As you are now the owner of this document which should have come to you for free, please consider making a donation of £1 or more for the upkeep of the (Radar) website which holds this document. I give my time for free, but it costs me money to bring this document to you. You can donate here https://blunham.com/Misc/Texas

Many thanks.

Please do not upload this copyright pdf document to any other website. Breach of copyright may result in a criminal conviction.

This Acrobat document was generated by me, Colin Hinson, from a document held by me. I requested permission to publish this from Texas Instruments (twice) but received no reply. It is presented here (for free) and this pdf version of the document is my copyright in much the same way as a photograph would be. If you believe the document to be under other copyright, please contact me.

The document should have been downloaded from my website https://blunham.com/, or any mirror site named on that site. If you downloaded it from elsewhere, please let me know (particularly if you were charged for it). You can contact me via my Genuki email page: https://www.genuki.org.uk/big/eng/YKS/various?recipient=colin

You may not copy the file for onward transmission of the data nor attempt to make monetary gain by the use of these files. If you want someone else to have a copy of the file, point them at the website. (https://blunham.com/Misc/Texas). **Please do not point them at the file itself as it may move or the site may be updated.**

It should be noted that most of the pages are identifiable as having been processed by me.

I put a lot of time into producing these files which is why you are met with this page when you open the file.

If you find missing pages, pages in the wrong order, anything else wrong with the file or simply want to make a comment, please drop me a line (see above).

It is my hope that you find the file of use to you.

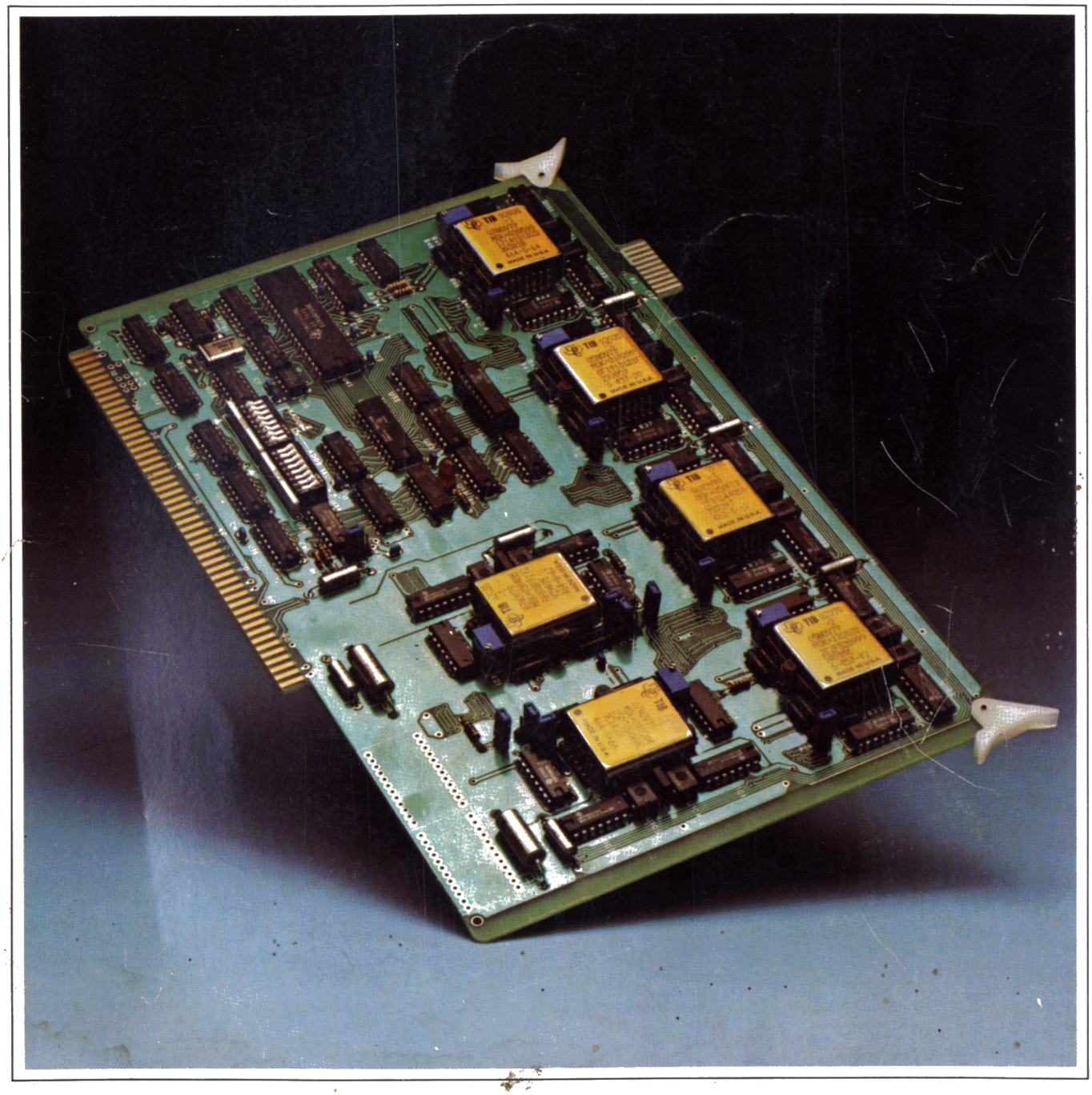
Colin Hinson
In the village of Blunham, Bedfordshire.

CL540A

REVISED

User's Manual TM990/210 Series Bubble Memory Systems

from Texas Instruments



IMPORTANT NOTICES

Information contained in this publication is believed to be accurate and reliable. However, responsibility is assumed neither for its use nor for any infringement of patents or rights of others that may result from its use. No license is granted by implication or otherwise under any patent or patent right of Texas Instruments or others.

Texas Instruments reserves the right to make changes at any time in order to improve design and to supply the best product possible.

Information contained herein supersedes previously published data on these devices.

Revisions : December 1979

: April 1980

: June 1980

C O N T E N T S

LIST OF ILLUSTRATIONS

LIST OF TABLES

SYMBOLS

The following symbols are used within the manual

uS	Microseconds
mS	Milliseconds
Kb/s	Kilobits per second
byte	8 bits
CR	Carriage return
>	Hexadecimal number (eg >400 = hex 400)
/	Refers to the inverted signal (e.g. PF/ means the inverted power fail signal)

Section 1

Introduction

1. INTRODUCTION

1.1 GENERAL

The TM 990/210 Bubble Memory Module provides non-volatile bubble memory storage for a TM 990 system using the well proven TIB0203S 92 Kbit, 3 micron magnetic bubble memory.

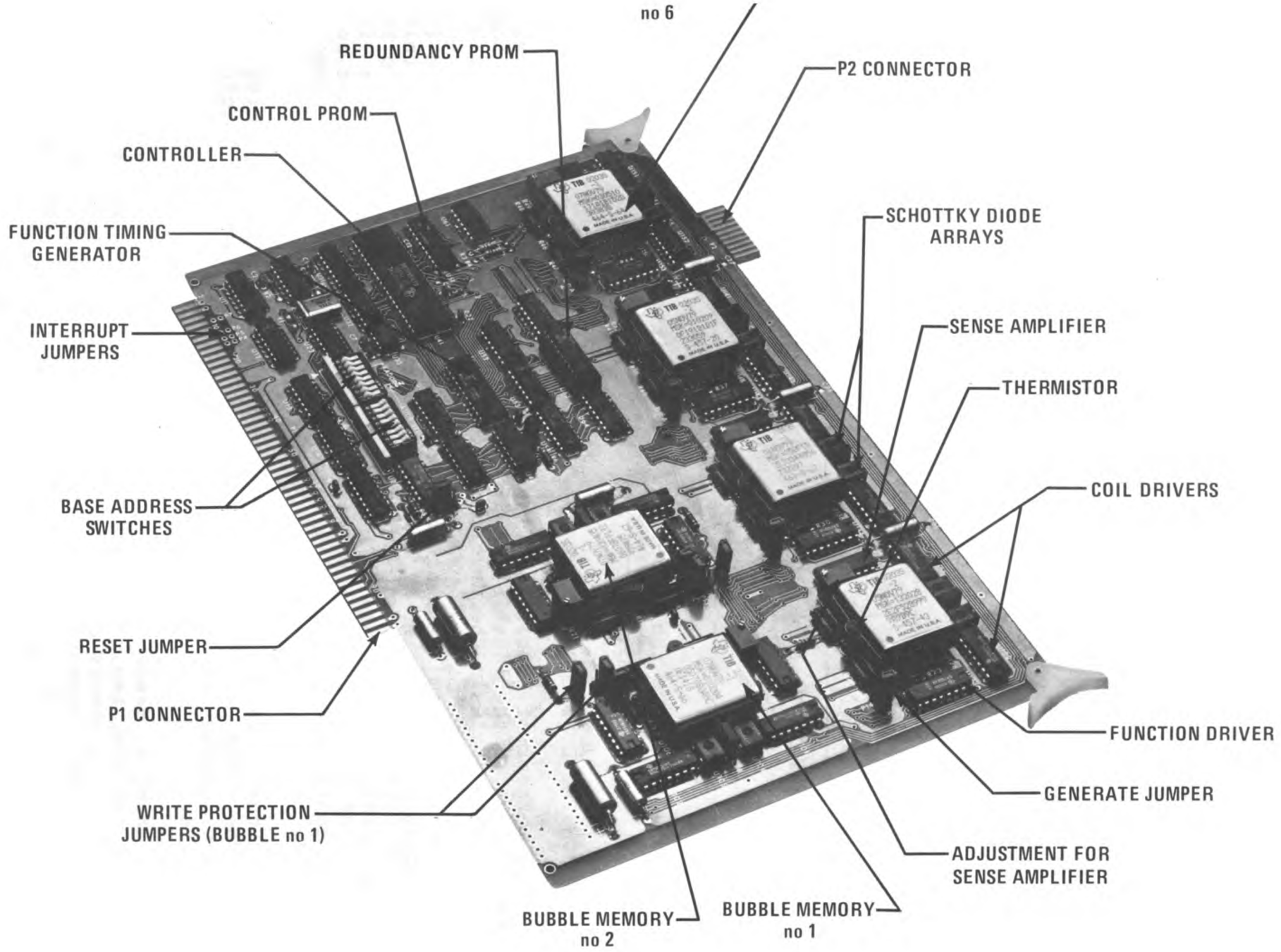
The board is 7.5 X 11 inches (19.05 x 27.94 cm) and it is both physically and electrically compatible with the TM 990 system. However this board may be easily interfaced with many other popular processor systems. Control of the bubble is achieved using a number of custom integrated circuits. Figure 1-1 shows the major components in the system.

Features of the TM 990/210 bubble memory board :

- Operates with the TM 990/100M or TM 990/101M CPU module
- 23 KBytes, 46 KBytes or 69 KBytes of non-volatile memory
- Access time is 860 uS min to 7.3 mS max
- Write-protect facility on bubbles #1 and #2
- Power fail protection
- Memory mapped CPU interface
- Single page or Multi-page transfers
- Continuous data transfer rate of 44 KBits / sec
- Data buffering of 18 bytes (1 bubble page)
- Redundancy handled by single programmed PROM
- Standard bus voltage requirements : +5, +12v, -12v

1.2 MANUAL ORGANIZATION

Section 2 of this manual describes the installation and operation of the TM 990/210 board. The power supply requirements are discussed in detail. An optional test and demonstration program (TIBUB) may be purchased for this board and its operation and use is described in section 3. The software interface is covered in section 4, and section 5 describes the operation of the hardware. The remainder of this section deals mainly with the bubble memory itself and its performance in the system. A glossary is included to cover some of the unique aspects of bubble memory technology.



TM 990/210-3 BUBBLE MEMORY MODULE
FIGURE 1-1

1.3 GENERAL SPECIFICATIONS

Nominal bubble memory field frequency : 100 KHz

Maximum continuous data rate : 44 Kb/s

Loop cycle time (for power fail) : 13 ms

Page size : 18 bytes

No of pages per bubble : 641

No of bubbles per board : 2,4, or 6 (see Section 1.4)

Bubble architecture : Major/Minor loop

Operating temperature : 0 deg C to +50 deg C

Storage temperature with data storage : -40 deg C to +85 deg C

Maximum ambient magnetic field : 20 Oersteds

CPU Interrupt level : jumper selectable (2,8,10 or 14)

Power Fail Warning : required 13 mS in advance

Power requirements : see table 1-1 below

TABLE 1-1 TM 990/210 POWER REQUIREMENTS

Storage Capacity in Bytes	23K (-1)	46K(-2)	69K(-3)
Vdd +12v +/- 3%	0.035 A	0.035 A	0.035 A
Vcc +5v +/- 3%	0.870 A	1.000 A	1.100 A
Vaa -12 +/- 3% (*Idle)	0.095 A	0.190 A	0.290 A
Vaa -12 +/- 3% (*Active)	0.245 A	0.330 A	0.420 A

* The system becomes active when the controller is rotating bubbles within a memory. The +5 and +12 volt currents are not significantly different between idle and active.

1.4 BOARD CHARACTERISTICS

The TM 990/210 bubble memory module is available in three versions. Two, four or six modular memory units (MMUs) may be supplied on one TM 990/210 module. Each MMU consists of a bubble memory, two coil drivers, two schottky diode arrays, a function driver and a sense amplifier.

TABLE 1-2 TM 990/210 MEMORY OPTIONS AVAILABLE

Model Number	Number of MMUs	Total storage in bytes
TM 990/210-1	2	23K
TM 990/210-2	4	46K
TM 990/210-3	6	69K

Figure 1-1 shows the TM 990/210 module fitted with 6 MMUs.

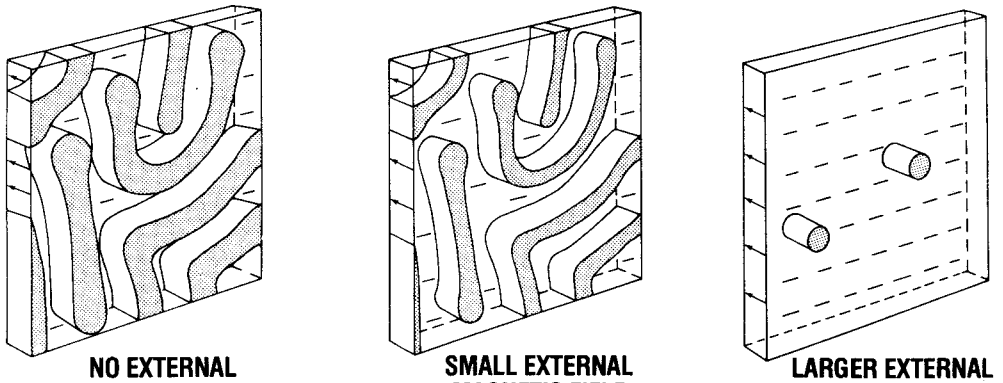
1.5 HOW THE BUBBLE MEMORY WORKS

The concept of bubble memories was originally conceived by Bell Laboratories in the late '60s. Since that time Texas Instruments has played a major role in the development of bubble memory technology.

Magnetic Bubbles are small cylindrical magnetic domains formed in single-crystal thick films of synthetic ferrites or garnets or in thin amorphous magnetic metal films when an external magnetic bias field is applied normal to the plane of the film. The domains are mobile in the plane of the film in the presence of a magnetic field gradient and can be controlled by special structures to perform logic or memory functions.

Each bubble memory on the TM 990/210 board stores 92,304 bits of data. The data is stored in the form of magnetic bubbles; the presence or absence of a bubble at a particular position defines the logic state.

The bubble memory housing contains the bubble memory chip, two permanent magnets, two mutually perpendicular coils and a magnetic shield which protects against external magnetic fields. The field of the two permanent magnets allow for the stable existence of magnetic bubble domains and therefore the non-volatility of the system. The orthogonal coils provide a rotating magnetic field used to propagate the bubbles on the chip in a shift register fashion.



EXTERNAL MAGNETIC FIELD SHRINKS RANDOM SERPENTINE DOMAINS OF MAGNETICALLY NEUTRAL CRYSTAL TO CYLINDRICAL FORM.

FIGURE 1-2

BUBBLE DOMAIN

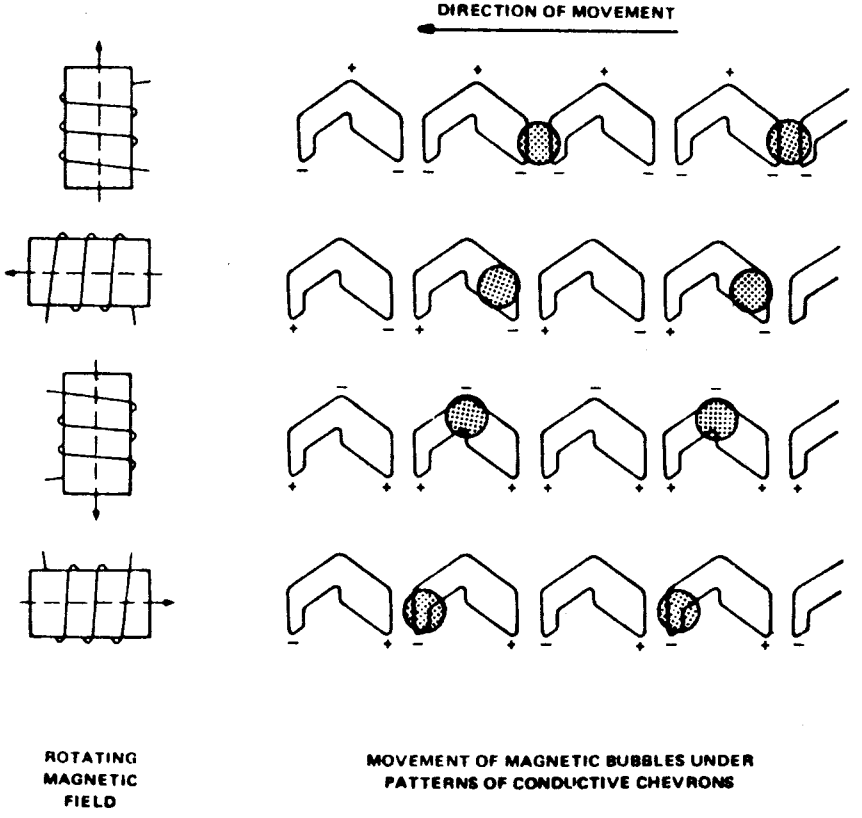


FIGURE 1-3

BUBBLE MOVEMENT UNDER CHEVRON PATTERNS

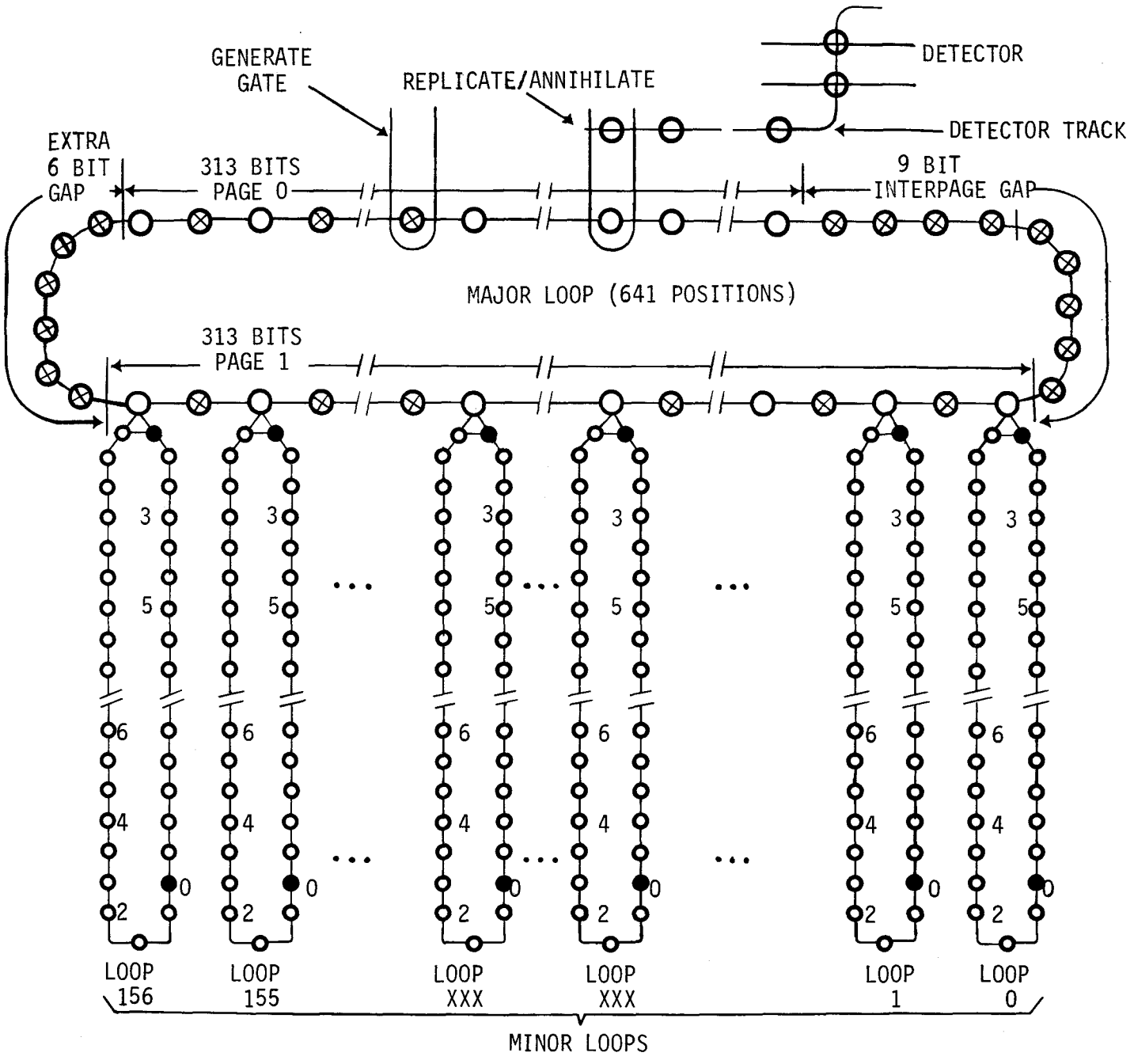
A necessary characteristic of the materials used in bubble devices is that they exhibit magnetic uniaxial anisotropy with the axis of easy magnetization perpendicular to the plane of the film. In the absence of the bias field, normally occurring, randomly distributed, serpentine domains exist (Figure 1-2). As the bias field is increased in strength, those domains that oppose the external field shrink in size until they form the cylindrical domains or bubbles. These bubbles are stable over a range of magnetic biases and can be made to move in the plane of the film with extremely small expenditures of energy. The bubbles, being magnetic dipoles, interact strongly; thus, in practice, propagation structures must keep the bubbles at least four bubble diameters apart, center to center. The bubble diameter is a function of the material composition and applied bias field.

In order to achieve practical Magnetic Bubble Memories, methods had to be devised to control Generation, Propagation, Transfer, Detection, Replication, Storage, and Annihilation of the Magnetic Bubbles. The general technique that is used to propagate bubbles from one position to another is known as the Field-Access Method. The propagation structures are made of permalloy which has been patterned by photolithographic techniques. These permalloy patterns act as small electromagnets whose polarity is induced and controlled by an external rotating field.

Typical field-access propagation structures are the T-Bar, the Y-Bar, the Contiguous Disk, the Chevron, and the Asymmetric Chevron. The asymmetric chevron pattern is currently the preferred method of bubble propagation, since the larger end of the chevron provides a strong force to pull the bubble between the chevrons and propagation is favored in one direction. The chevrons are arranged in a series of adjacent loops known as the minor loops. Figure 1-3 shows the movement of bubbles under chevron patterns. To obtain the best compromise between system performance and complexity an additional loop of chevrons is used to access the bubbles in the minor loops. This loop, known as the major loop, is used to generate, detect and annihilate the bubbles (see Figure 1-4). The bubbles may be transferred between the minor and major loops by applying a local field near the bubble using a thin electrical conductor, thus causing the bubble to change direction.

Bubble domains are created by a current pulse flowing through a generate loop. The intense localized magnetic field causes a magnetic domain inversion in the bubble material and under carefully controlled conditions the bubble will remain even when the current is removed. A data string equal in length to the number of minor loops is generated and this is known as a 'page' of data. The page is shifted around the major loop to align with the minor loops and the transfer gate is energized. The page of data is transferred to the minor loops and circulated. New data may be generated, shifted and transferred into each of the 641 different minor loop positions.

