60476080

CONTROL DATA CORPORATION

CDC® LARGE MEMORY SUBSYSTEM

AT429, BA222

GENERAL DESCRIPTION
OPERATION
INSTALLATION AND CHECKOUT
THEORY OF OPERATION
DIAGRAMS
MAINTENANCE
PARTS DATA

HARDWARE REFERENCE/MAINTENANCE MANUAL

REVISION RECORD	
REVISION	DESCRIPTION
A (10/81)	Manual released class A.
B (02/82)	Manual revised; includes ECO DS022167, providing a revised spare parts list and a new LMS address PWA assembly drawing and logic diagram.
C (10/82)	Manual revised; includes ECO DS022223, providing corrected pull-up resistor values on the address logic diagram.
D (04/83)	Manual revised; includes ECO DS022276, removing the FCC EMI warning label which is not required in this manual. The FCC EMI tested/complies warning label has been included in the system-level maintenance manual.
E (07/84)	Manual revised to incorporate changes/additions requested by memorandums, numbers 41 and 42 dated July 1984.

Publication No.
60476080

REVISION LETTERS I, O, Q AND X ARE NOT USED

by Control Data Corporation
Printed in the United States of America

Address comments concerning this manual to:

Control Data Corporation
Publications and Graphics Div.
2401 North Fairview Avenue
St. Paul, Minnesota 55113

or use Comment Sheet in the back of this manual.

MANUAL TO EQUIPMENT LEVEL CORRELATION SHEET

This manual reflects the equipment configurations listed below.

EXPLANATION: Locate the equipment type and series number, as shown on the equipment FCO log, in the list below. Immediately to the right of the series number is an FCO number. If that number and all of the numbers underneath it match all of the numbers on the equipment FCO log, then this manual accurately reflects the equipment.

EQUIPMENT TYPE	SERIES	WITH FCOs	COMMENTS
AT429-A	01		
BA222-A	02		

New features, as well as changes, deletions, and additions to information in this manual are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

PAGE	REV	PAGE	REV	PAGE	REV
Cover	-				
Title Page	-				
ii	E				
iii/iv	D				
v/vi	E				
vii/viii	D				
ix	A				
1-1/1-2	A				
2-1	A				
3-1	A				
3-2	E				
3-4/3-5	A				
3-6	E				
4-1 thru 4-53	A				
5-1 thru 5-60	A				
5-61/5-62	C				
5-63 thru 5-79	B				
6-1 thru 6-32	A				
7-1 thru 7-4	A				
7-5/7-6	B				
7-7 thru 7-14	A				
7-15 thru 7-19	B				
A-1 thru A-18	A				
B-1	A				
Comment Sheet	E				
Mailer	-				
Cover	-				

PREFACE

This manual contains the theory of operation, diagrams and maintenance information for the CONTROL DATA® Large Memory Subsystem. Information presented in this manual is intended for use by maintenance personnel in training and in the field.

Logic diagrams are provided in this manual. The interconnecting, block, and timing diagrams presented herein are intended to supplement the logic diagrams of the memory and interface. Installation and checkout information is contained in section 3 of this manual and should be used in conjunction with the system installation manual.

Additional information on the memory and memory interface is found in the publications listed below:

Publication	Publication Number
CYBER 18 Computer Systems Central Processor Field Repair Guide	60475001
CYBER 18 Computer Systems Overview Manual	60475000
CYBER 18 Computer Systems with MOS Memory Installation Manual	96768360

CONTENTS

APPENDIXES

FIGURES

TABLES

GENERAL DESCRIPTION 1

The large memory subsystem (LMS) consists of a data interface card, an address interface card, a modified backpanel, and from one to four large memory array (LMA) PWAs. All cards are standard 11- by 14-inch (279- by 356-millimeter) printed wire assemblies (PWAs) and each requires one slot in the processor chassis. Two LMS memory banks may be coupled together to double total memory size.

The address interface supplies timing and control for the LMS and processes memory requests and addresses from the CPU and peripheral devices. The data interface provides the buffers for write data inputs and read data outputs.

The memory arrays contain 128K words (K=1024) of dynamic RAM, for a total of 512K possible memory words in a four-array bank. The word size is 18 bits, which includes a 16-bit data word, a parity bit, and a protect bit. The total memory capacity with two banks is 1024K (1M; M = 1,048,576) words.

The modification to the backpanel adds address and control lines necessary to allow communication within the LMS.

The LMS PWAs operate on system power, are cooled by fans located in the processor chassis, and operate under the same environmental conditions as the system.

SYSTEM BLOCK DIAGRAM

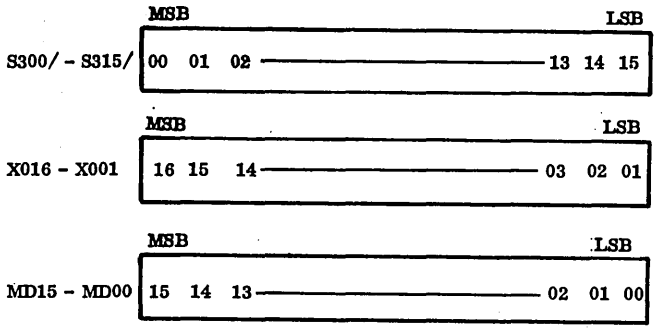
A block diagram of the LMS as part of an operational system is shown in figure 1-1.

MEMORY SIZE AND CAPABILITY

The LMS is configured as a one-bank, two-port memory that may be expanded to create a two-bank, three-port memory. Each bank provides from 128K to 512K 18-bit words of storage. The two banks are referred to as internal and external banks. Each bank may have only one memory request active at one time. However, each bank operates independently, thus permitting concurrent access by different ports to each bank. Each of the three ports in each bank (CPU, DMA, and external) has independent address, data, and control paths and may request memory independently of any operation currently in progress on another port.

TERMINOLOGY

The bit position terminology used in this manual is such that direct translation to a bit block is possible. Some typical bit statements and their corresponding blocks are as follows:



True/false terminology for signal state is used throughout this manual. A signal name with a slash (/) at the end (for example EXTSEL/) indicates the true (or active) state of this signal to be a TTL low (or zero) and the false (or inactive) state to be a TTL high (or one). A signal name without a slash (/) at the end (for example EXTSEL) indicates the true state of this signal to be a TTL high (or one) and the false state to be a TTL low (or zero).

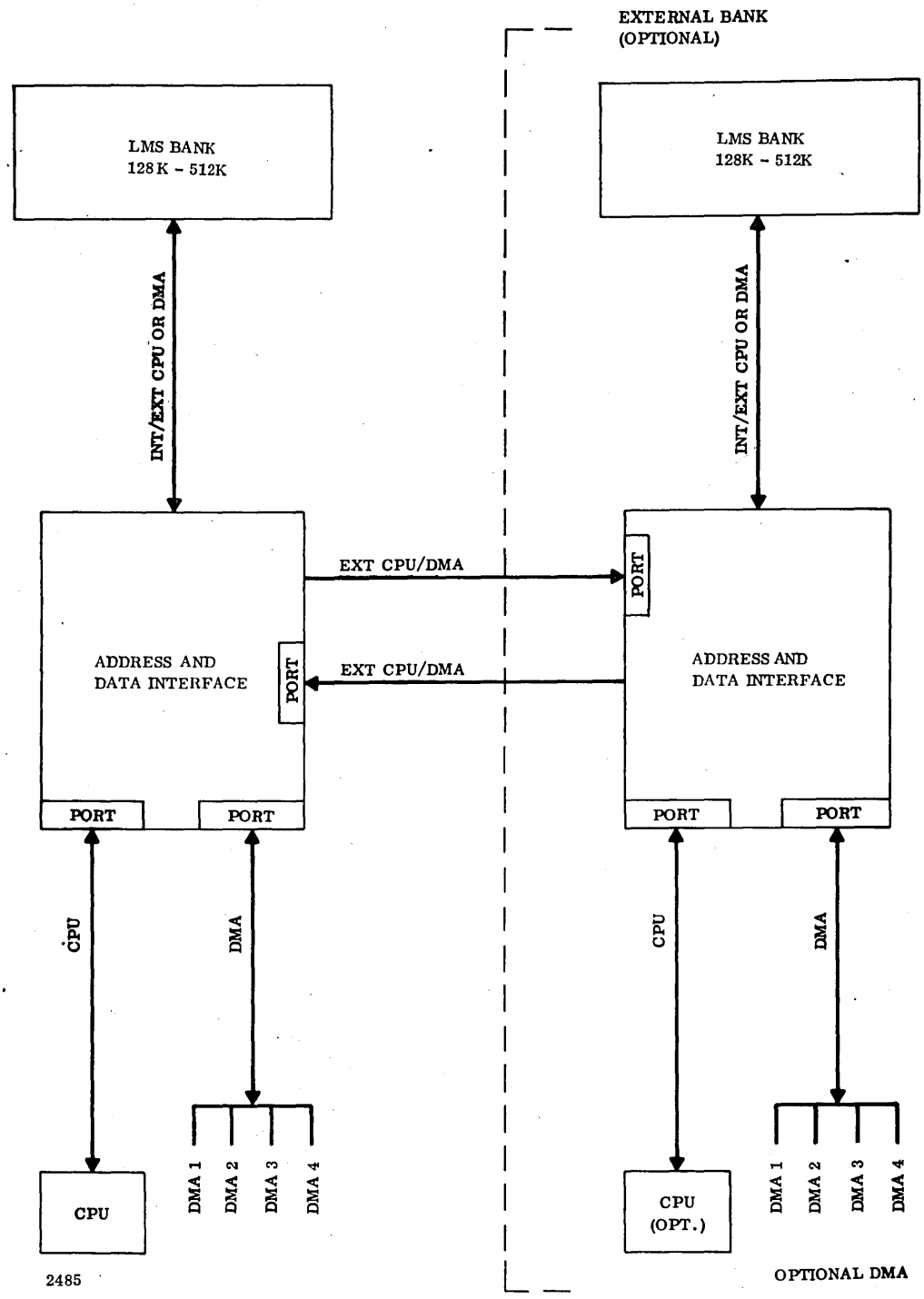


Figure 1-1. Large Memory Subsystem Configuration

OPERATION 2

There is no operating or programming information applicable to this equipment.

The large memory subsystem (LMS) PWAs replace the memory interface and memory array PWAs originally supplied with a CYBER 18 CPU. The LMS PWAs interface with the CPU via the PWA backpanel edge connector. Additional signal interface between the LMS PWAs and the CPU is accomplished via removal (table 3-1) and installation (table 3-2) of wire-wrapped backpanel connections on the CPU chassis backpanel.

NOTE

The CYBER 18 installation procedure assumes that the LMS is not already installed in the CYBER 18 CPU chassis.

CYBER 18 INSTALLATION

NOTE

Perform ODS diagnostic tests on memory, processor, and all DMA controllers to ensure that the system is operating properly prior to LMS installation.

UNPACKING

Before beginning the LMS installation, remove the LMS PWAs and wired panel assembly (backpanel) from the shipping container and inspect them for damage. If any assembly shows signs of damage, do not proceed with the installation. If the LMS assemblies are damage-free, perform the removal procedure.

INTERFACE AND ARRAY REMOVAL

Before the LMS PWAs can be installed in the CYBER 18 CPU chassis, all MOS memory array PWAs and memory interface PWAs residing in slots V, W, X, Y, Z, and AC must be removed. Remove the PWAs as follows:

1. Turn the system OFF.
   a. Press MASTER CLEAR
   b. Remove power from all peripherals.
   c. Set CPU AC POWER switch to off.
2. Remove right side and right rear panels. (On a dual-CPU system also remove the left side and left rear panels.)
3. Remove the cover from the CPU chassis. (On a dual-CPU system, remove the cover from both CPU chassis.)
4. Remove any and all PWAs from slots V, W, X, Y, Z, and AC (in both CPUs in dual-CPU systems).
5. Return the MOS memory array PWAs and the memory interface PWAs as directed by the customer or ESS management.

NOTE

If an LMS data PWA has not been supplied then the memory interface data PWA removed from slot V must be modified for LMS use. Set it aside for later modification.

BACKPANEL REMOVAL

Label and remove all peripheral cables, connectors, and adapters from the CPU chassis backpanel. (On a dual-CPU system, label and remove all connectors and adapters from both CPU chassis backpanels.)

Unplug all PWAs from the CPU backpanel (on a dual-CPU system, unplug all PWAs from both CPU backpanels).

Remove the peripheral screws from the backpanel and set aside for later use. Carefully remove the backpanel from the system. (On a dual-CPU system, remove the wire-wrap backpanels from both CPU chassis.)

WARNING

The removal and installation of a backpanel should only be performed by qualified personnel.

If a new packpanel is not supplied, modify the existing processor backpanel as follows:

1. Remove the wires listed in table 3-1.
2. Add the wires listed in table 3-2 at the wrap levels indicated. All wires should be routed in a direct path between points of connection and as close as possible to the backpanel. Keep point to point wiring as tight as possible.
3. Type part number 88919628 revision 01 on a blank adhesive label (part number 39397102) and affix the label to the backpanel over the old part number (located beside slot "AC").

MEMORY INTERFACE DATA PWA MODIFICATION

If an LMS data PWA has been supplied, skip this operation and go to LMS backpanel installation.

WARNING

Modification of the data PWA requires clad cuts and soldered wire jumper additions. This should be performed only by qualified personnel.

Obtain the data interface PWA previously set aside.

Using a sharp-bladed instrument (for example, an exacto knife), cut the clad as shown in figure 3-1.

TABLE 3-1. DELETE WIRES

Signal Name	From		To		Level
	Slot	Pin	Slot	Pin	
CHARINPUT/	AA	47	A	47	2
CHARINPUT/	C	47	D	47	2
CHARINPUT/	E	47	F	47	2
CHARINPUT/	G	47	H	47	2
CHARINPUT/	J	47	K	65	2
CS/	W	237	V	239	2
CHARINPUT/	C	47	A	47	1
CHARINPUT/	E	47	D	47	1
CHARINPUT/	G	47	F	47	1
CHARINPUT/	J	47	H	47	1
GROUND	W	87	W	101	1

TABLE 3-2. ADD WIRES

Signal Name	From		To		Level
	Slot	Pin	Slot	Pin	
CHARINPUT/	C	47	E	47	1
CHARINPUT/	F	47	J	47	1
CPUMAB20/	V	279	W	218	1
DMAMAB19/	D	47	G	47	1
DMAMAB19/	H	47	W	230	1
DMA1MAB20/	H	259	W	270	1
DMA2MAB20/	G	259	W	273	1
DMA3MAB20/	D	259	W	276	1
DMA4MAB20/	A	259	W	278	1
E64KNO-2/	Y	7	Y	207	1
E64KNO-3/	Z	7	Z	207	1
E64KNO-4/	AC	7	AC	207	2
RAS64KNO-2/	Y	27	Y	227	2
RAS64KNO-3/	Z	27	Z	227	2
RAS64KNO-4/	AC	27	AC	227	2
REFRESH	V	239	W	229	1
CHARINPUT/	AA	47	C	47	2
CHARINPUT/	E	47	F	47	2
CHARINPUT/	J	47	K	65	2
DMAMAB19/	A	47	D	47	2
DMAMAB19/	G	47	H	47	2

NOTE

Upon completion of adds and deletions in tables 3-1 and 3-2, this CPU backplane has 20-bit addressing capability for DMA devices (that is, GB145, FJ448 and FJ127). To enable 20-bit addressing for these equipments, refer to the appropriate sections of CYBER 18 Computer System with MOS Memory Installation Manual (see preface for the publication number).

