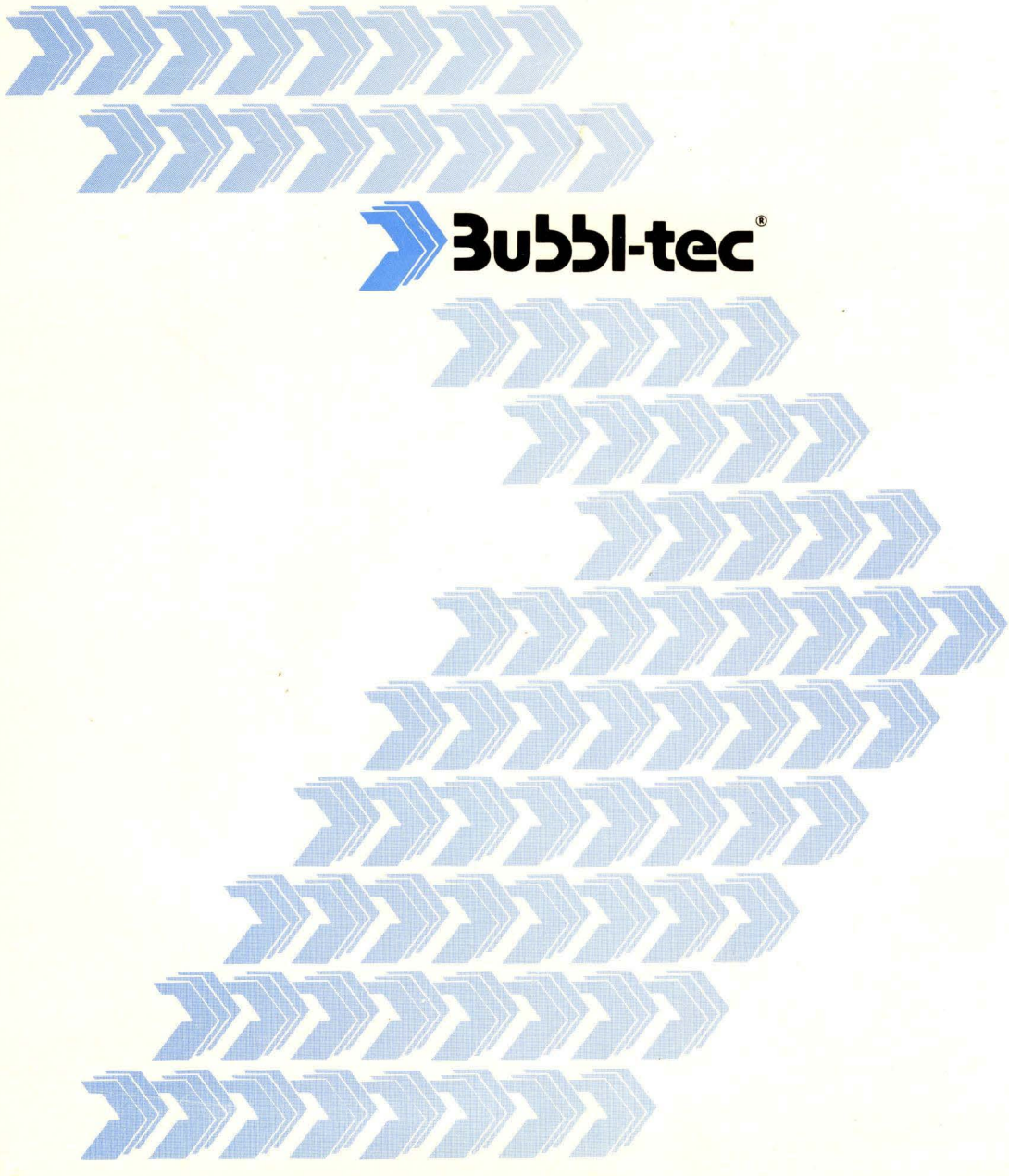
QBC-11/02

LSI-11 Compatible

BUBBL-MACHINE™



Technical Manual
for

QBC-11/02

LSI-11 Compatible

BUBBL-MACHINE™

Solid-State Mass Storage System

Bubbl-tec®
division of PC/M, Inc.
6800 Sierra Court
Dublin, CA 94568
Telephone: 415-829-8700
TWX/Telex: 910-389-6890

TABLE OF CONTENTS

SECTION 1.0

INTRODUCTION

The QBC-11/02 BUBBL-CONTROL™ and its companion QB-series BUBBL-BOARDs form a complete non-volatile mass-storage system implemented on dual-height modules that plug directly into an LSI-11 computer system. Each QB-series BUBBL-BOARD can provide as much as 512 kilobytes of mass storage in an all-solid-state package, completely eliminating all external chassis, power supplies, moving parts and mechanical storage media. The QBC-11/02 BUBBL-CONTROL module can control as many as 16 BUBBL-BOARDs, providing up to 8 megabytes of on-line mass storage.

The QBC-11/02 plugs directly into the LSI-11 backplane bus. All required power (+5 and +12 volts only) is derived from the bus. The BUBBL-BOARDs include on-board power-switching circuitry which the QBC-11/02 controller uses to power-down any BUBBL-BOARD module which is not currently in use. This technique greatly diminishes the overall system power requirement, especially when system storage capacity is extended beyond 1 megabyte. Very little additional power is required when more than two QB-series BUBBL-BOARDs are used in the system.

The QBC-11/02 controller connects to the companion BUBBL-BOARDs via a 26-conductor ribbon cable. The BUBBL-BOARDs receive all their control signals from this cable, and therefore require no connection to the LSI-11 backplane except to receive power. As a result, the BUBBL-BOARDs can be externally mounted, if desired.

The QBC-11/02 is capable of operation in direct-memory-access mode, and is compatible with DEC's RT-11 and RSX-11 operating systems. The bubble-memory system can be operated as either a system or non-system mass-storage device using DEC's standard "DY" floppy-disk handlers. The required bootstrap is stored in a PROM on the QBC-11/02 module, so no additional bootstrap module is required.

QB-series BUBBL-BOARD modules are removable and transportable without loss of data contained within the magnetic-bubble devices. Therefore, data can be moved from one machine to another simply by pulling a BUBBL-BOARD out of one machine and installing it in another.

The QBC-11/02 controller provides built-in self-diagnostics which can be invoked with simple ODT commands. The self-test mode is intended for use in acceptance testing, and as a field maintenance aid.

A green activity-indicator light (labeled RUN) on each QB-series BUBBL-BOARD indicates when the bubble devices on the module are being accessed. A red indicator labeled ERR illuminates if the controller ever detects an error in the bubble-stored data.

NOTE: DEC, LSI-11, RSX-11 and RT-11 are registered trademarks of Digital Equipment Corporation, Maynard, Massachusetts, USA.

1.1 IN CASE OF DIFFICULTY

If you have difficulty with your BUBBL-MACHINE system and/or decide to return one or more units to the factory for repair, you may do so after you have obtained a return authorization. Please note that no merchandise returns will be accepted without a prior return authorization having been issued. To obtain return authorization, simply call or write Bubbl-Tec and explain why it is necessary to make a return. You will then be promptly provided with the necesary Return Authorization Number.

When returning a module for repair, be sure to pack it properly in anticipation of rough handling in transit. Also, be sure you have prominently displayed the Return Authorization Number and your return address in at least two places on the outside of the shipping carton. Shipments not properly marked will not be accepted by the receiving department at Bubbl-Tec. Be sure to adequately insure your shipment, and prepay all necessary shipping and insurance charges. COD shipments will not be accepted at Bubbl-Tec.

When returning merchandise for repair, be sure to write a detailed letter to the attention of the Repair Department, describing the nature of the difficulties you have encountered. Also, include a purchase order or other memorandum authorizing us to perform the required work (warranty repairs are, of course, done without charge). Send this letter by AIR MAIL, so that it will arrive at the factory before, or coincident with, the arrival of the returned merchandise. Always mention the Return Authorization Number in your letter, and fully describe the merchandise you are returning for repair.

SECTION 2.0

INSTALLATION

The QBC-11/02 controller is intended to be installed in the card cage of the host LSI-11 microcomputer. It furnishes all required control signals for the companion QB-series BUBBL-BOARDs via a 26-conductor ribbon cable. To set up the bubble-memory system, the LSI-11 Device Address and Vector Address must be selected, the QB-series BUBBL-BOARDs must be properly strapped, and the power-down sequence must be checked. Each of these items is discussed below.

2.1 POWER SUPPLY REQUIREMENTS

Power consumption for the QBC-11/02 controller and typical companion BUBBL-BOARDs is shown in Table 2-1. Regulation, sensing and filtering must be provided on the +5 and +12 volt power supplies to assure that they remain within ±5% of their nominal values at all times while the system is operating.

The QBC-11/02 uses power-control circuitry on the BUBBL-BOARD modules to put them into power-down mode when they are not being accessed. The two most recently accessed BUBBL-BOARDs are always kept in standby mode (powered up, but not necessarily shifting bubbles). Bubbles are only shifted on one board at a time. When bubbles are shifted the module is said to be 'active'.

Model	Power-down		Stand-by		Active	
	+5	+12	+5	+12	+5	+12
QBC-11/02					1.4	
QBI-11/256	0.25	0.01	0.5	0.15	0.5	0.6
QBI-11/512	0.25	0.01	0.6	0.25	0.6	1.2

TABLE 2-1

Current Consumption by Board Type
(in amperes)

2.2 POWER-UP and POWER-DOWN SEQUENCING

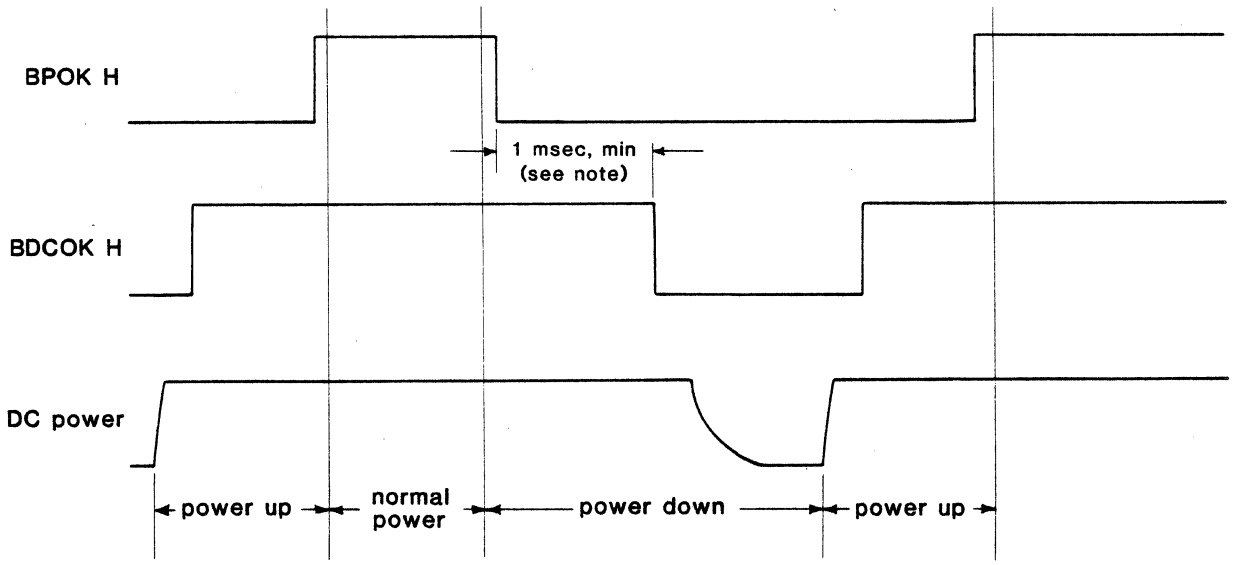
The QBC-11/02 controller executes a sequence of operations at power-up and power-down time which assures proper synchronization of the bubble storage medium. These special sequences are triggered by the BPOK H and BDCOK H signals on the LSI-11 bus, or (if these signals are not available) by power-fail detection circuitry on the BUBBL-BOARDs. The user must review his system's power supply circuitry to assure that either the BPOK and BDCOK bus signals are provided as described in Section 2.2.1, or his power supply provides sufficient "hold-up" at power-down time as described in Section 2.2.2.

2.2.1 BPOK and BDCOK BUS SIGNALS

The normal power-up and power-down timing using a DEC power supply is shown in Figure 2-1. During power-up, the BDCOK signal is not asserted high until both the +5 and +12 volt supplies have stabilized at their nominal values. Then, after several milliseconds, the BPOK signal is also asserted high, indicating that AC power is stable. The circuitry on the QBC-11/02 controller is held in a reset state until both BPOK and BDCOK are asserted high, at which point (after an additional brief delay) it begins its initialization sequence.

At power-down time, the BPOK signal is de-asserted at least 1 millisecond before BDCOK is de-asserted, and DC power begins to go out of tolerance. (One millisecond is required by the QBC-11/02; DEC power supplies normally provide 4 milliseconds.) The QBC-11/02 detects the de-assertion of BPOK and completes all necessary power-down operations before DC power begins to drop out of tolerance.

On some LSI-11 systems a front-panel DC-power switch, or a switch labeled "INIT" or "RESTART", is provided that shuts off the DC-power regulators and forces the BDCOK bus signal low. This switch will not disturb proper QBC-11/02 operation if it either de-asserts the bus BPOK signal at least 1 millisecond before the DC supplies are shut off, or allows the DC supplies to gradually decay, as explained in Section 2.2.2, below. A DC power switch that crow-bars the DC supplies without first de-asserting BPOK, however, could potentially cause loss of all stored data in the bubble devices, as well as loss of the boot loops in the bubble devices themselves. (If the boot loops are ever lost, they must be restored by the procedures described in Appendix C.)