Data is retrieved from the minor loops one page at a time. When the desired page rotates to the top of the minor loops, it is



NOTE:

○ → VALID BIT IN LOOPS (PRESENCE OR ABSENCE OF BUBBLE DOMAIN)

● → INACTIVE BIT IN LOOPS (BUBBLE DOMAIN READ FROM THIS BIT)

⊗ → EMPTY BIT POSITION IN MAJOR LOOP (WILL NOT BE WRITTEN OR READ)

FIGURE 1-4

TIB0203S BUBBLE MEMORY INTERNAL ARCHITECTURE

transferred to the major loop. There the bits move in serial form until the first bit arrives at the replicate/annihilate element. One of two control operations is now performed: replicate or annihilate. If the page of data is to be stored again after being read, a replicate current pulse is issued and the bubble is stretched and subsequently cut into two bubbles by the application of an opposite polarity current pulse in the same hairpin loop. The current pulse collapses the domain wall in the center of the loop. As the external bias magnetic field alone determines the size of the bubble, each of the new bubbles is the same size as the original bubble. One of these bubbles is diverted into the detector area where it is read and destroyed and the other continues to move around the major loop to be stored again. Thus it is possible to read a page of data, recirculate the information and put it back in the minor loops for non-volatile storage. Before new data can be entered into a particular page position in the minor loops, the old data must be removed by doing a destructive read operation. This is the same as an ordinary read operation except that the bubbles are annihilated by applying a current pulse to the replicate/annihilate gate that transfers the bubble out of the major loop and into the detector track where it propagates off the chip.

Detection of the bubble is acheived by the use of a magneto-resistive thin film. Upon reaching the detector element, the bubble is stretched to several hundred times its original size into a stripe domain to increase its stray flux by several orders of magnitude. Thus during the stretch operation the detector is operating as a magnetic-amplifier. The detector consists of two magnetoresistive permalloy arrays arranged in a bridge circuit. A signal of several millivolts is generated free from common mode noise. This signal is then amplified by the sense amplifier to a usable level.

1.6 BUBBLE MEMORY TIMING ANALYSIS

There are two basic methods of operating the bubble memory controller : single page mode or multi-page mode.

In single page mode, one page is transferred from the minor loops into the major loop, rotated past the replicate and generate elements, and then returned to the minor loops. This operation always rotates the bubbles through two complete revolutions of the loops. To address and return the correct page to the minor loops takes exactly one revolution, then an additional revolution is required for positioning the page in the minor loops. Since it takes 10uS for a bubble to move one position, the whole operation takes (641 x 2 x 10) uS, or 12.8 ms to read 18 bytes (one page).

During a multi-page transfer two pages reside in the major loop concurrently. The second page is transferred into the major loop before the first page is returned to the minor loops. This results in a considerable improvement in data throughput.

The interpage gap defines the number of positions or periods between the trailing bit of the first page and the leading bit of the second page. In a bubble memory using a major/minor loop architecture the interpage gap has a physical restraint in that the first page must be returned to the minor loops before the third page is transferred from the minor loops. Therefore, two pages will never be symmetrically arranged in the major loop.

The bubble memory has a page size of 157 bits (including redundancy), but when a page is resident in the major loop there is an empty bit position between each data bit position (see figure 1-4). Therefore the overall length of the data stream is 313 bit positions.

The major loop has 641 available bit positions and two pages will occupy 626 or [(157+156)x 2] physical bit positions . Therefore, the minimum physical interpage gap equals [(641-626)/ 2]=7.5. However it is desirable for consecutive multiple pages to lie in the same even bit positions so that bubbles resident in the major loop always reside in 'every other bit' positions. Therefore, the actual interpage gap, that is defined within the controller, is equal to 9 bit positions in this system (see Figure 1-4).

If a multi-page transfer is started from page 0 the timing is as follows:

Time in clock cycles	Comments
0	Page 0 is transferred to major loop.
1	Page 0 has moved 1 bit position in the major loop.
313	Last bit of page 0 clears the top of the minor loops
322	After the interpage gap of 9 positions, page 1 is transferred into the major loop. Although it is sequentially page 1, it is physically page 322 because the minor loops are displaced by 322 positions from their reference position.
635	Last bit of page 1 clears the minor loops.
641 (=0)	Page 0 is returned to the minor loops.
644 (=3)	After the interpage gap of 9 positions, page 2 is transferred to the major loop. Sequentially this page is number 2 but physically it is page 3.

In conclusion a single page transfer requires 1282 field clock cycles and a multi-page transfer requires 322 field clock cycles per page plus the clock cycles required to fetch the first and return the last page.

1.7 MAXIMUM DATA RATES

Although the nominal field frequency on the 92K bubble memory is 100 KHz, the peak data rate at the sense amplifier is only 50 KHz. When a page of data is transferred from the minor loops it occupies alternate positions within the major loop. Because of physical limitations on the bubble design it is not possible to arrange minor loops adjacent to every major bit position. Redundancy and the interpage gap in multi-page mode further reduce the average data rate.

The simplest method for deriving the data rate is shown below :

$$\text{Average Data Rate} = \text{Field Freq.} \times \frac{\text{no. of valid data bits per page}}{\text{total no of field cycles per page}}$$

(see section 1.6)

For single page mode	For multi page mode
$= 100\ \text{KHz} \times \frac{144\ \text{bits}}{641 \times 2}$	$= 100\ \text{KHz} \times \frac{144\ \text{bits (18 bytes)}}{322}$
$= 11.2$ K Bits/second	$= 44.72$ K Bits/second

Therefore, the maximum continuous data rate may only be achieved in multi-page mode.

1.8 GLOSSARY

ACTIVE : Within this manual the controller is considered to be active while it is rotating the bubbles within any bubble memory.

BUBBLE : Magnetic bubbles are mobile cylindrical domains that can be generated in a thin magnetic crystalline layer and moved within the layer by interaction with propagate patterns processed directly onto the layer and magnetized by external fields.

The TIB0203S magnetic bubble memory consists of a non-magnetic Gadolinium Gallium Garnet substrate upon which the magnetic epitaxial layer is grown and the propagate patterns are implemented in Permalloy (see section 1.5).

CONTAMINATION : If data is written into a defective loop there is a general risk of contamination across the bubble. The generated bubble may wander across the loops and affect other normally good loops, and additional stray bubbles may be replicated. If a page appears to have

a hard error in a particular bit position the bubble may be slightly contaminated. Major contamination presents itself as a large cluster of random errors. A contaminated bubble memory cannot be flushed out by just writing zeroes to the whole bubble; the newly formed bubble domains should be collapsed by using a calibrated magnet. If the external field is too low, the bubbles will not collapse, and if the field is too high the internal magnetic field may be impaired.

The TM 990/210 module has on-board redundancy circuitry to handle the bad loops. There is no risk of contamination unless the redundancy circuitry fails.

FIFO : Data is transferred between the CPU RAM and the bubble via a First-In, First-Out (FIFO) memory in the controller. The FIFO allows buffering between the regular transfer of data to and from the bubble and possibly irregular data transfers to and from the CPU. (The FIFO is only effective in single page mode.)

HARD ERRORS : A hard error (sometimes known as a data error) may be introduced during a write or a read cycle. A hard error may only be removed by rewriting the whole page of data again. Hard errors are most likely to occur during function drive such as transfer, generate, etc. Hard errors occur infrequently and typically a system may produce one hard error for every 100 soft errors, although this figure will vary greatly between systems.

PAGE : Data is accessed by transferring a single bit from each of the minor loops into the major loop. These 157 bits of data make up a page and this is the smallest block of data that the controller can handle.

PAGE POSITION COUNTER : The page position counter (PPC) is a register in the controller that tracks the position of the bubbles. The register is incremented at the end of each field cycle by the controller. Before the controller issues any command the PPC is set to zero. After an I/O operation the bubbles are always returned to their original position (i.e. PPC is zero).

PAGE SELECT REGISTER : The Page Select Register (PSR) is used to address a particular page of data within the bubble memory. As the bubbles are rotated in the minor loops the PPC register is incremented and compared with the PSR. When PSR = PPC the page requested is available for transfer (i.e. the page of data is adjacent to the major loop)

REDUNDANCY : A redundancy of minor loops negates the requirement for perfect bubble chips. Some defective minor loops result from manufacturing processes related to the fine geometries of the permalloy patterns. On the TIB0203S as many as 13 of the total 157 minor loops may be defective and a map of their locations is printed on the chip package. This map consists of the hexadecimal address of each loop starting at Hex 00.

The TM 990/210 board uses a single PROM to store the redundancy

map for all 6 bubble memories and this PROM is programmed at the factory.

REFERENCE POSITION : When the data in the bubble memory is sitting in the reference position, page 0 is immediately available for transfer into the major loop. At the start of a data transfer the controller assumes that the bubble is in the reference position and the controller always returns the data to the same position. To rotate one bubble around a minor loop back to its original position requires 641 field cycles. Therefore every operation always takes a multiple of 641 field cycles.

The bubble memory controller allows the reference position to be changed for a particular bubble memory, this is described in section 4.17.

SOFT ERRORS : A Soft error (sometimes known as a read error) is the most common type of error associated with a bubble system. A soft error may normally be corrected by rereading the data from the bubble. Most errors are attributed to the bubble sense operation. If the sense amplifier circuitry is exposed to a particularly noisy enviroment, soft random errors may be produced.

1.9 APPLICABLE DOCUMENTS

9900 FAMILY SYSTEMS DESIGN

TM 990/100M MICROCOMPUTER USER'S GUIDE

TM 990/101M MICROCOMPUTER USER'S GUIDE

TIB0203S 3u MAGNETIC BUBBLE MEMORY AND ASSOCIATED CIRCUITS

TMS 9916 (TIB0901) BUBBLE MEMORY CONTROLLER

TIB0833 SENSE AMPLIFIER

Section 2

Installation and Operation

2. INSTALLATION AND OPERATION

2.1 REQUIRED EQUIPMENT

* TM 990/510 4 slot OEM chassis
  or TM 990/520 8 slot OEM chassis

* TM 990/100M or TM 990/101M CPU card

* Power supply system (+5V, +12V, -12V) with A.C. power failure warning and a 13mS minimum hold-up time on all supplies.

* Data Terminal such as a :
  Decwriter 11
  Hazeltine 1500 series
  Lear Siegler ADM-1, ADM-2, ADM-3
  Soroc IQ 120
  Teletype model 3320 5JE
  Texas Instruments models 733 ASR or 745 KSR

2.2 POWER SUPPLIES AND POWER FAIL WARNING

The choice of the power supply for a TM 990 system using a TM 990/210 bubble memory module requires careful consideration.

During a bubble memory data transfer, data in the minor loops are transfered into the major loop. In the case of power failure pending it is necessary to flag the 210 module immediately and the power supply must maintain all DC supplies for at least an additional 13 ms to allow the controller to return the data 'back home' into the minor loops. Failure to do so will result in the loss of that page of data and the reference position for that bubble. The 13 ms is the worst case time for the controller to complete this operation.

Early power fail warning is most easily provided by monitoring either the AC line signal or the unregulated DC supplies generated within the power supplies. Many supplies using this method are capable of maintaining their DC outputs for 50 ms at nominal input voltage. Monitoring the DC supplies that feed the TM990/210 system is not recommended. This approach relies on the detection of 'out-of-tolerance' supplies before the controller begins execution of its power down sequence. All the supplies must be in tolerance for the full length of the power down sequence because sensitive operations such as the page transfer-in to the minor loops still have to be completed.

Power failure is signalled to the TM990/210 module by driving PF/ (pin 15 on P2) low. This input is standard LS TTL with a pullup resistor. On more recent revisions of the TM 990/210 module a facility has been added that allows interrupt level 1 to be connected

(XAEN) should be set to ON (enable).

In systems with only 64 KByte addressing capability U23 (S4) should be set to OFF (disable).

The remaining switches U23(S3,S2,S1) and U24(S8 through S1)should be used to set up the address lines MSB to LSB, respectively. As an example Figure 2-2 shows the switch settings to decode address hex C000.

FIGURE 2-2 EXAMPLE OF BASE ADDRESS SELECTION

(setting up base address of hex C000)

Switch U23								Switch U24							
S8	S7	S6	S5	S4	S3	S2	S1	S8	S7	S6	S5	S4	S3	S2	S1
||	||	||	||	||	||	||	||	||	||	||	||	||	||	||	||
XA0	XA1	XA2	XA3	XAEN	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
x	x	x	x	(1)	1	1	0	0	0	0	0	0	0	0	0
				off	off	off	on	on,	on	on	on	on,	on	on	on

x - Either state

2.4 SETTING THE INTERRUPT JUMPER

The TM 990/210 board activates the interrupt line (low level) when the CPU is expected to transfer data to or from the controller. The interrupt line may signal the CPU at one of four different levels. The appropiate jumper positions are shown below :

TABLE 2-1 INTERRUPT JUMPER OPTIONS

Interrupt level 2	- Jumper location	P4
Interrupt level 8	- Jumper location	P7
Interrupt level 10	- Jumper location	P6
Interrupt level 14	- Jumper location	P5

2.5 SELECTING THE RESET LINE

The reset line may be connected to either an EXTRST/ signal supplied to pin 19 on P2 or to IORST/, the general reset on the TM 990 bus, (pin 88 on P1).

The reset will halt the controller immediately and restore its normal power up state. In most applications the reset would probably not be used. If the reset is connected to the bus using IORST/ then the controller will be automatically reset when the CPU is reset, which may be undesirable. In this case the EXTRST/ line should be used.

--- WARNING ---

If reset is applied to the TM 990/210 board while the controller is active, the address reference position for the active bubble will be lost. Bubble contamination may also occur.

The reset jumper options are shown below :

TABLE 2-2 RESET JUMPER OPTIONS

EXTRST/ (Pin 19 on P2)	Connect P3A to P3B
IORST/ (Pin 88 on P1)	Connect P3C to P3B

2.6 ENABLING WRITE PROTECTION

Bubbles 1 and 2 are fitted with a write protect feature (see figure 1-1). Jumper options allow the generate and annihilate signals to be disabled. It is intended to protect system programs resident in bubble memory from accidental erasure. The following pairs of jumpers should be removed from the board to give write protection.

TABLE 2-3 WRITE PROTECTION JUMPER OPTIONS

Bubble #1	:	P8, P9
Bubble #2	:	P11, P12

2.7 REPLACING A BUBBLE MEMORY

A bubble memory may be replaced if it fails. It is necessary to set up the correct environment on the board prior to the installation of the the new bubble memory. The following steps are required :

1. Set the generate jumper for the correct current level. The TIB0203S bubble memories fall into three bands of required generate current and the subscript following the bubble memory type number identifies the correct category. Near each bubble on the 210 board there is a 3 way jumper with options A,D and C that correspond to the different generate currents. The jumpers are numbered P10,P13,P14,P15, P16,P17 for bubbles #1 through #6, respectively.

The jumper should be set as follows :

TABLE 2-4 GENERATE CURRENT JUMPER OPTIONS

Bubble type	Jumper Position	Generate current
TIB0203S-1	A	290 mA
TIB0203S-2	D	360 mA
TIB0203S-3	C	235 mA

2. Set the sense amplifier input offset voltage to zero. While the board is under power, but not active, the voltage between pins 12 and 13 on the sense amplifier should be adjusted to less than +/- 1.0mV. (This is not a critical adjustment.)

3. Program a new redundancy PROM (U63). The redundancy PROM holds the maps for six bubbles; each bubble uses one data line only from the PROM. Bubble cell #1 uses the least significant data bit in the PROM and the other bubble cells use adjacent bits.

TABLE 2-5 REDUNDANCY PROM FORMAT

Prom Data line	DO8	DO7	DO6	DO5	DO4	DO3	DO2	DO1
Equivalent 9900 Data Bit	MSB 0	1	2	3	4	5	6	LSB 7
Bubble Cell No	-	-	6	5	4	3	2	1

For every defective loop printed on the front of the package, a ONE is programmed into the corresponding address in the PROM. The address of each bad loop is printed as a two digit hexadecimal number on the front of the package. For example if the first minor loop was defective then '00' would appear on the front of the package.

Many methods may be used to blow a new redundancy PROM. Each bad loop for each bubble may be manually loaded into the new PROM using a standalone PROM programmer. The prom may be either a 74S471 or a 74S472 but the prom must be blown with a '471 programmer. The TM 990/210 addresses the prom as a '471. As the PROM is of the fusible link variety it is necessary to blow the maps for all six bubbles into the new redundancy PROM.

Alternatively if a 990/4 or 990/10 fitted with a fusible link PROM personality card is available then the following method is suggested.

1. Using the PROM programming utility read the old PROM into the system memory. (Using PROM or :PROM/SYS the OFFSET should be set to ALL, using :TXPROM/SYS the PROM bit step should be set to 8. Set the personality card switch to the C position)

2. Using the memory change command reset the data bits corresponding to the bad loops on the old bubble memory. At the addresses printed on the top of the old bubble the corresponding bit at the address should be a ONE, change this to a ZERO.

3. Repeat the above step with the new bubble memory but set the correponding bits to ONES.

4. Program a new PROM with the contents of the data buffer.

--- WARNING ---

The TM 990/210 board should not be operated with an incorrect or a missing redundancy PROM because contamination of the bubble may occur.

The TIB0203S 3 micron bubble memory cannot be replaced by its predecessor, the TIB0203. Although the architecture of both devices is basically the same, the 'S' version uses a smaller bar. As a result the control signals required to drive each bubble memory are different and the devices should not be interchanged.

If a bubble memory cell displays a dubious performance the solution is not necessarily to replace the bubble memory. Many faults in a bubble memory system can be identified as semiconductor failures within the support I.C.s. Many TIB0203S bubble memories fitted to the TM 990/210 bubble memory module have zero bad loops and therefore

these devices may be used to temporarily swap into a bad unit position. If the replaced bubble also fails in a particular position then the original bubble memory was probably not at fault.

2.8 TESTING THE INTERFACE TO THE BUS

If a non-standard hardware interface is being used with the the TM 990/210 module and there is no operational software available to test the system then the following tests may be performed to check out the interface prior to software debug.

These tests require the use of a memory inspect/change command alone. The following example uses the 'M' command available in TIBUG or TIBUB.

1. Set the Controller Base Address to an unused memory address. In this example the controller base address is set to hex C000; all switches in block U23 except S1 should be off, S1 on U23 and all switches in U24 should be on.

2. Reset the controller ; set the jumper P3 from B to C and reset the CPU board (see section 2.5).

3. Using TIBUG or TIBUB read Status Register #2 at hex C01F, see Figure 2-3 step a). The value returned should be hex FExx (the xx's may signify any value). This result means that the power fail latch is set and there is no interrupt.

4. Load location hex C01F with 0. Bubble module #0 will now be selected and the power fail latch is reset. This may be confirmed by reading Status Register #2 again in step c). The result should be hex FC00.

5. Status register #1 should now be tested as in step e). The result should be hex 42 as in step f). Two status bits are normally set after resetting the controller ; Page-Counter-Equals-Zero and Load/Increment. No more than one other status bit should be set.

If the status reads hex FF then the CPU is not communicating with the controller at all. Check the controller base address and the supplies to the system.

6. Load the least significant byte of the Page Counter register. This is achieved by modifying Register 6 (hex C00C) as in step g). In the example the Page Counter LSB is changed to 89xx. This particular register on the controller may be written to and read from. In step i) the Page Counter is read again and the value loaded in step g) can be seen again.

7. As the page counter has been loaded, the page-counter-equals -zero status bit is reset. In step j) Status Register #1 is read again and

normally the status is hex 12 or hex 02.

8. This last test executes one single page read on page 0. Steps l,m, and n) load the Page Select Register with a zero and execute a single page read. When the read command is loaded into the command register (hex C004) the LED on the board will flash once only. This test indicates that the controller will perform a simple accessing operation. The controller does not need to be initialized to perform this test.

It is difficult to perform further tests without operational software. If the system has responded correctly to all the above tests but completely fails to respond within the system then TIBUB is required to completely check out the system, see Section 3.

FIGURE 2-3 EXAMPLE OF INTERFACE TEST

```
                                        read Status Register #1
a)   ?M C01F                  <-----  request memory inspect
b)   C01F = FExx 0            <-----  enter new value + CR
c)   ?M C01F                  <-----  request memory inspect
d)   C01F = FC00              <-----  enter CR
e)   ?M C00A                  <-----  read Status Register #2
f)   C00A = 42xx (or 52xx)    <-----  enter space
g)   C00C = 00xx 8900         <-----  enter new value + space
h)   C00E = FCxx -            <-----  enter minus sign
i)   C00C = 89xx -            <-----  enter minus sign
j)   C00A = 12xx (or 02xx)    <-----  enter CR
k)   ?M C000
l)   C000 = FFxx 0            <-----  load PSR LSB with 0
m)   C002 = FFxx 0            <-----  load PSR MSB with 0
n)   C004 = FFxx 0200         <-----  enter new value + CR, OBSERVE LED
     ?                                 test completed
```

Section 3

TIBUB Monitor

3. TIBUB MONITOR (TM 990/431)

3.1 GENERAL

TIBUB is an interactive monitor intended to demonstrate and test the operation of the bubble memory module within a TM 990 system. TIBUB is purchased separately from the TM 990/210 Bubble Memory Module. TIBUB is in essence the same as the standard CPU monitor, TIBUG, and most of the commands are identical. However, additional commands, XOPs, and features have been added specifically to support the TM 990/210 bubble memory module .

TIBUB is shipped in two TMS 2716s. It will operate in sockets U43 (MSB) and U45 (LSB) on the TM 990/100 and TM 990/101 CPU cards while TIBUG is resident and this is described in section 3.2. Alternatively TIBUB may be operated without the TIBUG monitor and this is described in section 3.3. Interactive communication is achieved through a terminal with either an RS232 interface or a 20mA current loop interface.

3.2 TIBUB INITIALIZATION USING TIBUG

The CPU card should be configured to operate TIBUG (2708s) in the normal position. The TIBUB PROMs (2716s) should also be accommodated at a suitable memory address. Suitable configurations are shown below for the TM 990/100M and the TM 990/101M CPU boards.

On a TM 990/100M board :
- Set J2 to the 2716 position
- Set J3 to the 2708 position
- Set J4 to the 2716 position
- Place TIBUG in sockets U42(MSB), U44
- Place TIBUB in sockets U43(MSB), U45

On a TM 990/101M board :
- Set BANK1 to 2708 position
- Set BANK2 to 2716 position
- Set E9-E10 (2716 position)
- Set E13-E14 (EPROM ON)
- Set E16-E17 (RAM HI)
- Place TIBUG in sockets U42(MSB), U44
- Place TIBUB in sockets U43(MSB), U45

TIBUB now resides in the memory addressing area from 1000 to hex 1FFE on the system. TIBUB also expects the use of 128 bytes of RAM starting at address hex FE00.

* If the CPU is a TM 990/101M, ensure that switch #1 is in the off position. (A bootload will not be attempted.)

* Power up system with the CPU card and the bubble card installed

* Press the RESET button on the CPU card.

* Enter character 'A' on the terminal. TIBUG times the start bit to determine the baud rate of the terminal.

* TIBUG will then print "TIBUG REV A"

* Using the 'R' command, set up the PC to hex 1190 and the WP to hex FE00. Execute TIBUB using the 'E' command. The TIBUG command sequence is shown below.

```
?R                  <------- enter 'R'
WP=0400 FE00        <------- enter FE00 and a space
PC=6F32 1190        <------- enter 1190 and a CR
?E                  <------- enter 'E'
```

* Enter character 'A' on the terminal.

* TIBUB will then print "TIBUB rev 210/A FE00 1000"

* Commands may now be entered after the TIBUB prompt '??'

The two hex words printed after the banner message correspond to the the address of TIBUB's main workspace and the first location of the TIBUB PROMs.

3.3 TIBUB INITIALIZATION VIA A CPU RESET

In this mode of operation the TIBUG EPROMs are replaced with the TIBUB EPROMs. The CPU board should be configured to accept 2716s.