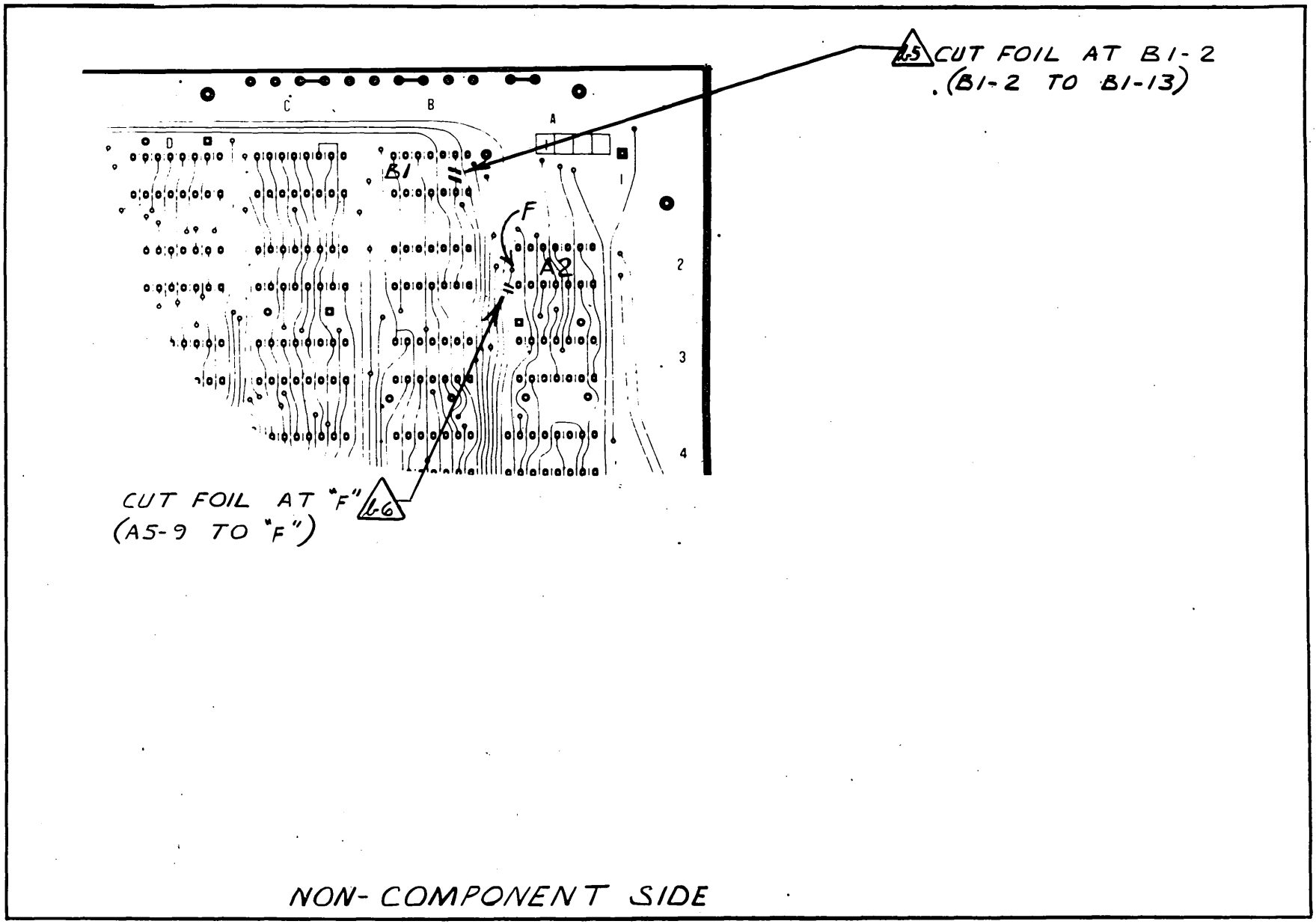


Figure 3-1. Memory Interface Data PWA Modification (Sheet 1 of 3)

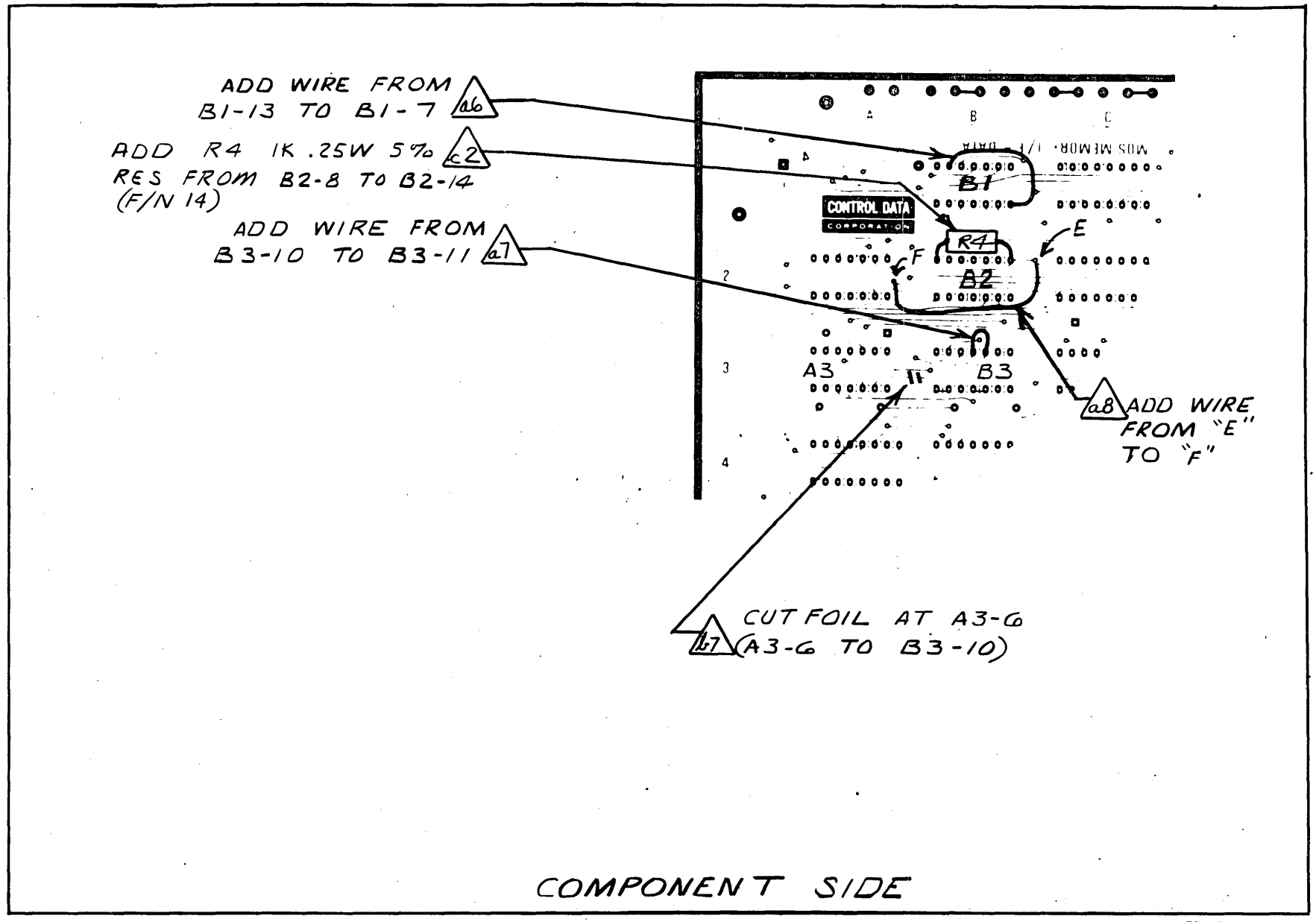


Figure 3-1. Memory Interface Data PWA Modification (Sheet 2 of 3)

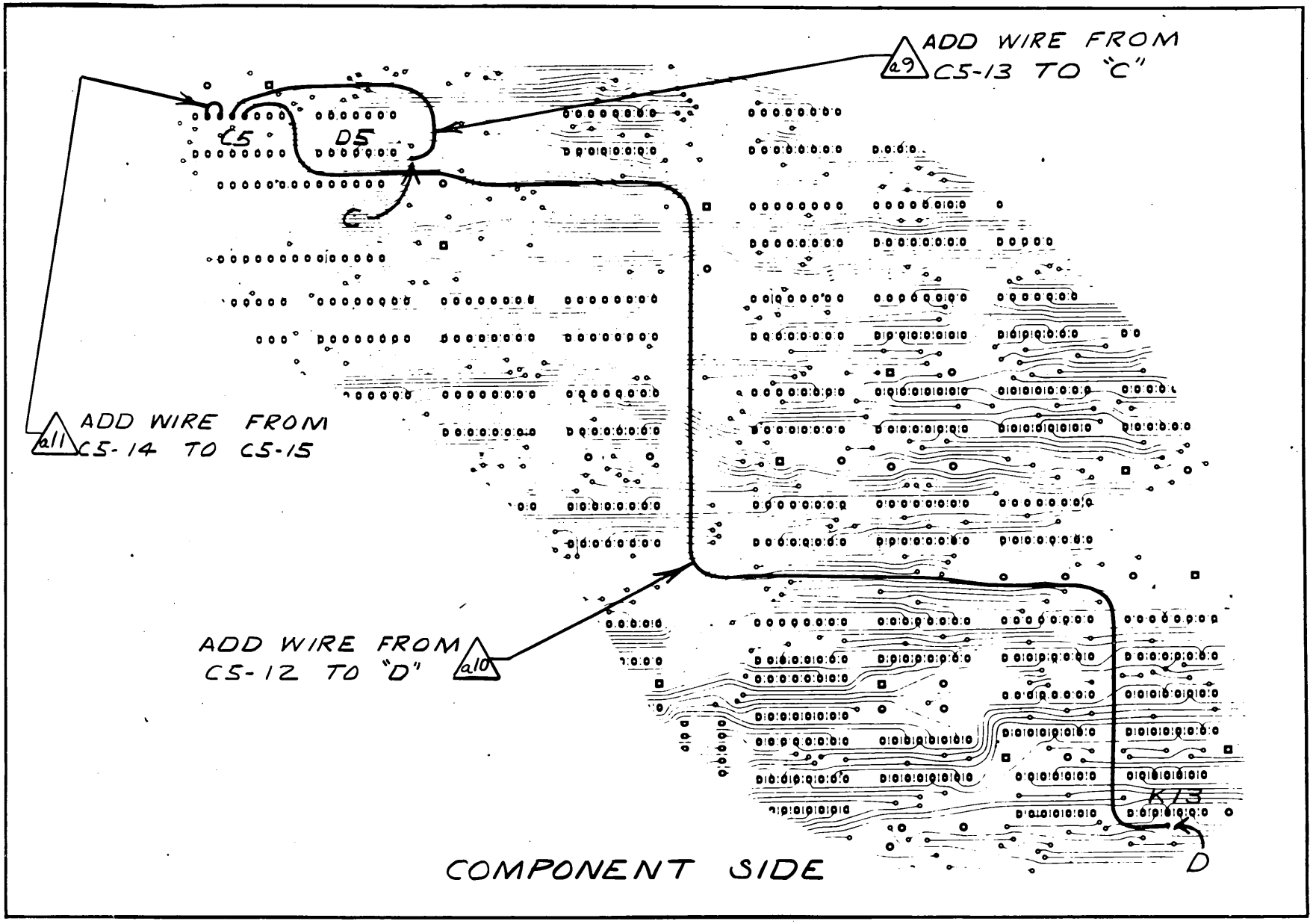


Figure 3-1. Memory Interface Data PWA Modification (Sheet 3 of 3)

Solder the white, 30 ga. wire supplied as shown in figure 3-1.

Re-check all clad cuts and wire-jumper additions.

NOTE

In a dual-CPU system, two data interface PWAs must be modified.

LMS BACKPANEL INSTALLATION

Install the new or modified LMS backpanel in the CPU chassis, adjusting it for proper PWA insertion. Install the previously removed peripheral screws in the new backpanel.

NOTE

The difference between a standard CYBER 18 backpanel and an LMS backpanel is the deletion, from the standard panel, of the wires listed in table 3-1 and the addition of the wires in table 3-2.

In a dual-CPU system repeat the above for the second CPU chassis.

LMS PWA INSTALLATION

Install the supplied (or modified) LMS data PWA in slot V of the CPU chassis.

Install the supplied LMS address PWA in slot W of the CPU chassis.

Install each LMS array PWA required by the system in the CPU chassis, starting with slot X and proceeding to slot Y, Z, and AC as necessary.

In a dual-CPU system repeat the above for the second CPU chassis.

Re-install all PWAs, backpanel peripheral connectors, cables, and assemblies previously removed.

CHECKOUT

After completing the installation of the LMS assemblies, perform the applicable ODS tests in accordance with the DDLTs found in section 6. Refer to appendix A for reference data associated with LMS ODS.

After successful completion of any and all diagnostic tests, replace the chassis covers and cabinet side panels. Return the system to on-line operation.

SWITCHES

Switch S10 at the top-front edge of the address PWA provides a status line into the control logic indicating External Bank is present. Switch S10-1 must be ON (CLOSED) in a dual CPU system. With a single CPU system, switch S10-1 is OFF (OPEN). Switches S10-2 through S10-4 are unused.

This section contains a functional description of the LMS. The description is related to the diagrams in section 5. Other illustrations are presented throughout this section.

NOTE

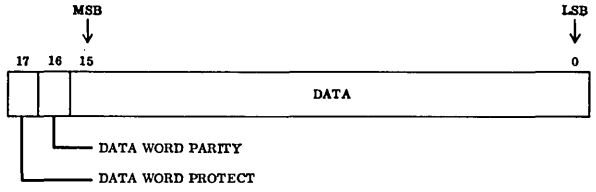
Refer to section 1 for information on Terminology.

ADDRESS, CONTROL, AND DATA FLOW

The LMS is a one-bank, two-port memory that may be expanded to create a two-bank, three-port memory. It consists of a data interface, an address interface, and from one to four memory arrays. The three ports permit three independent data, address, and control paths to memory. Any port may request a memory cycle at any time, but only one bank reference can occur at a time. Priority selection and multiplexing of concurrent requests is accomplished by the LMS. One port is used by the CPU, the second port is shared by four DMA subports, and the third port is used as a dual unidirectional interface with a second LMS. The simplified block diagram in figure 4-1 shows the address and data paths.

After priority selection, addresses from the CPU, DMA, or external ports are decoded to determine a word location in an LMA. Either the CPU or a DMA device may request an external bank address according to the address formats described later in this section.

The memory for each bank consists of from one to four LMA PWAs, each containing 128K words of storage, for a total of from 128K to 512K words. Each word contains 18 bits, according to the data format shown below. The word parity and protect bits are described later in this section.



A memory cycle can consist of a read operation, a write operation, or both a read and a write operation. Dynamic memory circuits also require a refresh operation on each row address at least once every 2 milliseconds to retain stored data.

During a read operation, data leaves the addressed location in the selected array to the data interface. There the read data is checked for parity and then directed to the requesting port. If an external bank was selected, data is accessed from the external port of the external bank by the external port of the internal bank, and then directed to the requesting CPU or DMA port.

During a write operation, the appropriate parity bit is generated for write data from the requesting port. The resulting 18 bits are sent to the selected array.

A split cycle (read followed by a write operation) is employed when requested by the CPU requesting read/modify/write cycle; or whenever a normal protected write cycle is requested; or when a character (half word) write cycle is requested. On a requested read/modify/write cycle, LMS activity stops between read and write portions of the cycle except for refresh requests. On a protected write cycle, the write portion of the cycle may be aborted if a protect fault is encountered. On a character write cycle, one of the characters read from memory is combined with the new character to be stored in the write portion of the cycle.

INTERFACE SIGNALS

The memory interconnecting diagrams, figures 5-1 through 5-4, show all data, address, and control lines associated with the data interface, address interface, and array PWAs. These diagrams include all origins, destinations, and pin numbers. Address and data paths are highlighted to provide a cross reference to figure 4-1. The functional relationship of these signals is shown in the functional block diagrams in section 5. A functional description of the block diagrams is provided later in this section. Tables 4-1 through 4-4 provide a brief description of each interface signal.

ADDRESS PATHS

CPU Port to Array

The CPU transmits both the address and write data over a 16-bit (S300/ through S315/) multipurpose path. Within the LMS, the address path taken depends upon the control signals from the CPU.

- Absolute addressing - The 16-bit path may be used to directly address the first 64K words in memory. The most significant five bits, S300/ through S304/ (LSB), are latched into the CPU address register in the data interface and are available to the address interface as lines CPUMAB16/ through CPUMAB12/. The least significant 11 bits, S305/ through S315/ (LSB), are latched into the CPU address register in the address interface.

  When the priority select circuits accept the CPU memory request, the 16-bit address is decoded to form the row address, column address, block select, and row address strobe as shown in figure 4-2.