NOTE: The indicated 1 millisecond minimum delay between BPOK and BDCOK is required by the bubble-memory system. DEC power supplies will provide a 4 millisecond delay.

FIGURE 2-1

Power-up and Power-down Bus Timing

2.2.2 BUBBL-BOARD POWER-DOWN REQUIREMENTS

If the BPOK and BDCOK signals are not provided on the backplane bus, or if the QB-series BUBBL-BOARDs are to be operated with a power supply separate from that which powers the Q-bus backplane, the power-up sequence will begin as soon as power is applied to the QBC-11/02 controller, but will not complete until both the controller and BUBBL-BOARDs have proper power applied.

If either the +5 or +12 volt supply to the BUBBL-BOARDs falls below the 94% point during normal operation, it will be detected by special circuitry on each BUBBL-BOARD and the on-board device-control circuitry will begin it's power-down sequence. See Figure 2-2. This sequence takes a maximum of 200 microseconds to complete, and **it is mandatory that both the +5 and +12 volt supplies stay above the 90% point for this entire 200 microseconds after the power-down sequence begins.**

For LSI-11 systems that do not provide properly sequenced BPOK and BDCOK signals on the backplane bus, and for systems in which the BUBBL-BOARDs will be powered from a separate power supply, it is essential that the user check

the BUBBL-BOARD power supplies to be sure they provide the required 200-microsecond hold-up at power-down time, or stored date can be scrambled, and the bubble-device boot loops lost.

The 200-microsecond requirement should be easily met by any linear 5-volt supply if at least 2000 microfarads of output filter capacitance is provided per load amp. Similarly, the 12-volt supply should provide at least 1000 microfarads per load amp. Almost any linear supply should have more than enough output capacitance to satisfy these requirements.

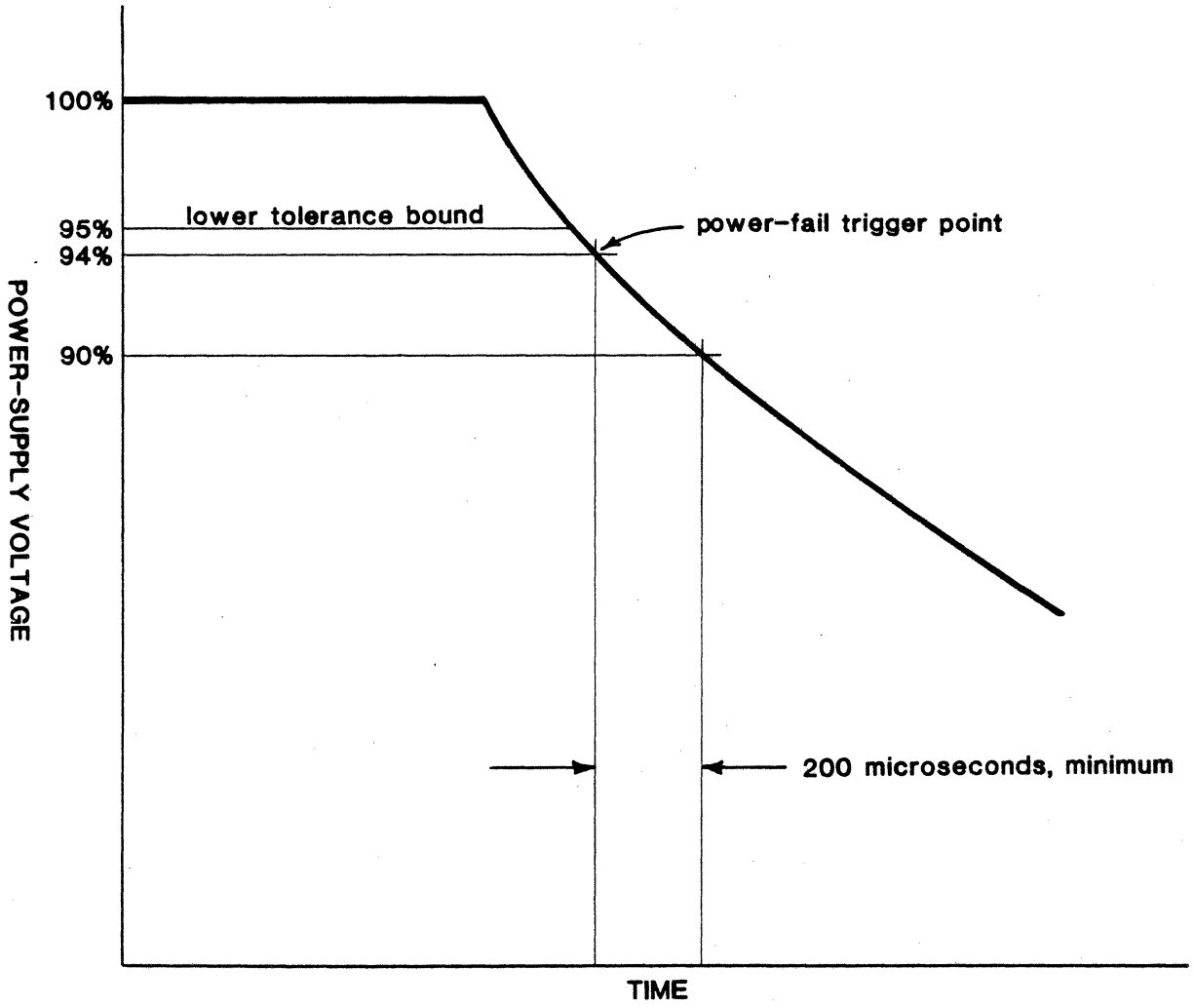


FIGURE 2-2
Required Power Supply Decay

The decay characteristic of switching-type power supplies varies greatly between designs, however, so the user of a switching supply should investigate the supply's power-down characteristics to assure that they meet the requirements shown in Figure 2-2. If necessary, additional output capacitance could be added to slow down the decay characteristic.

Pursuant to the above requirements, it is not advisable to remove or install either the QBC-11/02 controller, or any BUBBL-BOARD module, from/to the host equipment while any power supply voltage is on.

2.3 DEVICE ADDRESS SELECTION

Before operating the QBC-11/02 controller for the first time, the proper device address must be set-up on the printed-circuit module. Four addresses are available: 177150, 177160, 177170 and 177174. The selection of one of these four addresses is accomplished by installing or removing shunts at positions W1 and W2, as indicated in Table 2-2. The QBC-11/02 is delivered with all shunts installed (closed). Appropriate shunts may be opened by cutting the narrow center of the shunt with a sharp instrument, such as a screwdriver.

SELECTED FUNCTION	SHUNT POSITION W8	W7	W6	W5	W4	W3	W2	W1
Device Address 177170							CL	CL
Device Address 177150							CL	OP
Device Address 177160							OP	CL
Device Address 177174							OP	OP
Vector Address 000264					CL	CL		
Vector Address 000270					CL	OP		
Vector Address 000274					OP	CL		
Boot Address 173000			CL	CL				
Boot Address 171000			CL	OP				
Boot Address 175000			OP	CL				
Bootstrap Disabled			OP	OP				
RX01 Mode Operation:								
With only 1 or 2 drives	OP	CL						
With 3 or more drives	CL	CL						
RX02 Mode Operation:								
Single-sided Emulation	OP	OP						
Double-sided emulation	CL	OP						

CL = strap installed (closed) OP = strap omitted (open)
(Also see Sections 2.6 and 2.7 below for description of jumpers W9 and W10.)

TABLE 2-2

Strap Options for QBC-11/02 Controller

The Device Address selects the address for the RXCS Command and Status Register. The RXDB Register is the next even address above the RXCS address.

The standard device address for the RXCS register when the bubble-memory system is used as the RT-11 system device is 177170. For a second controller or non-system device, 177150 or 177174 are the preferred device addresses.

The QBC-11/02 uses the LSI-11 Q-bus signal BBS7 L to enable the address decoder. This signal insures that the decoder only responds to addresses in the highest bank of memory. The bubble-memory system may be used with LSI-11 systems that generate extended address bits without making changes on the module.

2.4 INTERRUPT VECTOR ADDRESS SELECTION

Three interrupt-vector addresses can be selected on the QBC-11/02: 264, 270 or 274. The selection of the Vector Address is made by installing or removing shunts at positions W3 and W4 on the printed-circuit board, as indicated in Table 2-2.

The standard Vector Address for the controller when it is used as a system device is 264. When a second (non-system) controller is installed, the preferred Vector Address is 270.

2.5 BOOTSTRAP ADDRESS SELECTION

The on-board RT-11 bootstrap is started at one of three memory addresses: 171000, 173000 or 175000. Selection is made by means of programmable shunts at positions W5 and W6 on the printed-circuit board, as shown in Table 2-2. Starting address 173000 should be used if the LSI-11 processor is strapped to boot automatically on power-up.

The bootstrap routine normally assumes that the address for the system device to be booted is 177170. The device address used by the bootstrap can be changed to 177150 by installing a jumper between wirewrap posts W11.

2.6 RX01 MODE and DOUBLE-SIDED DISK EMULATION IN RX02 MODE

When shunt W7 is installed, the QBC-11/02 controller emulates an RXV11/RX01 floppy-disk controller. With shunt W7 open (as it normally will be), the controller emulates the RXV21/RX02 double-density floppy controller.

When operating in RX01 mode, two logical drives are assigned to each 512K-byte QBI-11 Bubbl-Board module. If 256K-byte Bubbl-Boards are to be used, wire-wrap jumper W9 should be installed on the QBC-11/02 to make the

controller firmware assign logical drive numbers 0, 1 ,2 (etc.) to Bubbl-Boards 0, 1, 2 (etc.), respectively.

If more than two Bubbl-Boards are to be used in the system, W8 should be installed to allow the use of the high-order byte of the track address to select the Bubbl-Boards, as shown in Figure 3-2. In this case, the software driver should be written to put the correct drive number in the upper half of the track word. Word-move instructions should be used to write the track address into the RXDB register. Byte-move instructions should not be used, because both bytes of the word-length RXDB register are strobed when either byte is written. If DEC's 'DX' (RX01) drivers are to be used with jumper W8 installed, they will require revision to accomodate more than two drives, and to insure that the upper byte of the track address is written properly.

When shunt W8 is installed, the QBC-11 emulates double-sided disk operation when running in RX02 mode, and two 512-Kbyte BUBBL-BOARD modules are considered to be one logical unit. Double-sided operation should not be attempted when using 256-Kbyte BUBBL-BOARDs. Bit 9 (Side Select) of the RXCS Register is used as the least-significant board-address select bit. Shunt W8 is normally left open, i.e., single-sided operation is the default mode.

2.7 DIAGNOSTIC JUMPER

The wire-wrap jumper-post pair W10 is made available for diagnostic purposes. When jumper W10 is installed, the QBC-11/02 module will ignore commands sent to it by the LSI-11 host, and the internal write/read self-test function is invoked. The QBC-11/02 begins a continuous write/read test of the controller itself as well as all BUBBL-BOARDs to which it has access. If a data error is detected during these tests, the ERRor light on the offending BUBBL-BOARD will be illuminated. If a controller error is detected, the test will halt. See Section 4 for further discussion of the diagnostic self-tests.

2.8 INSTALLATION IN CARD CAGE

The QBC-11/02 can be installed anywhere in the LSI-11 Q-bus. However, care must be taken to insure that the Priority Interrupt Grant and Direct Memory Access (DMA) bus chains are not broken anywhere along the bus. Normally there must be a card in every slot lying between the CPU and cards that use interrupts or direct-memory-access.

The user should analyze the timing requirements of his system to determine the optimum placement of the QBC-11/02 within the interrupt-priority

and DMA-priority chains.

Due to the power-up and power-down sequencing required by the bubble devices, the QBC-11/02 module (and its companion BUBBL-BOARDs) should never be inserted into, or removed from, the LSI-11 system when power is present on the bus.

2.9 BUBBL-BOARD JUMPERS

Each QB-series BUBBL-BOARD module must be set-up to respond to a unique board address, from 0 to F (hex). Up to 16 BUBBL-BOARD modules can be connected to one QBC-11/02 controller. The board address is selected by installing shunts at locations W1 through W4 on the BUBBL-BOARD printed-circuit board. W1 selects the least-significant address bit. A closed shunt sets a logical zero for that address bit. Thus, a four-position shunt with all positions installed (closed) selects board-address 0; opening W1 (only) selects board-address 1; etc..

When the system is operated in RX01 mode, a 512-Kbyte BUBBL-BOARD module corresponds to two logical units (and a 256-Kbyte module corresponds to a single logical unit). All 512K-byte BUBBL-BOARDs should be assigned even addresses, and a maximum of eight BUBBL-BOARD modules is allowed in an RX01-mode system.

When only two bubble devices are installed on the BUBBL-BOARD module, strap W6 is installed on the BUBBL-BOARD to indicate to the QBC-11/02 that the module has only 256 Kbytes of storage capacity. If only 256K-byte modules are to be used, they may be assigned both odd and even addresses if W9 is installed on the QBC-11/02 controller.

2.9.1 STRAPPING FOR SINGLE BUBBLE-DEVICE OPERATION

In normal operation, no jumper should be installed at position W7 on the BUBBL-BOARD module, since this jumper is used for maintenance purposes only.

For diagnostic purposes it is sometimes desireable to operate each bubble device individually on a particular BUBBL-BOARD, in order to help isolate the cause of an operational problem. This is done by installing a shunt in position W7 to select single-device operation, and then selecting the desired bubble-device number with appropriate shunts at W5 and W6. W5 is the least-significant bubble-device selection bit. Again, an installed shunt selects a logical zero, so shunts installed at both W5 and W6 would select bubble-device number 0 on the BUBBL-BOARD; opening W5 (only) selects device number 1; etc.

