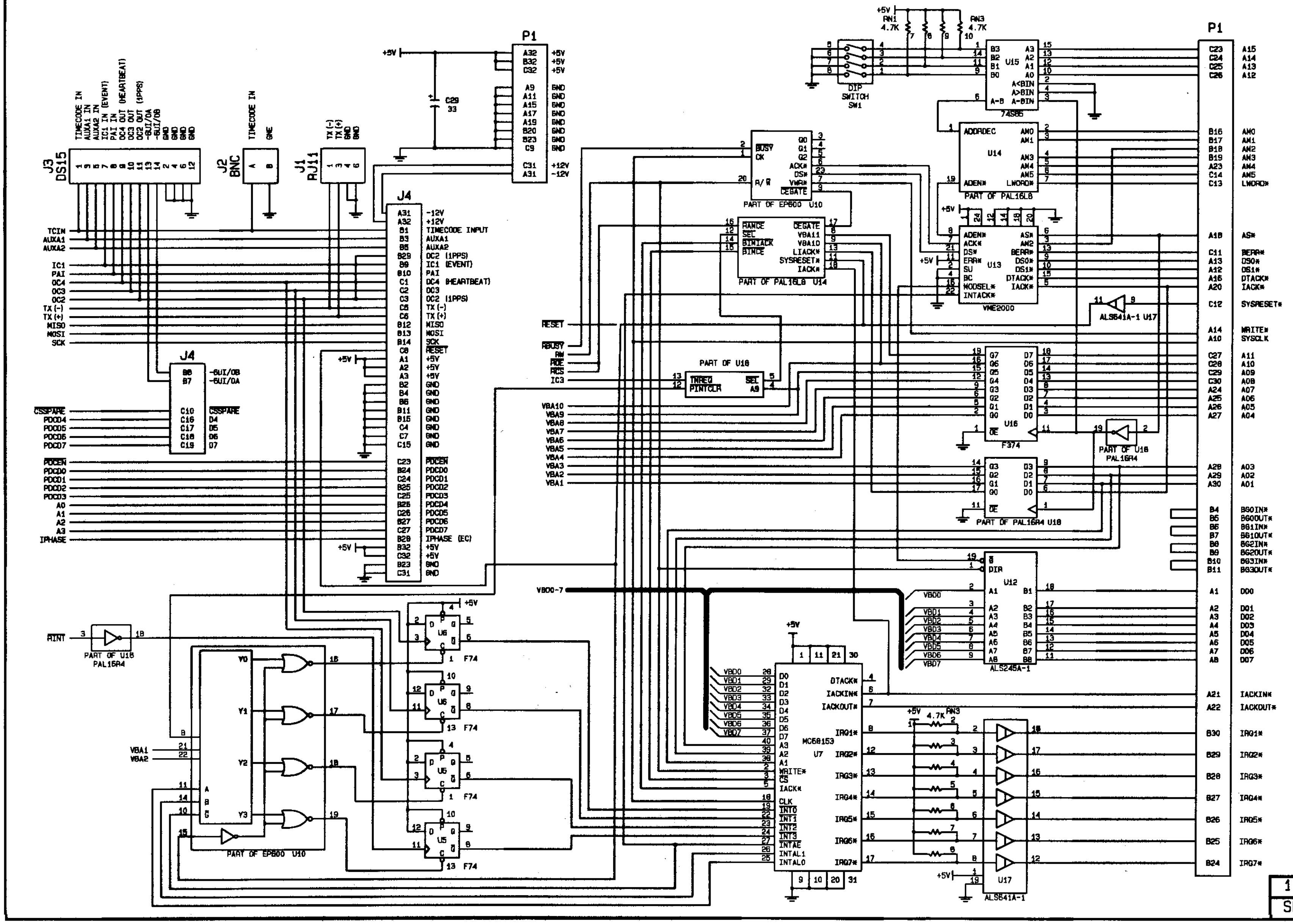
This chapter contains the schematic diagram, assembly drawing, and parts list for the bc330VME.

REVISIONS			
REV	DESCRIPTION	DATE	APPROVED
A	PRODUCTION RELEASE	05-02-90	D.W.
B	REV PER ECO# 173	06-29-90	D.W.
C	REV PER ECO# 180	09-11-90	D.W.
D	REV PER ECO# 236	02-02-94	J.F.
E	REV PER ECO# 246	09-13-94	D.W.



NOTES: UNLESS OTHERWISE SPECIFIED
1. ALL COMPONENT VALUES ARE IN OHMS AND MICROFARADS.