On a TM 990/100M board :
- Set J2 to the 2716 position
- Set J3 to the 2716 position
- Place TIBUB in sockets U42(MSB), U44

On a TM 990/101M board :
- Set BANK1 to 2716 position
- Set E9-E10 (2716 position)
- Set E13-E14 (EPROM ON)
- Set E16-E17 (RAM HI)
- Place TIBUB in sockets U42(MSB), U44

TIBUB now resides at memory locations 0 to hex 1000.

* If the CPU is a TM 990/101M, ensure that switch #1 is in the off position. (A bootload will not be attempted.)

* Power up system with the CPU card and the bubble card installed

* Press the RESET button on the CPU card.

* Enter character 'A' on the terminal. TIBUB times the start bit to determine the baud rate of the terminal.

* TIBUB will then print "TIBUB rev 210/A FE00 0000"

* Commands may now be entered after the TIBUB prompt '??'

TIBUB can also be inititiated directly using a LOAD. However in this mode the standard version of TIBUB will not operate within the standard memory maps available on either the TM 990/100M or the TM 990/101M CPU cards. TIBUB requires 128 bytes of RAM starting at hex DE00. TIBUB itself must reside between hex F000 and hex FFFE. If the RAM requirement is awkward in a particular system then the LOAD WP vector at location hex FFC in the PROMs may be modified accordingly. The RESET WP vector may also be changed from hex FE00 if required.

3.4 ABSOLUTE BUBBLE MEMORY ADDRESSING

In all TIBUB operations involving bubble memory data transfers the bubble may be assumed to be a large byte-addressable mass storage device. Page boundaries and bubble boundaries are transparent to the user. TIBUB assumes that the module is populated with six operational bubble memory cells. All six bubbles are initialized automatically as soon as the first write or read command is executed. If an attempt is made to access data from an unpopulated memory unit then TIBUB will generate a timing error, see Section 3.6. (The timing error is caused because the controller cannot access a redundancy map within the redundancy PROM.) If an attempt is made to access data outside the range of the six bubbles this returns a range error.

A page on the 92 K bubble is normally 18 bytes, however, TIBUB uses the last two bytes for error detection. The error detection is always operational and cannot be switched off. If TIBUB detects a read error the data is still transferred but an error is returned at the completion of the transfer.

The dump and load operations transfer the direct RAM image to and from the bubble memory module; 9900 tagged object code is not used.

Thus the maximum System storage with error detection is :

16 bytes x 641 pages x 6 bubbles = 61536 bytes

TIBUB considers the bubble address 0 to be the first byte of page 0 on bubble 1. The maximum storage for a given number of bubbles is shown below, the address of the last byte is one less than the storage size.

TABLE 3-1 TM990/210 BYTE STORAGE CAPABILITY

No of populated Memory Units	Total storage @18 bytes/page (hex)	(dec)	TIBUB storage @16 bytes/page (hex)	(dec)
1	2D12	11538	2810	10256
2	5A24	23076	5020	20512
3	8736	34614	7830	30768
4	B448	46152	A040	41024
5	E15A	57690	C850	51280
6	10E6C	69228	F060	61536

All initialization operations, etc. are done transparently. TIBUB sets an internal flag after the first operation to indicate that initialization is complete. TIBUB operates the controller in MULTI PAGE MODE ONLY but the controller is always reset into single page mode at the end of an operation.

3.5 TIBUB COMMANDS

3.5.1 GENERAL

The following table shows the full TIBUB command list.

TABLE 3-2 TIBUB COMMAND LIST

INPUT	RESULTS	SECTION
B	Execute and breakpoint on specified address	note 1
C	Inspect/change CRU locations	note 1
D	Dump RAM to Bubble	3.5.2
E	Execute	note 1
F	Find byte/word in CPU ram	note 1
H	Help	-
I	Initialize Bubble Memory System	3.5.3
L	Load CPU ram from bubble / memory	3.5.4
M	Inspect / change CPU memory	note 1
P	Load and execute Program specified in bootmap	3.7.4
R	Inspect / change CPU registers	note 1
S	Single step / trace program	3.5.5
T	Reserved for future use	-
U	Execute Internal 9900 Disassembler	3.5.6
V	Verify operation of Bubble Memory System	3.5.7
W	Inspect / change Workspace registers	note 1
X	Copy data block in CPU ram	3.5.8
Z	Inspect / change Bubble memory	3.5.9

Note 1 : These commands are also available in TIBUG and a detailed description may be found in the CPU manuals ´TM 990/100M MICROCOMPUTER USER´S GUIDE´ and ´TM 990/101M MICROCOMPUTER USER´S GUIDE´.

Each TIBUB command is executed by entering a single letter and it should be followed with any other parameters required. All numerical parameters entered are assumed to be hexadecimal. Parameters should be spaced apart using either a space or a comma.

The major differences between TIBUB and TIBUG are :

TIBUB does not support the following commands :

Dump to cassette - D
hexadecimal arithmetic - H
Terminal character rate - T

TIBUB has an upgraded single step/trace command (S).

TIBUB has an internal 9900 disassembler (U).

TIBUB is dynamically self-relocating - it will operate anywhere in memory.

TIBUB can be directly initialized using a CPU LOAD.

3.5.2 D, DUMP RAM TO BUBBLE

Syntax :

D < start address > < stop address > < storage address >

Data from RAM is stored in the bubble memory module starting at the storage address specified. There are no defaults, if less than three parameters are specified then TIBUB will generate an error message.

3.5.3 I, INITIALIZE BUBBLE MEMORY SYSTEM

Syntax :

I < controller base address >

The initialize command assumes the new controller base address, resets the controller and initializes six bubble memories, irrespective of the number fitted. If the base address is not entered then a default address of hex C000 is loaded.

3.5.4 L, LOAD RAM FROM BUBBLE / CASSETTE

Syntax :

```
L  < start address >  < stop address >  < storage address >
     - or-
L  < cassette load bias >
```

The load command may be used to load from cassette or the bubble module. A cassette load is assumed if only one parameter is supplied. If two parameters are entered, an error is returned. A load from bubble memory is initiated if the three parameters are entered.

3.5.5 S, SINGLE STEP / TRACE PROGRAM

Syntax :

```
S  < number of steps >
```

This command allows multiple stepping through the user program. After the number of specified instructions have been executed, the contents of the workspace, program counter, and status registers are printed and the next instruction to be executed is disassembled.

If the number of steps is not specified then TIBUB will trace the user program until the escape key is entered on the keyboard. The trace output is shown below.

```
?? S
        WP=FE00  PC=018C  ST=C400  0680  BL   R0
        WP=FE00  PC=FE00  ST=C400  045B  B    *R11
        WP=FE00  PC=018E  ST=C400  C24B  MOV  R11,R9
        WP=FE00  PC=0190  ST=C400  0229  AI   R9,>FE72
        WP=FE00  PC=0194  ST=3400  0229  CLR  12
??
```

Note that the single step command uses the load vectors at hex FFFC and hex FFFE. For successful operation RAM must be available at these locations or TIBUB must be resident at starting address hex F000.

3.5.6 U, EXECUTE 9900 DISASSEMBLER

Syntax :

U < start address > < finish address > CR

This command allows the 9900 code resident in the system RAM to be disassembled. If only the start address is specified then each line of code is disassembled when a CR is entered on the keyboard. If the start address and finish address are specified then the disassembly continues until the end address is reached or a character is entered on the keyboard.

An example of the output of the disassembler is shown below.

```
?? U F020 F030
F020 02E0 LWPI >EF60
F024 020C LI   R12,>0080
F028 C80C MOV  R12,@>EFB8
F02C 0208 LI   R8,>007C
F030 CE20 MOV  @F01C,*R8+
??
```

3.5.7 V, VERIFY OPERATION OF BUBBLE MEMORY SYSTEM

Syntax :

V < storage address > < data pattern > < / > CR

--- WARNING ---

This test destroys the contents of the bubble module.

This command allows the bubble memory system to be tested by writing and reading from the bubble memory. During the read operation the data is verified against the data pattern written in the bubble. After each page has been written or verified the data pattern is rotated through five bits. This is necessary to perform the worst case loading tests on the bubble memory.

The test operates in two modes. It may be used to test a specific two page block or alternatively it will test the complete memory system. The mode selected is determined by the number of numeric parameters entered with the 'V' command.

If a valid storage address and a data pattern are entered then the system executes a 2 page write cycle followed by a two page read cycle.

For example :

```
?? V 2830 AAAA              <---- enter carriage return
??
```

If the prompt returns immediately then no errors were detected. The operation of the test can be seen by confirmed by reading the data in the memory using the 'Z' command.

```
?? Z 2820 284F              <---- enter carriage return
2820 (1 001) : FFFF FFFF FFFF FFFF   FFFF FFFF FFFF FFFF
2830 (1 002) : AAAA AAAA AAAA AAAA   AAAA AAAA AAAA AAAA
2840 (1 003) : 5555 5555 5555 5555   5555 5555 5555 5555
??
```

Note that the data pattern has been shifted between the pages. If the test fails then the page containing the data error will be printed.

For example :

```
?? V 2830 AAAA              <---- enter carriage return
2830 (1 002) = AAAA AAAA AAAA AAAA A2AA AAAA AAAA AAAA AAAA
??     |  |
       |  |---- page number [0..280], hex
       |
       |------- bubble number [0..5]
```

The test may be repeated continually by entering a slash (/) after the command. The test may be halted by entering the ESC character.

For example :

```
?? V 2830 AAAA/             <---- enter ESC to terminate test
??
```

When the address and data word are specified and a slash is used to terminate the command entry (as in the above example) the error dump is inhibited to allow the waveforms on a bubble memory module to be inspected.

The verify command can be used to test the whole system by entering the data pattern only after the 'V' command and omitting the

storage address. In this mode the data pattern is written to all the bubble memories and then the data pattern is automatically verified by reading the entire memory. Any errors detected are printed using the same format as above. The test determines the size of the memory system automatically by testing for controller interrupts while the first page is written to bubble memories #3 and #5.

If the 'V' command is entered without any parameters then five selected data patterns are used in rotation to test the whole bubble memory system. The first four data patterns used are :

0001,FFFF,0000,FFFE

In the final test the address of the particular location is loaded as the data pattern. For example location 2F46 is loaded with data word 2F46. This complete test takes approximately 30 seconds per bubble memory fitted.

The test may be prematurely terminated by entering the ESC character and the test may be repeated continually by entering a slash after the data pattern. All data errors will be printed using the standard format.

Example of a system test using one data pattern :

```
?? V 5555                <---- enter carriage return
??                       <---- prompt printed after 1 minute
```

Example of a repeated system test using various data patterns :

```
?? V /                   <---- enter '/' to execute in loop mode
```

A Verify test should always be halted by entering the 'ESC' character before the system is powered down if the test is still running.

3.5.8 COPY DATA BLOCK IN CPU RAM

Syntax :

X < start address > < finish address > < destination address >

This command allows a block of data resident in system to be copied into another area of system RAM. The start address and finish address define the data pattern to be copied. The data pattern is copied into the new area starting at the destination address specified. All three parameters must be specified to execute the operation. There are no default values.

3.5.9 INSPECT AND MODIFY BUBBLE MEMORY DATA

Bubble memory Inspect / change syntax :

Z < storage address > CR

Bubble memory dump syntax :

Z < storage start address > < storage end address > CR

Bubble memory inspect / change reads the data directly from the bubble and allows modification of the word of data. The termination character determines the next step:

* If a carriage return, the prompt occurs for the next command.

* If a space, the next location is displayed.

* If a minus sign, the previous location is displayed.

If a hexadecimal value is entered before the termination character, the displayed memory location is updated to the value entered. The bubble and page numbers of the page accessed are also printed. If a checksum error occurs when the page of data is read then a colon replaces the equals sign to indicate caution.

Example :

```
?? Z 8C50                        <----- Carriage return entered
8C50 (3 142) : 2025              <----- note, checksum error
8C52 (3 142) = 8925  EE60        <----- New contents entered, space bar
8C54 (3 142) = 4309              <----- Minus sign entered
8C52 (3 142) = EE60              <----- New contents
8C54 (3 142) = 4309
8C56 (3 142) = 0032  6295        <----- Carriage return entered
??     |   |
       |   |---- page number [0..280], hex
       |
       |------- bubble number [0..5]
```

Bubble memory dump directs a display of bubble memory contents from 'start address' to 'stop address'. Each line of output consists of the absolute address of the first data word followed by the bubble number and page number in brackets. The page of data is then printed as eight hexadecimal words, the last word is reserved by TIBUB for checksums and not printed.

Example :

```
?? Z 8C40 8C5F
8C40 (3 141) = 2020 4745 3970 8EE4   DEAD EEA3 FE34 5112
8C50 (3 142) = 2025 EE60 4309 6295   E562 619D 5DE3 5DE3
??      ↑  ↑
        |  |
        |  |---< page number [0..280], hex
        |
        |------< bubble number [0..5]
```

This command enables the entire contents of the bubble memory system to be displayed directly on the terminal. If any character is entered on the terminal during the printout, the operation will be terminated before the 'stop address' is reached.

3.6 TIBUB ERROR MESSAGES

TIBUB will generate an error message if an incorrect operation occurs during either command processing or its execution. The word ERROR is printed followed by a single digit number to identify the type of error, see table 3-3.

TABLE 3-3 TIBUB ERROR MESSAGES

0	Invalid tag detected by cassette loader
1	Checksum error (bubble or cassette) detected
2	Invalid range or termination character detected
3	Null input field detected by load or dump routines
4	Invalid command detected
5	Bubble controller operation timeout
6	Bootstrap load map checksum error

3.7 USER ACCESSIBLE UTILITIES

3.7.1 GENERAL

TIBUB supports 7 XOP's that provide terminal control from the user program (identical to those in TIBUG) and two extra XOP's that support byte transfers to and from the Bubble Memory System.

XOP support within TIBUB is only directly available when TIBUB resides in memory starting at location 0. TIBUB will provide XOP support while resident elsewhere if the user loads the appropiate XOP vectors. The XOP vectors can be copied from the start of the TIBUB

EPROMS. The new address of the EPROMS must be added to each of the program counter vectors.

TABLE 3-4 TIBUB XOP SUPPORT

XOP	FUNCTION	paragraph
6	Write data block to Bubble Memory	3.7.2
7	Read data block from Bubble Memory	3.7.3
8	Write 1 hex character to terminal	note 1
9	Read one hex word from terminal	note 1
10	Write one hex word to terminal	note 1
11	Echo character	note 1
12	Write one character to terminal	note 1
13	Read one character from terminal	note 1
14	Write message to terminal	note 1
15	Reserved by TIBUB for breakpoint	-

Note 1 : XOPs 8 to 14 are covered in detail in the respective Microcomputer user's manual.

3.7.2 DUMP DATA BLOCK INTO BUBBLE MEMORY

XOP 6 copies a block of data from the CPU RAM to the bubble memory. There are no restrictions on either block size or starting address. The XOP returns an error if the controller times out or the storage address is too high.

The XOP uses a Supervisor Call Block (SCB) to transfer the parameters. Its format is shown in Table 3-5.

TABLE 3-5 SCB FORMAT FOR BUBBLE MEMORY XOPs

BYTE	DESCRIPTION
0	CPU RAM address
2	Number of bytes to transfer
4	MSW of storage address
6	LSW of storage address

The storage address is defined as 32 bits to allow for possible expansion to larger systems. For calls to TIBUB (TM 990/210 revision) the MSW (most significant word) of the storage adrress should be set to zero.

The error returns are implemented in a similar manner to XOP 9, (read a hex character from the screen). After TIBUB returns control to the caller, the immediate return address is assumed to point to the error handler and the following address is the normal return entry point. The Carry (C) status bit will be set if an attempt is made to access a storage address outside the range of the bubble memory module. On the TM 990/210 module the storage address must not exceed hex F05F. If the controller times out during the operation then the Equal (EQ) status bit is set, see Table 3-6. (A new status word is loaded into the 9900 by the RTWP instruction at the end of the XOP call.) In the following example a call is made to the XOP to transfer 205 bytes of data. If any type of error occurs then a message is generated.

```
*
SCB     DATA >4000              CPU RAM address
        DATA 205                number of bytes to transfer
        DATA 0                  storage address MSW
        DATA >CE00              storage address LSW
*
*
ENTRY   XOP  @SCB,6             Call bubble routine
        DATA ERR                Pointer to error handler
        MOV  a,b etc            Normal XOP return
*       EXIT
*
ERR     XOP  @TEXT,14           Error handler
*       EXIT
*
TEXT    'Bubble Data Transfer Error'
BYTE    0
*
*
```

3.7.3 LOAD DATA BLOCK FROM BUBBLE MEMORY

XOP 7 copies a block of data from bubble memory to CPU RAM. The format of the call is identical to the dump XOP described in section 3.7.2. The XOP uses an SCB to pass the parameters to the bubble routine. The SCB block is shown in Table 3-5.

The XOP uses two return addresses, the first address following the call is the error return address and the second is the normal return address. Three types of error may be returned, each sets a different status bit. Table 3-6 defines the possible errors and the status bits returned.

TABLE 3-6 ERROR RETURN FROM BUBBLE MEMORY XOPs

ERROR	STATUS BIT
Controller time out	EQ (D2) : Equal
Range error	C (D3) : Carry
Checksum Error detected	OP (D5) :*Odd parity

* Applicable to Load (XOP 7) only

If the type of error is unimportant then any error returned may be handled in the same way as described in Section 3.7.2. The following example shows how full error status may be recovered after the XOP call.

```
*
*
SCB     DATA >4205              CPU RAM address
        DATA 30                 number of bytes to transfer
        DATA 0                  storage address MSW
        DATA >AE00              storage address LSW
*
ENTRY   XOP  @SCB,7             Call bubble routine
        DATA ERR                Pointer to error handler
        MOV  a,b etc            Normal XOP return
*       EXIT                    carry on program execution
*
ERR     JEQ  ERR1               controller error
        JOC  ERR2               range error
        JOP  ERR3               checksum error
        JMP  $                  (system error)
*
ERR1    XOP  @TEXT1,14
        JMP  HOME
ERR2    XOP  @TEXT2,14
        JMP  HOME
ERR3    XOP  @TEXT3,14
        JMP  HOME
HOME    EQU  $
*       EXIT
*
TEXT1   TEXT 'Controller timeout'
        BYTE 0
TEXT2   TEXT 'Invalid address'
        BYTE 0
TEXT3   TEXT 'Checksum error'
        BYTE 0
```

3.7.4 BOOTSTRAP LOADING FROM BUBBLE MEMORY

TIBUB provides a bootstrap loading facility that will load a program from bubble memory into CPU RAM, and then directly execute the loaded program. This facility is restricted to use with the TM990/101M CPU card.

When TIBUB is initialized via a RESET or a LOAD, TIBUB will test switch # 1 on S2 on the CPU card. If the switch is ON then the first 22 bytes starting at storage address 0 will be read and examined to find a boot map. A bootload may also be initiated by entering 'P' on the keyboard when TIBUB is operating at command level. The following data block constitutes the boot map.

TABLE 3-7 BOOTSTRAP LOAD MAP

BYTE	DESCRIPTION
0	Primary Status (not used =0)
2	Secondary Status (not used =0)
4	Command Register (not used =0)
6	Storage address in bubble memory MSW
8	Storage address in bubble memory LSW
10	Size of program in bytes
12	CPU address of first byte of program MSW (= 0)
14	CPU address of first byte of program LSW
16	Status block return address MSW (not used =0)
18	Status block return address LSW (not used =0)
20	Checksum (16 bit)

The above data structure has been chosen to remain compatible with the TM 990/303 floppy disc controller in case the TM 990/210 module is required to be used with the MPP (Microprocessor Pascal) file manager at any time in the future.

The sum of the eleven words in the data block must be zero. If the checksum fails then a normal TIBUB reset will be attempted (wait on character 'A' being entered) and then an error message will be generated before the first prompt is generated. The unused data words may be used as required, but the checksum must always be correct. The first two words of the new program loaded into system RAM must be reserved as the WP and PC vectors for initial program execution. The boot loader executes a direct BLWP instruction at this location.

The bootstrap load may be initiated on a TM 990/100M by arranging for the CRU input bit hex 12 (R12=hex 24) to be low during the RESET or LOAD operations.

Section 4

Software Interface

4. SOFTWARE INTERFACE

4.1 GENERAL

Nearly all the command processing and control is handled on the TM 990/210 by the TIB0901/TMS9916 NMOS controller IC. There are seven basic programming steps required to complete a bubble data transfer from a 'cold start'.

1 . Load Bubble Select Register
2 . Initialize Controller and Bubble
3 . Load Page Select Register (bubble address)
4 . Load Page counter (multi-page mode only)
5 . Issue command
6 . Load / unload buffer
7 . Test power fail and status

Steps 5 and 6 are reversed for the single page write command. The controller will only transfer pages of data. One page of data on the TIB0203S is 18 bytes in length.

Before commencing a new command cycle, the controller must be allowed to complete any previous operations.

The next sections cover the register arrangement in the controller and the different modes of operation. Each of the programming steps is covered in subsequent sections.

4.2 CONTROLLER REGISTERS

The controller is a memory mapped device and occupies 16 consecutive locations in memory. The controller presents only 8 bits to the 16 bit TM 990 bus. These bytes are all located on consecutive even byte addresses and therefore in the case of the 9900 the controller registers are only seen in the most significant byte of each word. The first 15 registers are actually on the controller chip but the 16th is implemented in external circuitry on the TM 990/210 board. The base address may be set up as explained in section 2.3.

TABLE 4-1 CONTROLLER REGISTER FUNCTIONS

REG -hex-	9900 Subad.	Direction and Function	Normal value loaded
0	0	Write to Page Select Register (LSB, 8 BITS)	-
1	2	Write to Page Select Register (MSB, 2 BITS)	-
2	4	Write to Command register, see 4.3	-
3	6	Read from Fifo	
4	8	Write to Fifo	-
5	A	Read from Status register #1	
6	C	Write/read Page counter (LSB, 8 bits)	-
7	E	Write/read Page counter (MSB, 2 bits)	-
8	10	Write Minor loop size (LSB, 8 bits)	>81
9	12	Write Minor loop size (MSB, 2 bits)	>02
A	14	Write/read Page position counter (LSB, 8 bits)	-
B	16	Write/read Page position counter (MSB, 2 bits)	-
C	18	Write to Page size reg. in bytes	>12
D	1A	Read from ROM code (for testing only)	
E	1C	Write/read State address (for testing only)	-
F	1E	Write to Bubble select register (4 bits max)	-
	and	Read from Status register #2	

4.3 COMMAND REGISTER

All controller commands are executed by loading the appropiate opcode into this register. The command register is controller register 2 at 9900 subaddress 4. Table 4-2 shows the bit definition. To issue a command the appropiate bit is set to a ONE. More than one bit may be set high at the same time. For example multi-page read is >12.

TABLE 4-2 BIT DEFINITION OF THE COMMAND REGISTER

9900 bus	TIB0901 data bit		Function
D0	7	MSB	Mask interrupt
D1	6		Reset controller + interrupt
D2	5		(test internal ROM code)
D3	4		Select Multi-page mode
D4	3		Select Single page mode
D5	2		Write page
D6	1		Read page
D7	0	LSB	Initialize (takes 6.4mS)

The reset command halts the write, read and initialize sequences immediately and will clear an unserviced interrupt. The two modes of operation, single and multi-page are mutually exclusive. When the controller is reset, single page mode is selected.

4.4 STATUS REGISTERS

Controller status information is presented in registers 5 and 15 (see table 4-1) on the TM 990/210 board. The controller IC, TIB0901, presents the 8 status bits in register 5. Two additional bits are also presented in register 15. These have been added to the TM 990/210 board to allow greater flexibility.

TABLE 4-3 BIT DEFINITION OF STATUS REGISTER #1

9900 bus	TIB0901 data bit		Function
D0	7	MSB	Byte read / written
D1	6		Page counter = zero
D2	5		Controller busy
D3	4		Addressed page available
D4	3		Major loop shift counter = 0
D5	2		Inhibit
D6	1		Load / increment
D7	0	LSB	Page read / written

TABLE 4-4 BIT DEFINITION OF STATUS REGISTER #2

D0-D5		Not used, but normally high
D6		Power fail latch set
D7	LSB	Interrupt

The most useful and significant status bits are described below :

CONTROLLER BUSY : The busy status bit is set to a ONE when either a write, read or initialize command is being processed. At the end of the operation the busy bit is reset and no further bubble movement will occur. The CPU must allow 15 uS to elapse before the status bit is tested if a command has just been issued to the controller.