2.9.2 STRAPPING FOR WRITE PROTECTION

For normal RT-11 (or other) operation, shunt W8 should not be installed on any BUBBL-BOARD. If the user wishes to write-protect a particular BUBBL-BOARD module, however, jumper W8 should be installed on that board. This will permanently inhibit writes to the bubble devices on that module, protecting any previously-stored data from being altered.

2.10 ACCEPTANCE TESTING

Before operating the bubble-memory system for the first time, the user may wish to run a few preliminary checks to insure that it will operate properly in his system. To do this the QBC-11/02 module should be plugged into the Q-bus with the SELF-TEST shunt W10 installed (and all other QBC-11/02 jumpers omitted). One BUBBL-BOARD module should be installed in the Q-bus and connected to the controller using the ribbon cable provided. This BUBBL-BOARD should be configured as board-address 0 by installing shunts W1-W4. For a 512-Kbyte BUBBL-BOARD, all other shunts should be omitted. For a 256-Kbyte module, jumper W6 must also be installed. The LSI-11 system can then be powered up. The green RUN lamp on the BUBBL-BOARD should come on briefly and then go out. The internal self-test firmware will exercise all logic on the BUBBL-BOARD (turning the RUN lamp back on), and will turn on the ERRor lamp if the controller detects an uncorrectable error. The system should be run in this mode for several minutes to be sure that the system is running properly. If the system refuses to run as described, it usually indicates a controller error, or that the operator has not set the system up correctly.

After running the preliminary self-test, the user should power down the system, remove the SELF-TEST shunt, and then set up the QBC-11/02 for the desired device address, vector address, bootstrap address, etc. as discussed above. When the system is then again powered up, there will be a brief illumination of the RUN lamp on the BUBBL-BOARD. When the RUN lamp goes out, the controller has completed its initialization process. LSI-11 console ODT can then be used to access the RXCS register at the device-address location. The RXCS should contain 04440_8. The RXDB register, which is located at the next higher (even) address, should contain 244_8.

After successfully completing the above checks, the hardware should be ready for operation in the user's system.

SECTION 3.0

PROGRAMMING

The QBC-11/02 bubble-memory controller is designed to respond to the same command protocol as DEC's RXV21/RX02 floppy-disk controller. The QBC-11/02 contains a Command and Status Register (RXCS), and a Data Register (RXDB). These registers serve as communication paths between the LSI-11 and a Z-80 microprocessor located on the controller module. The Z-80 interprets the commands produced by the LSI-11 and controls the data transfers to and from the bubble storage medium.

This section, although not intended to be a general tutorial on mass-storage controller operation, will define the function of each bit in the Command and Status Register, and the sequence of commands that is expected by the Z-80 for correct system operation.

The QBC-11/02 can be strapped so that it emulates either an RX01 (single density) or RX02 (double density) floppy-disk system. Because of the inherent advantages, we will assume in this section that the controller is configured to emulate an RX02. Contact Bubbl-tec if you must run in RX01 mode and you do not have access to RX01 programming information.

3.1 THE COMMAND AND STATUS REGISTER (RXCS)

Whenever a command is written into the RXCS register the DONE flag is cleared. The clearing of the DONE flag is sensed by the Z-80, which reads the RXCS register and performs the desired function. Note that if the RXCS is written into by the LSI-11 while the Z-80 is executing a function, the new command will be ignored. Bits 0 through 3 and bits 12-14 are write-only bits, i.e., they can be written but not read by the LSI-11. Bits 5, 7, 11 and 15 are read-only bits for the LSI-11. Bit 6 is a read/write bit. Both bytes of the RXCS are written whenever either byte is written by the LSI-11. Bits 5, 7, 11 and 15 of the RXCS are writeable only by the Z-80 on the QBC-11/02 module. Bits 4, 8 and 9 are writeable by the LSI-11 and are readable at the completion of a function (when DONE is set).

The format for the RXCS register is shown in Figure 3-1. Except for the

Interrupt Enable and Transfer Request bits, none of the bits read by the LSI-11 should be considered asserted unless the DONE bit is also asserted.

15	14	13	12	11	09	08	07	06	05	04	03	02	01	00
ERR	RX INIT	EXT ADDR		RX02	SIDE SEL	DEN	TR	INT ENB	DONE	UNIT SEL	FUNCTION			GO

FIGURE 3-1

RXCS Register Format

Definitions of the RXCS bits are:

BIT NUMBER	NAME	DEFINITION
0	GO	Must be set to enable Function Codes in bits 1-3.
1-3	FUNCTION	Defines operation to be performed by QBC-11/02 controller, as follows (left-hand bit in CODE column below is bit 3; right-hand bit is bit 1):

CODE	FUNCTION
000	Fill Sector Buffer
001	Empty Sector Buffer
010	Write a sector of data into bubble memory
011	Read a sector of data from bubble memory
100	Diagnostic and power-control functions
101	Read status from RXES register
110	Write a deleted data sector mark (not implemented)
111	Read Error Register

4	UNIT	This bit is used to select Drive #0 or Drive #1 in the RX02 floppy-disk system. See the description of the Read Sector and Write Sector functions in Section 3.3 for information on selecting 'drives' 2 through 15.
5	DONE	If=0, the controller is busy. If=1, the controller has completed a function. This bit is cleared whenever anything is written into the RXCS register by the LSI-11, and may be set or cleared by the QBC-11/02 controller.
6	INTERRUPT ENABLE	If=1, an interrupt will be generated when DONE is asserted.

7	TRANSFER REQUEST	If=1, the controller is ready to receive data from, or send data to, the LSI-11 via the RXDB register.
8	DENSITY	For an RX02, this bit determines the density of the diskette being accessed. For the QBC-11/02, this bit must be set for all READ, WRITE, FILL BUFFER, and EMPTY BUFFER functions. It is a read/write (R/W) bit, only valid when DONE is set, and will always read 1.
9	SIDE SEL	When double-sided-disk emulation is enabled via jumper W8, two BUBBL-BOARD modules constitute one logical unit. Bit 9 then becomes the least-significant board-address select bit.
10	Unused	To be RX02 compatible, always write as 0.
11	RX02	Read-only bit which identifies the device as an 'RX02'. Always set for the QBC-11/02 in RX02 mode.
12-13	EXT ADDR	Two most significant bits (write-only) of the starting DMA address for data transfers.
14	INITIALIZE	Asserting this bit will initialize the bubble-memory system without initializing other LSI-11 devices. This software INIT performs the same function as a hardware INIT received from the Q-bus. DONE will first be negated, and the controller will initialize the bubble devices to determine the current position of the bubble pages. Sector 1, Track 1 will then be read into the sector buffer. The QBC-11/02 controller will then assert DONE and set the BUBBLE READY and INITIALIZE DONE bits in the RXES register. The RXES contents will then be placed in the RXDB register.
15	ERROR	When this bit is asserted by the QBC-11/02 controller, an error has occurred during a command execution. This bit will be cleared when another command or an INIT pulse is sent to the QBC-11/02 controller.

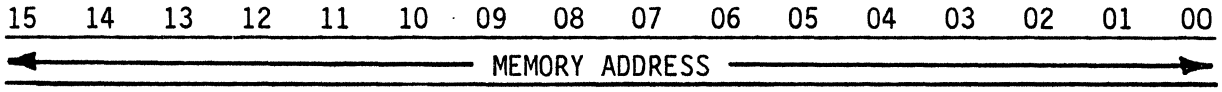
3.2 THE DATA REGISTER (RXDB)

The RXDB Data Register serves as the data path between the LSI-11 and the Z-80 on the QBC-11/02 controller module. Two 16-bit registers are actually used, one that can be written by the LSI-11 and read by the Z-80, and one that can be written by the Z-80 and read by the LSI-11. The LSI-11 address of the RXDB is always the next even-numbered address above that of the RXCS register.

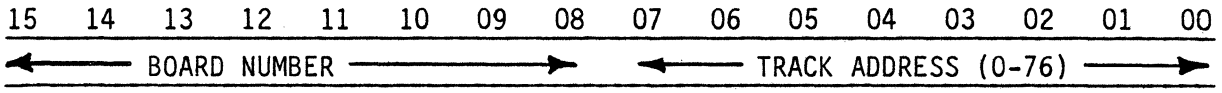
Different information will be placed in the RXDB register at different times depending upon the function that is being executed. During the Fill Buffer and Empty Buffer functions, starting address and word count are transferred between the LSI-11 and the Z-80 via the RXDB. During the Write and

Read Sector functions, the sector and track addresses are sent to the Z-80 from the LSI-11 through the RXDB. For the Read Status function, the RXES Status Register is placed in the RXDB by the Z-80. For the Diagnostic Function, parameters defining the diagnostic operation to be performed are placed in the RXDB register as described in Section 4.2. At the end of execution of each function, the RXES Status Register contents are loaded into the RXDB to be read by the LSI-11.

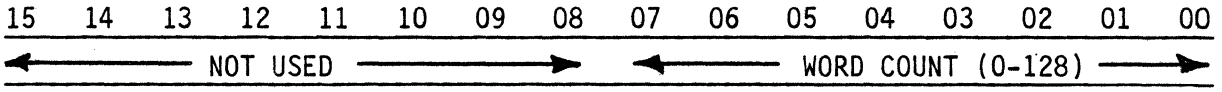
The formats for the data transfers and track and sector addresses are shown in Figure 3-2:



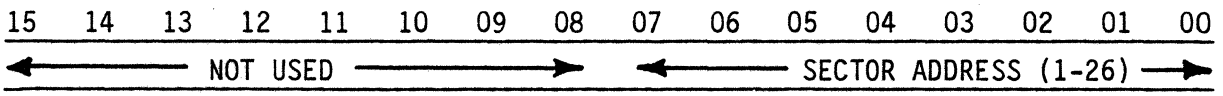
Format for RXBA DMA Memory Address



Format for RXTA Track Address



Format for RXWC DMA Word Count



Format for RXSA Sector Address

FIGURE 3-2
RXDB Register Formats

The Error and Status Register (RXES) is displayed in the RXDB at the completion of each function and after a Read Status function command. The format for the RXES Register is shown in Figure 3-3:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
← LAST FUNCTION →				NXM	WC OVFL	HD SEL	UNIT	BUB RDY	DD	DRV DEN	DEN ERR	AC LO	ID	SIDE 1 RDY	DAT ERR

FIGURE 3-3
RXES Register Format

The definitions for each of the applicable RXES bits are:

BIT NUMBER	NAME	DEFINITION
00	DATERR	When set, an uncorrectable data error has been detected during the reading of a sector of data from bubble memory. The ERROR and DONE bits are set in the RXCS register. (For DEC RX01 systems, this bit is called "CRC Error".)
01	SIDE 1 READY	When double-sided disk emulation is enabled by jumper W8, this bit will be set.
02	ID	The INITIALIZE DONE bit is set after the completion of the INIT function which occurs after power-up or when either a software or a hardware (bus) INIT is received by the controller.
03	AC LO	Set when a power failure is detected by the QBC-11/02.
04	DEN ERR	Set when a READ, WRITE, FILL, or EMPTY is attempted without the RXCS DENSITY bit set.
05	DRV DEN	Always 1.
06	DD	The DELETED DATA bit used in RX02 floppy-disk systems will always be set to 0 by the QBC-11/02.
07	BUBRDY	The BUBBLE READY bit will be set whenever the bubble-memory system is ready for operation. This bit is valid only after a Read Status or INIT function. (For DEC RX02 systems, this bit is called "DRVRDY".)

08	UNIT	The least-significant bit of the selected BUBBL-BOARD number.
09	HD SEL	When double-sided disk emulation is enabled by jumper W8, the side selected for the last function executed will be indicated by this bit.
10	WC OVFL	Set when a FILL or EMPTY buffer command is attempted with a word count larger than the sector size.
11	NXM	Set when any DMA operation detects non-existent memory.
12-15	CMD	These bits echo the last function executed by sending back the four least-significant bits of the RXCS Register.

3.3 FUNCTION CODES

A 3-bit function code is sent to the controller in bit positions 03, 02, and 01 when the RXCS is written by the LSI-11. Bit 00 must also be set to enable the Z-80 to perform the function. The function codes are described below. (Refer also to Section 3.1.)

3.3.1 FILL BUFFER (000)

Before a sector of data can be written into bubble storage, the Sector Data Buffer on the QBC-11/02 controller must be loaded with the 128 sixteen-bit words of data. This is accomplished with the Fill Buffer function.

When the controller receives a Fill Buffer command, it negates DONE and asserts the Transfer Request (TR) flag in the RXCS register. The LSI-11 senses the TR flag and loads the word count into the RXDB, clearing the TR flag in the process. The word count is normally 128 for a full sector transfer. The Z-80 senses the clearing of the TR flag and stores the count internally. It then re-asserts the TR flag. The LSI-11 then sends the 16 least-significant bits of the starting DMA address for the data transfer, which clears TR. The Z-80 then fills the Sector Buffer from memory. If the word count is less than 128, zeroes fill the remainder of the buffer. The DONE flag is then asserted by the Z-80, signifying completion of the function.

3.3.2 EMPTY BUFFER (001)

This function transfers data from the Sector Buffer to the host. Word count and starting address are sent via the RXDB as for FILL BUFFER function.