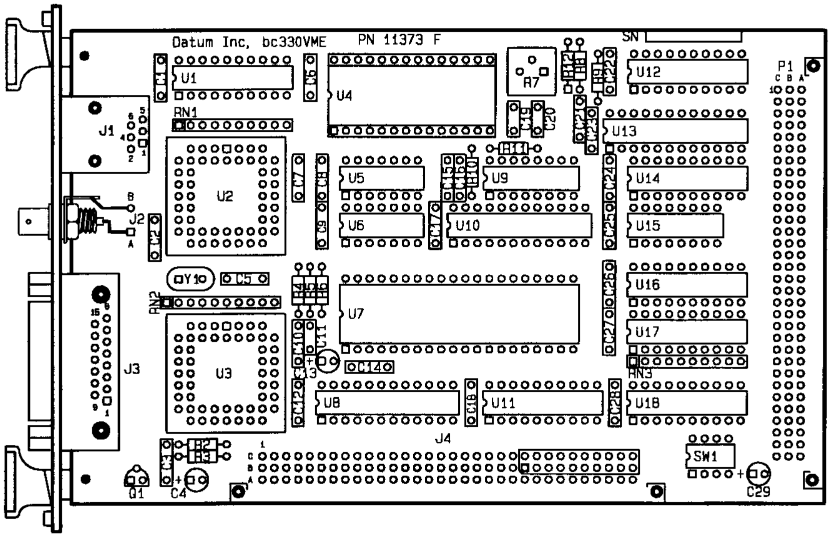
UC100 FILE: S330_1E.SCH	Datum Inc. BANCOMM DIVISION
DRWN BY: JML	SCHEMATIC DIAGRAM, bc330VME TIME CODE PROCESSOR
DATE: SEPT 13 1994	
APPROVED:	SHEET 1 OF 2 — 11370 REV-E



11370 REV E

SHEET 2 OF 2

REVISIONS			
LTR	DESCRIPTION	DATE	APPVD
A	FIRST RELEASED	02 MAY 90	D.W.
B	REV PER ECO# 173	29 JUN 90	D.W.
C	REV PER ECO# 180	11 SEP 90	D.W.
D	REV PER ECO# 236	02 FEB 94	J.F.
E	REV PER ECO# 246	13 SEP 94	D.W.
F	REV PER ECO# 284	14 NOV 95	M.B.



NOTE: bc330V-3U FRONT PANEL SHOWN. OPTIONAL bc330V-6U STYLE FRONT PANEL INSTALLED WHEN EITHER OF THE FOLLOWING PC BOARD OPTIONS ARE REQUESTED. -6UP, -6UD, -6UTCG, & -6UVTI.

UC100 BCM067.ZIP FILE: A330V_E1.SCH	Datum Inc BANCOMM DIVISION
DRWN BY: JWL	ASSEMBLY, bc330VME TIME CODE PROCESSOR
DATE: 14 NOV 1995	
APP. BY:	SHEET: 1 OF 2 | 11373 REV-F