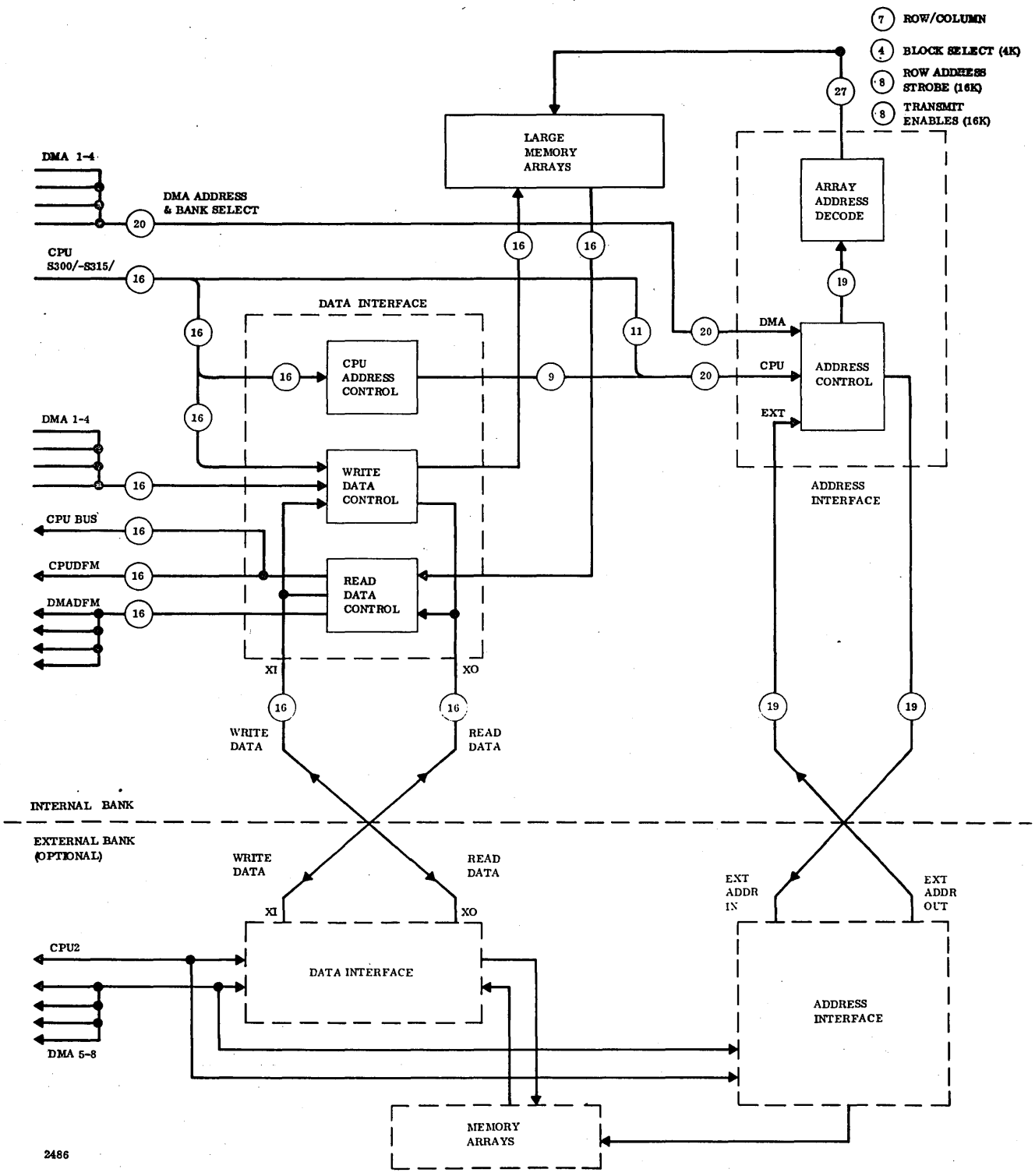


Figure 4-1. Large Memory System Address and Data Paths

TABLE 4-1. MEMORY INTERFACE SIGNALS: CPU

Signal	Source†	Destination†	Description
BUSxx	D	CPU	Data output lines 00 (MSB) through 15 to CPU three-state bus.
CPUCHAR0 CPUCHAR1	CPU	A	Write character 0 or character 1. Enables S300/ through S307/ or S308/ through S315/, respectively, into memory during a write cycle.
CPUBKPWRT	A	Brkpt Contlr	Signal to breakpoint controller to enable OPERAND breakpoint compare.
CPUDFMxx	D	CPU	CPU data output lines 16 through 01 (LSB) from memory. May also contain breakpoint address.
CPUMDS/	A	CPU	Memory data resume. Read data from memory is available to the CPU.
CPUPROTECT/	CPU	A	Program protect line. Indicates the protect status of a CPU operation.
CPUREQ/	CPU	A	Memory request. Requests a memory cycle.
CPUSC/	CPU	A	Split cycle request. Requests the read portion of a read/modify/write cycle.
CPUWRITE	CPU	A	Read/write control. On a read/modify/write cycle, requests write portion.
CPUWRTPB	CPU	A	Protect bit enable. Directs protect control to write protect bit (bit 17) into memory.
ENMEMBUS/	CPU	D	Enables memory data to lines BUS00 through BUS15.
ENPRTSYS	CPU	A, D	Enable protect system line. Overrides normal protect system when false.
EVENPAR	D	CPU	Word parity status line. Shows even number of 1 bits in word stored in memory during a write cycle. Does not include protect or ECC bits.
GATEAB/	CPU	A, D	Memory address enable. Loads S300/ through S315/ into the memory address register.
GATELBDS/	CPU	D	Lower bounds enable. Loads S300/ through S315/ into the lower bounds register. Selects status to CPUDFMxx lines if MIR28 is true.
PROTECT/	CPU	A	FCR control of protect system. Forces ENPRTSYS false if false.
GATEUBDS/	CPU	D	Upper bounds enable. Loads S300/ through S315/ into the upper bounds register. Enables loading of memory page file if MIR28 is true. Enables loading of address mode register if MIR29 is true.
MC/	CPU	A, D	Master clear. Resets latches.
MEMPROTBT/	D	CPU	Memory protect bit status. Same as memory bit 17 except always false if a CPU operation is in bounds.
MIRxx'	CPU	D	Inputs from CPU micro instruction register bits 28 through 31. MIR28' enables loading of memory page file status outputs. MIR29' enables loading of the AMR register, MIR30' selects page mode operation, and MIR31' selects program files when loaded into the AMR register.
MULTIND	CPU	D	Multilevel indirect mode line. When true, inhibits the use of the most significant address bit from line S300/.

†A: Address interface
D: Data interface

TABLE 4-1. MEMORY INTERFACE SIGNALS: CPU (Contd)

Signal	Source†	Destination†	Description
S300/ - S315/	CPU	A, D	Multipurpose path from the CPU, used for data and addresses.
SECSTATUS/	D	CPU	Status line indicating a single bit error was detected.
SETSM105/	A	CPU	Program protect fault line. Write attempt made to a protected location.
SETSM108/	A	CPU	Parity error status line. Indicates a parity error in read data.
STBCPUBKP/	A	CPU	Strobe line to CPU panel interface to compare breakpoint address. Sets CPU status latch SM104.
TO/, TO'/ T2ODDCLK/ T3MEM/	CPU	A, D	Timing signals from CPU.

†A: Address interface
D: Data interface

TABLE 4-2. MEMORY INTERFACE SIGNALS: DMA

Signal	Source†	Destination†	Description
DMADxx/	DMA	D	Write data lines 16 through 01.
DMADFMxx	D	DMA	DMA data output lines 16 through 01 from memory.
DMAMABxx/	DMA	A	Address lines 20 through 01.
DMAMAE/	A	DMA	Memory address error. Nonexistent external bank selected.
DMAMDS/	A	DMA	Memory data resume. Read data from memory is available to the DMA.
DMAMPB/	D	DMA	Memory protect bit status. Same as memory bit 17.
DMAMPE/	A	DMA	Parity error status line. Indicates a parity error in read data.
DMAMPL/	D	DMA	Word parity status line. Shows even number of 1 bits in word stored in memory during a write cycle. Does not include protect bit.
DMAPROT/	DMA	A	Program protect line. Indicates the protect status of a DMA operation.
DMAPF/	A	DMA	Program protect fault line. Write attempt made to a protected location.
DMAWRITE	DMA	A	Read/write control. On a read/modify/write cycle, requests write portion.
DMAxMAB20/	DMA	A	Local bank request from individual DMA device (1 through 4). False selects external bank.
DMAxMR/	DMA	A	Memory request from individual DMA device (1 through 4).
DMAxRA/	A	DMA	Request accept to individual DMA device (1 through 4).

†A: Address interface
D: Data interface

TABLE 4-3. MEMORY INTERBOARD SIGNALS

Signal	Source†	Destination†	Description
BLKAB/	A	D	Inhibits loading of CPU addresses during CPU read/modify/write cyle operation.
CAS/	A	M	Column address strobe to memory arrays.
CHR0/ CHR1/	A	D	Character 0 and character 1 write data enable lines to write data multiplexer.
CPUMABxx/	D	A	CPU memory address lines 20 through 12. CPUMAB20/ selects the external bank.
CPUNBNDS/	D	A	CPU address in bounds. Emulates a true CPUPROTECT/ line.
REFRESH	A	D	True (high) only during refresh cycle. Inhibits parity check during refresh operation.
DMAREQACT/	A	D	DMA internal request cycle in process. Enables DMA write data to memory.
E64KNx-x/	A	M	Read data transmit enables 0-1 through 1-4. Enable read data from selected 64K of memory (one of eight).
EMPF	D	A	Status line indicating a load memory page file operation.
ENCPWRTDO/	A	D	Enables CPU write data to external bank over lines XO16 through XOO1.
ENDMWRTDO/	A	D	Enables DMA write data to external bank over lines XO16 through XOO1.
ENEXTRDO/	A	D	Enables read data to external bank over lines XI16 through XIO1.
ENMOS250	M	A	Fast array status line. Speeds up memory timing chain, if all arrays are fast.
ENMOSWRTD/	A	D	Enables write data to memory arrays.
EXTREQACT/	A	D	External request cycle in process. Enables external write data to internal memory bank.
MDxx	D, M	Bidirec.	Data lines to/from memory array (15 through 00).
MOSMAxx/	A	M	Column and row address lines 06 through 00 to memory arrays.
MPAR	D, M	Bidirec.	Parity bit line to/from memory arrays.
MPBL	D	A	Protect bit from read data.
MPE+DED/	D	A	Parity error.
MPROT	D, M	Bidirec.	Protect bit line to/from memory arrays.
RAS64Kx-x/	A	M	Row address strobes 0-1 through 1-4. Eight lines strobe one of eight selected 64K groups of memory.
SELCPUEXT/ SELCPUINT/	A	D	Select lines to place internal or external read data on the CPUDFMxx output lines. If both are false, breakpoint address is selected.
SLTBLKx/	A	M	Block select lines 0-3. Selects one of four 16K blocks on each half of array.

† A: Address interface
D: Data interface

TABLE 4-3. MEMORY INTERBOARD SIGNALS (Contd)

Signal	Source†	Destination†	Description
STBCPUDFM/	A	D	Strobes selected read data to CPU output data register, CPUDFMxx.
STBDFM/	A	D	Strobes to latch corrected read data for possible restore operation.
STBDMADFM/	A	D	Strobes selected read data to DMA output register, DMADFMxx.
STCP/EXD/	A	D	Strobes write data into CPU, DMA, and external write data input registers.
TOMEM/	A	D	TO pulse from timing chain used to latch write data parity.
WRTMPF/	A	D	Timing signal used to load the memory page file or generate a read-only page interrupt.
WRTPB/	A	D	Enables writing of protect bit to memory.
WTMOS/	A	M	Read/write control line to memory arrays.

†A: Address interface
D: Data interface
M: Memory array

TABLE 4-4. INTERNAL/EXTERNAL MEMORY BANK INTERFACE SIGNALS

Signal	Interface Board†	Description	External Bank Signal Name
CH0/PF	A	Input: Write character 0. Enables XI16 through XI09 into memory. Output: Protect fault. Write attempt made to a protected location.	PF/CH0
CH1/$\overline{PE}$	A	Input: Write character 1. Enables XI08 through XI01 into memory. Output: Parity error detected.	$\overline{PE}$/CH1
EXTIMABxx/	A	Input: Memory address lines 19 through 01 from external bank.	EXTOMABxx/
EXTOMABxx/	A	Output: Memory address lines 19 through 01 to external bank.	EXTIMABxx/
EXTREQIN/	A	Input: Memory cycle request from external bank.	EXTREQOUT/
EXTREQOUT/	A	Output: Memory cycle request to external bank.	EXTREQIN/
EXTRESIN/	A	Input: Memory data resume. Read data available from external bank after trailing edge.	EXTRESOUT/
EXTRESOUT/	A	Output: Memory data resume. Read data available to external bank after trailing edge.	EXTRESIN/
EXTWRTIN	A	Input: Read/write control. On a read/modify/write cycle, requests write portion.	EXTWRTOUT
EXTWRTOUT	A	Output: Read/write control. Derived from CPUWRITE or DMAWRITE inputs.	EXTWRTIN
MPB/PRF	A, D	Input: Memory protect bit status. Same as memory bit 17. Output: Program protect line. Derived from CPUPROTECT/or DMAPROT/ inputs.	PRF/MPB

†A: Address interface
D: Data interface

TABLE 4-4. INTERNAL/EXTERNAL MEMORY BANK INTERFACE SIGNALS (Contd)

Signal	Interface Board†	Description	External Bank Signal Name
MPL/SC	A, D	Input: Word parity status line. Generates EVENPAR or DMAMPL/ output. Output: Read/modify/write cycle request. Derived from CPUSC/.	SC/MPL
$\overline{\text{PE}}$/CH1	A	Input: Parity error detected by external bank. Generates SETSM108/ or DMAMPE/ output. Output: Character 1 write enable. Derived from CPUCHAR1 input.	CH1/$\overline{\text{PE}}$
PF/CH0	A	Input: Program protect fault line from external bank. Generates SETSM105/ or DMAPF/ outputs. Output: Character 0 write enable. Derived from CPUCHAR0 input.	CH0/PF
PRF/MPB	A, D	Input: Program protect line. Output: Memory protect bit status. Same as memory bit 17.	MPB/PRF
SC/MPL	A, D	Input: Split cycle request from external bank. Output: Word parity status line.	MPL/SC
XIxx	D	Input: Write data from external bank (16 through 01). Output: Read data to external bank (16 through 01).	XOxx
XOxx	D	Input: Read data from external bank (16 through 01). Output: Write data to external bank (16 through 01).	XIxx

†A: Address interface
D: Data interface

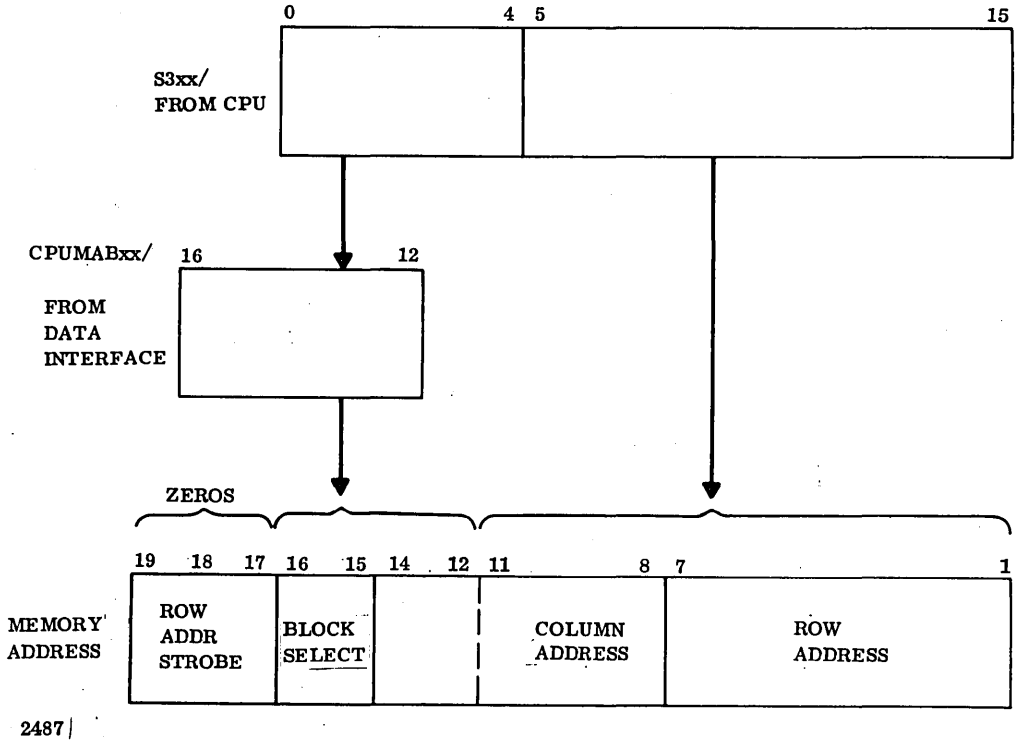


Figure 4-2. CPU Absolute Addressing

Since bit 17 and higher of the CPU memory address do not exist, internal memory beyond the first 64K words is not accessible with this addressing method. Therefore, CPU absolute addressing cannot be used to access the external bank.

- Bounds address loading - The path S300/ through S315/ is utilized to load either the 16-bit lower bounds register or the 16-bit upper bounds register in the data interface. These two registers set the upper and lower boundaries of the unprotected memory area available to the CPU, and a limit comparison is made each time a CPU address is loaded.

- Page mode addressing - In this addressing method the least significant 11 bits of the path, S305/ through S315/ (LSB), are strobed into the CPU address register in the address interface as in absolute addressing, and are used to form the least significant 11 bits of the memory address. However, the upper eight bits plus the external bank select bit are provided by the memory page file located in the data interface. They are available to the address interface as CPUMAB20/ through CPUMAB12/ with CPUMAB20/ as the external bank select bit. The memory page file is a 64-word by 9-bit RAM employed to utilize the large storage capacity of the LMS; it is discussed later in this section. The file is divided into two 32-word halves; the lower addressed half is known as page set zero, the upper as page set 1. The file address is supplied by the most significant five bits of the CPU multipurpose path, S300/ through S304/, plus the latched MIR31' line from the CPU micro instruction register. All 1M words of memory may be addressed utilizing page mode. Figure 4-2 shows both page mode addressing and memory page file loading address formats.

- Memory page file loading - When loading the memory page file with page addresses, the most significant five bits, S300/ through S304/, are the file address, and S305/ is the upper- or lower-half select bit. The page address, which will be written into the memory page file, is formed from the nine bits S307/ through S315/. The level on S307/ is stored as the external bank select bit. S306/ is the tenth bit, but it is not used and is not stored.