PAGE COUNTER ZERO : The page-counter-equals-zero status bit is set when the page counter is decremented to zero while in multi-page

mode. The bit is reset when the page counter is reloaded.

INTERRUPT : The controller uses the interrupt bit to signal the CPU that the next byte should be loaded/unloaded when the controller is operating in multi-page mode.

POWER FAIL LATCH : During power up on the 210 module the power fail latch is set and all bubble I/O operations are inhibited. Loading the Bubble Select Register will reset the latch. The power fail latch status bit allows the software to test for a power fail glitch during the last data transfer to the bubble.

The other status bits are covered in more detail in the TMS9916/TIB0901 Bubble Memory Controller specification.

4.5 SINGLE PAGE OR MULTI-PAGE ?

The controller may be operated in two different ways: single or multi-page mode. In single page read mode the controller transfers one page of data only and the FIFO memory is loaded with 18 bytes of data. When the FIFO is full, an interrupt is issued to the CPU. The CPU does not have to service the interrupt immediately because no more data will be loaded into the FIFO. A new command cycle cannot be commenced until the present cycle is completed and the controller becomes idle again. In the single page write mode the interrupt is generated after the FIFO has been unloaded, and therefore it may be reloaded but a new command may not be issued until the controller becomes inactive again.

In multi-page read mode the controller accesses the first page of data and as soon as the first byte of data is available in the FIFO an interrupt is issued to the CPU. This byte must be read before the next 8 bits of bubble data are received by the controller, i.e. within 160uS (8x2x10uS). Data does not accumulate in the FIFO; only one byte is ever available. The CPU must always be capable of unloading the FIFO within 160uS of receipt of the last byte. The multi-page write mode of operation is basically the same, a byte must be loaded into the FIFO for each interrupt that the CPU services.

From the above it can be seen that there are two distinct modes of operation, and the correct mode should be chosen for the application.

Consider a simple host system that does not have interrupts. In such a system multi-page mode can be utilized for single page transfers. Large blocks of data may be transferred in multi-page mode at the maximum data rate (45 Kbits/sec). The software may poll the interrupt line and wait for it to go active before the next byte is transferred to or from the FIFO.

In a multi-tasking or interrupt driven system in which the bubble memory controller has a low priority, single page mode should be chosen. Otherwise the CPU may not be able to service the controller

within the critical period of 160 uS. In single page mode the interrupt is generated when the FIFO is full and the data may then be unloaded. But before a new command is issued to the controller the present cycle must be completed. The controller must poll the busy bit until it is not busy.

If the access time to the first data bit is not important a second approach may be adopted. Each single page operation takes exactly 12.8 mS. After the 12.8 ms has elapsed the data has been read or written, the controller becomes idle and it is ready to start a new command cycle. If a programmable timer is available in the system such as the TMS9902 or TMS9901 interval timers, the timer may be used to interrupt the CPU after the 12.8 mS has elapsed. This approach requires the minimum CPU overhead. In a read operation the read command is issued and the timer is immediately loaded. When the CPU is interrupted the data may be loaded/unloaded and a new command cycle can be started immediately. The operation can be pipelined by reducing the timer interval slightly and unloading the last page or loading the next page into the FIFO before the controller actually halts. (The Page Select Register may also be reloaded.) When the controller stops another command cycle may be started immediately if required.

In a multi-tasking, interrupt driven system in which the bubble memory controller has a high priority, multi-page mode will produce the best performance if the controller service routine is efficiently coded. Other lower priority tasks or interrupts may still be handled in the time remaining from the CPU servicing the interrupt from the controller. As an approximate guide the 9900 will execute about 30 instructions in the 160uS time frame but this assumes no memory wait states and typical instruction useage. The controller generates interrupts to the host CPU and the CPU then services the interrupt by transferring a single byte of data. The interrupt is reset when the byte is transferred.

4.6 LOADING THE BUBBLE SELECT REGISTER (BSR)

It is only possible to select one memory unit at any given time. To select bubbble memories #1,2,3,4,5,or 6, register 15 on the controller is loaded with 0,1,2,3,4, or 5, respectively. If a number from 8 to 15 is loaded into the register, all bubbles are deselected. Loading data into the BSR also serves to reset the power fail latch. The bubble select register is write only; the address is shared with two read only status bits.

Example of 9900 coding :

```
LI   R0,3           Load no. of desired bubble into R0
LI   R8,BASE        Load controller base into R8
SWPB R0             Move number into MSB
MOV  R0,@>1E(R8)    Move number into base + offset * 2
```

CONTROLLER POWER UP INITIALIZATION

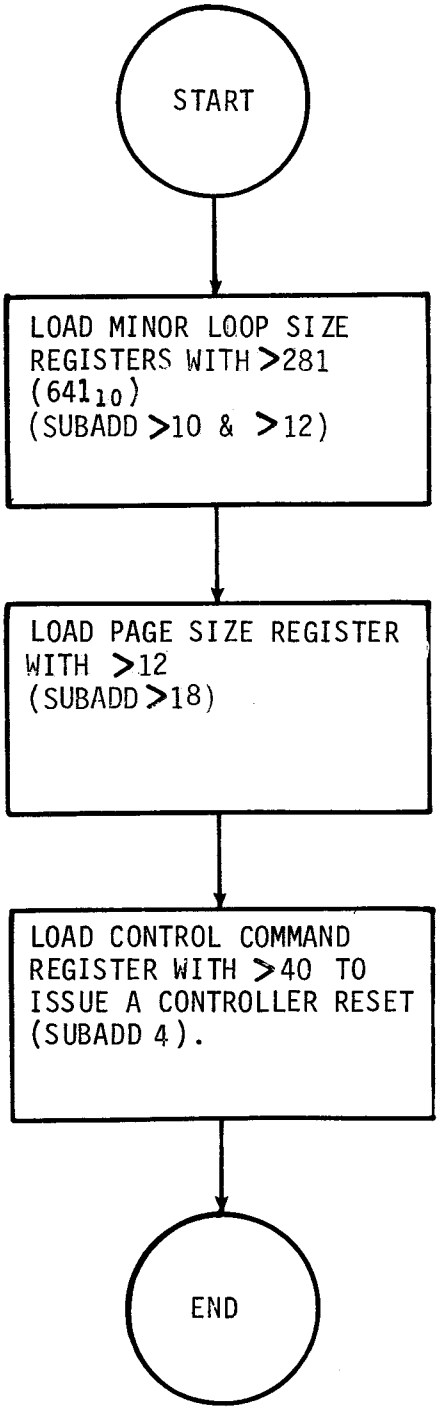


FIGURE 4-1

BUBBLE MEMORY POWER UP INITIALIZATION

EACH BUBBLE IN THE SYSTEM MUST BE INITIALIZED.

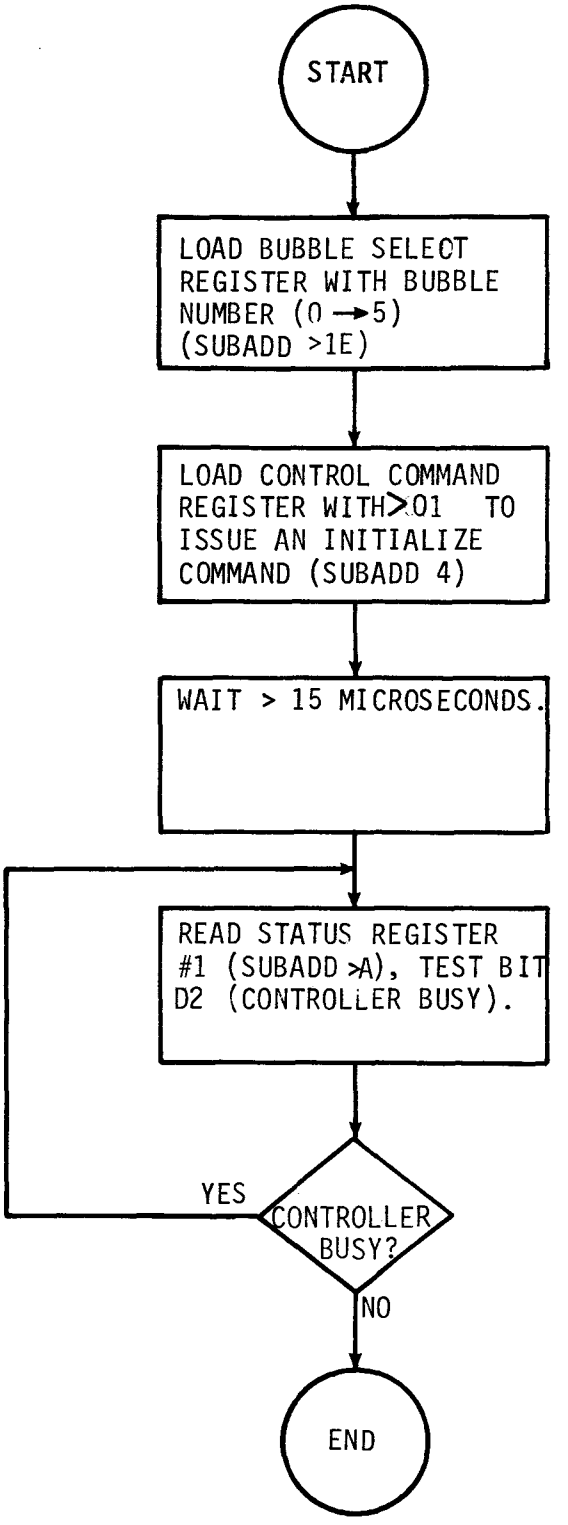


FIGURE 4-2

4.7 CONTROLLER INITIALIZATION

The controller is intended to operate with any bubble using a major/minor loop architecture irrespective of size. Therefore the registers must be preloaded with the dimensions of the TIB0203 architecture. If a user program fails to work correctly or displays abnormal phenomena, then incorrect initialization can often be found to be the root of the problem.

The following steps are necessary after each power up.

(See Figure 4-1)

1. Load minor loop size registers MSB (2) and LSB(>81)
2. Load page size (>12)
3. Issue controller reset (>40)

The next steps are required for every bubble in the system.

(See Figure 4-2)

4. Load the bubble select register
5. Issue controller initialization (>01)
6. Test controller status until it is not busy

The following 9900 code may be used to initialize up to 6 bubble memories.

```
*       Initialization table
*       Address, Data
*
TOP     BYTE >12,>02        load minor loop size MSB
        BYTE >10,>81        load minor loop size LSB
        BYTE >18,>12        load page size
        BYTE >04,>40        issue reset command
        BYTE >04,>01        issue initialize command
        BYTE 0,0
*
SR1     EQU  >0A
BSR     EQU  >1E
*
INIT    EQU  $
        CLR  R3             clear bubble register (0)
*
DELA    CLR  R5             clear timer
DELA1   INCT R5             increment timer   T5=0..>7FFE
        JLT  TIMMOT         timeout has occurred,T5=>8000
        MOVB @SR1(BAS),R1   read status register
        SLA  R1,3           wait for busy reset
        JOC  DELA1          still busy, loop back
*
EOB     CI   R3,>600        last bubble was 5 ?
        JHE  FINI           yes, so finished
```

```
*
BIN     LI   R4,TOP        get pointer to table
        MOVB R3,BSR(BAS)   select bubble
L1      MOVB *R4+,R5       get address byte
        JEQ  NEXBUB        its a zero
        SRA  R5,8          address to LSB (arithmetic)
        A    BAS,R5        calculate port address
        MOVB *R4+,*R5      load data byte
        JMP  L1            next byte
*
NEXBUB AI    R3,>100       increment bubble counter
        JMP  DELA          and repeat loop
*
FINI    EXIT               initialization complete
*
```

In the above example it is assumed that the base address points to register 0 on the controller. It may be more desirable to set the base register to point to the FIFO read address or the status register. This sometimes result in a small saving of code or easier coding. The above software routine incidentally reloads the registers on the controller six times.

The reset command resets any pending interrupts, stops any bubble operations immediately, and selects single page mode. The software reset command does not reset the FIFO pointer. The reset command should never normally be issued while the controller is active or the reference position for the particular bubble that is being accessed will be lost.

The initialization command rotates the bubbles in the minor loops through one complete cycle (641 positions). It also flushes out any bubbles in the major loop by annihilation. The LED on the TM 990/210 module will flash when the controller executes the initialization.

4.8 LOADING THE PAGE SELECT REGISTER (PSR)

Pages of memory are addressed by loading registers 0 and 1 with the LSB and MSB (2 bits) respectively of the required page.

Example : to select page >214

```
        LI     R1,>214    Load 214 into register 1
        LI     R8,BASE    Load base address into register 8
        MOVB   R1,@2(R8)  Load register 1 with MSB (>02)
        SWPB   R1         Move LSB to MSB in register
        MOVB   R1,@0(R8)  Load register 0 with LSB (>14)
```

However the page select register refers to the position of the page in the minor loops, that is page 1 is located next to page 2. But in multi-page mode, the pages are not read in this sequence. The multi-page operation transfers the next convenient page from the minor

loops as soon as the previous page clears the top of the minor loops. The next convenient page in the minor loops is, in fact, on the other side of the minor loops from the original page. Between the two sequential pages there are 322 bit postions.

Therefore, if four pages are read starting at page zero, these pages will be accessed in the following sequence :

0 , 322 (0+322), 3 (322+322-641), 325 (3+322)

This sequence is used to acheive the highest data throughput in multi-page mode. In applications that utilize the multi-page mode of operation, the pages should be numbered in the sequence that they would be read or written in multi- page mode and the page select register should be loaded with the equivalent physical position of the page in the minor loops. For example if page 1 is required then 322 is loaded into the Page Select Register. This means that if, instead, two pages were requested and the first was page 0, then the second would also be the same page 1 or physically page 322 in the minor loops. TIBUB as described in section 3 of this manual operates only in this manner. Note that in single page or multi-page mode, the controller always accesses the same first page.

The general relationship between the physical position of the page (PSR value) and the sequence number is shown below.

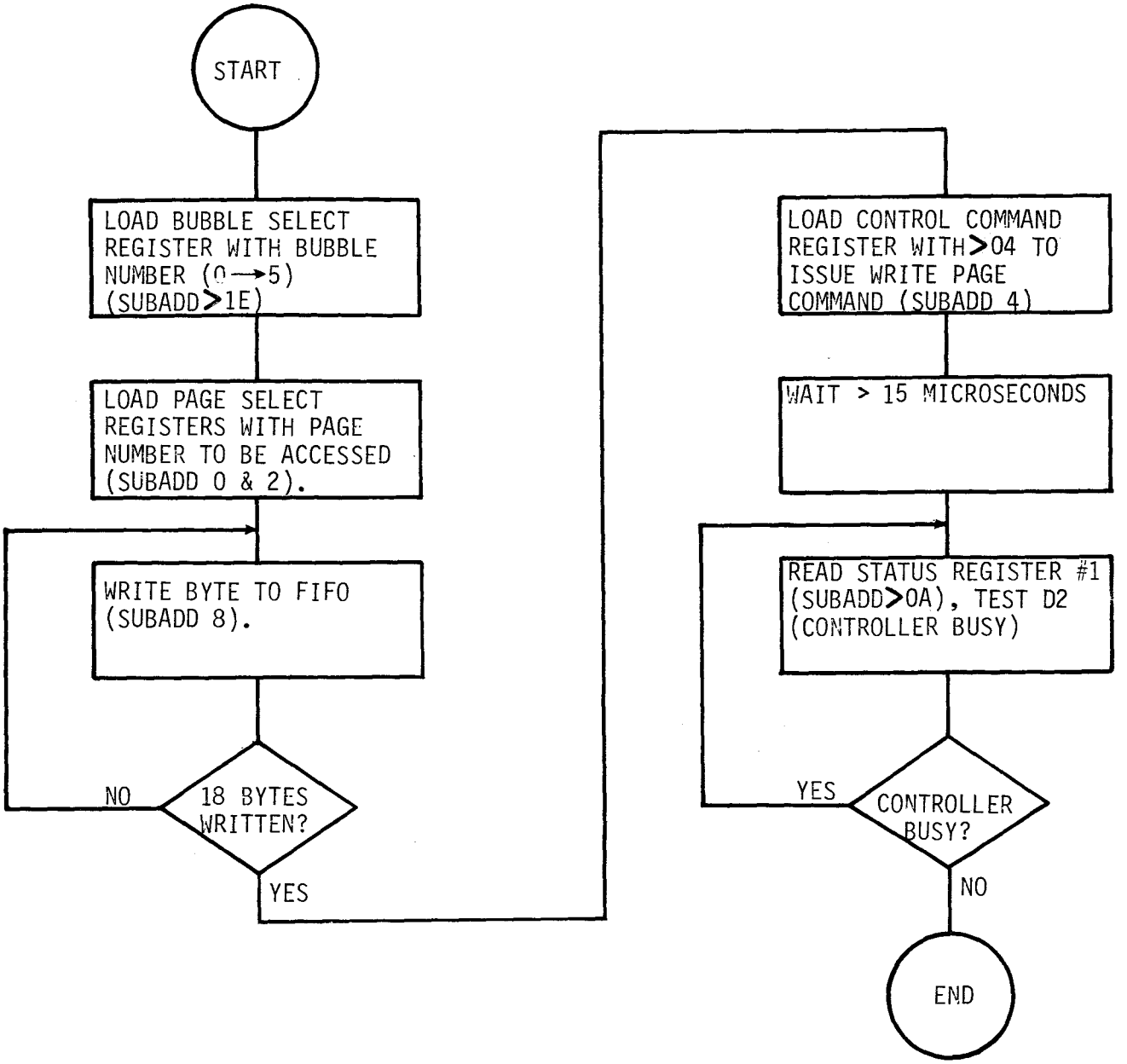
PSR value = (Page sequence no. x 322) MOD 641

(A MOD B refers to the remainder after integer division of A by B)

As the 9900 microprocessor has implicit multiply and divide instructions, this algorithm may be implemented simply as follows :

```
ENTRY     EQU
          LI     R1,>87     Consider selection of page >87
          LI     R6,>142    Store bit position jump (=322)
          LI     R7,>281    Store no. of pages      (=641)
*
          MPY    R6,R1      Leave 32 bit result in R1  R2
          DIV    R7,R1      Divide by no. of pages
*                           and leave remainder in R2
*
          LI     R8,BASE    Set up controller base address
*
          MOVB   R2,@2(R8)  Load PSR (msb)
          SWPB   R2         Align LSB to controller port
          MOVB   R2,@0(R8)  Load PSR (lsb)
```

If the page select register is loaded with a value greater than 640 (hex 280) and a read or write command is executed then the controller will "hang up". The controller may be stopped by issuing a reset command.



SINGLE PAGE WRITE (STATUS DRIVEN)

FIGURE 4-3

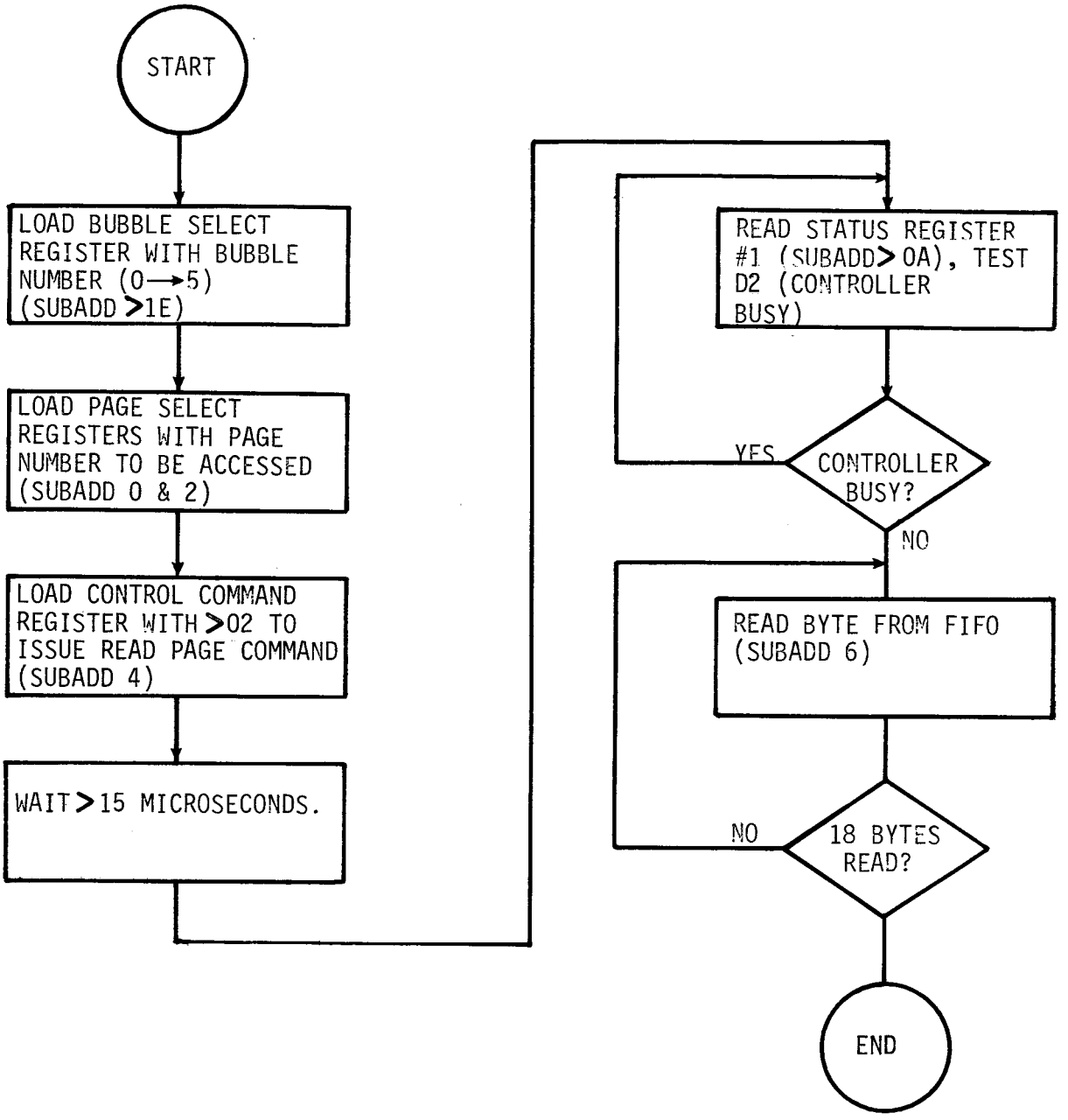
4.9 SINGLE PAGE WRITE

Before any data transfers are executed the page select and bubble select registers should be loaded (see Figure 4-3). If the controller has been previously operated in multi-page mode, then single page mode should now be reselected. The FIFO should be loaded with 18 bytes and a single page write command issued. Suitable 9900 code is shown below:

```
BUFFER    BSS     18
SINGPG    BYTE    >08
WRCMD     BYTE    >0C
*
ENTRY     EQU
SINGWR    LI      R8,BASE          Set up controller base
          LI      R4,BUFFER        Load buffer address
LOOP      MOVB    *R4+,@8(R8)      Load FIFO from buffer
          CI      R4,BUFFER+18     Was it last byte ?
          JNE     LOOP             No so loop back
*
          MOVB    @WRCMD,@4(R8)    Issue single page write command
*
DELAY     SETO    R2               Allow the controller several
DEL       SRL     R2,1             Microseconds to set the
          JOC     DEL              Busy bit high.
*
TEBUSY    MOV     @>0A(R8),R6      Read status and store in R6
          SLA     R6,3             Busy bit set ?
          JOC     TEBUSY           Yes,  so read again
*         EXIT
```

At this stage the controller will become busy and the write page command will be executed. The operation will take 12.8 ms (641 x 2 x 10 us). In single page mode 18 bytes should always be transferred to or from the FIFO. If the FIFO is left partially full then the FIFO pointer will not be reset thus the following operation is jeopardized. The pointer may be reset by executing a single page read and not transferring the data or by applying a hardware reset to the TM 990/210 board. The FIFO is actually a register bank addressed by a recirculating pointer.

During program development it is quite acceptable to read the FIFO either immediately before a write command is issued or after a read command is issued. The data in the FIFO can be read repeatedly because the FIFO pointer underflows from the bottom of the FIFO back up to the top of the FIFO.