3.3.3 WRITE SECTOR (010)

The Write Sector function transfers data from the Sector Buffer to the bubble device. When the QBC-11/02 controller receives a Write Sector command from the LSI-11, it negates the DONE flag and asserts the TR flag to request the sector address. When the LSI-11 loads the sector address into the RXDB register, TR is negated by the Z-80. The QBC-11/02 will then assert the TR flag again to request the track address.

The track address has a special format, since the QBC-11/02 can support sixteen 'units' instead of only two. The most-significant byte of the track address is interpreted as a 'board number' used to select the unit for data transfer. The QBC-11/02 logically OR's this byte with the Unit Select bit in the RXCS, so units 0 and 1 can be selected in the standard way. Do not use byte instructions (e.g., MOVB) to write the track address since the LSI-11 will write the same byte to both halves of the RXDB, usually resulting in erroneous BUBBL-BOARD selection.

After the LSI-11 loads the track address into the RXDB, the QBC-11/02 will check to see that the track address is greater than or equal to zero, and the sector address is in the range 1-26. It also checks to see that the combined track and sector address actually exists within the bubble-memory system. If any of these conditions is not met, the 040 error code is placed in the Error Register. The QBC-11/02 also checks to see if the selected BUBBL-BOARD actually exists in the system, and if it is write-protected (jumper W8 installed). If the BUBBL-BOARD does not exist, an 060 error code is placed in the Error Register. If the BUBBL-BOARD is write-protected, an 065 error code is placed in the Error Register. If any of the above three errors are detected, the ERROR flag is set in the RXCS register, and DONE is asserted. An interrupt is also generated if the Interrupt Enable bit is set.

If no errors are detected, the green RUN light will be turned on, and the QBC-11/02 controller will write the data into bubble storage. Four bubble devices are operated in parallel to store the 128-word sector in one 64-byte page of each bubble. An error-correction code is also written into each page of the sector.

After the data has been written to bubble storage, the QBC-11/02 controller turns off the RUN light, asserts the DONE flag, and generates an interrupt if the Interrupt Enable bit (bit 6) is set in the RXCS register. The contents of the Sector Buffer are not destroyed during the Write Sector operation.

3.3.4 READ SECTOR (011)

The Read Sector function is used to transfer information from bubble storage to the QBC-11/02 Sector Data Buffer. When the LSI-11 writes the Read Sector command into the RXCS register, the DONE flag is negated by the Z-80. The QBC-11/02 controller then asserts the TR flag twice, as in the Write Sector function, to obtain the sector address and then the track address for the read operation. The addresses are checked for out-of-range conditions as in the Write Sector function, and the DONE and ERROR flags are set if an error is detected. Also, if an addressing error is detected, an 040 or 060 error code will be placed in the Error Register.

If the address is within range, the QBC-11/02 controller reads the page corresponding to the specified sector-address from bubble storage, checking the data for errors as it is read. Correctable errors are automatically corrected. If an un-correctable error is detected, the QBC-11/02 will re-read the sector. This automatic retry will be repeated until either the sector is read without error, or the re-try count equals four. If after four retries an uncorrectable error is still detected, the RXES register contents (with the DATA ERROR bit set) are loaded into the RXDB register, and the DONE and ERROR flags are set in the RXCS register. If no errors are detected, the DONE flag is set, and an interrupt is generated if the Interrupt Enable bit is set in the RXCS register.

3.3.5 POWER CONTROL and DIAGNOSTIC FUNCTIONS (100)

When the QBC-11/02 receives a function code four (100), it will assert the Transfer Request (TR) flag to get further information from the LSI-11. The LSI-11 processor should respond by placing in the RXDB either a diagnostic-function select code as described in Section 4.2, or one of the two power-control select codes.

Normally, as explained in Section 2.1, only two (at most) BUBBL-BOARDs are powered up at any given time. However, each time a BUBBL-BOARD is brought from the powered-down state to the standby state, it must be re-intialized by the controller. Since the initialization process takes as much as several hundred milliseconds, overall system speed can be enhanced by keeping more than two BUBBL-BOARDs in stand-by. The power-control commands make it possible to take advantage of this possibility - trading off additional power consumption to gain an increase in throughput.

To put a given BUBBL-BOARD in standby mode, the user should place the octal power-on code 400*b+10 in the RXDB in response to the TR flag, where 'b' is the board address for the particular BUBBL-BOARD. The QBC-11/02 will then leave that BUBBL-BOARD powered up until a power-off code is received.

To turn power off on a given BUBBL-BOARD, the LSI-11 should place the power-off code 400*b+11 in the RXDB in response to the Transfer Request flag.

Function code four (100) is also used to implement several diagnostic functions; see Section 4.2. Both the power-control codes and the diagnostic codes are non-DEC commands; they are intended for use only with bubble memory.

3.3.6 READ STATUS (101)

The Read Status command will cause the QBC-11/02 controller to place the contents of the RXES Error and Status Register into the RXDB. The RXES contents will contain information produced as a result of the last command executed by the QBC-11/02. The DONE flag is negated by the Z-80 when the LSI-11 loads the Read Status command into the RXCS register, and is re-asserted by the controller after the RXES has been loaded into the RXDB. As usual, an interrupt will be generated if the Interrupt Enable bit is asserted when the DONE flag is set.

3.3.7 WRITE DELETED DATA SECTOR (110)

The Write Deleted Data Sector function is the same as the Write Sector function. No deleted data marks are written by the QBC-11/02 controller.

3.3.8 READ ERROR CODE (111)

When the Read Error Code command is written into the RXCS by the LSI-11, the DONE flag is negated, and TR is asserted. The LSI-11 sends a starting address to the RXDB, and the QBC-11/02 dumps four words into memory. The first byte sent contains definitive error codes which are defined below. This error code is placed in the RXDB Register, DONE is asserted, and an interrupt is generated if the Interrupt Enable bit is set.

The error codes are:

Code	Error
001	Correctable Read Error
002	Uncorrectable Read Error
004	BMC Timing Error
010	Initialization Error
040	Track and/or sector address out of range
060	No such board (unit)
065	Addressed BUBBL-BOARD is write-protected
230	DMA word count too large
240	Density error
250	Non-existent-memory (NXM) error during DMA
260	BUBBL-BOARD power bad

The other information returned is:

Word	Bits	Meaning
0	15-8	Word Count
1	7-0	Unit 0 Track Address
1	15-8	Unit 1 Track Address
2	7-0	Last Track Addressed
2	15-8	Last Sector Addressed
3	15-0	Not used

3.4 SECTOR INTERLEAVING

In order to allow the LSI-11 to keep up with a rotating floppy disk during read and write operations, DEC's RT-11 floppy-disk handler provides 2:1 sector interleaving of the 26 sectors on each track, as well as a 6-sector track-to-track skew. The QBC-11/02, however, stops rotating the bubble device at the end of each sector transfer, and is ready to instantly access the next sector no matter how long the LSI-11 takes to fill or empty the Sector Data Buffer. To optimize throughput of the QBC-11/02, the sector addresses within the bubble device have been arranged by the Z-80 to correspond to a sector interleave as produced by the RT-11 handler.

Programmers using a non-RT-11 handler which does not incorporate sector interleaving should set the most-significant bit (bit 7) of the track-address byte passed to the QBC-11/02 during write-sector and read-sector operations. Setting this bit will cause the QBC-11/02 to bypass its interleave algorithms, thereby speeding operations with a non-RT-11 handler. See the example non-RT-11 handler in Appendix B.

3.5 PROGRAMMING EXAMPLES

Examples of the programming sequences required to fill the Sector Buffer, write a sector, read a sector, and empty the Sector Buffer may be found in the sample stand-alone driver described in Appendix B. More examples may be found in Chapter 3 of DEC's RXV21 Floppy Disk Operating Manual.

> NOTE
>
> If you are using an LSI-11/23 or are writing your own device drivers, please note that the BIS and BIC instructions will not work on the QBC-11/02 registers as may be expected. This is because these commands use the DATIO bus cycle (read/modify/write) and most of the bits of the status registers are either read-only or write-only bits, as they are in the RXV21/RX01 floppy-disk system. Also, byte commands should not be used to write into the RXCS or RXDB.

SECTION 4.0

DIAGNOSTICS

Two levels of diagnostics are available to aid the user in assuring that the bubble-memory system is working properly. The first level is built into the QBC-11/02 firmware and does not require any LSI-11 host software. These "self-test" functions can be triggered by installing a jumper strap on the QBC-11/02, and powering up the module. Alternatively, self-testing can be started with LSI-11 console ODT commands.

If a self-test is started by installing the jumper strap, it is not even necessary that the QBC-11/02 module be installed in an LSI-11 system. Simply apply power to the printed-circuit board, and provide a pull-up on the BINIT L bus line (line AT2). An overall test result can be obtained by viewing the LED indicators on the attached BUBBL-BOARD(s).

A second level of diagnostics uses LSI-11 host software to test the entire system, including the interface to the Q-bus. Each of these diagnostic procedures will be described in this section.

4.1 MANUALLY-INITIATED SELF TEST

To manually start a self-test, the user must install a shunt in location W10, and power up the module. The QBC-11/02 will light the green RUN led on BUBBL-BOARD 0, perform its normal initialization, and then start a write/read test using a random data pattern. The controller will sequentially test all BUBBL-BOARDs in the system starting with BUBBL-BOARD 1. The test first writes a pattern in all sectors of the bubble device, and then reads it back.

If an uncorrectable data error is detected, the red ERR light on the board being tested is turned on and left on. The RUN light is also left on to indicate that the error was a data error, and testing continues.

If a fatal error is detected, such as the Z-80 not being able to communicate with the bubble device controller on a particular BUBBL-BOARD, the ERR light will be turned on, the green RUN light will go out, and testing will cease on that BUBBL-BOARD. If neither the ERR nor RUN light ever comes on, the QBC-11/02 module is completely inoperative, with the Z-80 unable to execute any instructions.

The QBC-11/02 will ignore any commands from the LSI-11 host processor as long as the W10 shunt is installed.

> CAUTION!
>
> The manually-initiated self-test destroys data in all BUBBL-BOARD(s) attached to the QBC-11/02. Use ODT or BUBL initiated self-tests if you want to test the system without losing previously-stored user data.

4.2 ODT-INITIATED SELF TEST

The QBC-11/02 firmware has a set of diagnostic routines which can be triggered by commands entered via the LSI-11 console ODT, or by higher-level software such as the BUBL diagnostic program described in Section 4.3, below. To initiate a diagnostic command, the user should send a function code 4 (100_2) to bits 3 through 1 of the RXCS register with the GO bit (bit 0) set; i.e., send an octal 011.

The QBC-11/02 controller will respond by clearing DONE and then setting the Transfer Request flag. The user should then send the code word defining the specific diagnostic function, as listed in Table 4-1. If more parameters are required to further define the function, the QBC-11/02 will raise the TR flag for each parameter, until all are transferred. The DONE flag is set in the RXCS Register at the completion of each diagnostic function, and the RXCS ERROR bit is set if an error is detected during the diagnostic function that was executed.

> NOTE
>
> All data errors are classified as soft (automatically corrected by the controller simply by re-reading the data), correctable (automatically corrected by the controller hardware), or un-correctable. Only in the latter case has data actually been lost.
>
> When operating the bubble-memory system at temperatures approaching the upper end of the specification range, it is normal for the correctable and/or soft data-error rate to increase. These errors are all automatically corrected by the controller, however, and are not cause for concern. Uncorrectable data errors will not normally be found within the specified temperature range.

OCTAL CODE	FUNCTION
0	READ ERROR LOG - Makes the 10-byte Error Log available for transfer to the LSI-11 host processor, via the RXDB register. The Transfer Request (TR) flag is asserted in the RXCS register for each byte to be transferred. (NOTE: The Error Log is cleared at power-up.) The Error Log bytes are: Byte # — Description 1-4 — Number of sectors read since power-up. Byte 1 is least-significant byte. 5-6 — Number of sectors read having correctable (automatically corrected) data errors. Byte 5 is LSB. 7-8 — Number of sectors read having soft data errors (automatically corrected by re-read). Byte 7 is LSB. 9-10 — Number of sectors read having un-correctable data errors. Byte 9 is LSB.
400*b + 1	READ-VERIFY TEST - Reads all sectors on the BUBBL-BOARD selected by 'b'. At completion of an entire read pass, sets DONE flag. Sets ERROR flag in RXCS register if any data error is detected. Also turns on the red ERR light and updates Error Log if an error is detected. This is a "read-only" (non-destructive) test.
400*b + 2	WRITE/READ TEST - Writes a random data pattern into every sector on the selected board and then reads all sectors. At the completion of the test, the DONE flag is set. If any data error is detected, the ERROR flag is set in the RXCS register. Turns on the red ERR light and updates the Error Log if an error is detected.
400*b + 3	BOOTLOOP READ FUNCTION - Reads the 40-byte boot map from the bubble device and makes the 40 bytes available for transfer to the LSI-11, via the least-significant byte of the RXDB register. The TR flag is set for each byte to be transferred. Turns on the ERR light and sets the ERROR flag in the RXCS register if an error is detected. Sets DONE flag at completion. Hardware straps determine which device on board 'b' is accessed (see Appendix C).
400*b + 4	BOOTLOOP WRITE FUNCTION - Accepts 40 bytes from the LSI-11, via the least-significant byte of the RXDB register. The TR flag is set for each byte transferred. Then writes the 40 bytes into the boot-loop of the bubble device, and sets DONE flag. (Read Appendix C before using this function.) WRITING THE BOOT LOOP SHOULD NOT BE ATTEMPTED WITHOUT A THOROUGH UNDERSTANDING OF THE PROCEDURES INVOLVED.