Once the 20-bit memory address has been formed, if the external bank select bit from the memory page file is true and a CPU (external) memory request has been accepted, a 19-bit address is sent to the external bank over the lines EXTOMAB19/ through EXTOMAB01/ (LSB) (see figure 4-3). The treatment of this address path, which is received by the external bank as the lines EXTIMAB19/ through EXTIMAB01/ (LSB), is discussed below.

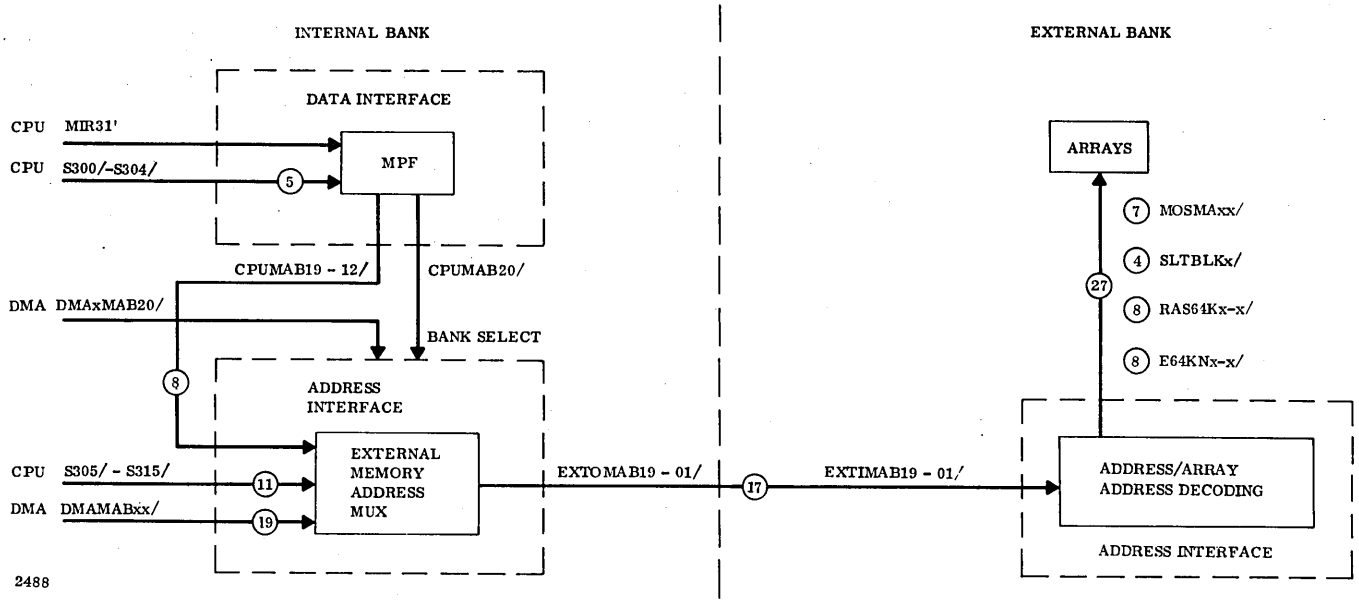


Figure 4-3. External Bank Address Path

A CPU-internal bank request, when accepted, processes the 19-bit memory address as follows:

1. The three most significant bits (19 through 17) of the memory address are decoded to enable one of eight row address strobes or transmit enables, which selects one of up to eight sets of 64K-word memory sections.

2. The next most significant two bits (16 and 15) select one of the four 16K-word (16,384-word) blocks within a 64K-word section.

3. The next most significant seven bits (14 through 8) determine the column address within the 16K block. Each block is divided into 128 columns and 128 rows.

4. The least significant seven bits (7 through 1) determine the row address within the block.

The row and column addresses are sent to an array over a common 7-bit bus, MOSMA06/ through MOSMA00/. The row address is placed on the bus first and is strobed into the array by the selected row address strobe. Then the column address is placed on the path and strobed into the array by the column address strobe, CAS/.

DMA Port to Array

The DMA devices transmit an address over a common open collector 19-bit bus, DMAMAB19/ through DMAMAB01/ (LSB).

External bank selection is made by individual lines from the four DMA subports, DMA1MAB20/ through DMA4MAB20/. If a true level is detected on the appropriate line selected by the DMA priority circuit, then when the DMA external request is honored, the incoming 19-bit DMA address is placed on lines EXTOMAB19/ through EXTOMAB01/ to the external bank.

If the DMAxMAB20/ line is false, a DMA internal request processes the 19-bit memory address as discussed above for a CPU internal request.

External Port Array

Addresses from CPU external or DMA external memory requests are received on lines EXTIMAB19/ through EXTIMAB01/. As the external request is accepted, the 19-bit address is processed as discussed above for a CPU internal request.

READ DATA PATHS

The 18 read data bits enter the data interface from an array over the 16-bit bidirectional data path, MD15 through MD00 (LSB), plus the parity bit line MPAR and the protect bit line MPROT. The data and parity bits enter the parity check decoding network, and any single-bit errors are detected. The resulting data is available to the CPU, DMA, or external ports, or to the array for a restore or character write operation.

Read data from the external bank appears at inputs XO16 through XO01 (LSB), and is available to the CPU or DMA ports. The parity input is the line MPL/SC, and protect bit input is line MPB/PRF.

Read Data to CPU Port

Read data from memory is available to two CPU data output paths: the 16-line data path CPUDFM16 through CPUDFM01 (LSB), and the 16-line operand data path BUS00 through BUS15.

The CPUDFMxx lines are stable with read data a maximum of 5 nanoseconds after the trailing edge of the CPUMDS/ signal. They remain stable until the leading edge of the next CPU memory request.

On a CPU internal read or protected write cycle, the read data is from the internal bank. For a CPU external read or protected write cycle, the read data is derived from the external write data/read data inputs on XO16 through XO01.

Also available over the CPUDFMxx lines are the last address from the CPU to be loaded via the path S300/ through S315/.

A second path for read data to the CPU is the set of lines BUS00 through BUS15, which feeds the main CPU three-state bus directly. Read data is stable at this output a maximum of 120 nanoseconds after the leading edge of the CPUMDS/ signal (for a read/modify/write cycle request, until the leading edge of the CPUMDS/ signal that follows the leading edge of the write initiate line). The source of data for this path is the CPUDFMxx lines, but the output path may also be used to read the memory page file or the status of a word read from memory.

Read Data to DMA Port

Read data from memory is available to the DMA devices over lines DMADFM16 through DMADFM01 (LSB). On a DMA internal read or protected write cycle the read data is from the internal bank. For a DMA external read or protected write cycle the read data is derived from the external write data output/read data inputs on XO16 through XO01. DMA read data is stable a maximum of 17 nanoseconds after the leading edge of the DMAMDS/ signal, and remains stable a minimum of 340 nanoseconds.

External Bank Read Data Path

The external read data path is shown in figure 4-4. Data stored in the external bank is read by the external bank data interface, output on external bank lines XI16 through XI01, and received by the internal bank inputs XO16 through XO01. Read data is stable on these lines, at the trailing edge of the EXTRESOUT/ signal from the external bank, which is received by the internal bank as the EXTRESIN/ input, and remains stable for a minimum of 80 nanoseconds. The same set of data lines is also used to carry write data to the external bank, but on a protected write cycle the write data will have been latched into a register in the external bank before the read data is placed on the lines. Note the timing diagram for an external bank protected write cycle.

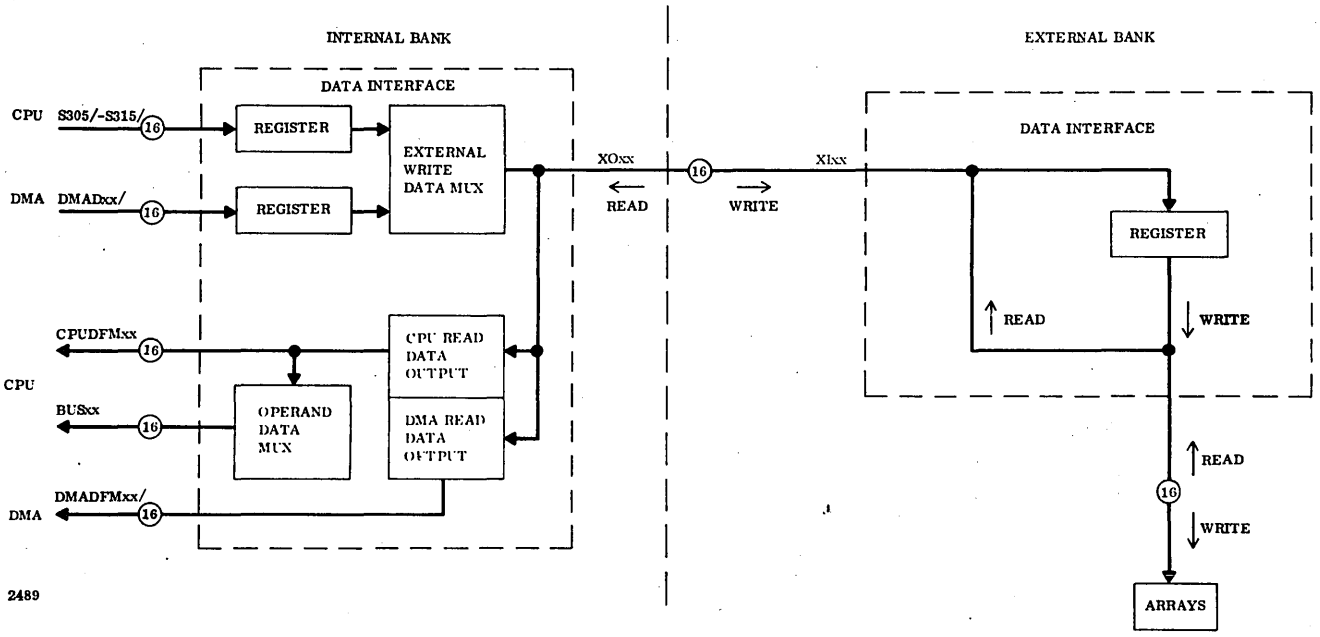


Figure 4-4. External Bank Read/Write Data Path

The external bank read data is available to the requesting CPU or DMA port as described in the preceding paragraphs.

The read data path for an external request made by a CPU or DMA device connected to the external bank is handled in identically the same manner as described here, with the internal and external banks switching roles. Note that a completely separate set of address, data, and control lines is used for this operation, and that the two functions may be executed simultaneously (for example, a request from an external DMA device for the internal bank may be processed concurrently with an internal CPU or DMA request for the external memory bank).

From the standpoint of the requesting device, internal and external requests are similar except that data word status is not available to the CPU for an external bank cycle.

WRITE DATA PATHS

The data interface accepts write data from four sources: the CPU, the DMA devices, the internal arrays, and the external bank.

Read data from either the read portion of a split cycle request or the first portion of a protected write or character write cycle is stored in the data interface. A write data multiplexer for each half (character) of a 16-bit word selects either the read data or one of the other three write data inputs to form a write data word. For example, on a CPU character 0 write request, the most significant eight bits (character 0) are selected from the CPU write data input, while the least significant eight bits (character 1) are selected from the read data to restore that character in memory.

Once selected, the write data word is sent to the selected array over lines MD15 through MD00. A word parity bit (MPAR) and protect bit (MPROT) are generated and sent to the array along with the write data.

CPU Write Data Path

Multipurpose path S300/ through S315/ (LSB) from the CPU is used for write data; it must be stable within 170 nanoseconds maximum after the leading edge of CPUREQ/ and remain stable until the CPUMDS/ signal (for a split cycle request, from 170 nanoseconds after the leading edge of CPUWRITE until the following CPUMDS/ signal). For character write operation, the CPU places the desired character in the character 0 or character 1 position. Data on the lines is latched after the CPU request has been accepted and is sent to the write data multiplexer.

DMA Write Data Path

The common DMA open collector data lines DMAD16/ through DMAD01/ must be stable with write data a maximum of 50 nanoseconds after the leading edge of the request accept, DMAxRA/, and remain stable until DMAxRA/ goes false. Data on the lines is latched as the memory cycle starts and is sent to the write data multiplexer.

External Bank Write Data Path

The external write data path is shown in figure 4-4. Write data from either the S300/ through S315/ lines from the CPU or the DMAD16/ through DMAD01/ lines from the DMA devices is enabled to the bidirectional write data out/read data in lines XO16 through XO01.

Write data is received by the external bank on input lines XI16 through XI01. After the external bank begins the external request cycle, the write data is latched in the external write data register and the signal EXTRESOUT/ is sent to the internal bank, which is received on the internal bank input EXTRESIN/. At this time the write data is removed from XO16 through XO01 so that read data may be received over the same path.

The write data path for an external request made by a CPU or DMA device connected to the external bank is handled in the same manner as described here, with the internal and external banks switching roles. A second set of address, control, and data lines between the banks is employed for this purpose.

REQUEST AND PRIORITY SELECTION

Each of the three ports requests a memory cycle via a request control line. For the CPU, the CPUREQ/ line is brought true. The four DMA devices have individual request control lines, DMA1MR/ through DMA4MR/. Requests from external bank CPU or DMA devices are received by the internal bank over the EXTREQIN/ line.

Since only one request of a single bank can be processed at one time, a priority scheme is employed to select one among two or more simultaneous inputs. Requests are honored according to the following order:

1. Refresh requests - Each 15.6 microseconds a portion of the RAM is refreshed, and a cycle is requested internally by the refresh clock. A refresh request has first priority.

2. External requests - Requests from external bank CPU or DMA devices have priority over internal CPU or DMA requests. External requests are received by the internal bank over the EXTREQIN/ line.

3. DMA requests - A separate priority selection is made among the four DMA request inputs, DMA1MR/ through DMA4MR/ (DMA1MR/ has the highest priority). Once the highest priority DMA request has been selected, request accept signal DMAxRA/ is sent to the requesting device, gating data, address, and control signals from that device to the lines common to all DMA devices. Then, either a DMA internal or external request is generated, which in turn competes with refresh, external and CPU requests. For a DMA internal request, refresh and external requests have higher priority. A CPU internal request has lower priority unless a DMA cycle is in progress, in which case a waiting CPU request takes precedence over a waiting DMA request. A separate priority selection is made between DMA and CPU external requests, again with the DMA request having higher priority than a CPU request unless a DMA cycle is in process.

4. CPU requests - CPU requests have lowest priority, except immediately following a DMA cycle, as discussed above. They are received over a CPUREQ/ line from the CPU.

MODES OF OPERATION

READ REQUESTS

Read or write operation is controlled by the write line from the requesting device. If the associated write line is false, a read operation is specified.

CPU Read Request (Internal Bank)

For a CPU read cycle the line CPUWRITE must be false within 25 nanoseconds following the leading edge of CPUREQ/, and remain false until the leading edge of CPUMDS/. A CPU read timing diagram is shown in figure 4-5; the signals are described in table 4-5.

After the CPU request has been selected by the priority circuits, the eight-phase timing chain is started and the memory address is sent to the selected array. Read data from an array is gated to the bidirectional MDxx lines by one of the E64KNx-x/ lines. It is then strobed into the CPU data output register by the second STBCPUDFM/ signal. A CPUMDS/ signal is sent to the CPU, the leading edge indicating that the incoming control lines are no longer needed, and the trailing edge indicating that read data is available over the lines CPUDFMxx and available to the BUSxx outputs. The parity error line SETSM108/ goes true for 80 to 100 nanoseconds within 100 nanoseconds following the CPUMDS/ signal if a parity error has been detected.

CPU Read Request (External Bank)

When a CPU request is received and the external bank is addressed, the requirements for the control and data lines remain unchanged but the timing is significantly altered. A CPU read (external) timing diagram is shown in figure 4-6 for the local bank signals. External bank signal activity is shown in figure 4-7. The signals are described in tables 4-6 and 4-7.

After the CPU request is selected by the CPU/DMA external bank priority circuits, a request is sent to the external bank over the line EXTREQOUT/, which is received by the external bank over the line EXTREQIN/. The address is gated to lines EXTOMABxx/. When the external bank completes the memory cycle, an EXTRESIN/ signal is received by the internal bank, and read data is available over lines XOxx. This data is strobed into the CPU data output register by the second STBCPUDFM/ signal, and the leading edge of CPUMDS/ indicates that incoming control signals are no longer needed and that read data is available over the CPUDFMxx lines and available to the BUSxx output. If a parity error is detected by the external bank, the line PE/CH1 is true following the EXTRESIN/ signal, and SETSM108/ to the CPU is set true for 80 nanoseconds within 40 nanoseconds after the leading edge of CPUMDS/.