SINGLE PAGE READ (STATUS DRIVEN)

FIGURE 4-4

4.10 SINGLE PAGE READ

The single page read operation is similar to single page write except the sequence of events is reversed (see figure 4-4). Before the buffer is unloaded, the status bit must be tested to ensure that the operation is complete. Appropriate 9900 code is shown below for a single page read (assuming that PSR and BSR een have been loaded).

```
BUFFER    BSS     18
SINGPG    BYTE    >08
RDCMD     BYTE    >0A
*
ENTRY     EQU
SINGRD    LI      R8,BASE           Set up controller base
          MOVB    @RDCMD,@4(R8)     Issue single page read command
*
DELAY     SETO    R2                Allow the controller several
DEL       SRL     R2,1              Microseconds to set the
          JOC     DEL               Busy bit high.
*
TEBUSY    MOV     @>A(R8),R6        Read status and store in R6
          SLA     R6,3              Test the busy bit
          JOC     TEBUSY            Still busy, so read again
          LI      R4,BUFFER         Set up pointer to buffer
LOOP      MOVB    @6(R8),*R4+       Unload Fifo into buffer
          CI      R4,BUFFER+18      Was it last byte ?
          JNE     LOOP              No, so loop back
*         EXIT
```

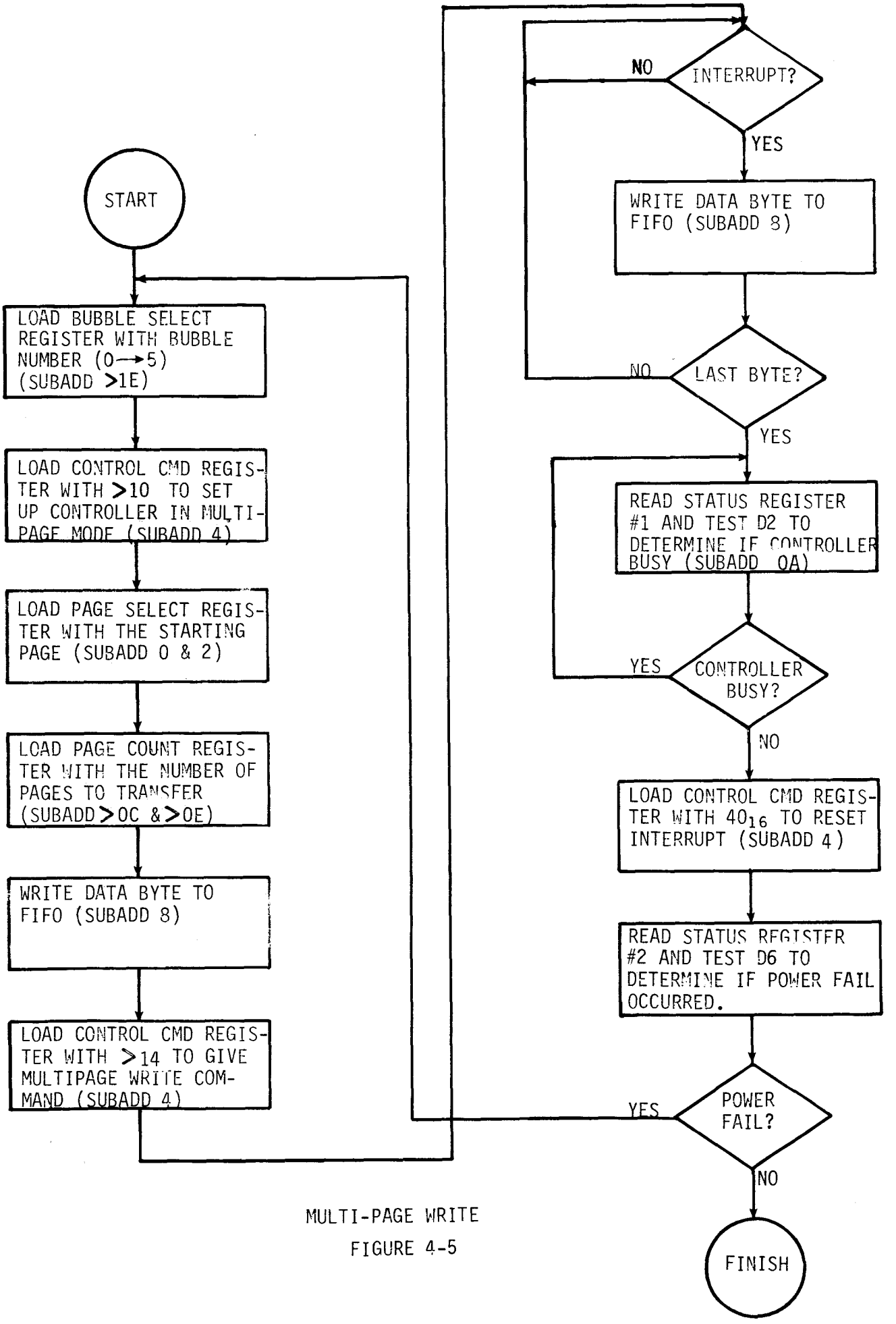
4.11 LOADING THE PAGE COUNTER REGISTER (PCR)

Before a multi-page command is executed the page counter must be loaded. During a multi-page command the page counter may be cleared and the controller will halt within 12.8 ms.

During multi-page read and write operations the page counter is decremented each time that a new page is transferred out of the minor loops. When the page counter register is decremented to zero the controller halts as soon as the present page of data is returned to the minor loops.

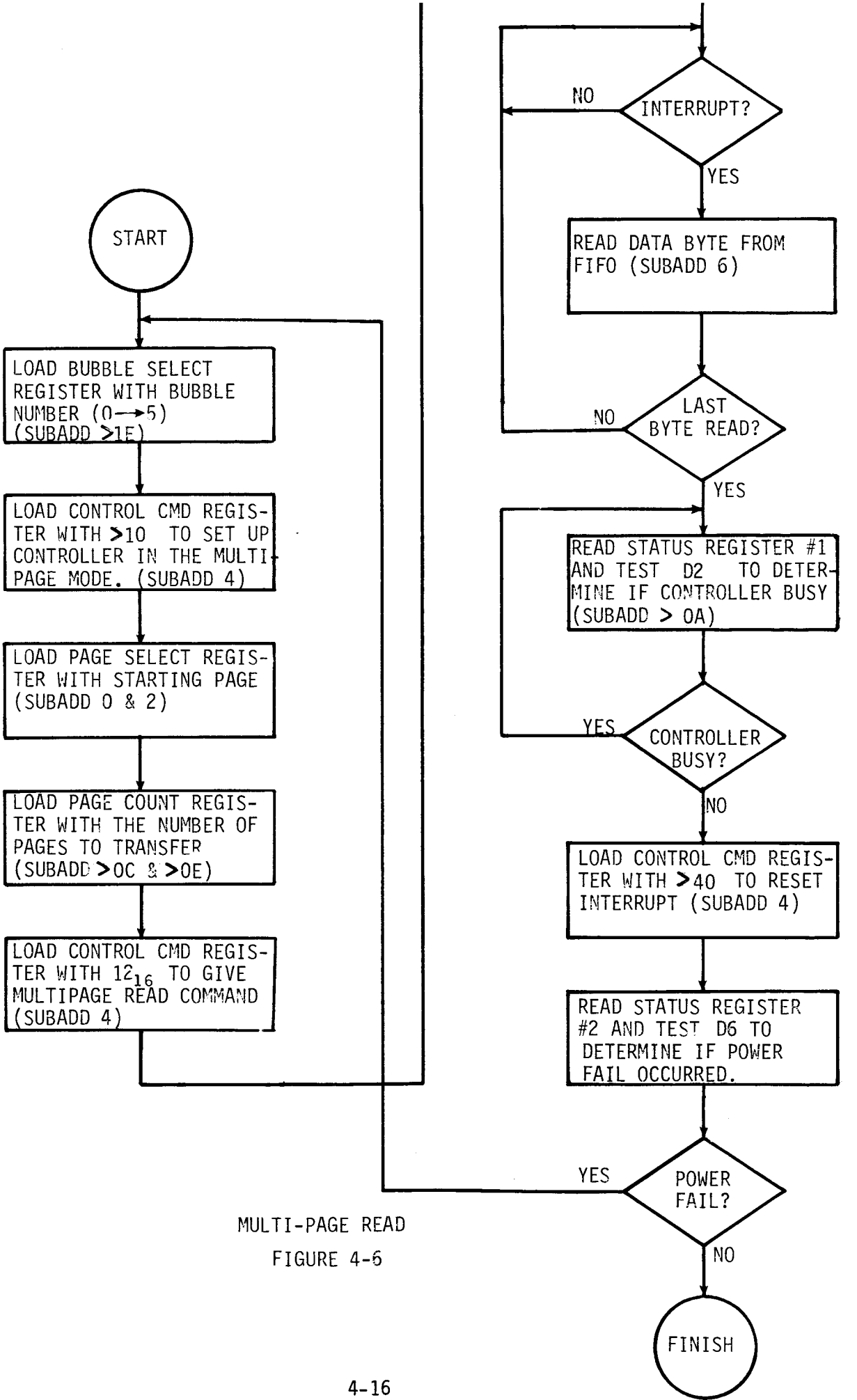
Following is an example of 9900 code suitable for loading the page counter with 340.

```
*
NOPAGE    DATA    340
*
ENTRY     LI      R8,BASE           Load controller base address
          LI      R4,NOPAGE
          MOVB    *R4+,@>0E(8)      Load MSB of page counter
          MOVB    *R4,@>0C(8)       Load LSB of page counter
*         EXIT
```



MULTI-PAGE WRITE

FIGURE 4-5



MULTI-PAGE READ

FIGURE 4-6

4.12 MULTI-PAGE WRITE

The following steps are required to execute a multi-page write :

1. Load BSR, PSR, and PCR.
2. Select multi-page mode
3. Load the first byte of data
4. Issue write command
5. Load remaining bytes one byte per interrupt.

(see figure 4-5)

The cycle is basically straight forward but the processor must load the first byte of data AFTER multi-page mode has been selected and BEFORE the write command is issued. The interrupt line may be reset by moving a byte into the FIFO.

4.13 MULTI-PAGE READ

The following steps are required to execute a multi-page read : (See figure 4-6)

1. Load BSR, PSR, and PCR.
2. Select multi-page mode
4. Issue read command
5. Unload bytes one byte per interrupt.

At the end of the read operation the final interrupt from the controller must be reset by issuing a RESET command to the controller. The interrupt line may be reset by moving the byte from the FIFO.

4.14 POWER FAIL

In the case of total power failure in which the CPU has been reset normally, the system will start up again as if it had been cold started.

A situation may arise in which the primary power source is interrupted for several milliseconds, PF/ line becomes active and the power fail latch is set. If PF/ becomes active while the controller is idle then subsequent operations will be inhibited because all bubble modules will be deselected.

If PF/ becomes active while the controller is executing a multi-page transfer, then the operation will be terminated prematurely. Power fail has no effect on a single page transfer in progress.

At the start of the following bubble operation all bubble modules are deselected and the data transfer is inhibited.

However the power fail latch may be reset at any time by loading the Bubble Select Register. Therefore at the end of any operation the power fail status should be checked. If power fail has occurred the subsequent course of action varies with the application. The state of the power fail latch can be tested on bit D6 of status register #2.

Irrespective of the action of the CPU, data up until the time of power fail is safe; only the page or pages that were still to be transferred will be lost. If the power remains bad then loading the Bubble Select Register will fail to reset the power fail latch and further bubble data transfers will still be inhibited. If required the software can test for this condition by polling the power fail status bit again after the register is loaded.

In some applications it may be acceptable to ignore the status bit completely and reload the Bubble Select Register at the start of any operation. If the power is restored and /PF is no longer active, then loading BSR will reset the power fail latch. In data critical applications it is safest to assume that none of the previous data was transferred in either direction and repeat the whole operation again.

4.15 TESTING FOR BUSY STATUS AND CLEARING INTERRUPTS

Before a new command cycle is entered the controller must be allowed to finish any previous command cycle. This may be done by checking that the 'BUSY' status bit is set low. If it is high then the processor must wait. The status bit may be tested at the start of a new command cycle or at the end of a previous command cycle. If the controller has executed a multi-page operation the controller should be reset to clear any unserviced interrupts. An example of suitable 9900 code is shown below.

```
*
*
RESET     DATA    >4000
*
ENTRY     EQU
          LI      R8,BASE           Set up controller base
TEBUSY    MOV     @>0A(R8),R6       Read status and store in R6
          SLA     R6,3              Test the busy bit
          JEQ     TEBUSY            Still bit, so read again
          MOVB    @RESET,@4(R8)     Reset controller
          EXIT
*
```

4.16 DIAGNOSTICS FOR PROGRAMMING ERRORS

Certain programming errors may be quickly diagnosed. Following is a list of common errors found.

TABLE 4-5 PROGRAMMING ERROR DIAGNOSTICS

SYMPTOM	CAUSE
The controller never becomes active irrespective of command	Check that XAEN is set correctly. Check that software and hardware base addresses match. Check that command register offset is 2 (subaddress 4).
The controller remains permanently active and 'locks up'.	The page select register is being loaded with a value that is out of range. The page select register may be reloaded or the controller can be reset.
During single page read the data is mainly zeros with an occasional correct page.	The processor is not allowing the busy bit to be set before it is read.
All zeros are read in any mode of operation.	Check that a valid bubble address is selected.
During a multi-page read the data has slipped by one byte.	The processor is not resetting the controller before the operation.
The correct data appears in cross patterns across the pages.	Incorrect parameters may be loaded into the bubble size registers.
The controller does not generate enough interrupts to allow the FIFO to be loaded/unloaded	Check that bubble register is not loaded with 6 or 7. If the redundancy PROM is not fitted the controller will assume that it is never addressing a good loop.
The pages appear to have fallen out of sequence.	The particular bubble memory may have lost its reference position. The cause may be power fail

	without sufficient warning. See Section 4.17.
There is incorrect data in the last one or two bytes of a page	The FIFO pointer may not be set to zero before the write data is loaded. Ensure that 18 bytes are always transferred to or from the FIFO. The FIFO pointer may be reset by issuing a fake single page read command.
The bubble system is operating correctly most of the time but very occasionally pages of data will be lost or the bubble select register appears to have been deselected.	Over sensitive power fail monitor can trip the power fail latch without reason, and the software might not be polling the PF bit. Check out power supply monitor.

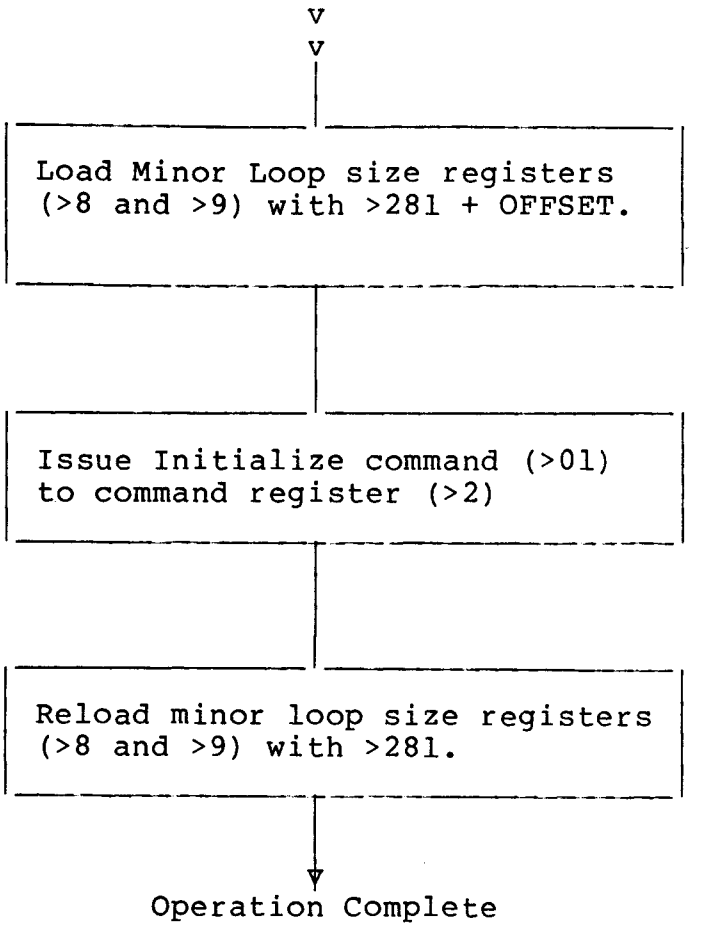
4.17 RESTORING THE BUBBLE REFERENCE POSITION

If necessary the reference position may be restored on a bubble memory without difficulty. The significance of the reference position is explained in the Glossary in section 1.8.

In the normal operating mode the controller always returns the data in the minor loops back to its original position. The controller achieves this by always rotating the bubbles through a multiple of the number of bit positions in the minor loop. The minor loop size is normally loaded into the controller during initialization. If therefore, the minor loop size registers are temporarily modified and a bubble operation is performed the bubbles will me moved into a different minor loop position.

For example, to advance the data by two pages the minor loop size registers should be loaded with 641+2 or alternatively just 002. If an initialize command is now issued the data will be moved two pages. The controller registers should now be returned to their normal value. The initialize command is the most suitable operation as this rotates the data through all the minor loop bit positions just once, that is 641 field cycles.

FIGURE 4-7 REFERENCE RESTORATION SOFTWARE FLOWCHART



NOTE : The offset required for the above operation is the displacement of the pages in the minor loops. If the PSR value is calculated from the required page as suggested in section 4.9 then the offset must also be calculated using the same algorithm.

Section 5

Theory of Operation

5. THEORY OF OPERATION

5.1 GENERAL

The control function is achieved using the TIB0901 NMOS LSI device. Figure 5-1 shows the internal arrangement of the controller. (Some boards may be fitted with parts identified as TMS9916 and TMS5502; these parts are functionally identical to the TIB0901).

The controller has major responsibilities including :

1. Starting and stopping the rotating magnetic field.
2. Tracking the position of the bubbles in the loops.
3. Determining the cycle timing for function drive (e.g Generate).
4. Data formatting and buffering between the 8 bit processor bus and the single bit bubble signals.

The timing of events occurring in the bubble memory is based on the rotating magnetic field. A field cycle is the time in which the magnetic field rotates though 360 degrees and a bubble will move from one chevron to another, that is one bit position.

Figure 5-2 shows the exact relationship between the different control signals generated by the controller. For every page accessed there is one transfer-in pulse and one transfer-out pulse. The other control signals operate on every bit in the page and therefore each control line generates a pulse train of 313 cycles. The 313 field cycles include 157 operations on valid data from the minor loops and 156 void operations.

The Replicate and Annihilate Enables are delayed 68 cycles from the time of the transfer of the page into the major loop, so that the first bit of the serially shifting page of data will be positioned under the Replicate/Annihilate gate on cycle 69 (see Figure 1-4). Similarly, the Detect Enable is delayed 86 cycles, so that the first valid data bit has the additional 20 cycles to travel under the Detector Track from the Replicate/Annihilate gate. The data will begin arriving at the Detector Bridge on cycle 87. Also, the Write Data signal is delayed 296 cycles in order to synchronize the the arrival of the first data bit position with the incoming data at the Generate Gate.

Other remaining key functions are handled by discrete TTL circuitry on the board. The bubble selection, redundancy circuitry and function timing in particular are handled outside the controller IC (see the TM 990/210 schematics in appendix E).

The amplitude, position and length of the bubble control signals are very critical in a bubble memory system. Minor inaccuracies can seriously downgrade the system performance. This precise control of the bubble interface signals is achieved by combining the intelligence