TABLE 4-1

Available Diagnostic Functions

4.3 LSI-11 HOST DIAGNOSTIC SOFTWARE

An LSI-11 diagnostic program called BUBL is provided with each QBC-11/02 shipment to a new customer, on an RT-11 compatible single-density floppy diskette. BUBL contains two routines to exercise the bubble-memory system, and facilitates using the diagnostic functions described in Section 4.2.

To run BUBL, load the diskette into floppy drive 1 after booting RT-11. Then type:

```
RUN D?1:BUBL            (? is either X or Y)
```

The program will respond by typing the list of commands that BUBL can execute, and will then print a '>' as a prompt, and await a command to be entered. The commands available are shown in Table 4-2. Legal abbreviations of the commands are shown as underlined letters. Multiple commands and parameters may be placed on a single line, separated by spaces. The commands will not be executed until a carriage-return is typed.

COMMAND	DESCRIPTION	
SHOW	Displays a list of current parameters used by test routines.	
HELP	Displays list of valid commands.	
SET	Sets values in parameter list.	
CO2BUB	Runs bubble diagnostic routine.	
WIZBUB	Runs bubble-memory reliability test.	
ERRLOG	Prints 10-byte Error Log	(DIAG 0)
ROTEST	Runs read-only data-verification test	(DIAG 1)
WRTEST	Runs write-then-read self-test	(DIAG 2)
RDMAP	Reads and prints 40 bytes of hex boot-loop data	(DIAG 3)
WRMAP	Accepts 40 bytes of hex data & writes boot-loop	(DIAG 4)

TABLE 4-2
BUBL Commands

When the SHOW command is typed, BUBL displays the list of parameters used by the test routines, with their current values:

```
>S
 VARIABLE        VALUE      USED BY          VARIABLE        VALUE      USED BY
RXCS             177174     C  W
UNIT             0          C  W
FIRST TRACK      0          C  W             FIRST SECTOR    1          C
LAST TRACK       0          C  W             LAST SECTOR     1          C
PASSES           1          C  W  S
CORRECT ERR      T          C
READ ONLY        F          C
RANDOM DATA      T          C
PRINT FLAG       T          C
PATTERN          0          C

        C=CO2BUB          W=WIZBUB          S=Self-Tests (ROTEST and WRTEST)
```

In the list above, RXCS defines the QBC-11/02 Device Address. FIRST TRACK, FIRST SECTOR, LAST TRACK and LAST SECTOR define the range for the test. For use with the QBC-11/02, CORRECT ERR should be always be set to True. PASSES defines the number of times that the test will be repeated before returning to command mode. READ ONLY limits CO2BUB to the verification of previously written data if set to True, and allows a write-before-read test if set to False. RANDOM DATA causes CO2BUB to use a random data pattern if set to True, or the fixed octal pattern defined by PATTERN if set to False. Setting the PRINT FLAG to False suppresses error printouts so that the test can proceed faster while using an oscilloscope for de-bugging. The default parameters used by the test routines are as shown in the list above. Only the underlined character(s) need be typed to specify a parameter for modification.

The SET command can be used to change the parameters to new values, as shown in the following example:

```
>SE F T 128 SE F S 1 SE L T 206 SE L S 16
```

The command line above will allow the entire bubble-storage area on one BUBBL-BOARD to be tested (128 has been added to the track addresses to defeat the controller's de-interleaving algorithm, which is needed only when using the RT-11 handler). Note that a space delimits all commands and parameters.

C02BUB should be used to get detailed information concerning QBC-11/02 behavior. As can be seen in the parameter list, C02BUB has many operational modes, so it can be set up for specific tests as required. After setting the parameter list to the desired values, typing C followed by a carriage-return will cause BUBL to respond as follows:

```
>C
ACCESSING AN RX02       (if the QBC-11/02 is in RX02 mode)
WRITING PASS 1
```

At the completion of the write portion of the test, WRITING PASS 1 is over-written with READING PASS 1. At the completion of each pass, the pass number is updated. If any errors are detected, the track, sector and byte numbers are printed along with the data patterns written and read back.

WIZBUB is designed for longer term reliability testing. After setting up the parameter list, typing W followed by a carriage-return will cause BUBL to respond with:

```
>W
ACCESSING AN RX02
WRITING PASS 1   UNIT 0   TRK 0
```

The track number is updated after writing into each track in the bubble device. WRITING is changed to READING when the last track specified has been written. The pass number is updated at the end of each complete pass.

If any errors are detected, WIZBUB will print an error count for each track, as well as an error code as defined in Section 3.3.8.

> NOTE
>
> When operating the bubble-memory system at temperatures approaching the upper end of the specification range, it is normal for the correctable and/or soft data-error rate to increase. These errors are all automatically corrected by the controller, however, and are not cause for concern. Uncorrectable data errors will not normally be found within the specified temperature range.

The RDMAP and WRMAP commands use the UNIT number to access a particular BUBBL-BOARD. Jumpers on the board determine which of the four bubble devices is selected (see Section 2.9).

SECTION 5.0

OPERATION WITH RT-11

The QBC-11/02 bubble-memory controller is designed to respond to DEC's RXV-21/RX02 floppy-disk command protocol. Thus, it will operate with RT-11, either as the system device or as a non-system, file-oriented device. To operate as a system device, a method must exist to load RT-11 into the bubble-memory system. Suggested techniques for transferring RT-11 system and data files from an existing floppy disk into the bubble-memory system are discussed in this section.

5.1 DEVICE HANDLERS

To operate with a floppy disk and the bubble-memory system simultaneously installed, RT-11 must be configured to handle two floppy-type controllers. Two methods are available to add a second controller to the system. First, the standard DEC RX02 handler, DY.SYS, may be set up to handle two controllers. This technique has limitations, however, in that the QBC-11/02 controller would then be limited to controlling a maximum of two BUBBL-BOARDs.

Another alternative is to add a second (non-system) handler to RT-11. A description of the relatively simple procedures required to modify DY.MAC to change the device and vector addresses and device name so that it may be compiled and installed in the system is contained in Appendix A. Adding a second handler has the advantage of allowing full access to as much as 8 megabytes of storage through a single QBC-11/02 controller.

> CAUTION
>
> If you plan to use the ROM-stored bootstrap to directly boot from the QBC-11/02 controller, you must use a device address that corresponds with one available in the ROM on the QBC-11/02 module. The QBC-11/02 ROM normally assumes the device address is 177170, but may be set to use 177150 by installing jumper W11. It may be convenient to generate two 'BY' handlers, one for copying to the bubbles, and one for use as a system device.

5.2 DEVICE INITIALIZATION

After provision has been made for the second controller, the bubble system must be set up for file directories. The BUBL diagnostic program (see Section 4.3) should first be run to insure that the QBC-11/02 is responding properly to LSI-11 commands. The RT-11 INIT command should then be used just as with a new diskette that is being used for the first time with RT-11.

In the following discussion, it will be assumed that a second handler called BY.SYS (using device address 177150) was added to the system so the device names controlled through the QBC-11/02 are BY0 .... BY7.

First, install the handler, using the command:

```
.INSTALL BY
```

Then, initialize the directory with the command:

```
.INIT BY0:
```

The system will ask:

```
BY0:/Init are you sure?
```

You should respond with a Y and a carriage return.

The INIT command should take about five seconds to execute. If the system 'hangs', check to see that the device and vector addresses are properly set up on the QBC-11/02 module, and that the second handler has been correctly modified and installed.

Repeat the INIT for each bubble board in the system. After the bubble-memory system has been initialized, try getting a directory with:

```
.DIR BYk:              (k is any valid board number)
```

The system should type (for example):

```
15-Apr-82

0 FILES, 0 Blocks
974 Free blocks
```

5.3 COPYING FROM FLOPPY DISK TO BUBBLE MEMORY

The standard RT-11 COPY command may be used to transfer files from the floppy disk to bubble memory. Remember to use the /SYS switch when transferring system files, as in the following:

```
.COPY/SYS DY0:RT11SJ.SYS BY0:

.COPY DY0:DIR.SAV BY0:
```

After copying all of the files that are needed, the bubble-memory system may be set up for bootstrapping with:

```
.COPY/BOOT BY0:RT11SJ.SYS BY0:
```

5.4 BOOTSTRAPPING RT-11

If you have done the copy operations described above, including the COPY/BOOT, you can easily switch to the bubble system by typing

```
.BOOT BY0:
```

If you had copied DY.SYS to the bubbles, you can still access your floppy.

If you have installed jumper strap W11 on the QBC-11/02 module, the bootstrap ROM will assume the RXCS is at 177150, and you can boot directly from it.

SECTION 6.0

OPERATION WITH RSX-11

The QBC-11/02 bubble-memory controller is designed to respond to DEC's RXV21/RX02 floppy-disk command protocol. Therefore, integrating the QBC-11/02 into an RSX-11 system requires taking essentially the same steps one would take to integrate an RX02 floppy into the RSX environment.

What follows is a brief summary of the steps that might be taken to accomplish this objective. It is assumed that the reader is familiar with the necessary utilities that DEC supplies with the RSX software. Since the user's specific objectives can vary depending on his application, four different cases are discussed below to illustrate typical methods for integrating the QBC-11/02 into RSX.

6.1 SUBSTITUTING A QBC-11/02 FOR AN EXISTING RX02

System integration is a trivial task for this case because no software changes are required. Simply remove the RX02 controller from the system, and insert the QBC-11/02, making certain the selected RXCS address and interrupt vector match those of the replaced unit. Remember that the QBC-11/02 has an on-board bootstrap ROM which must be set to the correct address, or possibly disabled if it conflicts with a bootstrap elsewhere in the system.

Turn on the system, initialize the directory structure on the QBC-11/02, and you are "on the air".

6.2 ADDING THE QBC-11/02 TO A SYSTEM CURRENTLY WITHOUT RX02's

The word here is SYSGEN. Although it may be possible to create a loadable version of DYDRV, the process is very cumbersome and not recommended except as an excercise for aspiring wizards. Do a SYSGEN including DY support, then proceed as in Case 1.

6.3 ADDING THE QBC-11/02 TO A SYSTEM WITH RX02's

Here, too, do a SYSGEN, increasing the number of DY controllers to include the number of QBC-11/02's being added to the system.

6.4 USING THE QBC-11/02 AS AN RSX-11S BOOT DEVICE

Proceed in easy stages. First build an RSX-11M system which includes the QBC-11/02. Doing this falls into the operations outlined above. Transfer the RSX-11S system image to bubble storage using the VMR utility. Assuming the bootstrap on the QBC-11/02 is enabled, the -11S system may now be booted.

Note that a DY driver is not required in the RSX-11S system if the only use of the bubble device is system booting. If, however, the bubbles are to be accessed by the system after the boot, the appropriate driver must be SYSGEN'd into the system. (Remember that -11S supports only a limited sub-set of File Control Services.)

6.5 CORRECTING FOR DEVICE SIZE DIFFERENCES

One QBI-11/512 contains 1000 data blocks, while an RX02 diskette holds 988. To access this extra capacity, the driver DYDRV.MAC must be edited and assembled. Change the phrase 'SINDOU = 988.' to 'SINDOU = 1000'. Then do a SYSGEN. (Another alternative is to patch the system data structures, but this is not recommended for any but the expert user.)

SECTION 7.0

THEORY OF OPERATION

This section describes the fundamental operation of the QBC-11/02 bubble-memory controller, as well as the companion QBI-11 BUBBL-BOARD.

As can be seen in the block diagram, Figure 7-1, the QBC-11/02 controller is designed around a Z-80 microprocessor. The Z-80 receives commands from the LSI-11 Q-bus, interprets the commands, and performs the requested function. These functions include filling the Sector Buffer with 256 bytes of information from the LSI-11, mapping floppy-disk track and sector addresses into bubble-memory page addresses, seeking a sector, writing a sector into bubble memory, reading a sector from bubble memory, and emptying the Sector Buffer by transferring data to the LSI-11.

In a typical write operation, the LSI-11 first sends a Fill Buffer command to the LSI-11 Command Buffer register on the QBC-11/02, and clears the DONE flag in the handshaking logic. The Z-80 senses the clearing of the DONE flag and reads the command from the Command Buffer. The Z-80 then determines that it is a Fill Buffer command and raises a Transfer Request (TR) flag in the handshaking logic, which the LSI-11 can sense. When the LSI-11 senses the transfer request, it loads the direct-memory-access (DMA) word count into the Data Buffer, and clears the TR flag. The Z-80 senses the clearing of the Transfer Request flag, reads the count from the Data Buffer and stores it. It then raises Transfer Request to notify the LSI-11 that it is ready for the DMA starting memory address. After the LSI-11 loads the memory address into the RXDB, the Z-80 transfers the data from LSI-11 memory to its own memory, via DMA. At the completion of the operation, the Z-80 sets the DONE flag in the RXCS Register.

The LSI-11 may then request a Write Sector function by loading a command into the LSI-11 Command Buffer, thereby clearing DONE. The Z-80 interprets the Write Sector command after it senses the clearing of the DONE flag, and raises the Transfer Request (TR) flag to get the sector address from the LSI-11. The LSI-11 loads the sector address into the RXDB in response to the transfer request, and clears the TR flag. The Z-80 reads the sector address,

and repeats the transfer request handshaking operation to get a track address. The Z-80 then maps the track and sector address into a bubble-memory page address, selects a BUBBL-BOARD (powering it up if necessary), and starts the bubble-memory seek and write sequence.

The Z-80 starts a write sequence by sending a Write command to the bubble device controller (BDC) shown in Figure 7-2. The BDC sets a flag to indicate that it is ready for the Z-80 to load data into the BDC's FIFO buffer. The Z-80 supplies bytes to the BDC every time the Data Request flag is raised, until the entire 256-byte sector has been transferred.