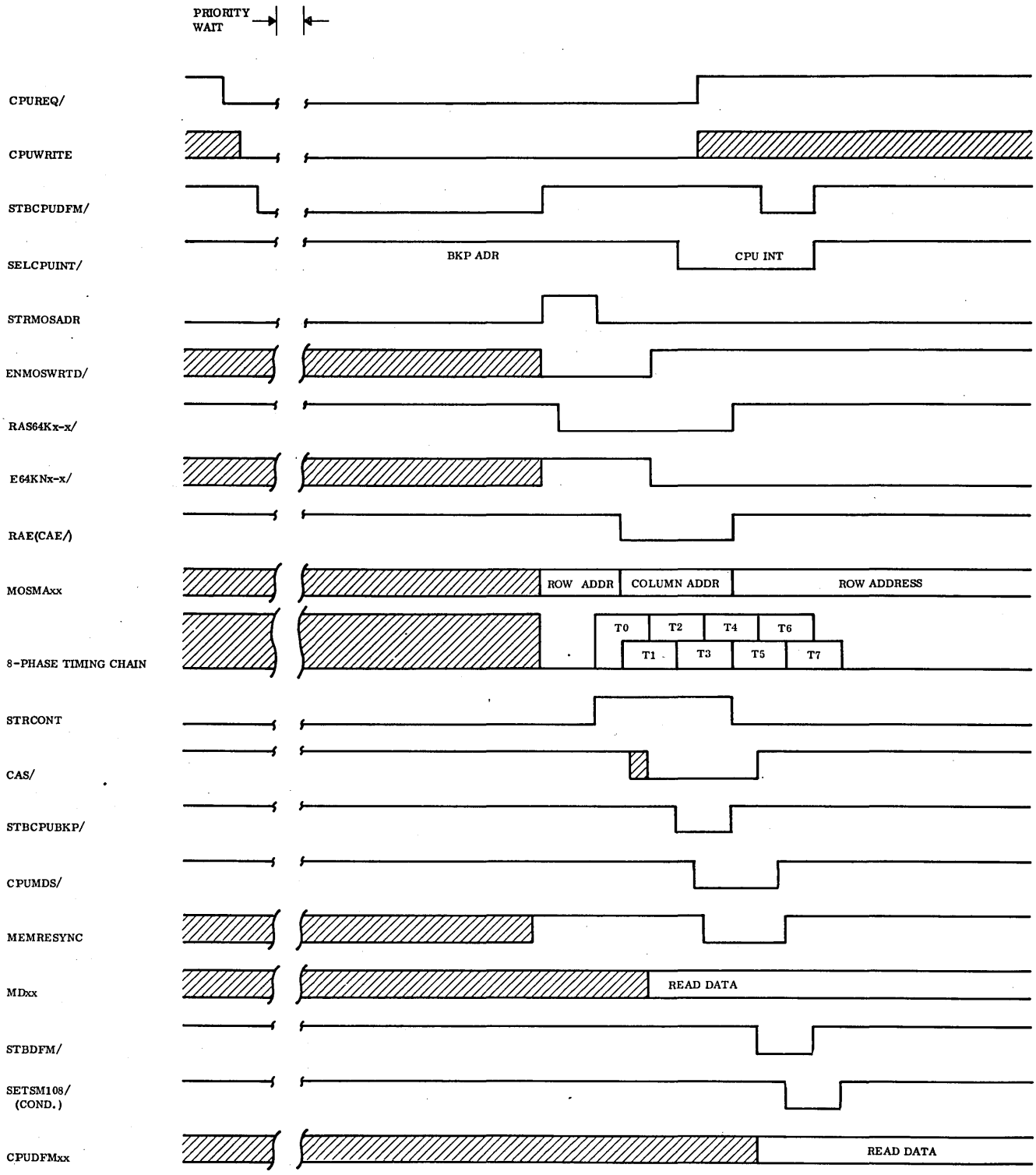


Figure 4-5. CPU Read Cycle Timing (Internal)

TABLE 4-5. CPU READ CYCLE (INTERNAL)

Signal	Description
CPUREQ/	CPU memory request. Must remain stable until leading edge of CPUMDS/.
CPUWRITE	Read/write control. Must remain false from 25 nanoseconds following CPUREQ/ until leading edge of CPUMDS/.
STBCPUDFM/	First pulse strobes breakpoint address into CPU output register. Second pulse strobes read data into CPU output register.
SELCPUINT/	Selects internal read data to CPU output register when true, breakpoint address when false.
STRMOSADR	Begins memory cycle following acceptance of request. Latches selected address in address register.
ENMOSWRTD/	When false, enables selected (read data) E64KNx-x/ output.
RAS64Kx-x/	Selected row address strobe.
E64KNx-x/	Enables read data to MDxx lines.
RAE ($\overline{CAE}$)	Row address enabled to MOSMAxx/ lines when true, column address when false.
MOSMAxx/	Row and column address lines to memory arrays.
T0 - T7	Eight-phase timing chain; starts approximately 50 nanoseconds after leading edge of STRMOSADR.
STRCONT	Timing pulse used to set status latches.
CAS/	Column address strobe to memory array.
STBCPUBKP/	Leading edge indicates to CPU that breakpoint address is available on CPUDFMxx lines until trailing edge.
CPUMDS/	Memory data resume. Control lines from CPU need no longer be stable. Read data available 5 nanoseconds after trailing edge.
MEMRESYNC	Resync signal to priority select circuit.
MDxx	Bidirectional data lines 15 to 00 from memory, contain read data beginning at T2 time.
STBDFM/	Timing signal to data interface.
SETSM108/	Goes true at T6+50 nanoseconds for 80 to 100 nanoseconds if a parity error is detected.
CPUDFMxx	CPU data output lines 16 through 01 from data interface. Contain read data 5 nanoseconds after trailing edge of CPUMDS/.

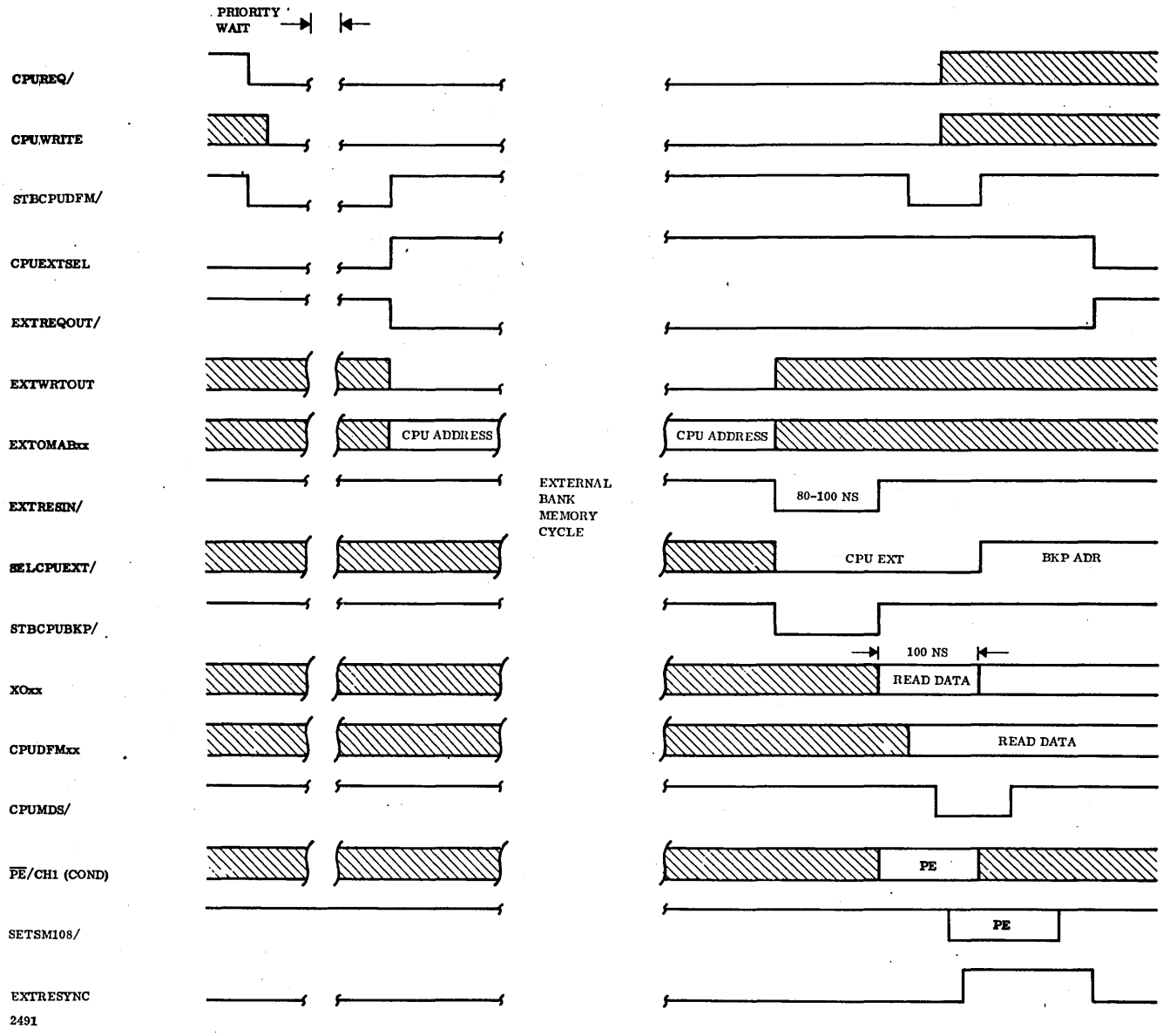


Figure 4-6. CPU Read Cycle Timing (External)

TABLE 4-6. CPU READ CYCLE (EXTERNAL)

Signal	Description
CPUREQ/	Same as internal read cycle.
CPUWRITE	Same as internal read cycle.
STBCPUDFM/	Same as internal read cycle.
CPUEXTSEL	Enables CPU memory address to lines EXTOMABxx/ and CPU control lines to external bank.
EXTREQOUT/	Memory cycle request sent to external bank.
EXTWRTOUT	Read/write control line to external bank. Set to level of CPUWRITE by CPUEXTSEL.
EXTOMABxx/	Memory address lines 19 through 01 to external bank.
EXTRESIN/	Memory data resume from external bank.
SELCPUEXT/	Selects external read data to CPU output register when true, breakpoint address when false.
STBCPUBKP/	Same as local cycle.
XOxx	Read data input lines 16 through 01 from external bank. Contains read data for 100 nanoseconds following trailing edge of EXTRESIN/.
CPUDFMxx	CPU data output lines 16 through 01 from data interface. Contains read data at leading edge of CPUMDS/.
CPUMDS/	Memory data resume. Control lines from CPU no longer needed. Read data available on CPUDFMxx.
$\overline{\text{PE}}$/CH1	Parity error input from external bank. Enabled for 100 nanoseconds following trailing edge of EXTRESIN/.
SETSM108/	Goes true 60 nanoseconds following trailing edge of EXTRESIN/ if the $\overline{\text{PE}}$/CH1 line is true.
EXTRESYNC	Resync signal to external priority select latches. Trailing edge allows next CPU or DMA external request to be processed.

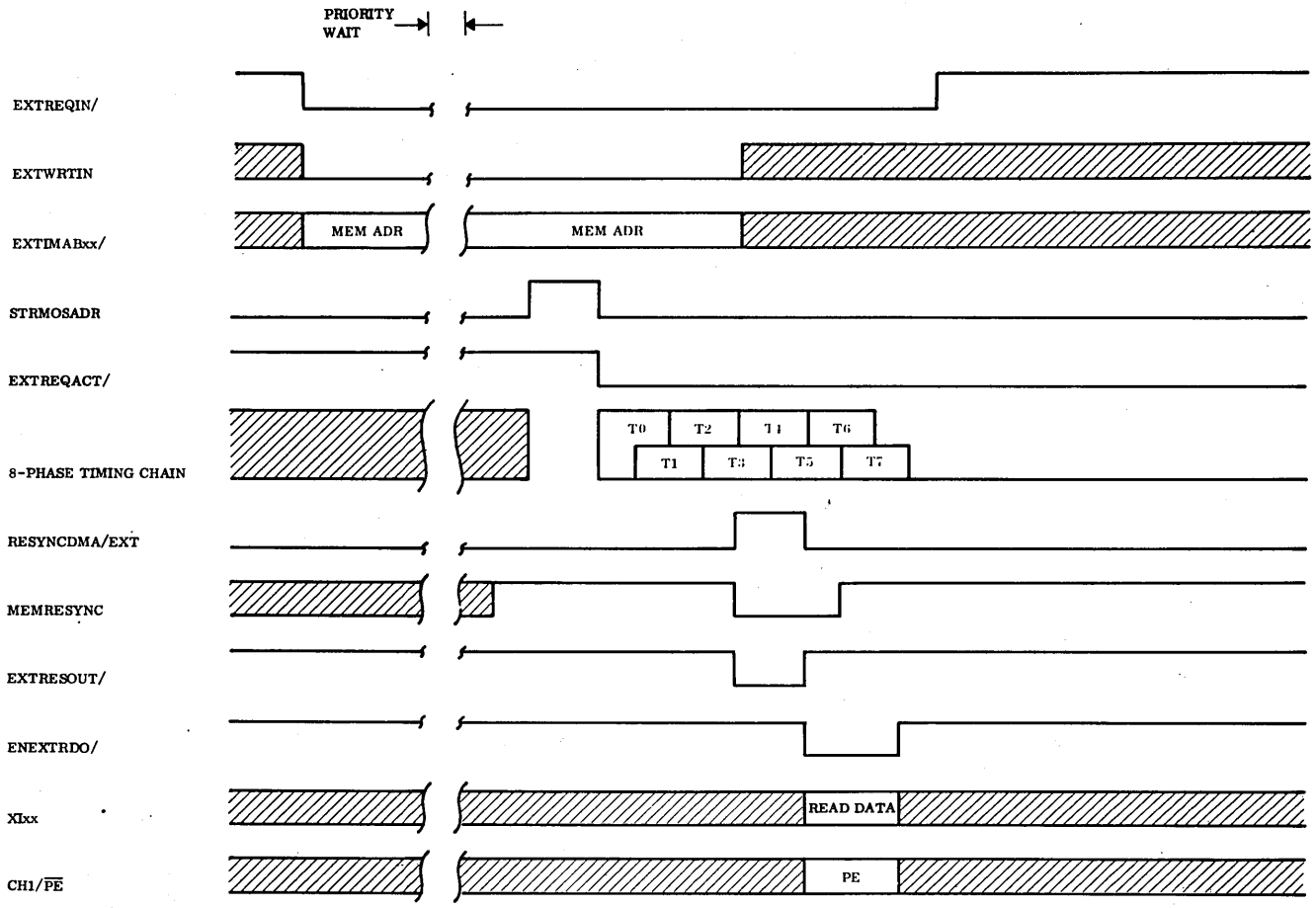


Figure 4-7. External Read Cycle Timing

TABLE 4-7. EXTERNAL READ CYCLE TIMING

Signal	Description
EXTREQIN/	Memory cycle request from external bank.
EXTWRTIN	Read/write control line from external bank; false for a read cycle.
EXTIMABxx/	Memory address lines 19 through 01 from external bank.
STRMOSADR	Begins memory cycle following acceptance of request. Latches memory address in address register.
EXTREQACT/	Status latch set at start of timing chain.
RESYNCDMA/EXT	Leading edge resets refresh, external, and DMA request latches in priority circuitry at T4 time.
MEMRESYNC	Resync signal to priority select circuit.
EXTRESOUT/	Memory data resume to external bank, sent at T4 time.
ENEXTRDO/	Enables read data from memory arrays to lines XIxx from T6 time to T6+100 nanoseconds.
XIxx	Read data output lines 16 through 01 to external bank. Contain read data during ENEXTRDO/.
CHI/$\overline{PE}$	Parity error output. Conditional signal enabled during ENEXTRDO/.

External Read Request

An external read cycle is requested by a true EXTREQIN/ line. Incoming read/write control line EXTWRTIN is also false to request a read operation. An external read cycle is shown in figure 4-7; the signals are described in table 4-7.

After the external request has been selected by the priority circuits, the eight-phase timing chain is started and the address from lines EXTIMABxx/ is decoded and sent to the selected array. Read data from the array is placed on the XIxx lines to the external bank by the ENEXTRDO/ signal, a 100-nanosecond pulse that begins at the trailing edge of the EXTRESOUT/ signal. A parity error is shown by a true level on the CH1/PE output during the same 100-nanosecond period.

DMA Read Request (Internal or External)

A DMA read cycle is requested by one of the four DMAxMR/ signals. The internal/external bank select line DMAxMAB20/ from the highest priority DMA request causes either a DMA internal or DMA external request to be generated. DMAxMAB20/ must stay stable until DMAxRA/ goes false after going true. The DMA device selected must then place control and address information on the lines common to all DMA devices. For a read cycle request, the DMAWRITE line must be false until the leading edge of DMAMDS/. A DMA internal read cycle timing diagram is shown in figure 4-8, with signal descriptions in table 4-8. A DMA external read cycle is similar to an internal cycle for signals to and from the DMA devices, and internal timing is similar to a CPU internal read cycle, discussed previously.