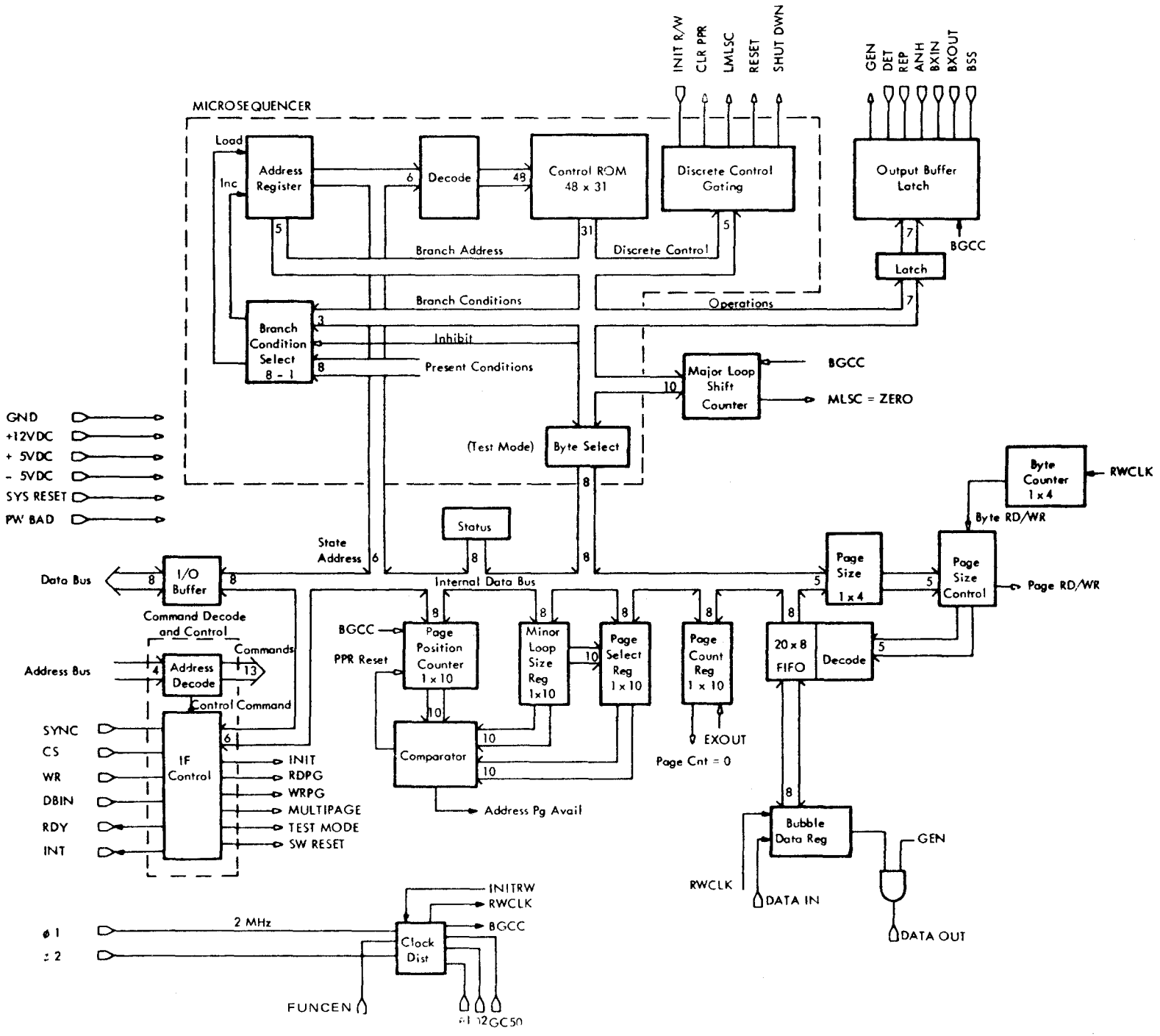


FIGURE 5-1 CONTROLLER BLOCK DIAGRAM

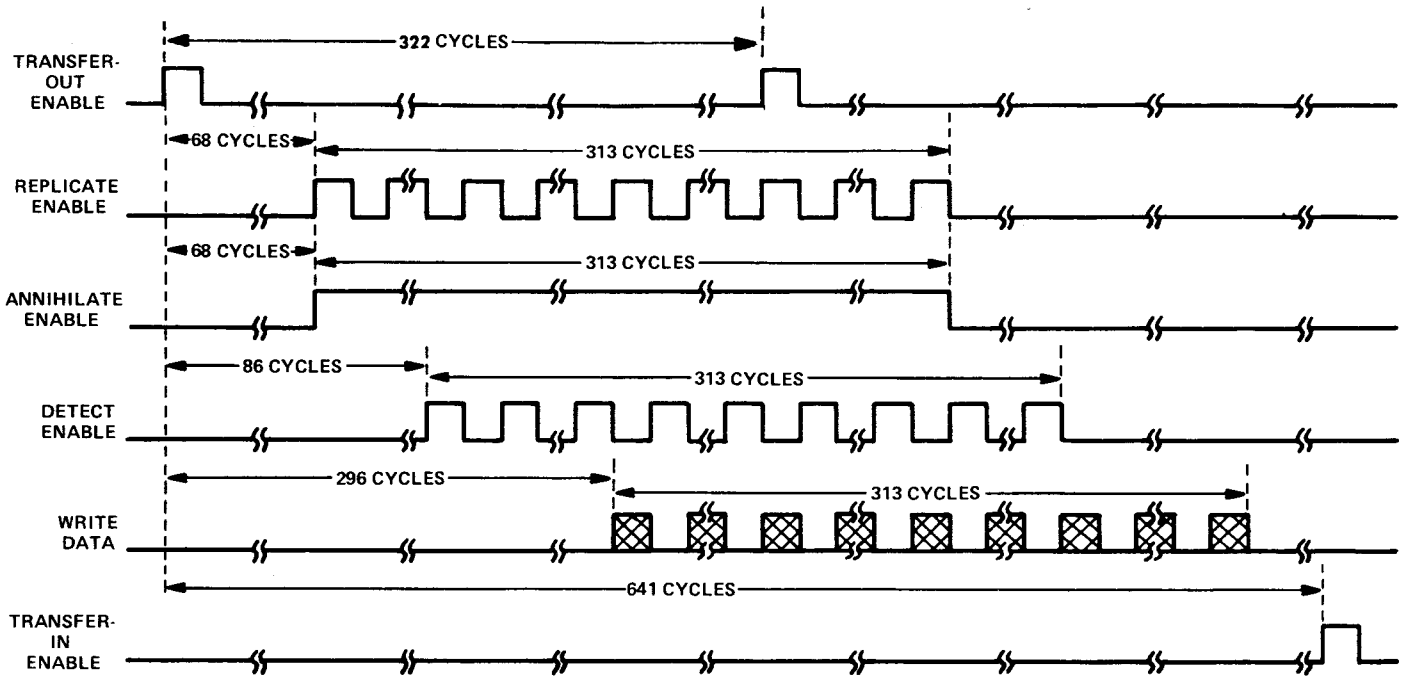


FIGURE 5-2. BUBBLE MEMORY CONTROL TIMING

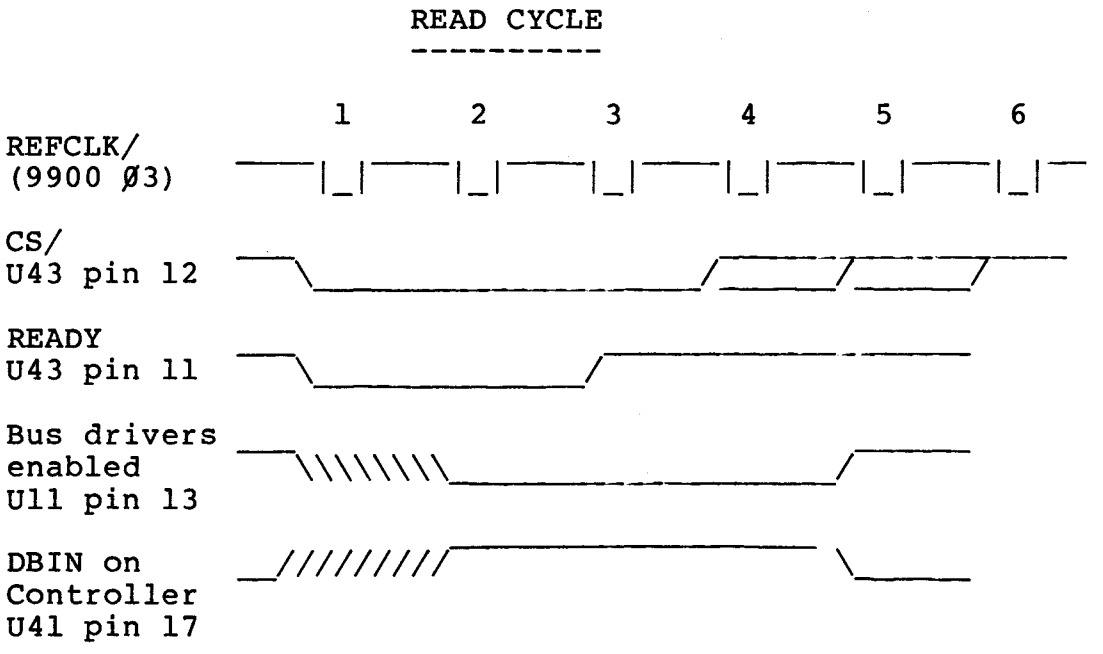
of the controller with the precision of the function timing generator, FTG. The controller provides cycle timing information for all functions and the function timing generator provides the phase timing within the cycle.

Each bubble memory requires its own individual function driver, coil driver and sense amplifier, because unlike the controller and FTG, these parts may not be easily multiplexed between the bubble modules.

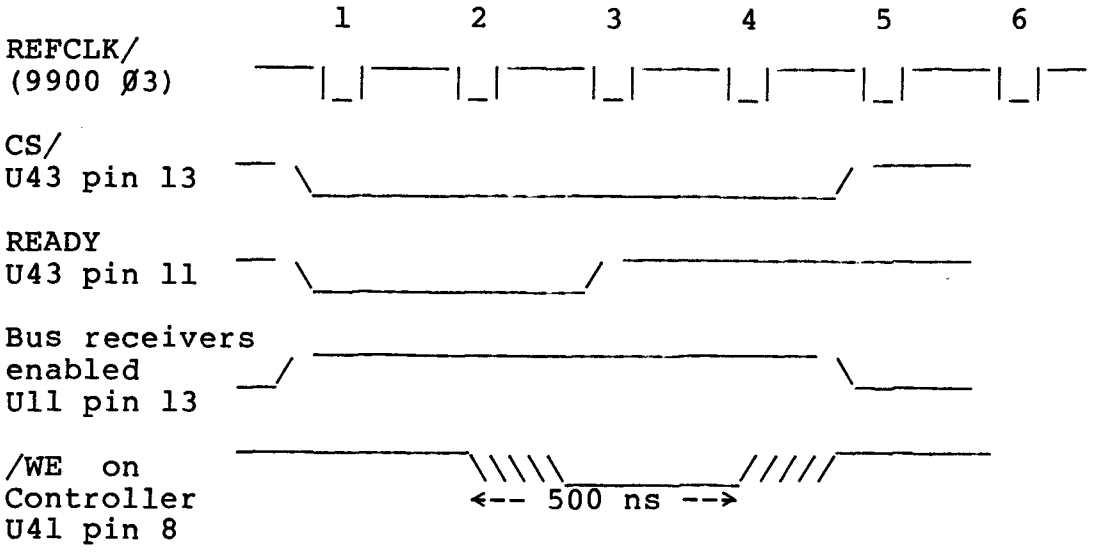
5.2 CPU INTERFACE

The TM 990/210 interfaces to the CPU via a 19 bit address bus, an 8 bit data bus and the four major control lines, all of which are TTL buffered. The board may be run asynchronously from the CPU clocks. The ready signal allows the use of fast processors. Figure 5-3 shows the main interface signals with reference to the bus reference clock, REFCLK (normally Ø3 on 9900 systems). A more detailed analysis is shown in figure 5-4, in these diagrams the signals are referenced to the on-board controller clocks.

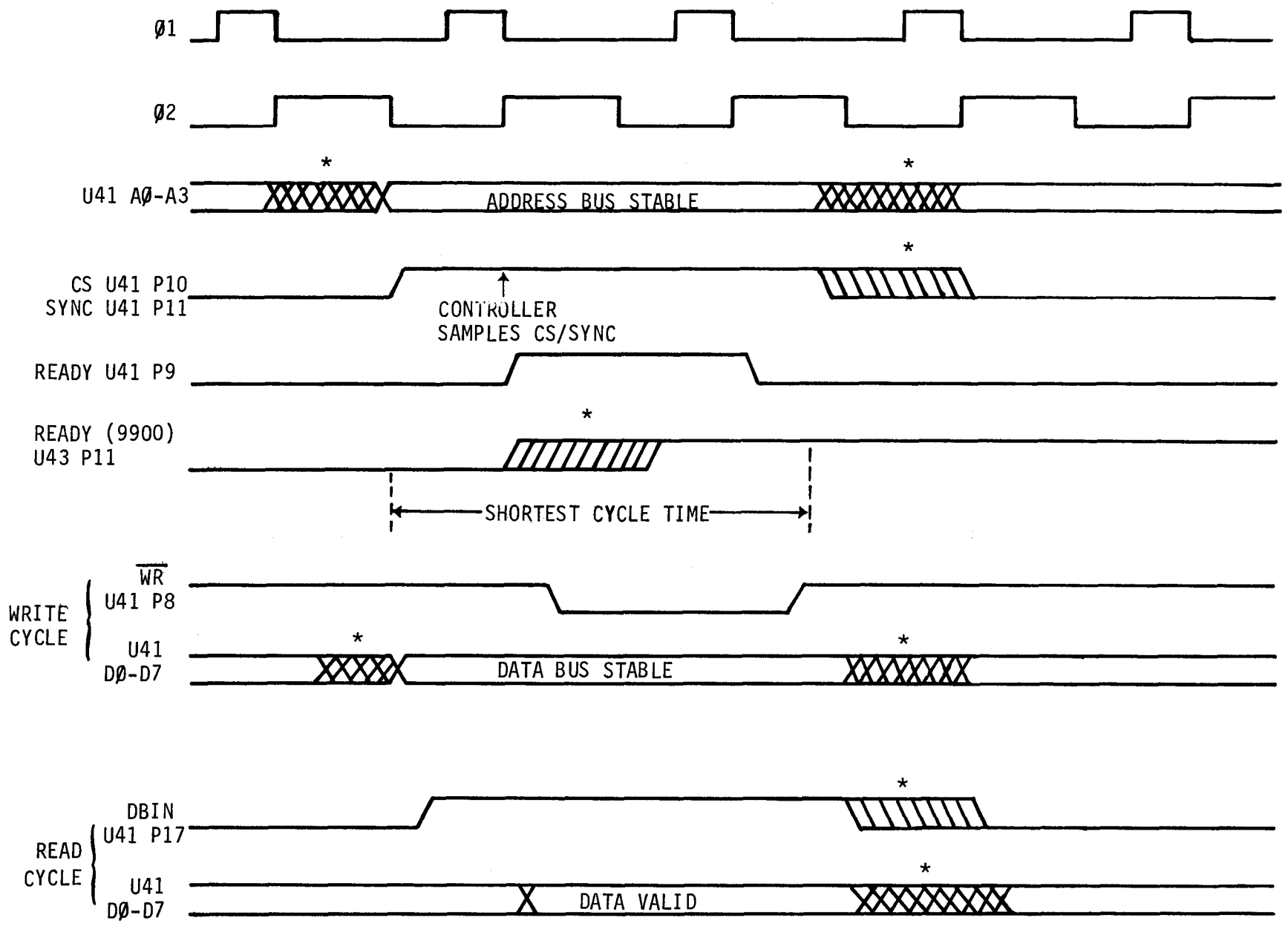
FIGURE 5-3 INTERFACE BUS TIMING DIAGRAMS
(with respect to the bus clock)



WRITE CYCLE



To communicate with the CPU, the controller IC is equipped with an eight-bit bidirectional data port. U11 and U12(LS243) provide buffering between the controller and the system bus. Sixteen memory mapped registers are presented to the CPU through 16 contiguous memory locations (= 32 bytes) whose base address is user selectable. Fifteen bits of the system address bus are compared with the address set up on the switches U23 and U24 using U13-U15(LS85) and U22. If the addresses are equal then the three-state buffer U43/LS125 is enabled and pin 11 is driven low. The ready signal is seen by the host processor which subsequently generates several wait states and the bus is held static (apart from /WE) until the signal is removed. Chip Select and SYNCH on the controller provide the chip enable function which is sampled internally on the falling edge of Ø1. CS and SYNCH must not change during its sample time and therefore U62(LS74) is used to synchronise CE onto the trailing edge of Ø2.



* THE EXACT TIMING IS DEPENDENT ON PHASE DIFFERENCE BETWEEN REFCLK (Ø3) AND THE CONTROLLER CLOCKS.

FIGURE 5-4 BUS INTERFACE TIMING DIAGRAM

Normally the 9900 MOVB instruction is used to load and unload the controller registers. When data is transferred to a controller register, the 9900 will first read the register and then write to the same register. This situation results in the shortest cycle time possible between two consecutive accesses. The rising edges of SYNCH for the two cycles are seperated by 2 us, the minimum time acceptable to the controller is 1.5 us.

Although the controller IC presents 16 registers to the CPU, only 15 are utilized and the 16th is implemented in discrete logic around the controller. When the 16th register is accessed the data port on the controller must be disabled. This is achieved using U51(S472), a fusible link PROM which decodes the 16th address. The PROM also serves to decode the 9900 control signals and appropiately access the controller IC, the bubble select register or the two external status bits. The two status bits (status register #2) are returned by enabling U43(LS125).

TABLE 5-1 PROM (U51) ADDRESS LINE ASSIGNMENTS

Pin no	S472 address		Function
19	I	MSB	not used = 0
18	H		Controller Chip enable
17	G		Controller ready
16	F		not used
5	E		DBIN (from BUS)
4	D		A11 (from BUS)
3	C		A12 (from BUS)
2	B		A13 (from BUS)
1	A	LSB	A14 (from BUS)

TABLE 5-2 PROM (U51) DATA

This Table has been edited from the 9900 source used to generate the PROM.

```
0000  IDLE   EQU  >14              No operation
0001  BUSOUT EQU  >01  U51 pin 6   Enable output drivers
0002  BUSIN  EQU  >02  U51 pin 7   Enable input  drivers
0004  STENB  EQU  >04  U51 pin 8   Enable/  SR #2
0008  MODSTB EQU  >08  U51 pin 9   Strobe bubble select
0010  WRB    EQU  >10  U51 pin 11  WE/  on  controller
0020  DBIN   EQU  >20  U51 pin 12  DBIN on controller

001E  FFWR   EQU  BUSIN+STENB+MODSTB+WRB   Write to   reg #15
0011  FFRD   EQU  BUSOUT+WRB               Read from  reg #15
0006  CONTWR EQU  BUSIN+STENB              Write to  controller
0035  CONTRD EQU  BUSOUT+STENB+WRB+DBIN    Read from controller
0016  IN     EQU  BUSIN+STENB+WRB          Enable bus
```

Start Address	Finish Address	Reference Data
0 hex	7F hex	IDLE
80	8F	IN
90	9E	CONTRD
9F	9F	FFRD
A0	BF	IDLE
C0	CE	CONTWR
CF	CF	FFWR
D0	DE	CONTRD
DF	DF	FFRD
E0	1FF	IDLE

The controller write and read cycles are basically similar, see figure 5-4. In the read cycle, DBIN is applied to the controller at the same time as CS. In the write cycle, WE/ is applied to the controller only while ready is high. In both cycles the data bus drivers are enabled as soon as CS is applied to the controller; the direction of the drivers is determined by the state of DBIN. The PROM (U51) generates invalid outputs while the address bus is settling. Therefore NAND gate U32 is used to inhibit the bus drivers from inadvertently turning onto the data bus.

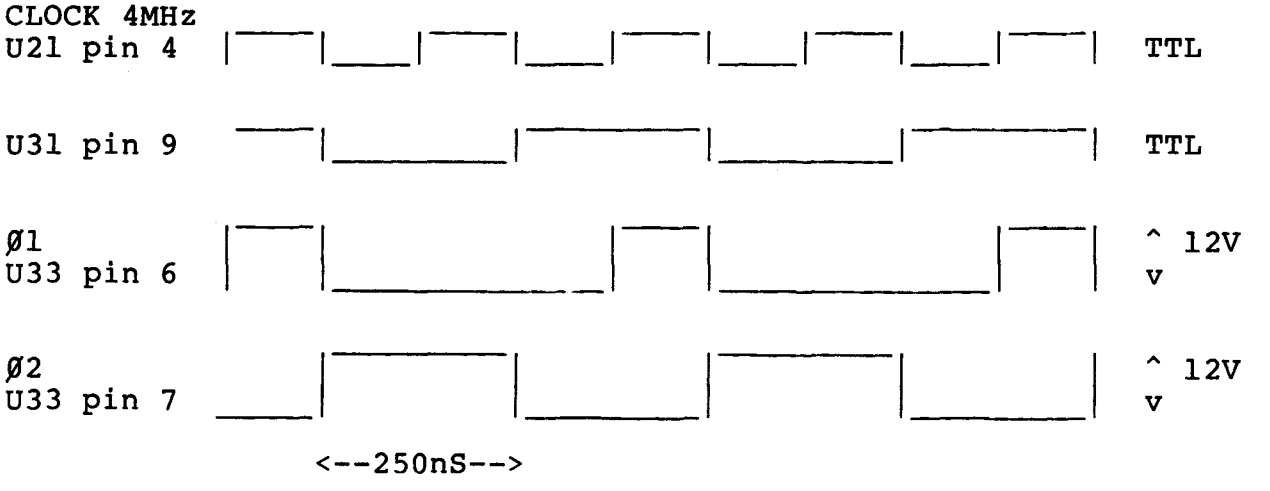
Irrespective of the register accessed, the overall cycle timing is controlled by the controller IC. The controller drives the ready signal (pin 9 on the controller) high and on the following controller clock cycle it is driven low. (The ready signal only remains high for

one controller clock cycle even though CS may still be present on the device). The original rising ready edge takes pin 3 on U31/LS112 high and on the falling edge of /Ø3 (REFCLK- system bus) the ready signal is returned high. The processor samples ready once again (on falling edge of Ø1), /MEMEN is removed and the cycle is completed.

5.3 OSCILLATOR AND CLOCK DRIVER

The oscillator is crystal controlled using a conventional arrangement of two S04 gates with feedback. The controller I.C requires two clocks (Ø1 and Ø2) at 2 MHz. U31 divides the 4MHz clock down to 2 MHz which is used for Ø2, see Figure 5-5. U32 serves to gate Ø2 with the 4 MHz clock to generate Ø1. These TTL signals are boosted by clock driver U33 to produce clocks with a peak amplitude of 12V to drive the NMOS controller.

FIGURE 5-5 CLOCK GENERATOR TIMING DIAGRAMS



5.4 REDUNDANCY HANDLING CIRCUITRY

This circuitry performs two functions. It flags the controller on pin 33 (FUNCEN) to indicate that a defective loop is being addressed and during a write cycle it inhibits the generation of a bubble destined to lie in a defective loop.

A counter implemented using two halves of an LS393(U53) is incremented on every other field cycle (50 KHz). The counter is used to directly address a PROM (U63/S472). When the first bit of a new page is about to be written or read in the major loop, the redundancy counter is cleared. As each bit is generated or detected the counter

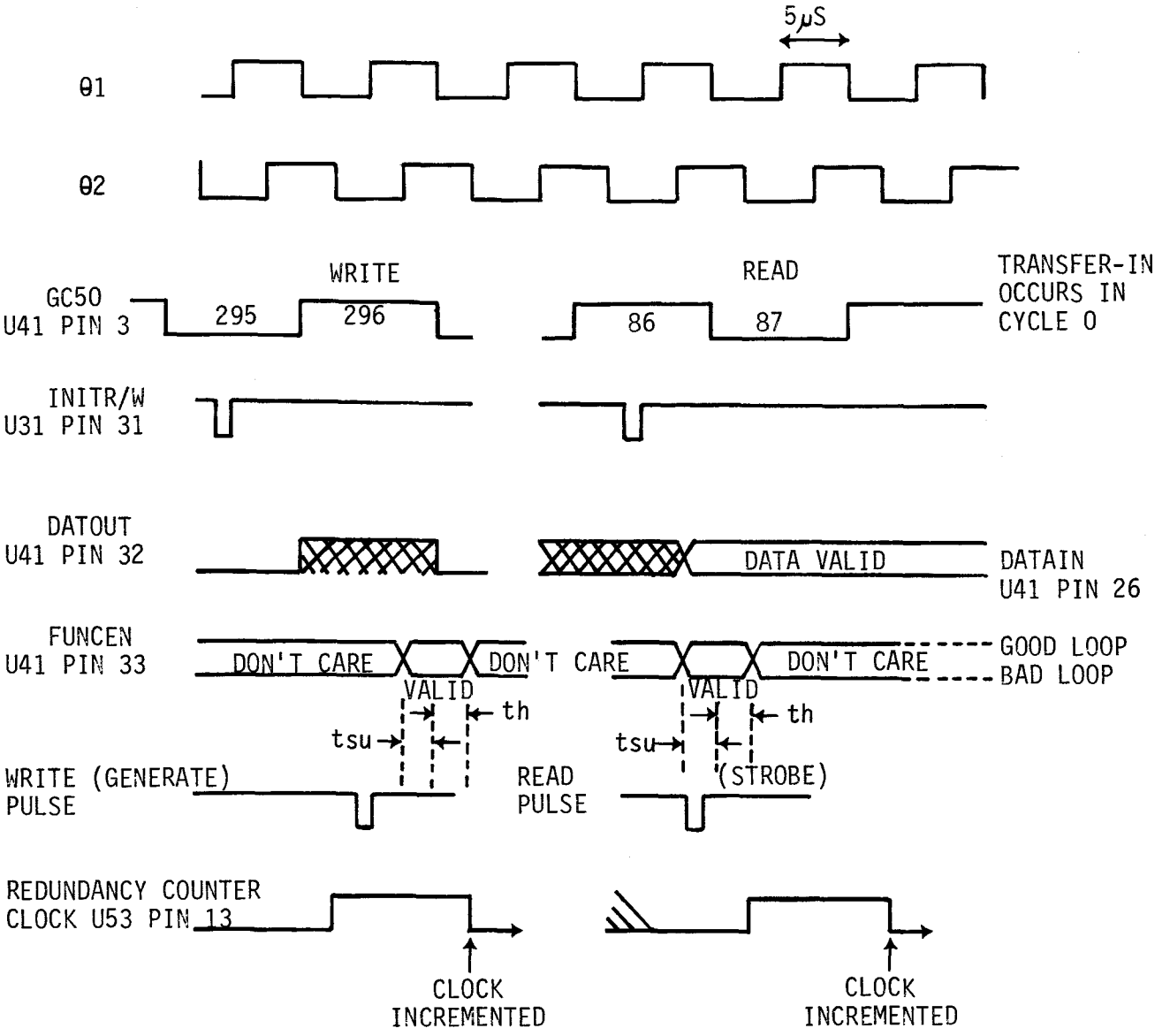


FIGURE 5-6 REDUNDANCY TIMING DIAGRAM

is incremented by one. When a defective loop is being addressed the output of the PROM goes high and this signal is seen by the controller via the demultiplexer U64/LS151. The demultiplexer is addressed by the bubble select register and it allows one of six programmed maps in the PROM to be read. When the controller sees a zero on FUNCEN (pin 33, U41) that particular field cycle is skipped and data is not moved into or from the controller's bubble data register. The counter is synchronized to the bubble operation by the INITR/W signal on pin 31 on the controller. Figure 5-6 shows the relationship between INITR/W and the redundancy PROM clock. The signal GC50 is also shown as a reference signal.

In the write operation the redundancy counter is cleared in the cycle preceeding the generation of the first bubble. The first and subsequent bubbles are generated while GC50 is high. At the end of the generate cycle, FUNCEN is sampled by the controller to test if the particular bit position was valid. (FUNCEN is sampled on the falling edge of GC50). If the data bit was valid then the data bit is shifted from the bubble data shift register in preperation for the next bubble cycle. If the data bit position had been invalid then generation would have been inhibited externally by NAND gate, U32. The controller never allows data to be generated into the alternate empty bit positions and therefore generate pulses must always be at least 20 uS apart.

In the read operation the redundancy counter is cleared in the same cycle that the first bubble is passing under the detector. The sense amplifier is strobed during every cycle that the redundancy counter clock is low. At the end of the read cycle, the sense amplifier data on DATAIN (pin 26) and FUNCEN is sampled on the falling edge of GC50. If FUNCEN is high, the data on DATAIN is shifted into the bubble data shift register.