The BDC provides timing pulses to the Function Driver, Coil Pre-driver and Formatter-Sense-Amplifier (FSA) for each of the bubble devices on the board. Data is shifted serially from the BDC to another FIFO built into the FSA. The FSA passes the data on to the Function Driver, appending an error detection and correction code in the process. The Function Driver produces the current pulses necessary to write data into the bubble device itself.

The bubble device is organized as 320 minor loops similar to shift registers. Each minor loop contains 4096 bit positions. In each bit position, the existence of a magnetic domain or "bubble" represents a "1" and the absence of a bubble represents a "0". All of the loops are shifted simultaneously by a rotating magnetic field.

To write into the bubble device, the Function Driver sends a current pulse to a "Generate" coil which splits an existing "seed" bubble and puts the "new" bubble onto a write track. This action is repeated for every "1" in a 320-bit page of data, as the bits are shifted down the Write Track by the rotating magnetic field. When the bits on the Write Track are properly aligned, a "Swap" gate transfers all of the bits in parallel into one bit position on each of the 320 minor loops.

A read operation is the inverse of the write operation. The host LSI-11 sends a Read Sector command to the Command Buffer. The Z-80 then asks for the track and sector address via the Transfer Request flag handshaking protocol, calculates the board and page addresses, and issues a Read command to the BDC on the selected BUBBL-BOARD. The BDC starts sending timing pulses to the Coil Drivers to rotate the minor loops so that the addressed page of 320 bits appears under the "Replicate" gates. Current pulses sent to the Replicate Gates split the bubbles, transferring a "new" bubble to the Read Track (leaving the original in the minor loop) for every loop containing a bubble in that bit position. After the page has been transferred to the Read Track it

is shifted past a magneto-resistive detector, which produces a pulse of a few millivolts when a magnetic bubble passes under it. The FSA detects the pulse, and shifts the bit into its FIFO buffer.

Although there are 320 minor loops in the bubble device, some of the loops may be defective and cannot be used to store data. The manufacturer identifies up to 48 loops as "bad", and stores a bit for each such loop in a special loop called the "Boot Loop". The Boot Loop also contains information for shifting the minor loops to a known "synchronizing" position at power-up time. During the power-up sequence, boot-loop information is read from the Boot Loop into the FSA and stored in a register. Logic in the FSA then is used to inhibit writes-to and reads-from the bad loops defined by the boot-loop data, or "bootmap".

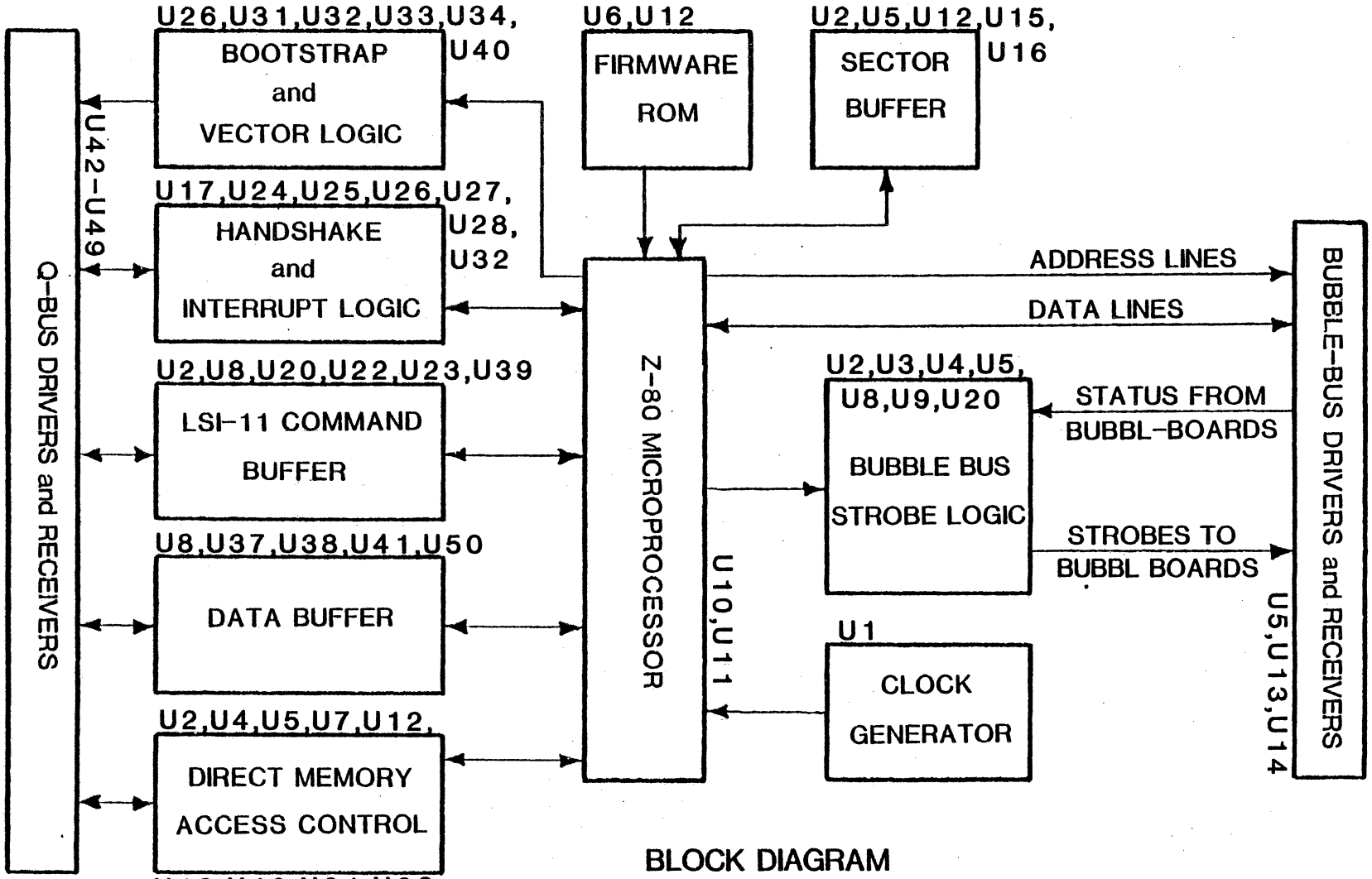
The FSA also performs the error detection and correction function. A 14-bit error-correction code is stored with each page of data, and the data is checked as it is read from the bubble device into the FSA. If an error is detected, the data is recycled through the error-correction circuitry to correct the error before it is sent on to the BDC and the Z-80. If the error is found to be uncorrectable by the FSA circuitry, the Z-80 will try to read the data four more times before notifying the LSI-11 host that an error has been detected. If a correctable error is detected, the Z-80 will automatically re-write the sector with the corrected data so permanent hard errors will not remain in the stored data.

The coil drivers rotate the magnetic field at a 50-KHz rate. Because the bubble device contains two write tracks and two read tracks that are operated in parallel, the peak data transfer rate is 100 K-bits per second. There are, however, overhead functions involved that reduce the effective overall data-transfer rate. It takes 369 or 371 field rotations to transfer a complete page from the minor loops and shift it by the detectors. Each page contains 512 data bytes, excluding the redundant-loop bits and error-correction code, so the effective data rate is 512 bits per page times 50,000 rotations per second divided by 370 rotations per page, or about 69,000 bits per second.

When four 1-megabit bubble devices are installed on the BUBBL-BOARD, all four devices are accessed in parallel during RX02-mode read and write operations. Each bubble device stores 64 bytes of data in each page, so the full 256-byte sector is transferred during a single bubble page access, taking less than 9 milliseconds. The transfer rate between the BUBBL-BOARD and the Z-80

is thus in excess of 275 kilobits per second.

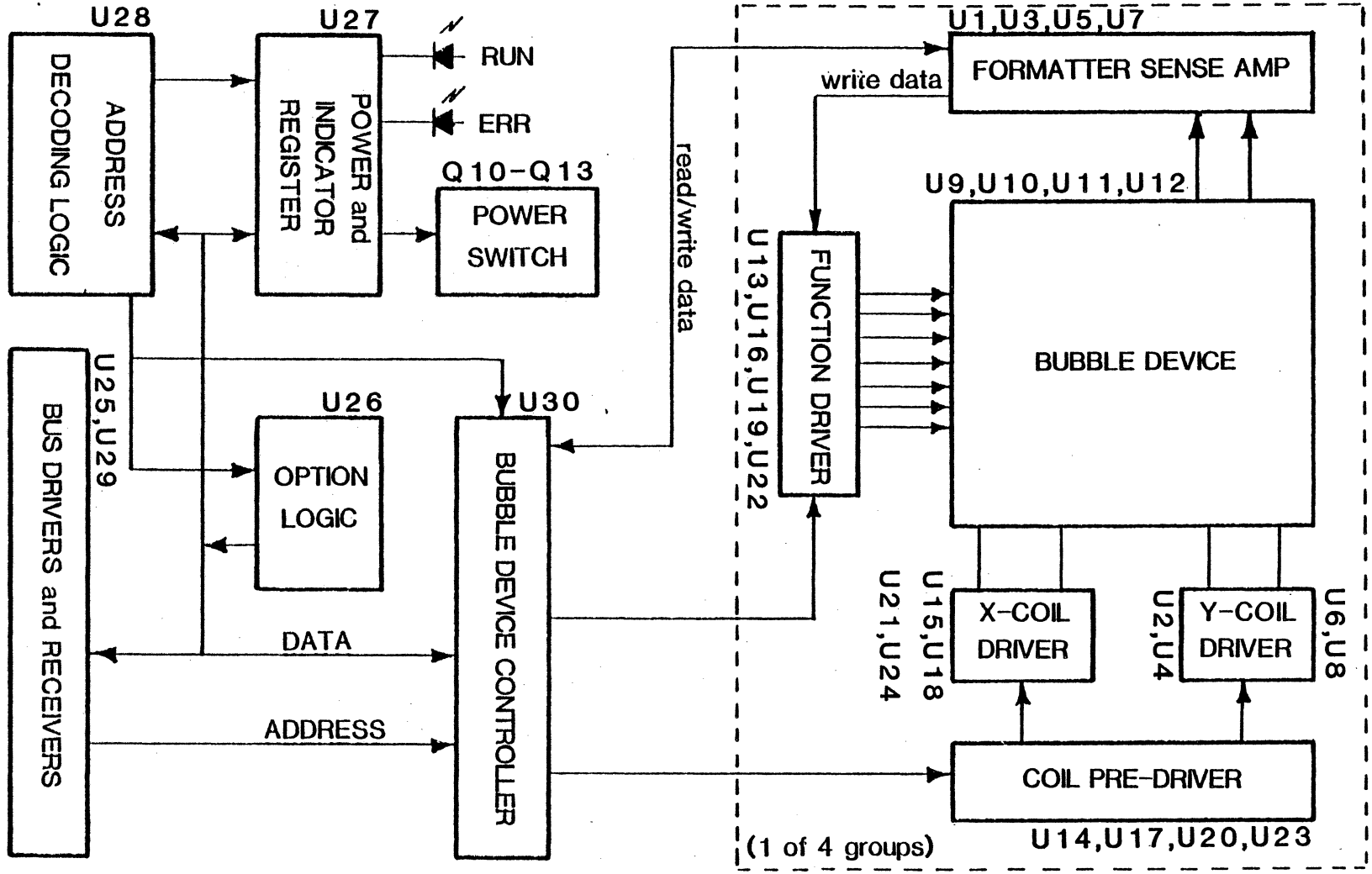
When operating in RX01 mode, or when only two bubble devices are installed on the BUBBL-BOARD, two bubbles are operated in parallel. Since 128 bytes can be stored in each page (64 x 2), only one page is accessed in RX01 mode for each 128 byte sector. In RX02 mode, two successive pages are accessed for a full 256-byte sector.



BLOCK DIAGRAM

QBC-11/02 CONTROLLER

FIGURE 7-1



BLOCK DIAGRAM

BUBBL-BOARD

FIGURE 7-2

APPENDIX A

ADDING AN RT-11 HANDLER FOR THE QBC-11/02

Adding a handler to RT-11 to accomodate the QBC-11/02 controller involves editing a copy of DY.MAC, then assembling, linking and installing the edited file. Three levels of revision are explained in this section. The first involves merely changing the default device address and vector so the QBC-11/02 can be used without conflicting with an RX02 floppy system which may also be present in the system. (Of course if the QBC-11/02 will use device address 177170 and vector 264 no changes are necessary to the handler; the QBC-11/02 is simply strapped to use these addresses. The bubble system is then operated as the 'DY' device.)

More extensive alteration of DY.MAC (herein termed level-2) allows the QBC-11/02 (strapped for single-sided operation; jumper W8 omitted) to operate with up to eight unit numbers (BY0: thru BY7:) Each unit corresponds to one 512-Kbyte BUBBL-BOARD module. (After the changes are made, the handler will still work with an RX02 floppy system, but will map all even unit numbers to Drive 0, and all odd unit numbers to Drive 1.)

The most extensive handler changes (level-3) allow up to sixteen BUBBL-BOARDs to connect to the QBC-11/02, but strict floppy compatiblity is lost. With these changes, pairs of 512K BUBBL-BOARDs (0-1, 2-3, etc.) are accessed as 1-megabyte units 0, 1, ... 7. Each unit contains 1976 blocks of storage.