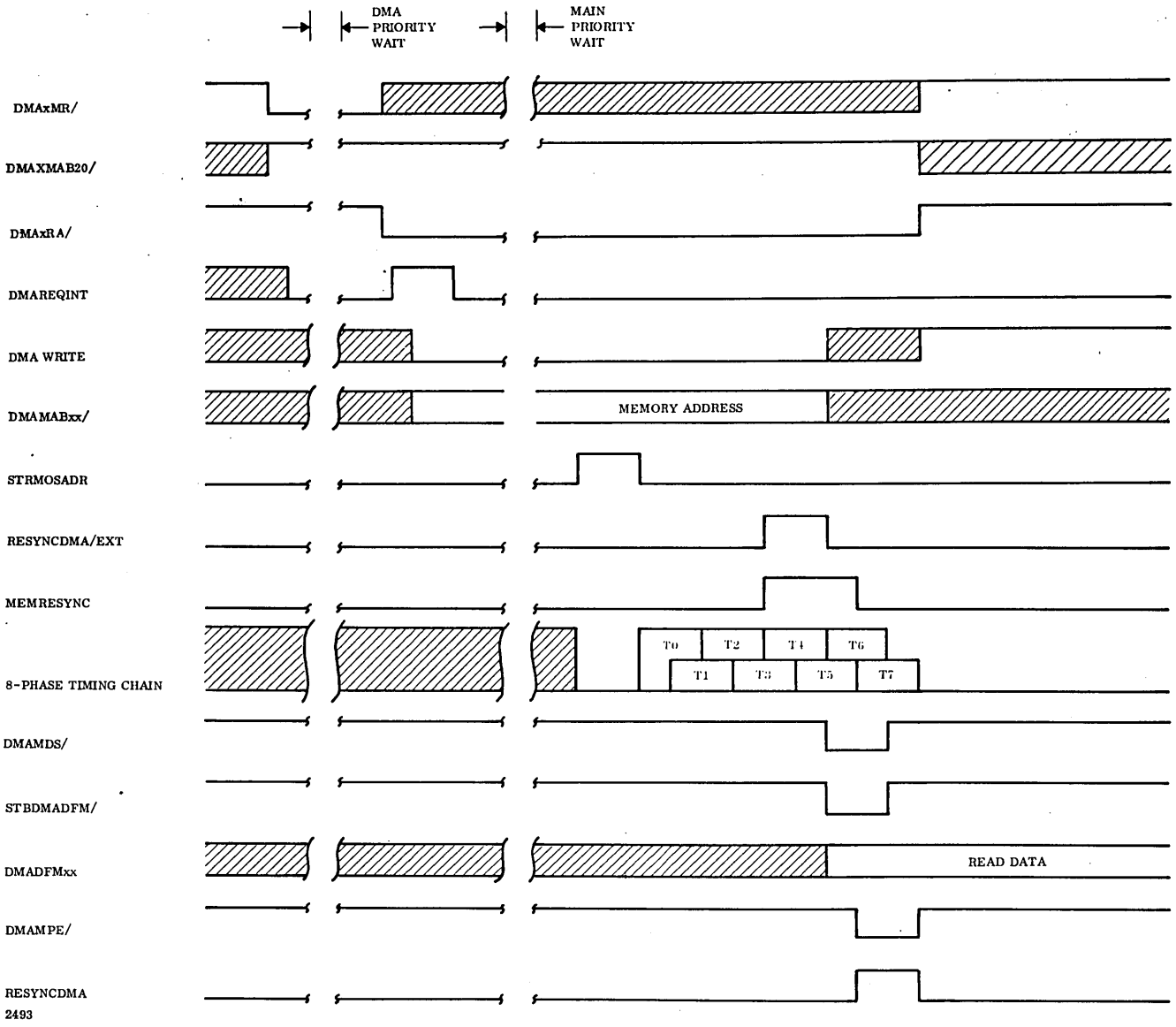


Figure 4-8. DMA Read Cycle Timing (Internal)

TABLE 4-8. EXTERNAL READ CYCLE TIMING

Signal	Description
DMAxMR/	Memory cycle request from one of four (1-4) DMA devices. Must remain true until the associated DMAxRA/ signal is sent, and must return false within 50 nanoseconds after the leading edge of DMAMDS/.
DMAxMAB20/	Internal/external bank select line (1-4). Must remain false for internal bank request from DMAxMR/ going true to DMAxRA/ going false.
DMAxRA/	Request accept signal (1-4). Set true when the request is accepted by the DMA priority circuits; instructs DMA device to place address and control levels on common lines within 50 nanoseconds. Returns false at T6+50 nanoseconds trailing edge.
DMAREQINT	DMA internal request pulse to main priority select circuits following selection of highest priority DMA request.
DMAWRITE	Read/write control line. Must be false within 50 nanoseconds following DMAxRA/, remain false until DMAMDS/, and must be disabled within 50 nanoseconds following the trailing edge of DMAxRA/.
DMAMABxx/	DMA memory address bus (19-01). Must contain memory address within 50 nanoseconds following DMAxRA/, remain stable until DMAMDS/, and be disabled within 50 nanoseconds after the trailing edge of DMAxRA/.
STRMOSADR	Begins memory cycle following acceptance of DMALRQ request. Latches memory address in address register.
RESYNCDMA/EXT	Leading edge resets refresh, external, and DMA request latches in priority circuit at T4 time.
MEMRESYNC	Resync signal to priority select circuit.
DMAMDS/	Memory data strobe. Control lines from DMA device no longer needed. Read data stable within 17 nanoseconds after leading edge.
STBDMADFM/	Strobes read data into DMA output register at T6 time.
DMADFMxx	DMA read data lines 16 through 01. Stable within 17 nanoseconds of DMAMDS/ for a minimum of 340 nanoseconds.
DMAMPE/	Memory parity error line. Goes true at T6+50 nanoseconds if a parity error is detected.
RESYNCDMA	Trailing edge resets DMA priority latches, allowing next DMA request to be selected.

If an external bank is selected but no external bank is connected, memory address error line DMAMAE/ is set true within 50 nanoseconds after DMAxRA/. An internal read cycle from the same address is performed, and the DMAMAE/ signal is cleared at DMAMDS/ time.

Read data is strobed into the DMA data output register by the STBDMADFM/ signal, a DMAMDS/ signal is sent to indicate address and control lines are no longer needed, and read data is available on the DMADFMxx outputs. Parity error signal DMAMPE/ is an 80- to 100-nanosecond pulse within 100 nanoseconds after the leading edge of DMAMDS/ if a parity error exists.

WRITE REQUESTS

A write cycle is initiated by a memory request plus a true write line from the requesting port. Two types of write cycles are used. The protected or automatic split cycle write cycle consists of a read followed by a write portion; the read portion detects parity errors or protect faults and may abort the write portion. The read portion in character write mode also saves the unchanged character of the addressed word so that it may be restored in the write portion. If ENPRTSYS is false and a word write operation is specified, a shortened write cycle without a read portion is performed (full or unprotected write) and the protect bit is automatically set, regardless of the condition of previously stored data.

ENPRTSYS is forced false by clearing the FCR protect switch bit 8 (J20G at the keyboard).

CPU Write Request (Internal Bank, Protected)

For a CPU write cycle, control line CPUWRITE must be true within 25 nanoseconds after the leading edge of CPUREQ/ and remain true until the leading edge of CPUMDS/. Write character lines CHAR0 and CHAR1 plus the program protect line CPUPROTECT/ must also be stable over the same time period. Write data is stable on the multipurpose path S300/ through S315/ within 170 nanoseconds of CPUREQ/ until at least 25 nanoseconds after the leading edge of CPUMDS/. A timing diagram is shown in figure 4-9 and signals are described in table 4-9.

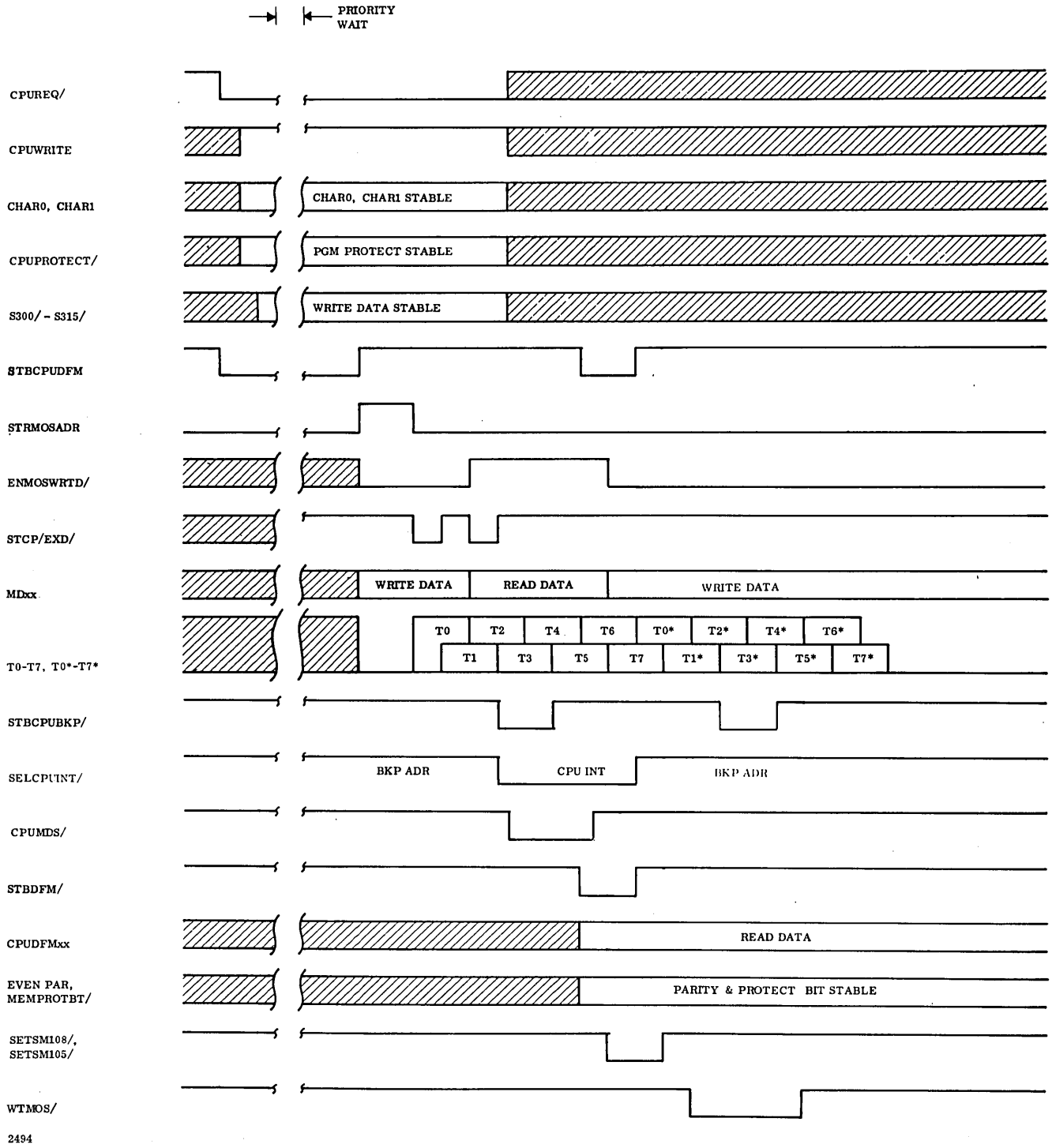


Figure 4-9. CPU Write Cycle Timing (Internal, Protected)

TABLE 4-9. CPU WRITE CYCLE TIMING (INTERNAL, PROTECTED)

Signal	Description
CPUREQ/	CPU memory request. Must remain stable until leading edge of CPUMDS/.
CPUWRITE	Read/write control. Must remain true from 25 nanoseconds after CPUREQ/ until leading edge of CPUMDS/.
CHAR0, CHAR1	Write character 0 and character 1 control. Must remain stable from 25 nanoseconds after CPUREQ/ until leading edge of CPUMDS/.
CPUPROTECT/	Program protect line. Must remain stable from 25 nanoseconds following CPUREQ/ until leading edge of CPUMDS/.
S300/ - S315/	CPU write data lines. Write data must be stable from 170 nanoseconds (should be less than 50 nanoseconds to generate parity) after leading edge of CPUREQ/ until 25 nanoseconds after leading edge of CPUMDS/.
STBCPUDFM/	First pulse strobes breakpoint address into CPU output register if previous CPU cycle was a read cycle. Second pulse strobes read data into CPU output register.
STRMOSADR	Begins memory cycle following acceptance of request. Latches selected address in address register.
ENMOSWRTD/	Enables write data to generate parity bit until T2 time, then enables selected E64KNx-x line to place read data on MDxx lines until T6+50 nanoseconds, then enables write data to MDxx lines.
T0-T7, T0*-T7*	Eight-phase timing chain. T0* to T7* is second cycle of chain. T0* starts at end of T6.
STCP/EXD/	Strobes write data from S300/ to S315/ into CPU write data input register.
MDxx	Bidirectional data lines 15 through 00 to/from memory arrays. Contain read data from T2 to T6+50 nanoseconds time, then write data.
STBCPUBKP/	Leading edge of each pulse indicates to CPU that breakpoint address compare is available on CPUDFMxx lines until trailing edge.
SELCPUINT/	Selects internal read data to CPU output register when low, breakpoint address when high.
CPUMDS/	Memory data strobe. Control lines from CPU need no longer be stable. Read data available 5 nanoseconds after trailing edge.
STBDFM/	Latches read data for use in character write or restore operation.
CPUDFMxx	CPU data output lines 16 through 01 from data interface. Contain read data 5 nanoseconds after trailing edge of CPUMDS/.
EVENPAR	Write data word parity. True if number of 1 bits in write data is even.
MEMPROTBT/	Memory protect bit output line.
SETSM105/	Protect fault line. Goes true at T6+50 if a protect is fault detected.
SETSM108/	Goes true at T6+50 nanoseconds for 80 to 100 nanoseconds if a parity error is detected.
WTMOS/	Write enable line to memory arrays, false from T2* to T6* time.

After the CPU request has been accepted, the eight-phase timing chain is started and write data is latched in the CPU write data input register and temporarily gated to the MDxx lines to determine the word parity of the incoming write data. Read data is then applied to these lines and strobed into the CPU data output register by the line STBCPUDFM/. A CPUMDS/ memory resume is sent to the CPU, the leading edge indicating the end of the need for input control and write data, and the trailing edge indicating read data, write word parity, and the memory protect bit status are available to the CPU over the lines CPUDFMxx, EVENPAR, and MEMPROTBT/, respectively. Read data output is also available to the BUSxx output. The parity error (SETSM108/) and protect fault (SETSM105/) lines indicate the appropriate status approximately 100 nanoseconds later by an 80 to 100 nanosecond pulse.

Write data is then again gated to the MDxx lines and the eight-phase timing chain started on a second cycle. The WTMOS/ control line to the arrays is set true for a 160- to 200-nanosecond period to write the data in memory. This signal is inhibited, leaving the memory contents unchanged, if a protect fault or protect parity error is encountered (see discussion on protection later in this section).

CPU Write Request (External Bank, Protected)

When a CPU write request is received and the external bank is addressed, the requirements for the control and data lines remain unchanged from internal bank requirements. A CPU write (external) timing diagram is shown in figure 4-10 for the internal bank signals, with descriptions in table 4-10. External bank signal activity is as shown in external write request timing.

After the CPU request is selected by the CPU/DMA external bank priority circuits, a request is sent to the external bank over the line EXTREQOUT/, which is received by the external bank as the signal EXTREQIN/. The address is gated to the EXTOMABxx/ lines, write data from lines S300 through S315/ to the XOxx lines, and CPU control lines to the appropriate external bank control lines.

When the external bank completes the memory cycle, an EXTRESIN/ signal is received by the internal bank, at which time the internal bank removes the address, write data, and control information from the internal/external bank lines. At the trailing edge of EXTRESIN/ read data is available over the XOxx lines, and parity and protect bit status over the appropriate control lines. The read data is strobed into the CPU data output register by the STBCPUDFM/ signal, and CPUMDS/ indicates that incoming control signals and write data are no longer needed and that read data is available over the CPUDFMxx lines as well as available to the BUSxx output. If a parity error or a protect fault is detected by the external bank, the appropriate SETSM108/ or SETSM105/ line to the CPU is set true for 80 nanoseconds within 40 nanoseconds following the leading edge of the CPUMDS/ signal.

If no external bank is connected, the external bank request is inhibited and the LMS performs a read cycle from the same address in the internal bank.

CPU Write Request (Protection Disabled)

A shortened write-only cycle is available if the normal write protection system is disabled by a false ENPRTSYS line. The protection scheme is overridden in that the previously stored protect bit is not read, no protect fault or write abort can occur, and the protect bit is set into memory bit 17 on every instruction. Only a memory addressing error (no external bank) defeats a write operation.

Processing of the request is the same as for a protected write request except that no protect fault signals will be generated and no read information is available. A timing diagram is shown in figure 4-11. Note that only one cycle of the eight-phase timing chain is utilized. Signal descriptions are included in table 4-11.

If the external bank is addressed and the ENPRTSYS line on the external bank is not also set false, a normal CPU (external) write cycle is executed (complete with read data) except that the external bank sees a true protect line (MPB/PRF). If ENPRTSYS is false on the external bank, the short cycle is executed and the protect bit is set even if ENPRTSYS was not set false on the internal bank.