5.5 BUBBLE SELECTION

Bubble selection is achieved by clocking in the address of the required bubble memory from the three least significant data lines into U34/LS174. The address is decoded by U35/LS138 and individual select lines are generated for each bubble memory module. U35 is also disabled by the power fail latch.

5.6 POWER FAIL LATCH

The power fail circuitry serves two functions at the moment that the power fail signal, PF/ becomes active. The present bubble operation is terminated and further operations are inhibited.

The main power fail latch utilizes half of U61/LS10. The latch is set at power up and by an active low PF/ signal. The latch is reset when the bubble select register is loaded. When PF/ becomes active, PWRBAD (U41 pin 29) is driven high on the controller. If the controller is operating in multi-page mode, the Page Counter is

effectively reduced to zero. Further transfer-out operations are terminated and any pages still resident in the major loop are returned into the minor loops. The controller finally returns the data in the minor loops back to the reference position. If the controller is operating in single page mode then PWRBAD has no effect on the operation because the single page cycle time is always 12.8 mS and it cannot be prematurely terminated without losing the reference position.

A secondary power fail latch utilizing one half of U54/LS112 is also set when the next bubble operation is started. Additional bubble memory data transfers are inhibited by deselecting any selected bubble memory on the leading edge of BSS. The bubble memory must not be deselected while the controller is active.

5.7 FUNCTION TIMING GENERATOR TIB0951 (LS361/U42)

The bubble memory system depends on the Function Timing Generator to initiate and synchronize the operation of various interface circuits.

The function timing generator is synchronized to the controller clocks by the rising edge of the run/stop signal (BSS) at the start of a new bubble operation. A counter and internal control flip-flops are used to control the starting, shifting and stopping sequences of the bubble field rotation. The counter divides each 10uS cycle into 40 intervals. The output of the counter accesses the decoder matrix on the rising edge of the clock pulse during each interval. The outputs of the decoder matrix are latched and then gated with signals from the controller to provide precise timing pulses to the coil drivers, sense amplifier and function driver.

The function timing generator also provides data protection during transient power conditions. An internal Vcc monitor network forces the coil drive outputs to the high impedance state when the Vcc supply drops to approximately 3.5 volts. The coil driver inputs sense this condition and prevent transient voltages from being applied to the coil inputs of the bubble memory.

5.8 FUNCTION DRIVER TIB0861 (75381)

The Function Driver converts the TTL logic pulses created by the function timing circuit into the current pulses required to operate the bubble memory. The device consists of five input logic gates which control five internal current generators. The constant current outputs drive the transfer gate, the generator element and the replicate element of the bubble memory.

Because the resistances of the elements vary considerably from device to device constant current sinks are necessary. Futhermore the bubble generator threshold levels vary because of normal variations in the magnetic epitaxial-film process. The TIB0203S is sorted into

three bands of bubble generate current and any mixture of bands may be used on the one board. The generate function is particularly temperature dependent. The function of the generate element is to produce only one bubble per pulse. If the current is too low or the temperature is too low then the bubble may not form or may collapse at the end of the generate pulse. But if the temperature or the current is too high the statistical possibility of generating two bubbles becomes realistic. So temperature compensation is provided by a thermistor and another resistor connected to the function driver. The second resistor is changed to select the band of generate element. The generate element has a low resistance and care should be taken to never expose the element to any high currents except in normal operation. This element will fuse quickly under adverse conditions.

5.9 SENSE AMPLIFIER TIB0833 (75285/X1075)

The passage of a magnetic bubble under the permalloy chevron detector array causes a reduction in electrical resistence of the detector element. A current passed through the detector can thus produce a corresponding change in voltage in a load resistor. The magnetic bubble is not the only changing magnetic field that affects the detector. The two drive coils in the bubble memory package also impose a signal upon the detector element.

The sense amplifier has a current mirror type input which supplies bias current to the detector resistors and a high impedance input to prevent loading of the bubble signal. The sense amplifier detect threshold is set to between 3 and 5 mV. Typical bubble signals are in the range of 8 mV.

When the restore (R) signal is active low it is used to charge the internal capacitor for the detection circuit. When restore is in the high state the sense amplifier is enabled for bubble detect. While the restore signal is enabled, the strobe signal is used to clock the amplified bubble signal into a flip-flop, the output of which goes to a three-state output buffer. The output enable signal allows bussing of the outputs.

The TIB0833 sense amplifier has a potentiometer connected accross the input which allows the current in each detector to be balanced accurately. For optimimum results the pot should be readjusted if the bubble is replaced. To adjust the voltage place a sensitive meter across the two inputs to the sense ampliier and rotate the pot until the voltage is less than 1.0mV.

Both detectors in the TIB0203S bubble memory lie in consecutive bit positions in the detector track. Therefore to sense a bubble, one position must be filled and the other must be empty. If both positions are filled the sense amplifier will respond by outputting a ZERO and not a ONE.

5.10 COIL DRIVER TIB0801A (75382A)

The coil driver circuit is used to generate triangular current waveforms for the bubble memory coils (see Figure 5-7). The currents are generated by applying a step voltage pulse to the coil. The coil's inductance integrates the voltage into a current ramp. When the pulse is switched off, the current is commutated by two diodes (one half of the VSB53 diode array) and allowed to ramp down to zero current. At that time the oppposite polarity pulse is applied to the coil and the current continues to ramp in the opposite direction. The coil drive circuit develops a peak field strength of 50 Oersteds. Ideally the upward slope should equal the downward slope but in fact the downward slope is greater because different voltages are produced during the two time periods under consideration. To avoid the possibility of oscillation when the commutation current is dying away, the drive field periods are slightly overlapped.

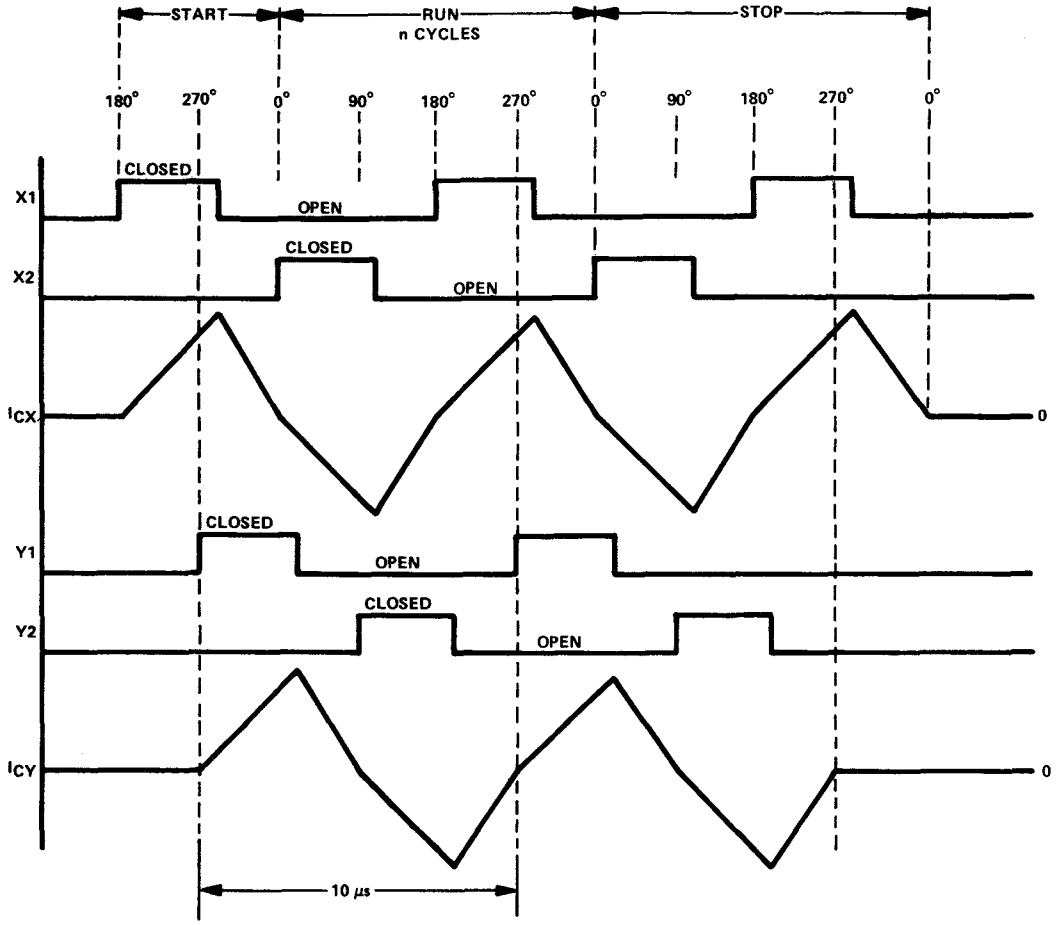


FIGURE 5-7. COIL DRIVE TECHNIQUE

Appendices

APPENDIX A TM 990 SYSTEM BUS PIN DEFINITION

The Pins marked with a '*' are used by the TM 990/210 board

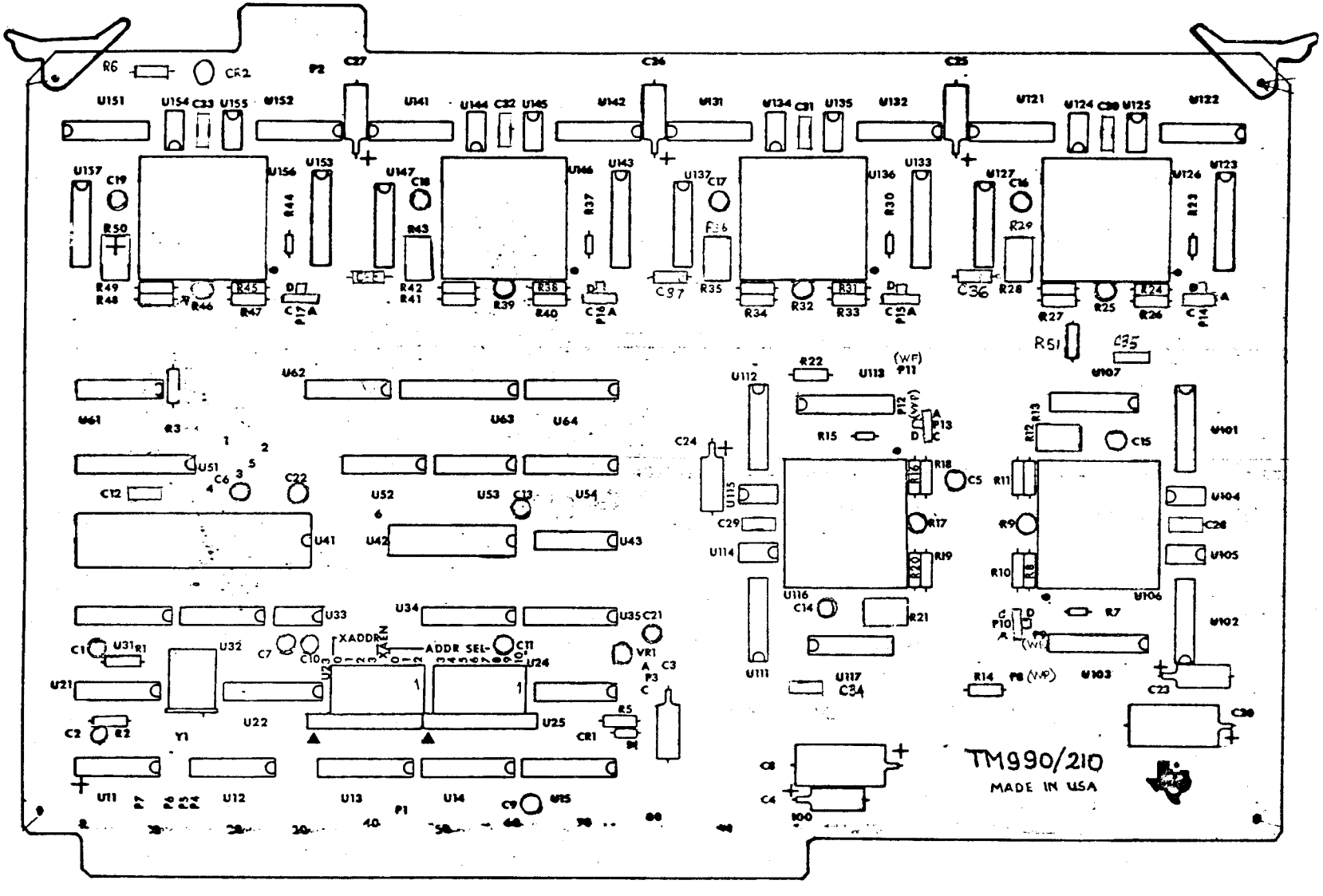
PIN		SIGNAL	GROUP
1/2	*	GND	Power/Ground
5	*	INT8/	
7	*	INT10/	
9		INT12/	
11	*	INT14/	Interrupt
13	*	INT2/	
15		INT3/	
17		INT5/	
19		IAQ	Control Bus
21	*	GND	
23	*	GND	Power/Ground
25	*	GND	
27	*	GND	
29		CRUIN	Cru Bus
31	*	GND	Power/Ground
33	*	D0	
35	*	D2	
37	*	D4	
39	*	D6	Data Bus
41		D8	
43		D10	
45		D12	
47		D14	
49		VAUX	Power/Ground
51		VBATT	
53	*	XA0	
55	*	XA2	
57	*	A0	
59	*	A2	
61	*	A4	Address bus
63	*	A6	
65	*	A8	
67	*	A10	
69	*	A12	
71	*	A14	
73	*	-12V	
75	*	+12V	
77	*	GND	
79	*	GND	
81	*	GND	
83	*	GND	
85	*	GND	
87		CRUCLK/	Cru Bus
89		GND	Power/ground
91		GND	
93		RESTART	Control/bus
95		GRANTOUT/	
97/98	*	+5V	Power/ground

PIN		SIGNAL	GROUP
3/4	*	+5V	Power/Ground
6		INT7/	
8		INT9/	
10		INT11/	
12		INT13/	Interrupt
14		INT15/	
16		INT1/	
18		INT4/	
20		INT6/	
22		BUSCLK/	(phase 1)
24	*	REFCLK/	(phase 3)
26		RESERVED	
28		RESERVED	
30		CRUOUT	Cru Bus
32		BUSY	Control Bus
34	*	D1	
36	*	D3	
38	*	D5	
40	*	D7	Data bus
42		D9	
44		D11	
46		D13	
48		D15	
50		VAUX	Power/Ground
52		VBATT	
54	*	XA1	
56	*	XA3	
58	*	A1	
60	*	A3	
62	*	A5	Address Bus
64	*	A7	
66	*	A9	
68	*	A11	
70	*	A13	
72		A15	
74	*	-12V	Power/Ground
76	*	+12V	
78		WE/	
80	*	MEMEN/	
82	*	DBIN	
84		MEMCYC/	
86		HOLDA	Control Bus
88	*	IORST/	
90	*	READY	
92		HOLD/	
94		PRES/	
96		GRANTIN/	
99/100	*	GND	Power/ground

APPENDIX B

PIN ASSIGNMENT ON P2 CONNECTOR

P2 PIN	SIGNAL
13	DISABLE/
15	BSS/
17	PF/
19	EXTRST/



COMPONENT LAYOUT FOR THE TM 990/210

NOTES:

1. INSTALL SOCKETS (ITEM 43) BEFORE INSTALLING ITEMS 20 AND 48
2. INSTALL SOCKETS (ITEM 42) BEFORE INSTALLING U51 AND U63 (ITEMS 2)
3. INSTALL SOCKET (ITEM 41) BEFORE INSTALLING U41 (ITEM 2)
4. INSTALL P10 AND P13 THRU P17 PER VIEW A-A
5. INSTALL CLIP (ITEM 45) PER VIEW B-B
6. -1, -2 AND -3 ASSY.
7. -2 AND -3 ONLY
8. -3 ONLY
9. A 74LS251 MAY BE SUBSTITUTED FOR THIS PART.
10. NOT TO BE INSTALLED UNTIL AFTER FLOW SOLDER.
11. ±20% OR BETTER TOLERANCE IS ACCEPTABLE

12. MASK HOLES PRIOR TO FLOW SOLDERING.
13. INSTALL ITEM 57 BEFORE INSTALLING U42 (ITEM 3)

REF DES	ITEM NO
U11	4
U12	4
U13	12
U14	12
U15	12
U21	17
U22	12
U23	46
U24	46
U25	14
U31	10
U32	14
U33	18
U34	9
U35	5
U41	2
U42	3
U43	7
U51	11
U52	16
U53	8
U54	10
U61	15
U62	13
U63	11
U64	6
U101	21
U102	21
U103	22
U104	24
U105	24
U106	20
U107	23
U111	21
U112	21
U113	22
U114	24
U115	24
U116	20
U117	23
U121	21
U122	21
U123	22
U124	24
U125	24

REF DES	ITEM NO
U126	20
U127	23
U131	21
U132	21
U133	22
U134	24
U135	24
U136	20
U137	23
U141	21
U142	21
U143	22
U144	24
U145	24
U146	20
U147	23
U151	21
U152	21
U153	22
U154	24
U155	24
U156	20
U157	23
C1	37
C2	35
C3	39
C4	39
C5	36
C6	36
C7	36
C8	38
C9	36
C10	36
C11	36
C12	36
C13	36
C14	36
C15	36
C16	36
C17	36
C18	36
C19	36
C20	38
C21	36
C22	36

REF DES	ITEM NO
C23	39
C24	39
C25	39
C26	39
C27	39
C28	34
C29	34
C30	34
C31	34
C32	34
C33	34
C34	56
C35	56
C36	56
C37	56
CR1	40
CR2	53
P3	51
~~P4~~	~~58~~
~~P5~~	~~38~~
P4/5/6	58
P7	49
P8	49
P9	49
P10	4
P11	49
P12	49
P13	4
P14	4
P15	4
P16	4
P17	4
R1	31
R2	31
R3	32
R5	30
R6	55
R7	25
R8	26
R9	48
R10	29
R11	27
R12	28
R13	47
R14	30

REF DES	ITEM NO
R15	25
R16	26
R17	48
R18	29
R19	27
R20	28
R21	47
R22	30
R23	25
R24	26
R25	48
R26	29
R27	27
R28	28
R29	47
R30	25
R31	26
R32	48
R33	29
R34	27
R35	28
R36	47
R37	25
R38	26
R39	48
R40	29
R41	27
R42	28
R43	47
R44	25
R45	26
R46	48
R47	29
R48	27
R49	28
R50	47
R51	32
C38	56

REF DES	ITEM NO
RP1	33
RP2	33
Y	44
VR1	19
P18	49

REF	REF	REF	SA		
DES	DES	DES	NO		
1	1	1	58	65610-403 (BERG)	PINS, DOUBLE ROW, .10 INCH
1	1	1	57	2932202	SOCKET, 22 PIN
5	3	1	56	1802Z5U05A473M	CAP., .047 µF (VARADYNE)
1	1	1	55	RC07G181	RESISTOR, 180 OHM, ¼W, 5%
2	2	2	54	CP-26	EJECTOR (WALDEN ELEC.)
1	1	1	53	TIL220	LED (TEXAS INSTRUMENTS)
11	9	7	52	65474-001 (BERG)	MINI-JUMPER, .100 INCH
7	5	3	51	65500-403 (BERG)	PINS, SINGLE ROW, .10 INCH
6	4	2	50	65500-401 (BERG)	PINS, SINGLE ROW, .10 INCH
6	6	6	49	65500-402 (BERG)	PINS, SINGLE ROW, .10 INCH
6	4	2	48	TSP102G	THERMISTOR, 1K OHM, 2% (SPECTROL)
6	4	2	47	64W104	POTENTIOMETER, 100K OHM
2	2	2	46	435166-5	SWITCH, DIP, (AMP)
4	4	2	45	10-31-0440-0913	CLIP, THERMISTOR
1	1	1	44	MP040	CRYSTAL, 4MHZ, (CTS)
96	64	32	43	75540-002	SOCKET, PIN, MINI-SERT,BERG
2	2	2	42	2932002	SOCKET, DIP, 20-PIN
1	1	1	41	2934002	SOCKET, DIP, 40-PIN
1	1	1	40	1N4148	DIODE (TI)
7	6	4	39	150D226X0015B	CAP., 22 µF, 15V, 20%
2	2	2	38	150D686X0015R	CAP., 68 µF, 15V, 20%
1	1	1	37	AVX34181U0A100K	CAP., 10 PF, 100V, 10%
16	14	12	36	AVX3429050E104M	CAP., 0.1 µF, 50V, 20%
1	1	1	35	AVX3418050E103M	CAP., 0.01 µF, 50V, 20%
6	4	2	34	AVX3430050E105M	CAP., 1.0 µF, 50V, 20%
2	2	2	33	750-101-102	RES., SIP, 1K OHM, (CTS)
2	2	2	32	RC07G102	RES., 1K OHM, 5%, ¼W
2	2	2	31	RC07G821	RES., 820 OHM, 5%, ¼W
3	3	3	30	RC07G472	RES., 4.7K OHM, 5%, ¼W
6	4	2	29	RN55D2742F	RES., 27.4K OHM, 1%, ¼W
6	4	2	28	RN55D4121F	RES., 4.12K OHM, 1%, ¼W
6	4	2	27	RN55D2001F	RES., 2K OHM, 1%, ¼W
6	4	2	26	RN55D3740F	RES., 374 OHM, 1%, ¼W
6	4	2	25	RN55D1001F	RES., 1K OHM, 1%, ¼W
12	8	4	24	VSB53	DIODE ARRAY, SCHOTTKY
6	4	2	23	TIB0833N	IC, SENSE AMPLIFIER, (X1075)
6	4	2	22	TIB0861N	IC, FUNCTION DRIVER, (75381)
12	8	4	21	TIB0801ANE	IC, COIL DRIVER, (75382A)
6	4	2	20	TIB0203S	IC, BUBBLE MEM., 92K SHRINK
1	1	1	19	MC79L05 ACP	IC, REGULATOR, -5V, ±5%
1	1	1	18	SN75361 AP	IC, TTL TO MOS CONV, DUAL
1	1	1	17	SN7404N	IC, INVERTER, HEX
1	1	1	16	SN74LS04N	IC, INVERTER, HEX
1	1	1	15	SN74LS10N	IC, GATE, TRIPLE, 3 IN NAND
2	2	2	14	SN74LS00N	IC, GATE, QUAD, 2 INPUT NAND
1	1	1	13	SN74LS74N	IC, FLIP-FLOP, D, DUAL
4	4	4	12	SN74LS85N	IC, COMPARATOR, 4 BITS
2	2	2	11	SN74S472N	IC, PROM, 8 X 512
2	2	2	10	SN74LS112N	IC, FLIP-FLOP, J/K, DUAL
1	1	1	9	SN74LS174N	IC, FLIP-FLOP, D, HEX
1	1	1	8	SN74LS393N	IC, COUNTER, DUAL, 4 BITS
1	1	1	7	SN74LS125N	IC, BUFFER, 3-STATE, QUAD
1	1	1	6	SN74LS151N	IC, DATA SEL/MUX, 8 LINES
1	1	1	5	SN74LS138N	IC, DEMUX, 3 TO 8 LINES
2	2	2	4	SN74LS243N	IC, BUS TRANCEIVER, 4 BITS
1	1	1	3	TIB0951N(45361)	IC, FUNCT. TIM. GENERATOR
1	1	1	2	TIB0901N	IC, CONTROLLER
-1	1	1	1	B90101	PRINTED WIRING BOARD
-3	-2	-1	ITEM NO.	ID#	DESCRIPTION

B90101 ESD, TM990/210 SCHEMATIC

REV			
A	[illegible]		
B	[illegible]		
C	46429	W.C. Purdon	2-13-80
D	46[illegible]	[illegible]	2-13-80
E	46437	W.C. Purdon	3-13-80

MATERIAL:

PROCESS #:

NEXT ASSEMBLY

TOLERANCES UNLESS SPECIFIED
.XXX ± .010
.XX ± .020
.X ± .030
ANGLES ±1°
BREAK ALL BURRS AND SHARP EDGES
REMOVE ALL BURRS AND SHARP EDGES
ALL DIMENSIONS ARE IN INCHES

HOLE TOLERANCES

.0135 THRU .126 THRU	.125 + .001 - .001	.250 + .001 - .001	.361 THRU .500 + .002 - .001
.126 THRU .251 THRU	.250 + .002 - .001	.360 + .004 - .001	.501 THRU .750 + .003 - .001
.251 THRU .501 THRU	.500 + .003 - .001	.750 + .005 - .001	.751 THRU 1.000 + .010 - .001
			1.001 THRU 2.000 + .012 - .001