To make the alterations:

```
.ASSIGN XX DX              (where XX represents the physical device for DY.MAC)

.COPY DY.MAC BY.MAC

.EDIT BY.MAC
*(do the editing here)

.MACRO BY

.LINK/EXE:SY:BY.SYS BY

.INSTALL BY                (and you are now "on the air".)
```

To make first-level changes, the editing to be done involves only one line. Search for

```
.DRDEF   DY,6,FILST$!SPFUN$,494.,177170,264
```

and change it to

```
.DRDEF   DY,6,FILST$!SPFUN$,494.,<address>,<vector>
```

where <address> and <vector> correspond to those chosen by the appropriate jumpers on the QBC-11/02 module. Terminate the edit here unless you want to increase the maximum number of units (BUBBL-BOARD modules) from two to eight. (Use 500. instead of 494. if you want to access the extra storage on each 512K byte board.)

To increase the maximum number of units to eight, search for the pattern 'C<7>' (without the quotes, of course). Display a few lines. They should be

```
      BIC       #^C<7>,R0
      MOV       R0,R2
      ASR       R0
.IF EQ DYT$0
      BNE       5$
.ENDC
      BCC       1$
```

Change these lines to read:

```
      BIC       #^C<7>,R0
      MOV       R0,R2
      MOVB      R0,UNIT
      ASR       R0
      BCC       1$
```

The above alteration takes the unit number as passed by RT-11 and stores it in the most-significant byte of the track-number word which will be sent to the QBC-11/02 later. Now search for 'ADD #27.,R3' and display

```
        ADD     #27.,R3
        .BR     DYDOFN
```

which becomes

```
        ADD     #27.,R3
        MOVB    R2,TRK
        MOV     (PC)+,R2
TRK:    .BYTE   0
UNIT:   .BYTE   0
        .BR     DYDOFN
```

This change takes the track number as computed by the handler and puts it into the lower order byte of the 'track' word. Finally, search for 'CSDN,CSDN', and display:

```
SAVDEN: .WORD   CSDN,CSDN
```

which should become:

```
SAVDEN: .WORD   CSDN,CSDN
        .WORD   CSDN,CSDN,CSDN,CSDN,CSDN,CSDN
```

We have increased the size of a table used to store density information. The level-2 edit is now complete.

To make third-level changes, search for:

```
.DRDEF  DY,6,FILST$!SPFUN$,494.,177170,264
```

and change it to:

```
.DRDEF  DY,6,FILST$!SPFUN$,988.,<address>,<vector>
```

where <address> and <vector> correspond to those chosen by the jumpers on the QBC-11/02 module. This increases the size of each 'unit' to about 1 megabyte.

Search for the pattern 'C<7>'. Display a few lines. They should be:

```
        BIC     #^C<7>,R0
        MOV     R0,R2
        ASR     R0
.IF EQ DYT$0
```

Change these lines to read:

```
        BIC     #^C<7>,R0
        MOV     R0,R2
        MOVB    R0,UNIT
        ASLB    UNIT
        BR      1$
.IF EQ DYT$0
```

This takes the unit number from RT-11 and multiplies it by 2 to form a bubble board number. Search for 'CLRB R3' and display a few lines:

```
        MOVB    R3,R2
        CLRB    R3
        SWAB    R3
```

Edit this to be:

```
        MOVB    R3,R2
        BIC     #177400,R2
        CMP     #76.,R2
        BGT     10$
        SUB     #76.,R2
        BISB    #1,UNIT
10$:    CLRB    R3
        SWAB    R3
```

The above checks the track number derived from the relative block number passed to the handler, and sets up to access the 'odd' board of each pair if the track is 76 or greater. (At this point in the handler the track number should be 0-75.) The rest of the changes match those for level-2 revisions. Search for 'ADD #27.,R3' and display:

```
        ADD     #27.,R3
        .BR     DYD0FN
```

which becomes:

```
        ADD     #27.,R3
        MOVB    R2,TRK
        MOV     (PC)+,R2
TRK:    .BYTE   0
UNIT:   .BYTE   0
        .BR     DYDOFN
```

Finally, search for 'CSDN,CSDN', and display:

```
SAVDEN: .WORD       CSDN,CSDN
```

which should become:

```
SAVDEN: .WORD       CSDN,CSDN
        .WORD       CSDN,CSDN,CSDN,CSDN,CSDN,CSDN
```

And the level-3 edit is complete.

APPENDIX B

NON-RT-11 DEVICE HANDLER

Two routines are listed below to aid the user in programming the QBC-11/02 controller (strapped for single-sided operation; jumper W8 omitted) for non-RT-11 applications. Both routines follow the standard FORTRAN calling conventions. For non-FORTRAN use, the routines may be called with:

```
JSR PC,RDBUB
```

for the read routine, and

```
JSR PC,WRBUB
```

for the write routine.

Before calling the routines, R5 should be loaded with a pointer to the argument table. The argument table consists of five words in the following order:

```
Number of arguments (4)
Pointer to Disk Unit number, 0 to 15.
Pointer to Track number
Pointer to Sector number
Pointer to first byte in data buffer
```

The write routine (WRBUB) will fill the Sector Buffer on the controller and then write the data into the specified sector in bubble memory. The read routine (RDBUB) will read the specified sector from bubble memory and then transfer the data from the Sector Buffer into LSI-11 memory. On return, both routines leave a completion code in R0 as follows:

```
 0 = Normal return when funtion is successfully completed.
-1 = Bubble memory not done when routine called.
-2 = DONE set when TR expected
-3 = TR set when DONE expected
-4 = QBC-11/02 error flag set after function
```

```
EXAMPLE QBC-11/02 DRIVER 1-AUG- MACRO V04.00  1-AUG-82 PAGE 1


                        .TITLE EXAMPLE QBC-11/02 DRIVER 1-AUG-82
000000                  .REPT 0
                    THIS IS AN EXAMPLE OF A SIMPLE QBC-11/02 DRIVER.
                    IT OBEYS RT-11 FORTRAN CALLING CONVENTIONS.  TWO
                    ROUTINES ARE INCLUDED AND CAN BE CALLED AS

                            IERR=RDBUB(IUNIT,ITRK,ISECT,IBUFF)
                            IERR=WRBUB(IUNIT,ITRK,ISECT,IBUFF)

                    ON RETURN, IERR = 0     FOR NORMAL COMPLETION
                                     -1     BUBBLE NOT 'DONE' AT CALL
                                     -2     'DONE' SET WHEN 'TR' EXPECTED
                                     -3     'TR' SET WHEN 'DONE' EXPECTED
                                     -4     QBC-11/02 ERROR FLAG SET

                    ALL TRANSFERS ARE ONE SECTOR AT A TIME (256 BYTES).
                    NOTE THAT IBUFF IS AN ARRAY OF BYTES.

                    REGISTERS R0-R5 ARE NOT PRESERVED BY THESE ROUTINES.
                    THE COMPLETION CODE IS RETURNED IN R0.  THE REGISTERS
                    ARE USED INTERNAL TO THE ROUTINES AS FOLLOWS:

                            R0   USED ONLY TO RETURN COMPLETION CODE
                            R1   READ/WRITE FUNCTION CODE
                            R2   WORD COUNT FOR FILL/EMPTY
                            R3   RXCS (BUBBLE COMMAND REGISTER ADDRESS)
                            R4   RXDB (BUBBLE DATA OR STATUS WORD)
                            R5   POINTER TO ARGUMENTS OR TO IBUFF

                    THE FIRMWARE IN THE QBC-11/02 NORMALLY COMPENSATES
                    FOR  THE INTERLEAVING ALGORITHM OF THE RT-11
                    RX02 HANDLER.  FOR STANDALONE APPLICATIONS WHERE
                    THIS PROCESSING IS NOT REQUIRED, IT IS DEFEATED
                    BY ADDING 128 TO THE TRACK NUMBER.  THIS EXAMPLE
                    HANDLER DOES IT IN THE SUBROUTINE 'SETUP', SO
                    THE USER CAN STILL CALL THE HANDLER WITH ITRK
                    IN THE RANGE 0-76.

                    THE UNIT NUMBER IS SENT TO THE QBC-11/02 AS THE
                    MOST SIGNIFICANT BYTE OF THE TRACK NUMBER FOR A
                    TRANSFER.  THE 'UNIT' BIT IN THE RXCS IS NOT SET,
                    SO IF THESE ROUTINES ARE USED WITH AN RX02 ALL
                    UNIT NUMBERS MAP INTO UNIT 0.

                    THIS HANDLER IS MEANT TO BE ILLUSTRATIVE AND
                    MAY NOT BE DIRECTLY APPLICABLE TO ANY SPECIFIC
                    APPLICATION.   THIS SOFTWARE IS NOT WARRANTED TO
                    BE FREE OF DEFECT, AND BUBBL-TEC ASSUMES NO
                    RESPONSIBILITY FOR ITS USE IN ANY CUSTOMER
                    PRODUCT OR APPLICATION.
                        .ENDR
```

```
EXAMPLE QBC-11/02 DRIVER 1-AUG- MACRO V04.00  1-AUG-82 PAGE 2


                    ;
                    ; SYMBOL DEFINITIONS
                    ;
177174              RXCS=177174             ;CONTROL REG ADDRESS
                    ;
000200              TR=200                  ;TRANSFER FLAG
000400              DD=400                  ;DOUBLE DENSITY BIT
000040              DONE=40                 ;DONE FLAG
000020              UNIT1=20                ;BIT TO SELECT UNIT 1
000407              READ=7+DD               ;FUNCTION CODE FOR READ
000405              WRITE=5+DD              ;CODE FOR WRITE
000403              EMPTY=3+DD              ;CODE FOR BUFFER EMPTY
000401              FILL=1+DD               ;CODE FOR BUFFER FILL
                    ;
177777              NOTDN=-1                ;NOT DONE AT CALL
177776              BADDN=-2                ;DONE SET WHEN TR EXPECTED
177775              BADTR=-3                ;WHAT DO YOU THINK?
177774              QBFLAG=-4               ;QBC-11/02 ERROR FLAG SET
                    ;
                    ; ROUTINE TO READ A SECTOR
                    ;
000000  012701      RDBUB:: MOV     #READ,R1
        000407
000004  004767              JSR     PC,SETUP
        000136
000010  001047              BNE     XIT             ;ERROR
000012  004767              JSR     PC,GO           ;DO THE READ
        000042
000016  001044              BNE     XIT             ;ERROR
000020  012713              MOV     #EMPTY,(R3)     ;ISSUE EMPTY COMMAND
        000403
000024  004767              JSR     PC,SETDMA
        000200
000030  001037              BNE     XIT             ;ERROR
000032  000425              BR      FINI            ;WAIT FOR DONE
                    ;
                    ; WRITE A SECTOR
                    ;
000034  012701      WRBUB:: MOV     #WRITE,R1       ;FUNCTION CODE TO R1
        000405
000040  004767              JSR     PC,SETUP        ;PRELIMINARY STUFF
        000102
000044  001031              BNE     XIT             ;ERROR
000046  012713              MOV     #FILL,(R3)      ;ISSUE FILL COMMAND
        000401
000052  004767              JSR     PC,SETDMA       ;SETUP DMA TRANSFER
        000152
000056  001024              BNE     XIT             ;ERROR
                                                    ;HELP, I'M FALLING THRU !!
```

```
                        ;
                        ; ISSUE COMMAND TO READ OR WRITE A SECTOR
                        ;
000060  010113  GO:     MOV     R1,(R3)         ;ISSUE COMMAND
000062  004767          JSR     PC,BWAIT
        000134
000066  100021          BPL     DERR            ;DONE MUST BE SET
000070  012714          MOV     (PC)+,(R4)      ;OUTPUT SECTOR
000072  000000  SECTOR: .WORD   0
000074  004767          JSR     PC,BWAIT
        000122
000100  100014          BPL     DERR
000102  012714          MOV     (PC)+,(R4)      ;TRACK OUTPUT
000104  000000  TRACK:  .WORD   0
                        ;
000106  004767  FINI:   JSR     PC,BWAIT
        000110
000112  100412          BMI     TERR
000114  005713          TST     (R3)            ;LOOK FOR ERROR FLAG
000116  100003          BPL     OKAY
000120  012700          MOV     #QBFLAG,R0
        177774
000124  000207          RTS     PC
                        ;
000126  005000  OKAY:   CLR     R0              ;NORMAL RETURN
000130  000207  XIT:    RTS     PC
                        ;
000132  012700  DERR:   MOV     #BADDN,R0
        177776
000136  000207          RTS     PC
                        ;
000140  012700  TERR:   MOV     #BADTR,R0
        177775
000144  000207          RTS     PC              ;ERROR
```



```
                        ;
                        ; ROUTINE TO SETUP REGISTERS
                        ;
000146  012703  SETUP:  MOV     #RXCS,R3
        177174
000152  010304          MOV     R3,R4           ;COPY
000154  032724          BIT     #DONE,(R4)+     ;LOOK FOR DONE BIT
        000040
000160  001415          BEQ     5$              ;BRANCH IF NOT OKAY
000162  005725          TST     (R5)+           ;ADVANCE PTR
000164  113567          MOVB    @(R5)+,TRACK+1  ;WHICH UNIT?
        177715
000170  113567          MOVB    @(R5)+,TRACK    ;SAVE TRACK
        177710
000174  013567          MOV     @(R5)+,SECTOR   ;AND SECTOR
        177672
000200  011505          MOV     (R5),R5         ;PTR TO DATA BUFFER
000202  012702          MOV     #^D128,R2       ;XFER COUNT
        000200
000206  060267          ADD     R2,TRACK        ;CHEATER WAY TO OFFSET
        177672
000212  000745          BR      OKAY
                        ;
000214  012700  5$:     MOV     #NOTDN,R0       ;ERROR CODE
        177777
000220  000207          RTS     PC              ;BACK TO ORIGINAL CALLER
                        ;
                        ; ROUTINE TO WAIT FOR DONE OR TR FLAG
                        ; RETURNS WITH 'N' FLAG SET IF TR FLAG SEEN
                        ; WARNING!  THIS IS A DANGEROUS WAY TO DO THIS
                        ;           TEST.
                        ;
000222  105713  BWAIT:  TSTB    (R3)
000224  001776          BEQ     BWAIT
000226  000207          RTS     PC
                        ;
                        ; PASS DMA PARAMETERS TO QBC-11
                        ;
000230  004767  SETDMA: JSR     PC,BWAIT        ;WAIT FOR DONE OR TR
        177766
000234  100336          BPL     DERR            ;DONE !!
000236  010214          MOV     R2,(R4)         ;WORD COUNT
000240  004767          JSR     PC,BWAIT
        177756
000244  100332          BPL     DERR            ;DONE !!!
000246  010514          MOV     R5,(R4)         ;STARTING ADDRESS
000250  000726          BR      OKAY
        000001          .END
```

APPENDIX C

READING AND WRITING BOOT-LOOPS

Every effort has been made in the design of this bubble-memory system to protect the bubble devices from loss of the vital bits in the boot-loops. The boot-loops contain a synchronization code and a map of the good and bad loops in each bubble device. If the data in a boot-loop is destroyed, the controller will not be able to locate the page corresponding to Track 0, Sector 0 on power-up, and/or may access bad loops within the bubble device. The boot-loop data may be lost if either of the power supplies suddenly drops out of tolerance, as described in Section 2, or if one of the integrated circuits on the printed-circuit board malfunctions.