ENPRTSYS is forced false if the FCR Protect Switch bit 8 is clear (i.e., J20G at the keyboard).

External Write Requests

An external write cycle is requested by a true EXTREQIN/ line. The read/write control line EXTWRTIN is true to indicate a write request. Control lines CH0/PF, CH1/PE, and PRF/MPB contain the control information dictated by the CPU or DMA control lines for write character and program protect. An external write cycle is shown in figure 4-12 and the signals described in table 4-12.

After the external request has been selected by the priority circuits, the eight-phase timing chain is started, and the memory address from EXTIMABxx/ and write data from XIxx are sent to the selected array. Read data from the array is placed on the same XIxx lines by the ENEXTRDO/ signal, a 100-nanosecond pulse that begins at the trailing edge of the EXTRESOUT/ signal. Parity and protect status information is transmitted over the bidirectional control lines CH0/PF, CH1/PE, and PRF/MPB during the same 100-nanosecond period.

DMA Write Requests (Internal and External)

A DMA write cycle is requested by one of the four DMAxMR/ signals. The internal/external bank select line DMAxMAB20/ from the highest priority DMA request causes either a DMA internal or external request to be generated. DMAxMAB20/ must stay stable until request accept line DMAxRA/ goes false after going true. The DMA device selected must then place control and address information on the lines common to all DMA devices. For a write cycle request, the DMAWRITE line must be true until the leading edge of DMAMDS/. The program protect line DMAPROT/ and write data lines DMADxx must also be stable over the same period.

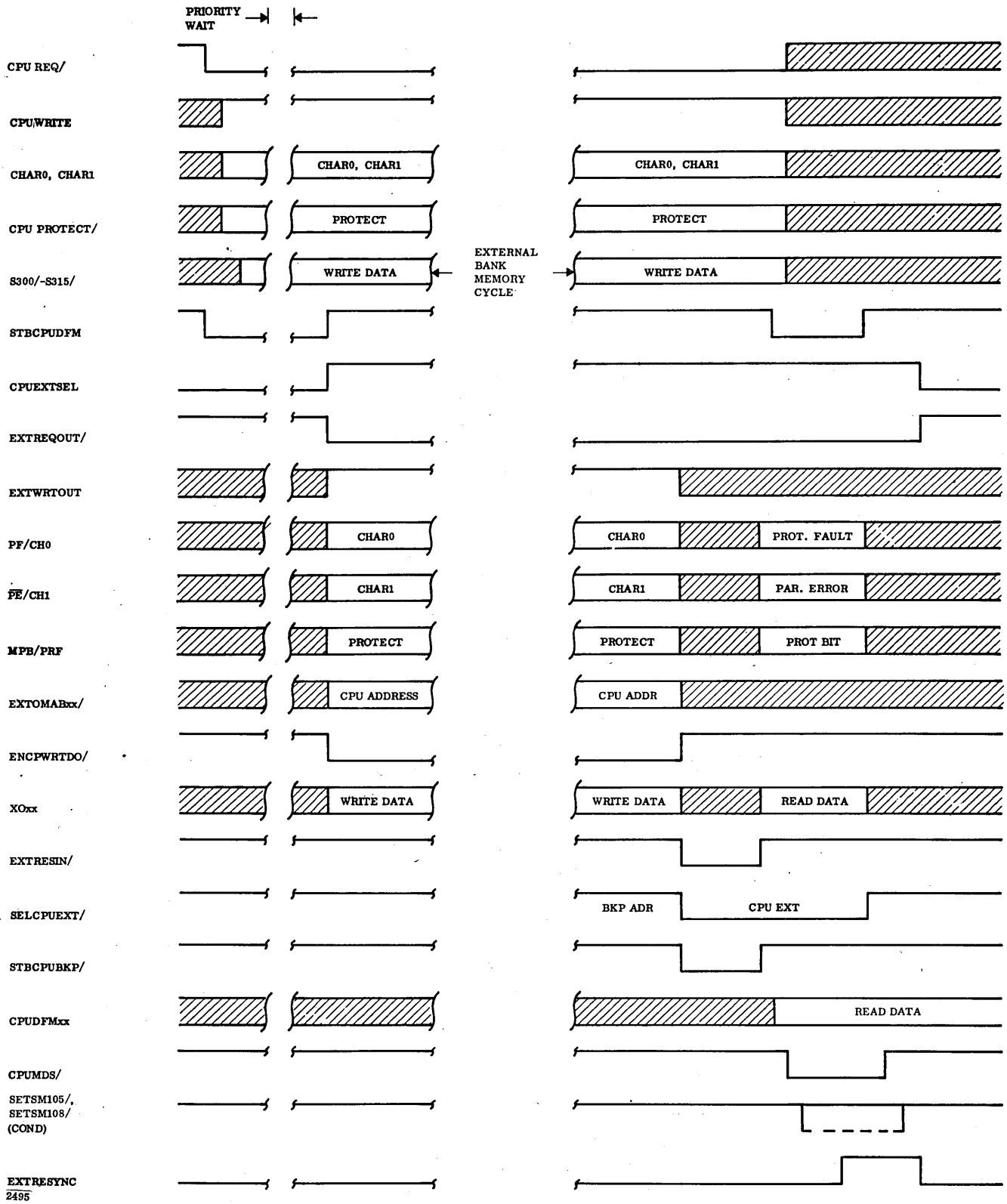


Figure 4-10. CPU Write Cycle Timing (External)

TABLE 4-10. CPU WRITE CYCLE TIMING (EXTERNAL)

Signal	Description
CPUREQ/	CPU memory request. Must remain stable until leading edge of CPUMDS/.
CPUWRITE	Read/write control. Must remain true from 25 nanoseconds after CPUREQ/ until leading edge of CPUMDS/.
CHAR0, CHAR1	Write character 0 and character 1 control. Must remain stable from 25 nanoseconds after CPUREQ/ until leading edge of CPUMDS/.
CPUPROTECT/	Program protect line. Must remain stable from 25 nanoseconds following CPUREQ/ until leading edge of CPUMDS/.
S300/ - S315/	CPU write data lines. Write data must be stable from 170 nanoseconds (should be less than 50 nanoseconds to generate parity) after leading edge of CPUREQ/ until 25 nanoseconds after leading edge of CPUMDS/.
STBCPUDFM/	First pulse strobes breakpoint address into CPU output register if previous CPU cycle was a read cycle. Second pulse strobes read data into CPU output register.
CPUEXTSEL	Enables CPU memory address to lines EXTOMABxx/ and CPU control lines to external bank.
EXTREQOUT/	Memory cycle request sent to external bank.
EXTWRTOUT	Read/write control line to external bank. Set to level of CPUWRITE by CPUEXTSEL.
PF/CH0	Dual-purpose line. Set to level of CHAR0 by CPUEXTSEL until EXTRESIN/. Contains conditional protect fault signal from external bank for 100 nanoseconds following trailing edge of EXTRESIN/.
$\overline{PE}$/CH1	Dual-purpose line. Set to level of CHAR1 by CPUEXTSEL until EXTRESIN/. Contains conditional parity error signal from external bank for 100 nanoseconds following trailing edge of EXTRESIN/.
MPB/PRF	Dual-purpose line. Set to complement of CPUPROTECT/ by CPUEXTSEL until EXTRESIN/. Contains level of protect bit from external bank for 100 nanoseconds following trailing edge of EXCTRESIN/.
EXTOMABxx/	Memory address lines 19 through 01 to external bank.
ENCPWRTDO/	Enables write data from lines S300/ through S315/ to lines XOxx.
XOxx	Dual-purpose data lines 16 through 01 to/from external bank. Contain write data from EXTREQOUT/ until EXTRESIN/. Contain read data for 100 nanoseconds after trailing edge of EXTRESIN/.
EXTRESIN/	Memory resume signal from external bank.
SELCPUEXT/	Selects external read data to CPU output register when true, breakpoint address when false.
STBCPUBKP/	Leading edge indicates CPU breakpoint address compare is available on CPUDFMxx lines until trailing edge.
CPUDFMxx	CPU data output lines 16 through 01 from data interface. Contain read data after leading edge of CPUMDS/.
CPUMDS/	Memory data resume. Control lines from CPU need no longer be stable. Read data available at leading edge.
SETSM105/	Protect fault line. Goes true 60 nanoseconds following trailing edge of EXTRESIN if the PF/CH0 line is true.
SETSM108/	Parity error line. Goes true 60 nanoseconds following trailing edge of EXTRESIN/ if the $\overline{PE}$/CH1 line is true.
EXTRESYNC	Resync signal to external priority select latches. Trailing edge allows next CPU or DMA external request to be processed.

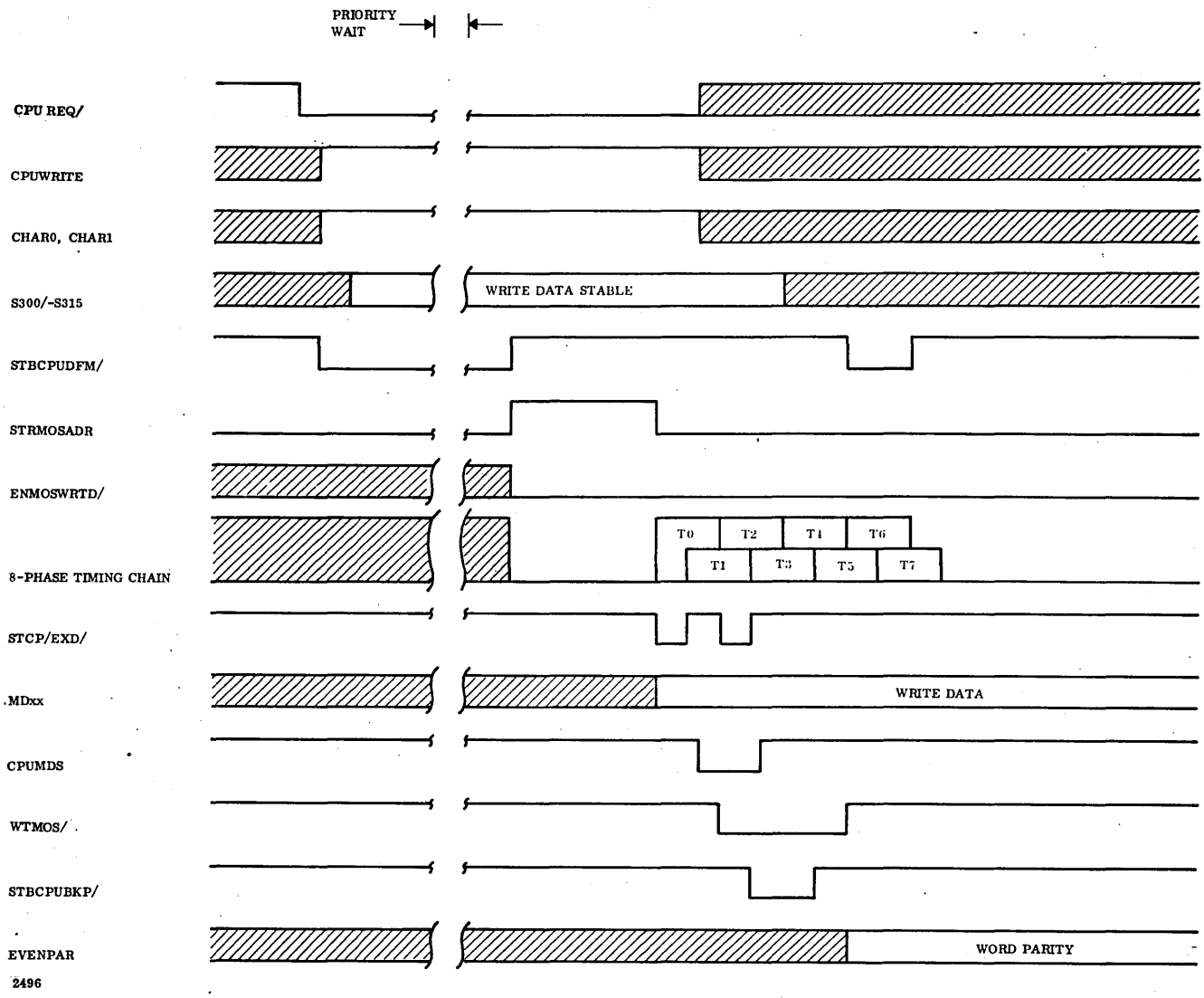


Figure 4-11. CPU Write Cycle Timing (Local, Protection Disabled)

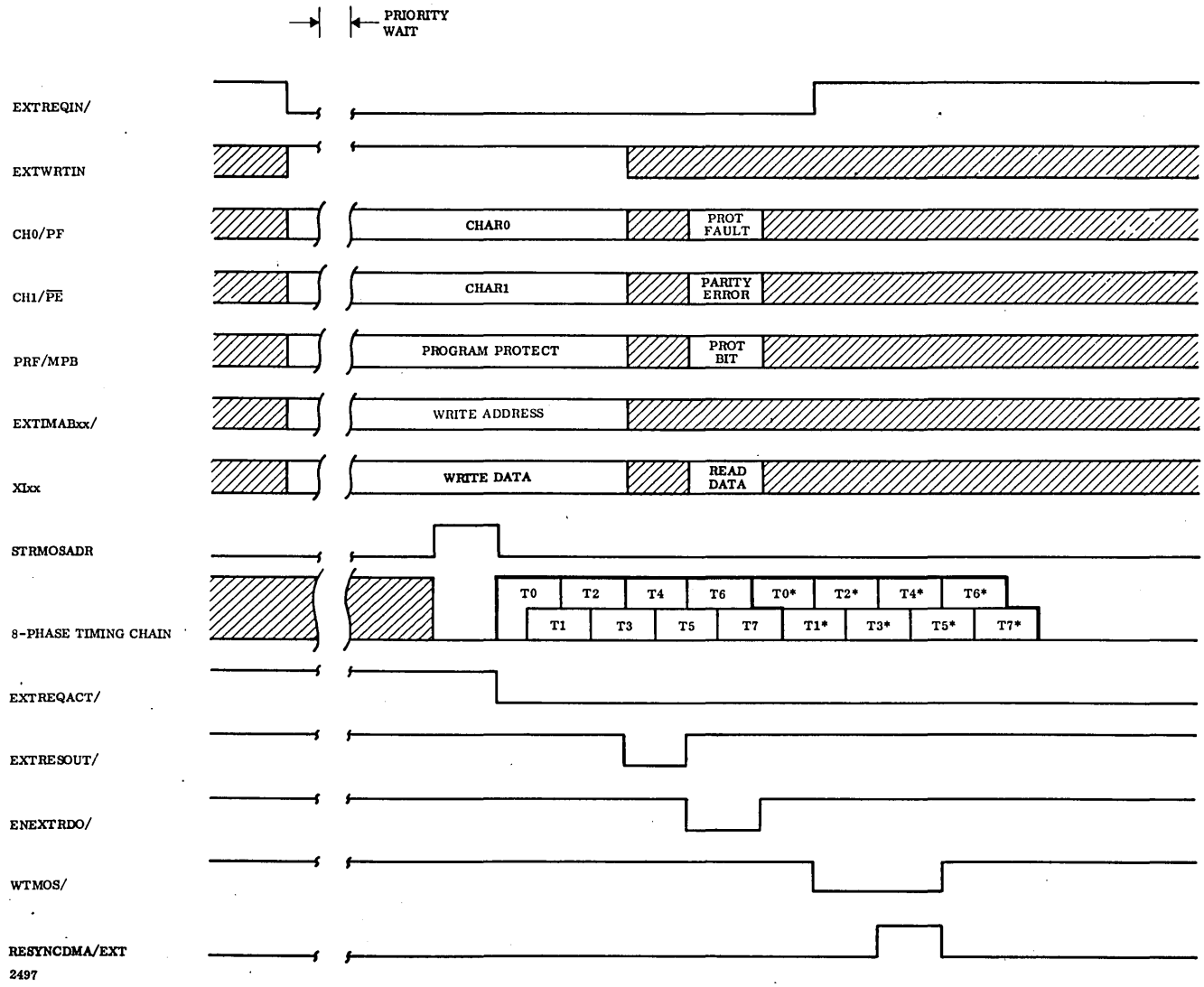


Figure 4-12. External Write Cycle Timing (Protected)

TABLE 4-11. CPU WRITE CYCLE (INTERNAL, PROTECTION DISABLED)

Signal	Description
CPUREQ/	CPU memory request. Must remain stable until leading edge of CPUMDS/.
CPUWRITE	Read/write control. Must remain true from 25 nanoseconds after CPUREQ/ until leading edge of CPUMDS/.
CHAR0, CHAR1	Both character write lines must be true from 25 nanoseconds after CPUREQ/ until leading edge of CPUMDS/.
S300/ - S315/	CPU write data lines. Must be stable from 100 nanoseconds after CPUREQ/ until 50 nanoseconds after trailing edge of CPUMDS/.
STBCPUDFM/	First pulse strobes breakpoint address into CPU output register if previous cycle was a read cycle. Second pulse unused.
STRMOSADR	Begins memory cycle following acceptance of request. Latches selected address in address register.
ENMOSWRTD	Enables write data to MDxx lines.
STCP/EXD/	Strobes write data from S300/ through S315/ lines into CPU write data input register.
MDxx	Write data lines to memory.
CPUMDS/	Memory data resume. Control lines from CPU need no longer be stable.
WTMOS/	Write enable line to memory arrays. True from T2 to T6 time.
STBCPUBKP/	If previous cycle was a read cycle, leading edge indicates breakpoint address available on CPUDFMxx lines until trailing edge.
EVENPAR	Write data word parity. True if number of 1 bits in write data is even.