Assembly, Parts Listing
bc330VME Time Code Processor

Ref: Drawing No. 11373 F

Ref: UC 100

Nov 14, 1995

AS	OPT	BC P/N	MANF P/N	MANUFACTURE	VALUE	DESCRIPTION	QTY#	REF DESIG.
		1501150	CM05CD150J03	CDE	15 PF, 500V	DIPPED MICA CAPACITOR	2.00	C2,5
		1503335	335RMR035M	IC	3.3 MF, 35V	ALUMINUM ELECTROLYTIC CAP.	2.00	C13,4
		1503336	336RMR025M	IC	33 MF, 35V	ALUMINUM ELECTROLYTIC CAP.	1.00	C29
		1506103	SR211C103KAA	AVX	0.01 MF, 100V	MONO CERAMIC CAPACITOR .2 R/L	1.00	C11
		1506153	SR211C153KAA	AVX	15000 PF, 100V	MONO CERAMIC CAPACITOR .2 R/L	1.00	C20
		1506562	SR211C562KAA	AVX	5600 PF, 100V	MONO CERAMIC CAPACITOR .2 R/L	1.00	C19
		1515104	MD015E104MAA	AVX/67349	0.1 MF, 50V	DIP GUARD CAPACITOR	22.00	C1,3,6-10,12,14-18,21-26
		1701100	11372F	BANCOMM DIV, DATUM	bc330V TCP	PRINTED CIRCUIT BOARD	1.00	PCB1
		2101003	31-221	AMPHENOL	50 OHM	FRONT MNT BNC BULKHEAD RECEP.	1.00	J2
		2104001	913346	ERNI	96 POS	DIN CONNECTOR, MALE	1.00	P1
		2124215	869521-1	AMP	15 POS	'D' SKT, .318 RTANG PCBMNT B/L	1.00	J3
		2149024	824-AG31D	AUGAT	24 POS	SLIM DIP SOCKET	2.00	REF: U8,10
		2150020	10620-01-445	ANDON/SPECIRA	20 POS	DIP SOCKET	2.00	REF: U14,18
		2150028	10628-01-445	ANDON/SPECIRA	28 POS	DIP SOCKET	1.00	REF: U4
		2152052	641748-2	AMP	52 POS	PLCC REC CHIP CARRIER	2.00	REF: U2,3
		2190004	SS-6466S-NF	STEWARD	RJ11 1PT,6 POS	SHIELDED PHONE JACK PCMNT.	1.00	J1
		2302005	DS-800	SEIKO	8MHz	CRYSTAL	1.00	Y1
		2404600	VME-6U-1450	PHILLIPS COMPONENTS		VME EXTRACTOR HANDLES KIT	1.00	BKT1
		2802002	3341-1S	3M		JACK SCREW KIT	1.00	BKT1
		4701102	RC07GF102J	ALLEN BRADLEY	1 K OHM, 1/4W	FIXED RESISTOR	1.00	R4
		4701103	RC07GF103J	ALLEN BRADLEY	10 K OHM, 1/4W	FIXED RESISTOR	5.00	R2,3,5,6,10
		4701153	RC07GF153J	ALLEN BRADLEY	15 K OHM, 1/4W	FIXED RESISTOR	1.00	R12
		4701204	RC07GF204J	ALLEN BRADLEY	200 K OHM, 1/4W	FIXED RESISTOR	1.00	R8
		4701472	RC07GF472J	ALLEN BRADLEY	4.7 K OHM, 1/4W	FIXED RESISTOR	1.00	R11
		4701513	RC07GF513J	ALLEN BRADLEY	51 K OHM, 1/4W	FIXED RESISTOR	1.00	R9
		4701685	RC07GF685J	ALLEN BRADLEY	6.8 M OHM, 1/4W	FIXED RESISTOR	1.00	R1
		4703103	72P103	BECKMAN	10 K OHM, 1/2W	SINGLE TURN POTENTIOMETER	1.00	R7
		4705472	710A472	ALLEN BRADLEY	4.7 K OHM, 1/8W	C-SIP RESISTORS, 10 PIN 'X'	3.00	RN1-3
		5108001	76SB04	GRAYHILL		4PST DIP SWITCH	1.00	SW1
		9006818	74F74	NATIONAL	14P DIP PKG	DUAL D FLIP FLOP	2.00	U5,6
		9006858	MM74F374N	NATIONAL	20P DIP PKG	OCTAL D FLIP FLOP	1.00	U16
		9009657	74HC373	VARIOUS	20P DIP PKG	OCTAL D TRANSPARENT LATCH, T/S	1.00	U1
		9015222	74S85	NATIONAL	16P DIP PKG	4-BIT MAGNITUDE COMPARATOR	1.00	U15
		9102002	68HC11A1FN	MOTOROLA	52P PLCC PKG	MICROCOMPUTER	1.00	U3 (SKT)
		9103031	MX68C153	MACRONIX	40P DIP PKG .6W	CMOS BUS INTERRUPT MODULE	1.00	U7
		9201030	MC34064P-5	MOTOROLA	3P CASE29-04 PKG	UNDERVOLTAGE SENSING DEVICE	1.00	Q1
		9207615	74HC244	NATIONAL	20P DIP PKG	OCTAL BUFFER/LINE DRIVER	1.00	U11
		9207920	SN74ALS245A-1N	TI	20P DIP PKG	OCTAL BUS TRANSCEIVER	1.00	U12
		9207925	SN74ALS641A-1N	TI	20P DIP PKG	OCTAL BUS TRANSCEIVER	1.00	U17
		9211001	VME2000-45	PLX TECHONLOGY	24P DIP	VME SLAVE INTERFACE	1.00	U13
		9307030	XR2212CP	EXAR	16P DIP PKG	PHASE LOCKED LOOP	1.00	U9
		9405001	EP600DC-3	ALTERA	24P DIP PKG .3W	EPLD	2.00	REF: U8,10 (SKT)
		9405015	PAL16L8B	MMI	20P DIP PKG .3W	PAL	1.00	REF: U14 (SKT)
		9405020	PAL16R4ACN	MMI	20P DIP PKG .3W	PAL 35 NS	1.00	REF: U18 (SKT)
		9406040	27C256	VARIOUS	28P DIP PKG .6W	32 K BYTE, CMOS EPROM	1.00	REF: U4 (SKT)
		9407660	IDT7130LA70J	IDT	52P PLCC PKG	1K X 8 DUAL PORT RAM	1.00	REF: U2 (SKT)
	-3U					-3U ASSEMBLY OPTION		
	-3U	2109020	3592-5002	3M		20 PIN RTANG HEADER	1.00	J4
	-3U	2401603	11375D	BANCOMM DIV, DATUM	bc330VME	-3U FRONT PANEL	1.00	BKT1
	6UP					-6UP ASSEMBLY OPTION		
	6UP	2111020	3592-6002	3M	20 POS	CONTACT HEADER	1.00	J4
	6UP	2401604	11376D	BANCOMM DIV, DATUM	bc330VME	-6UP Blank Ft. Panel Option	1.00	BKT1
00						STANDARD ASSEMBLY		
00		2104001	913346	ERNI	96 POS	DIN CONNECTOR, MALE	1.00	J4