SURFACE ROUGHNESS

DRAFTSMAN: [illegible] DATE 1/15/80
DESIGNER: [illegible] DATE
PROGRAMMER: [illegible] DATE
APPROVED: [illegible] DATE
RELEASED: W.C. Purdon 2/13/80

TEXAS INSTRUMENTS
INCORPORATED
SEMICONDUCTOR OPERATIONS

CODE IDENTITY NUMBER 01295

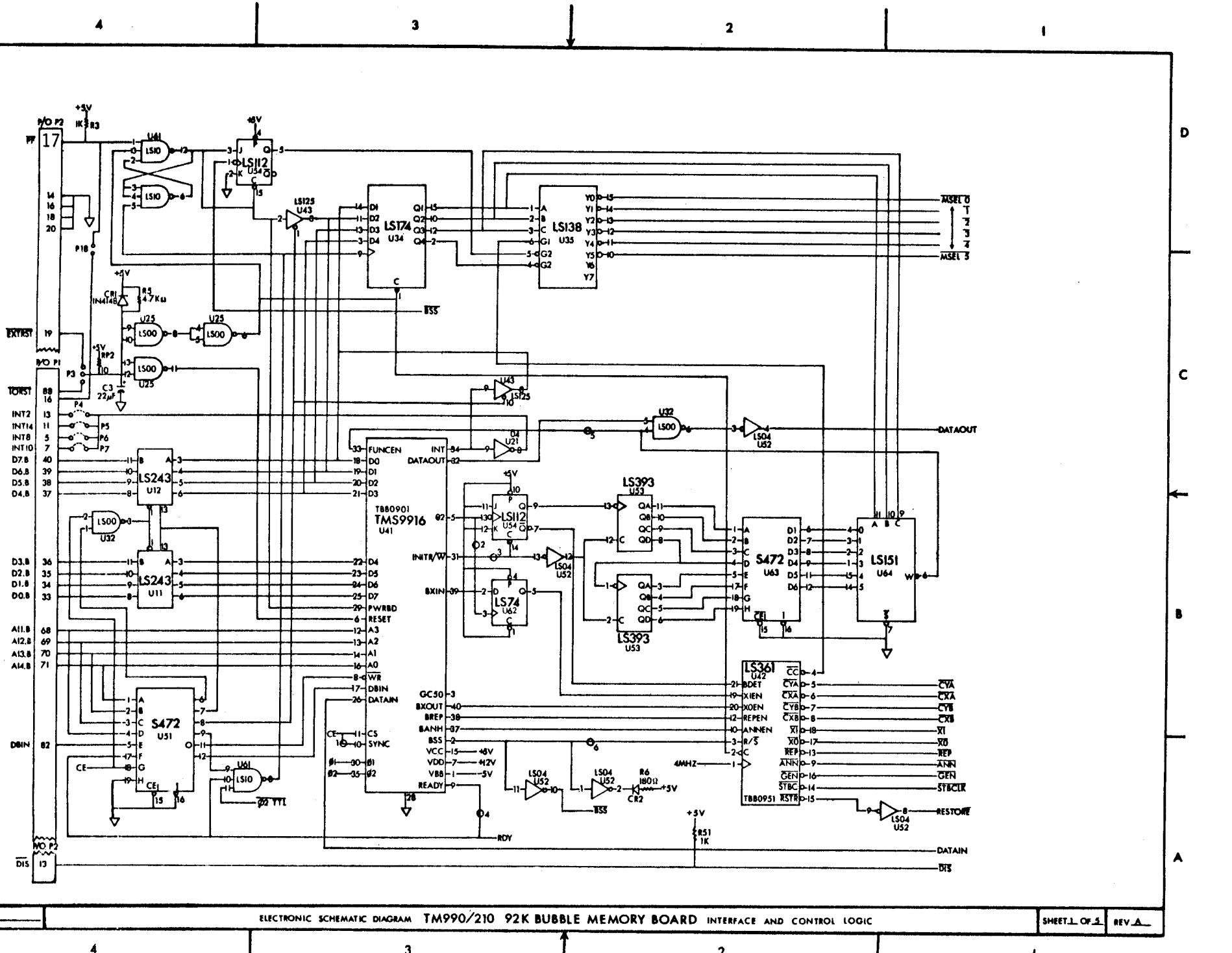
TITLE: TM990/210 ASSY. DRWG.

SCALE 1 1/2 SIZE C B90101 SHEET 1/2

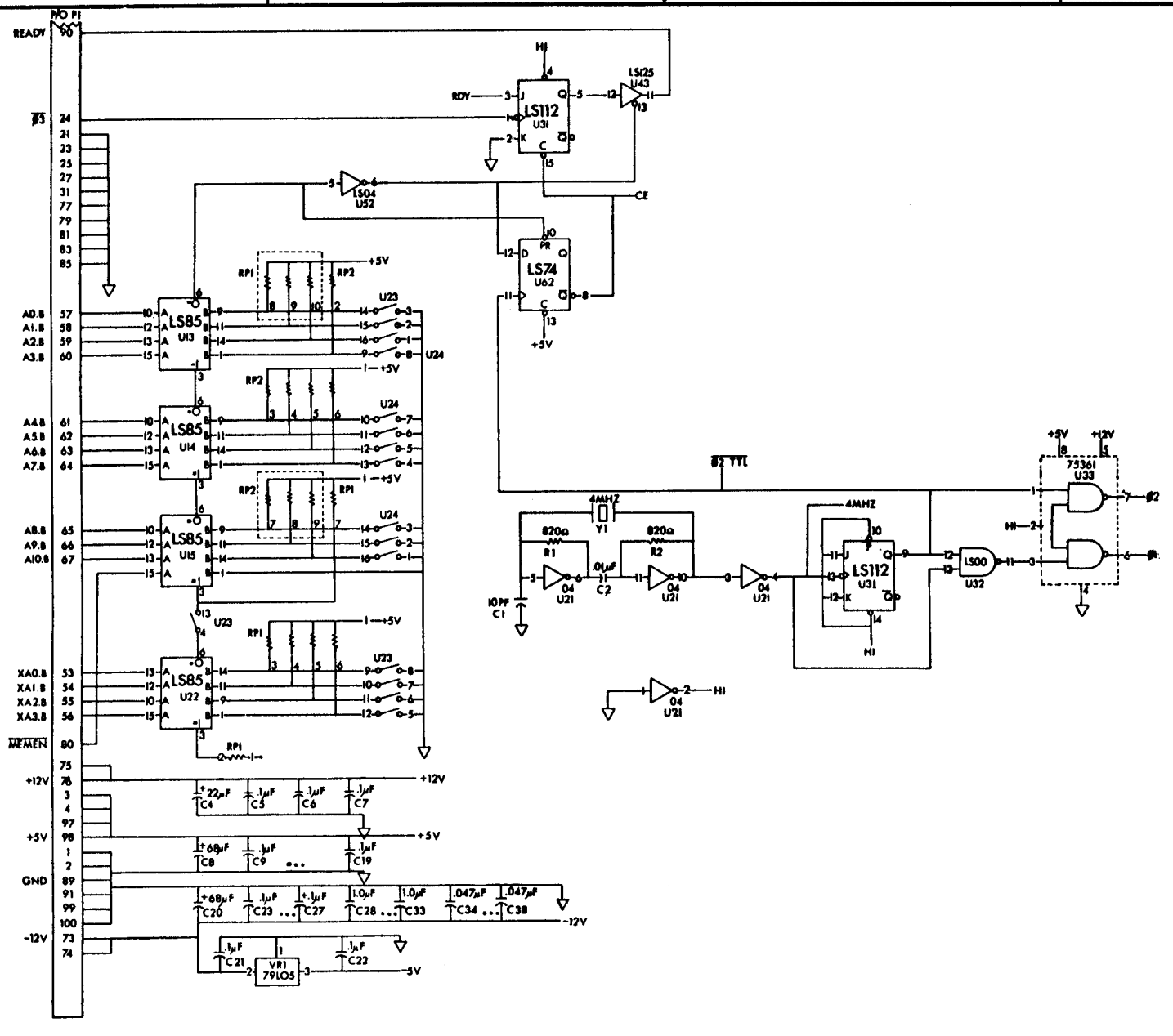
D-1

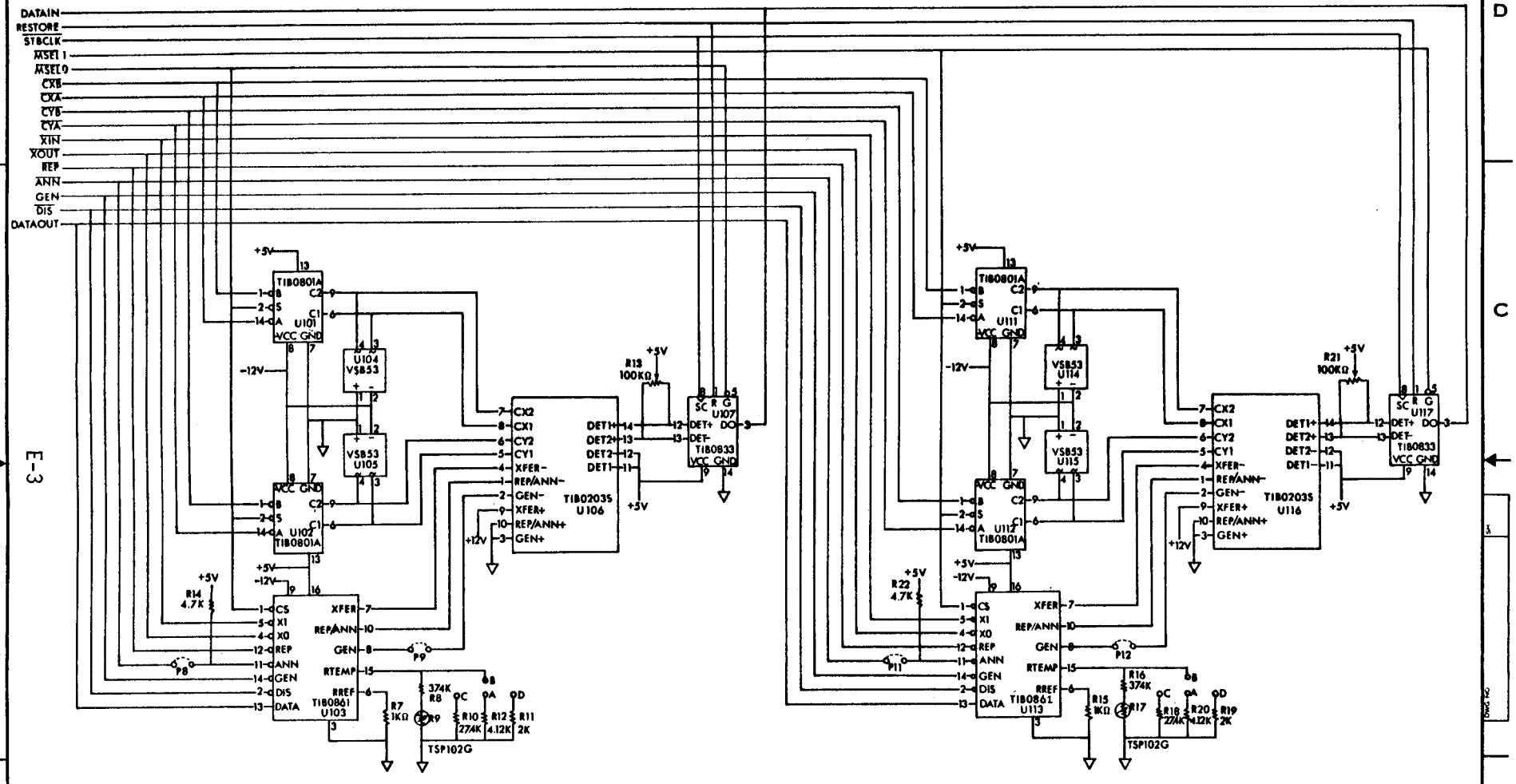
SHEET

SIZE C

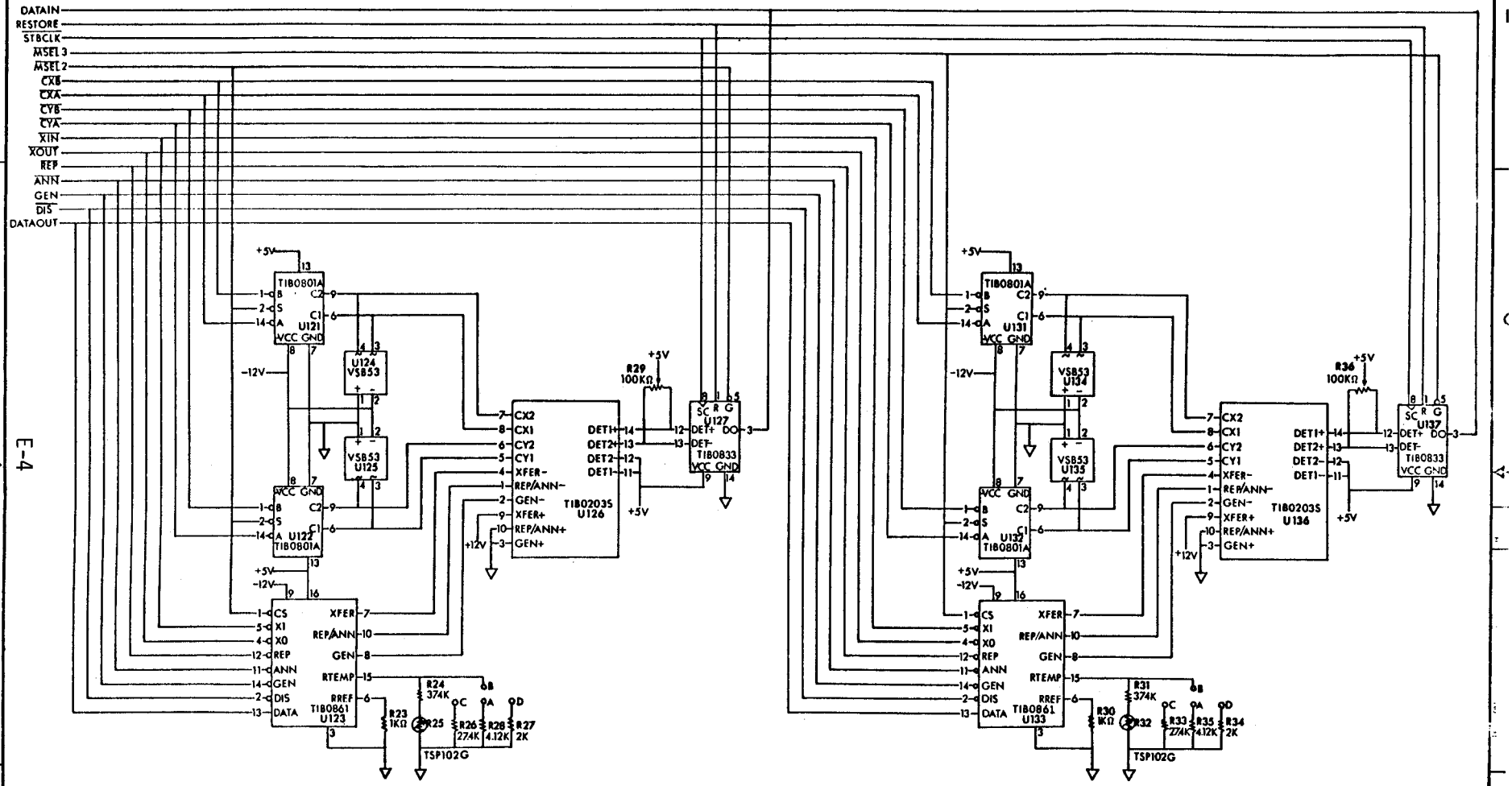


ID NO.	ELECTRONIC SCHEMATIC DIAGRAM TM990/210 92K BUBBLE MEMORY BOARD INTERFACE AND CONTROL LOGIC	SHEET 1 OF 1	REV A

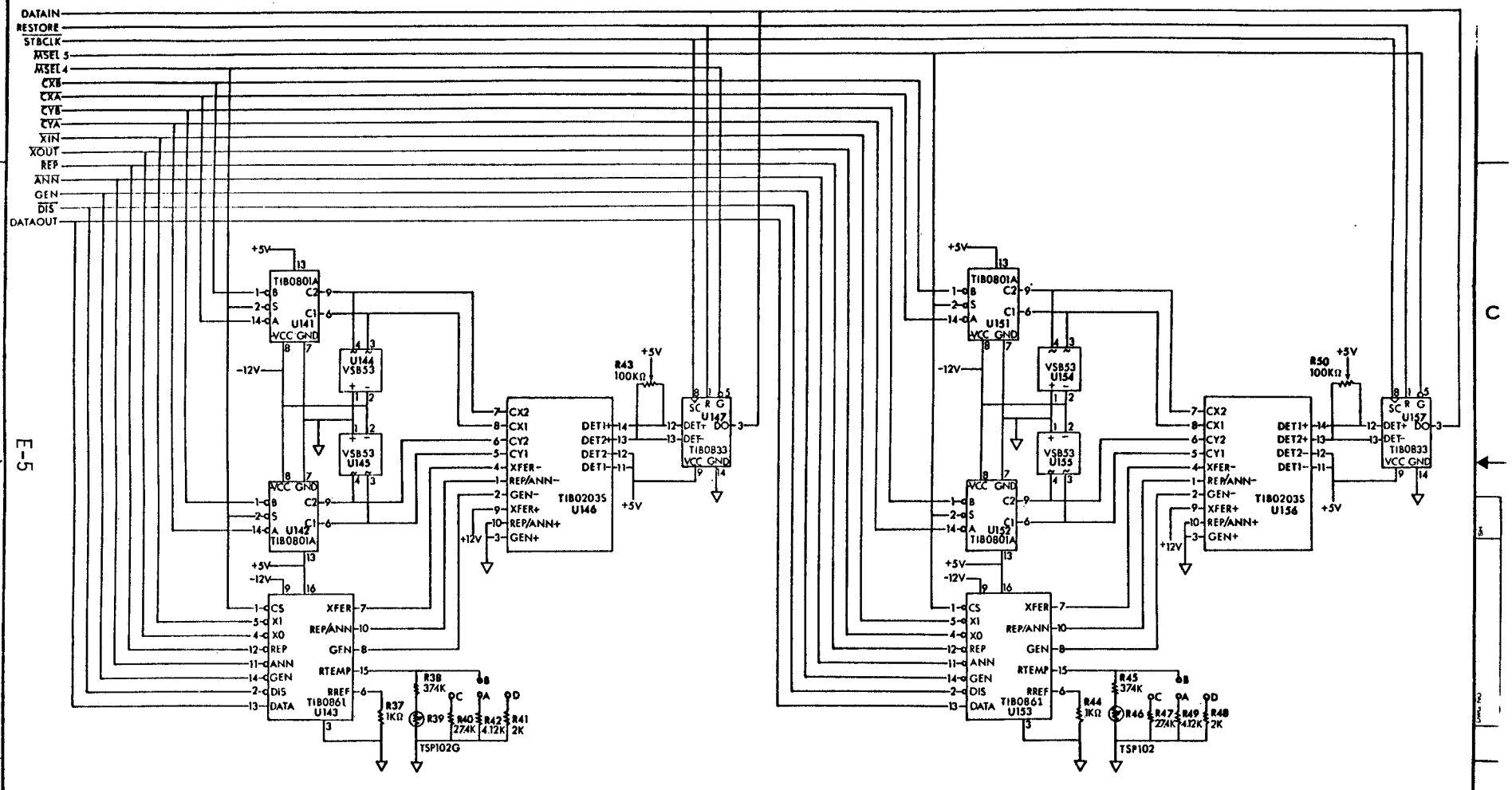

P/O P1
READY
RDY
LS112 U31
LS125 U43
LS04 U52
CE
LS74 U62
+5V
RP1
RP2
LS85 U13
LS85 U14
LS85 U15
LS85 U22
U23
U24
A0.B
A1.B
A2.B
A3.B
A4.B
A5.B
A6.B
A7.B
A8.B
A9.B
A10.B
XA0.B
XA1.B
XA2.B
XA3.B
MEMEN
+12V
GND
-12V
C4 22µF
C5 .1µF
C6 .1µF
C7 .1µF
C8 68µF
C9 .1µF
C19 .1µF
C20 68µF
C23 .1µF
C27 .1µF
C28 1.0µF
C33 1.0µF
C34 .047µF
C38 .047µF
C21 .1µF
C22 .1µF
VR1 79L05
-5V
Ø2 TTL
4MHZ
Y1
820Ω R1
820Ω R2
.01µF C2
10PF C1
04 U21
LS112 U31
LS00 U32
75361 U33
Ø2
Ø1
HI



ID NO.	ELECTRONIC SCHEMATIC DIAGRAM	TM990/210 92K BUBBLE MEMORY BOARD	BUBBLE MEMORY UNITS	SHEET 3 OF 5	REV



ID NO.	ELECTRONIC SCHEMATIC DIAGRAM	TM990/210 92K BUBBLE MEMORY BOARD	BUBBLE MEMORY UNITS	SHEET 4 OF 5	REV G



P NO.	ELECTRONIC SCHEMATIC DIAGRAM — TM990/210 92K BUBBLE MEMORY BOARD — BUBBLE MEMORY UNITS	SHEET 5 OF 5	REV A

APPLICATION SUPPLEMENT

TO THE TM 990/210 USER'S MANUAL:

ACCESSING PAGE 640

When an access is made to any except the last page in the bubble memory (PSR value = 640), the controller generates a transfer-in enable with a nominal duration of 10 uS. Following an access to page 640, this transfer-in enable remains active until well into the next transfer. This may result in both a transfer-in and a transfer-out enable occuring simultaneously during this next page cycle, resulting, in some cases, in questionable data integrity. The following instructions offer simple and convenient solutions to prevent this occurence.

If the system is operated in the single page mode, then this issue can be addressed by either of two methods. One is that accesses to page 640 can be avoided altogether. The other, if the use of page 640 is desired or required, is that such access (read or write) to page 640 can be followed by issuing an initialize command to the controller. The bubble module should be de-selected while the initialization is performed.

If the system is operated in the multi-page mode, the situation is analogous to that above with one exception. It must be remembered that pages are accessed sequentially and automatically and that absolute page number 640 is actually in the sequential position 213 (hex D5) relative to 0. The issue of the transfer-in enable behavior described above only arises when the last page in a multi-page operation is this relative position 213 (absolute page 640). In this case, either of two approaches can be followed. The first is to test and recognize when this relative position 213 (absolute 640) is the last page accessed in the transfer, and then issue an initialize command to the controller following the transfer and after de-selecting the bubble module. The second approach is to increment by one the page count register in the controller for the case where the last page would otherwise be 213 (absolute 640). This results in the last page of a multi-page transfer never being 213.

MANUAL UPDATE

The following changes or additions have been made in this edition (CL540A)

PAGE NO.	CHANGE OR ADDITION
2-8	In this example, the base address (hex C000) should be set up as shown in Figure 2-2.
3-2	To execute TIBUB from TIBUG, the PC should be set to 1190 and not 1090 as stated.
3-5	Table 3-2 shows TIBUB's standard command set. TIBUB II does not support the Verify command. The following command should be added to TABLE 3-2: P Execute a Bootstrap Load operation.
3-15	Table 3-6 changed to show that the 'carry' status bit is also set by a 'Range error'. In the example shown below the table, a range error may be detected using the 9900 instruction 'JOC' and not 'JOV'.
3-16	The bootstrap load map on Table 3-7 changed to meet the recent file management standard. The checksum should be placed in bytes 20 and 21
4-6	On figure 4-1 the subaddress of the page size register changed to >18.
4-21	On figure 4-7 the minor loop size registers changed to 8 and 9.
B-1	The pin assignments, 15 and 17 on the P2 connector are shown reversed. The correct pinout is :

P2 SIGNAL	SIGNAL
13	DISABLE/
15	BSS/
17	PF/
19	EXTRST/

E-1	The circuit diagram changed to show that the Power Fail warning signal is connected to pin 17 and not pin 15.
ADDENDUM	Added: Application Supplement - Accessing Page 640.

(Back of Manual)