Some symptoms of bad boot-loop data are:

1. Initialization Error message.
2. The system hangs up with a green RUN light on continuously when an attempt is made to access the bubble devices.
3. Very high hard-error rates.

To check the boot-loop, run the LSI-11 BUBL diagnostic program described in Section 4.3, and use the RDMAP command. RDMAP will cause a 40-byte boot map for one bubble device in the system to be printed out. The particular device is selected by setting the appropriate jumpers on the desired BUBBL-BOARD module (see Section 2.9.1), and setting the UNIT variable in BUBL to the board address for that BUBBL-BOARD. Note that the map is printed in hexadecimal format with two hex digits per byte. The data are printed in five lines of 8 bytes each, exactly the same as the 5-line format of map data printed on the label of the bubble device (spaces between bytes on the device label are omitted, however). Compare the data with the 40 bytes printed on the label of the bubble device. Binary 0's in the pattern correspond to bad loops in the bubble device, and binary 1's correspond to good loops. Thus, the data should be predominantly 1's.

If the boot-loop data does not exactly agree with the pattern on the label, the boot-map must be re-written. Use BUBL's WRMAP command to load the

40 bytes of boot-loop data from the bubble device label into the QBC-11/02. After giving the WRMAP command, type the 40 bytes in the same 5-line format that the RDMAP type-out produced. The (H) shown by the WRMAP command on each line just reminds you this is hex data. After exactly 40 bytes have been typed in, BUBL will ask if the data are all correct. You should check the data you just typed for correctness, then type either Y or N. Typing Y causes the new boot-map to be written to the selected bubble device. Typing N aborts the entire operation.

Unless other problems still exist in the system, the QBC-11/02 should now be ready for proper operation.

NOTE

Although the information given above for writing a boot-loop is sufficient and complete, it is recommended that a new user call the engineering department at Bubbl-Tec before attempting to write a boot-map for the first time. Our number is (415) 829-8705.

APPENDIX D

Technical Specifications

QBC-11/02 Bubble-Memory Controller

Software Compatibility:	Command compatible with DEC RXV21/RX02 floppy-disk software.
Memory Capacity:	Up to 8 megabytes (when operated with sixteen 512-Kbyte BUBBL-BOARDs).
Command Functions:	Fill Buffer Empty Buffer Write Sector Read Sector Read Status Read Error Message Initialize Enable/Disable Interrupt Select Unit Number Run Self-Test
Sector Buffer Size:	256 bytes.
Device Address Select:	177150, 177160, 177170, and 177174.
Interrupt Vector Select:	264, 270 or 274.
Bootstrap:	Built-in, no extra hardware required.
Bootstrap Address Select:	171000, 173000, 175000, or disabled.
Bubble Device Formatting:	Automatic synchronization by controller at power-up.
Power Required (maximum):	+5 volts DC at 1.4 amps.
Size:	Standard LSI-11 dual-height module.
Operating Temperature:	0 to 70 degrees Celsius, maximum, ambient.
Weight:	10 ounces.

--These specifications are subject to change without notice--

APPENDIX E

Technical Specifications

QBI-11 BUBBL-BOARD

Memory Capacity:	QBI-11/512: 512 Kilobytes. QBI-11/256: 256 Kilobytes.
Bubble Device Access Time:	Less than 41 msec average, 82 msec maximum.
Transfer Rate:	QBI-11/512: In excess of 270,000 bits/second. QBI-11/256: In excess of 235,000 bits/second.
Sector Transfer Time:	QBI-11/512: Less than 9 milliseconds. QBI-11/256: Less than 18 milliseconds.
Error Protection:	14-bit Fire Code. Corrects error bursts up to 5 bits long. Automatic re-try for soft errors.
Bubble Device Formatting:	Automatic synchronization by controller at power-up.
Power Required (maximum):	Powered down: 5 volts at 0.06 amps. 12 volts at 0.01 amps. Standby: 5 volts at 0.6 amps. 12 volts at 0.25 amps. Active: 5 volts at 0.6 amps. 12 volts at 1.2 amps.
Size:	Standard dual-height (double width) module.
Operating Temperature:	0 to 70 degrees Celsius, maximum, ambient.
Storage Temperature:	-40 to 100^{o} C, without loss of stored data.
Magnetic Environment:	Less than 20 Oersted, at bubble device, max.
Weight:	21 ounces.

--These specifications are subject to change without notice--

* - Order replacements directly from Bubbl-Tec.

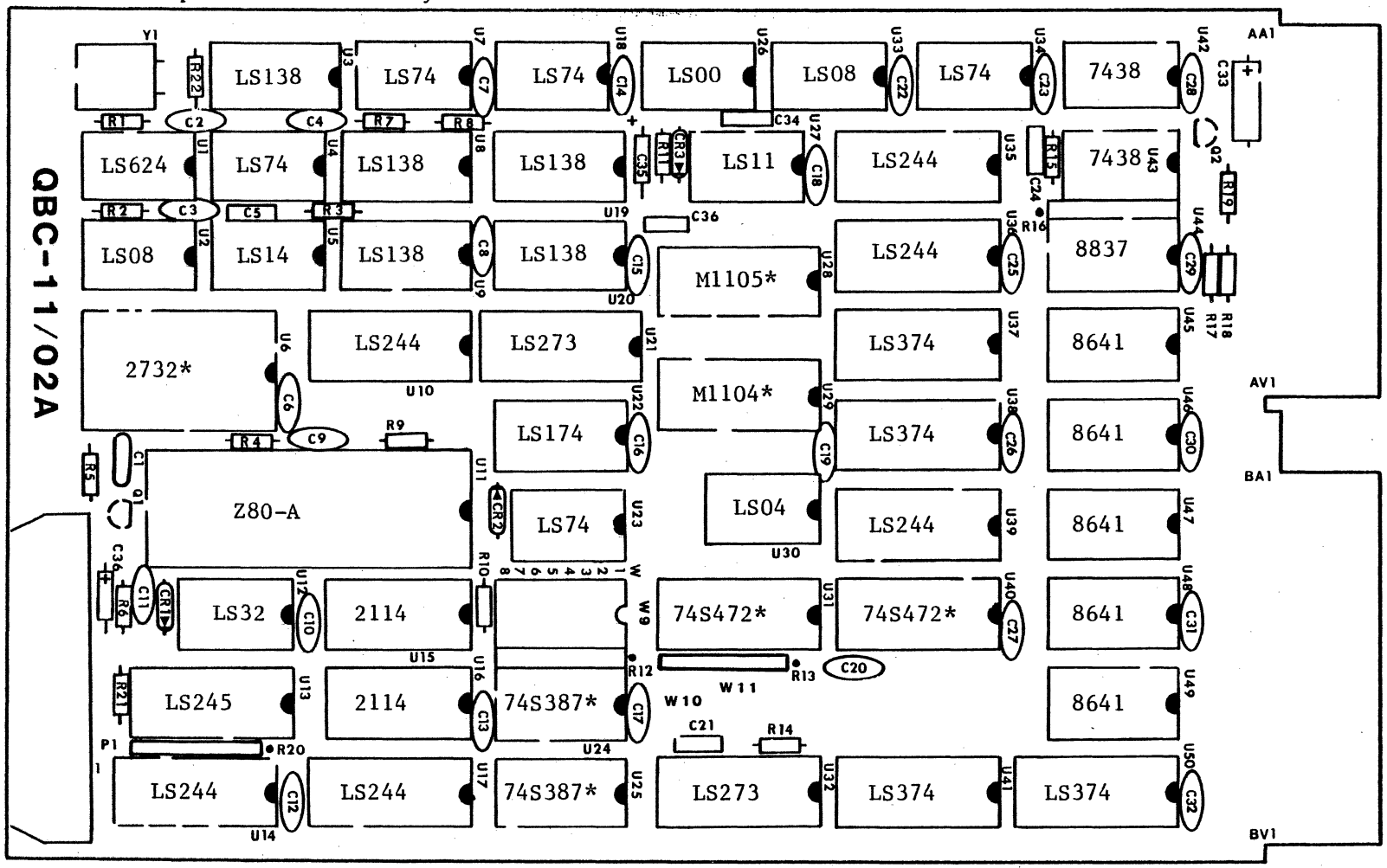


FIGURE F-1

Printed-Circuit Board Layout for QBC-11/02 BUBBL-CONTROL

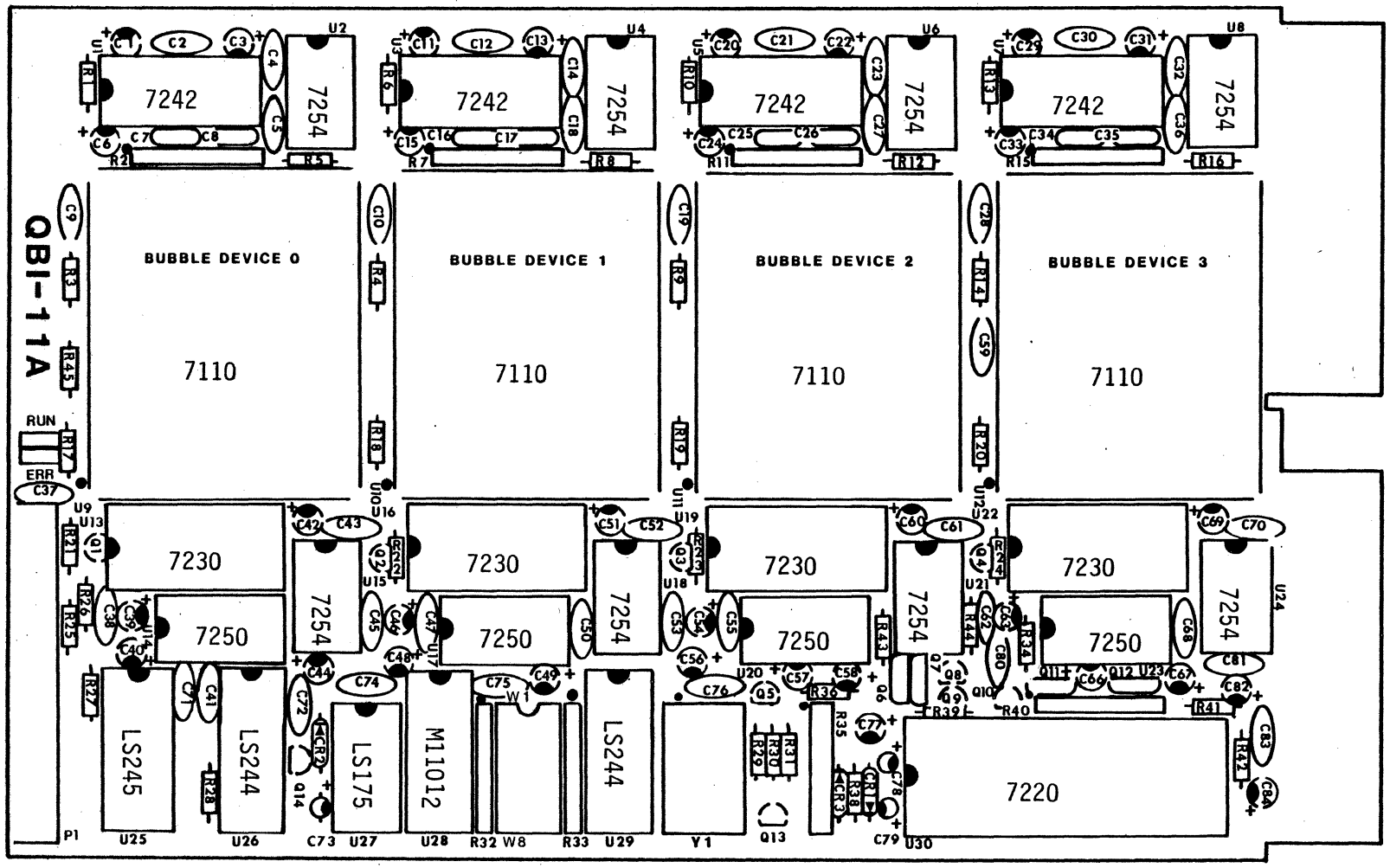


FIGURE G-1

Printed-Circuit Board Layour for QBI-11 BUBBL-BOARD