TABLE 4-12. EXTERNAL WRITE CYCLE TIMING (PROTECTED)

Signal	Description
EXTREQIN/	Memory cycle request from external bank.
EXTWRTIN	Read/write control from external bank; true for a write cycle.
CH0/PF	Write character 0 control line until T4 time; protect fault conditional output during ENEXTRDO/.
CH1/$\overline{PE}$	Write character 1 control line until T4 time; parity error conditional output during ENEXTRDO/.
PRF/MPB	Protect control line until T4 time; memory protect bit 17 during ENEXTRDO/.
EXTIMABxx	Memory address lines 19 through 01 from external bank.
XIxx	Write data input lines 16 through 01 from external bank until T4 time; read data output lines during ENEXTRDO/.
EXTREQACT/	Enables write data from XIxx lines to memory arrays.
ENEXTRDO/	Enables read data from memory arrays to lines XIxx from T6 time to T6+100 nanoseconds.
WTMOS/	Write enable line to memory arrays; true from T2* to T6* time.
RESYNCDMA/EXT	Leading edge resets refresh, external and DMA request latches in priority circuit at T4* time.

A timing diagram for a DMA internal protected write cycle is shown in figure 4-13, with signal descriptions in table 4-13. A DMA external write cycle is similar to an internal cycle for signals to and from the DMA devices, and internal timing is similar to a CPU internal write cycle, discussed previously. A shortened DMA unprotected write cycle is possible when the ENPRTSYS line is false, with internal timing similar to a CPU unprotected write cycle.

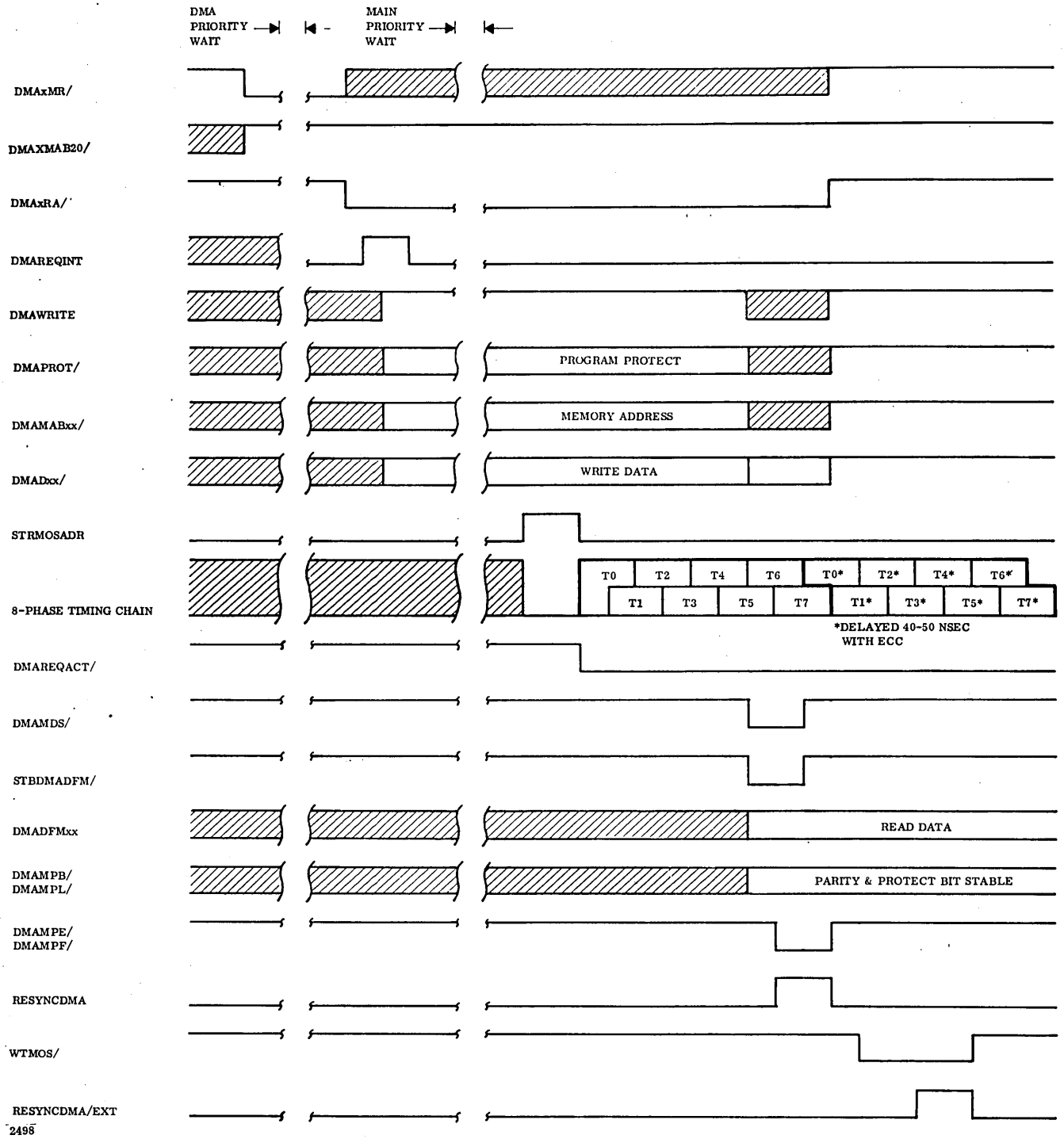


Figure 4-13. DMA Write Cycle Timing (Internal, Protected)

TABLE 4-13. DMA WRITE CYCLE TIMING (INTERNAL, PROTECTED)

Signal	Description
DMAxMR/	Memory cycle request from one of four (1-4) DMA devices. Must remain true until the associated DMAxRA/ signal is sent, and must return false within 50 nanoseconds after leading edge of DMAMDS/.
DMAxMAB20/	Local/external bank select line (1-4). Must remain false for internal bank request from DMAxMR/ going true until DMAx RA/ goes false.
DMAxRA/	Request accept signal (1-4). Set true when the request is accepted by the DMA priority circuits, and instructs DMA device to place address, write data, and control levels on the common lines within 50 nanoseconds. Returns false at T6+50 nanoseconds, trailing edge.
DMALRQ	DMA local request pulse to main priority select circuits following selection of highest priority DMA request.
DMAWRITE	Read/write control line. Must be true within 50 nanoseconds following DMAxRA/, remain true until DMAMDS/, and must be disabled within 50 nanoseconds after the trailing edge of DMAxRA/.
DMAPROT/	Program protect line. Must remain stable from 50 nanoseconds following DMAx RA/, remain stable until DMAMDS/, and be disabled within 50 nanoseconds after the trailing edge of DMAxRA/.
DMAMABxx/	DMA memory address bus (19-01). Must contain memory address within 50 nanoseconds following DMAxRA/, remain stable until DMAMDS/, and be disabled within 50 nanoseconds after the trailing edge of DMAxRA/.
DMADxx/	DMA write data lines. Write data must be stable within 50 nanoseconds following DMAxRA/, remain stable until DMAMDS/, and be disabled within 50 nanoseconds after the trailing edge of DMAxRA/.
STRMOSADR	Begins memory cycle following acceptance of DMALRQ request. Latches memory address in address register.
DMAREQACT/	Enables write data from DMADxx/ lines to memory arrays.
DMAMDS/	Memory data resume. Write data and control lines from DMA device no longer needed. Read data stable within 17 nanoseconds after leading edge.
STBDMADFM/	Strobes read data into DMA output register at T6 time.
DMADFMxx	DMA read data lines 16 through 01. Stable within 17 nanoseconds of DMAMDS/ for a minimum of 340 nanoseconds.
DMAMPB/	Memory protect bit output line.
DMAMPL/	Write data word parity output line. True if number of 1 bits in write data word is even.
DMAMPF/	Protect fault line. Goes low at T6+50 nanoseconds if a protect fault is detected.
DMAMPE/	Goes true at T6+50 nanoseconds for 80 to 100 nanoseconds if parity error is detected.
RESYNCDMA	Leading edge resets DMA priority latches, allowing next DMA request to be selected.
WTMOS/	Write enable line to memory arrays; true from T2* to T6* time.
RESYNCDMA/EXT	Leading edge resets refresh, external, and DMA request latches in priority circuit at T4* time.

If an external bank is selected but no external bank is connected, the memory address error line DMAMAE/ is set true within 50 nanoseconds after DMAxRA/. An internal read cycle from the same address is performed, and the DMAMAE/ signal is cleared at DMAMDS/ time.

After the DMA request has been accepted by both the DMA and main priority select circuits, write data from the lines DMADxx/ is latched in the DMA write data input register and the parity is determined. Read data is strobed into the DMA output register by the line STBDMADFM/. A DMAMDS/ signal is sent to indicate that address, write data, and control lines are no longer needed from the DMA; read data is available on the DMADFMxx output, and word parity as well as memory protect bit status are available over lines DMAMPL/ and DMAMPB/. The parity error (DMAMPE/) and protect fault (DMAMPF/) lines indicate the appropriate status approximately 50 nanoseconds later by an 80- to 100-nanosecond pulse.

Write data is then gated to an array and the WTMOS/ control line is set true to write the data into the selected location. This signal is inhibited, leaving the memory contents unchanged, if a protect fault or protect fault or protect parity error is encountered (see Protection below).

SPLIT CYCLE OPERATION

Two types of split cycle operation are utilized in the LMS. The automatic read/write split cycle, used in a protected write operation, is discussed above under Write Requests. However, a read/modify/write cycle may be requested that allows certain controls and data to be changed between the read and write portions. The request is made via the split cycle control line from the CPU or EXT port, and either the internal or remote bank may be addressed.

The first portion of the cycle is a read portion, with the same input and output requirements as a standard read cycle, except that the split cycle control line (CPUSC/) must remain true throughout the read portion and the read/write control line must stay false.

After the read cycle has ended, the LMS halts. The read/write control line must be held false during the modify period, as this line going true triggers the write portion of the cycle. As the read information is processed, changes in the write data and character write control lines may be made. The write portion may not be aborted, but the original data may be restored to memory by having both write character lines pulled false by the requesting device.

The write portion of a read/modify/write split cycle is initiated by the write line going true, and the split cycle control line must return false within 50 nanoseconds of that time. Normally, a read/write automatic split cycle constitutes the write portion of a read/modify/write requested split cycle, as shown in the timing diagram of figure 4-14, and in the signal descriptions of table 4-14.

WORD/CHARACTER MODE

Certain applications utilize memory more efficiently in eight-bit characters than in 16-bit words. The character mode of operation allows memory information to be modified one 8-bit character at a time.

Character 0 is the most significant eight bits of the data word, received by the CPU port on lines S300/ through S307/. Character 1 is the least significant eight bits. Read operations are not affected by the character mode, and the CPU must extract the desired character from the full 16-bit word. Character write operations are accomplished by reading the addressed word, then writing a new word consisting of the one new character along with the remaining eight bits just read from the addressed location. This rewriting of read data is known as a restore operation. Thus, in a character 0 write cycle character 0 is written with new information while character 1 is restored.

The character write control lines CHAR0 and CHAR1 direct write operations in memory according to the following truth table:

CHAR0	CHAR1	Write Function
True	True	Full word write
True	False	Write character 0, restore character 1
False	True	Write character 1, restore character 0
False	False	Restore entire word

The full word restore operation may be used in read/modify/write split cycle requests when the CPU chooses not to write new information, thus effectively aborting the write portion of the cycle.

When either or both of the character write control lines is false, a read/write (automatic split cycle) is always performed during a write request, even if protection is disabled by a false ENPRTSYS line.

ABSOLUTE /PAGE MODE

All addresses from the DMA and external ports use a direct 19-bit address, and the DMA devices include an extra line for bank selection so that the full 1M-word expanded memory may be addressed. The CPU, however, is limited to a 16-bit address and therefore must use an indirect addressing scheme to fully utilize the memory capability. The page mode is used only for CPU requests.

Absolute mode refers to the use of a direct 16-bit addressing method that the CPU may utilize to access the first 64K words of memory. Page mode refers to the indirect addressing method in which the least insignificant 11 bits (2K words) are directly addressed but the most significant nine bits are supplied by a random-access memory known as the memory page file. Thus, the memory is divided into 2K-word segments, or pages, with the page address stored in the memory page file.

Up to 64 page addresses may be stored in the file, out of the 512 pages possible. The memory page file is divided into two 32-word locations, with the upper half known as page set zero and the lower half as page set one. Each 9-bit word in the memory page file is a 9-bit page address. The memory page file is accessed and controlled through micro instruction register lines MIR28' through MIR31'.

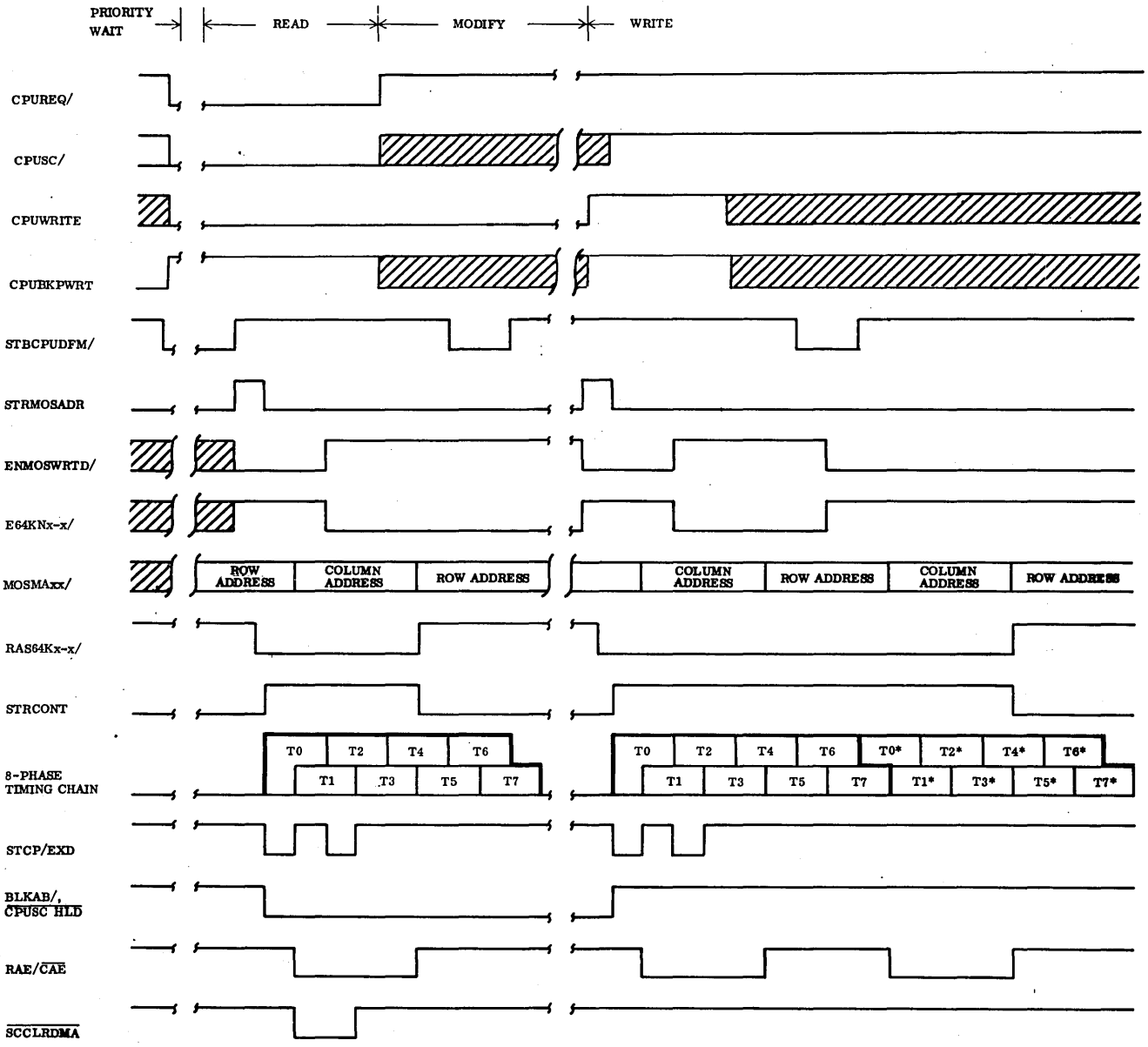


Figure 4-14. CPU Split Cycle Read/Modify/Write (Protected) (Sheet 1 of 2)

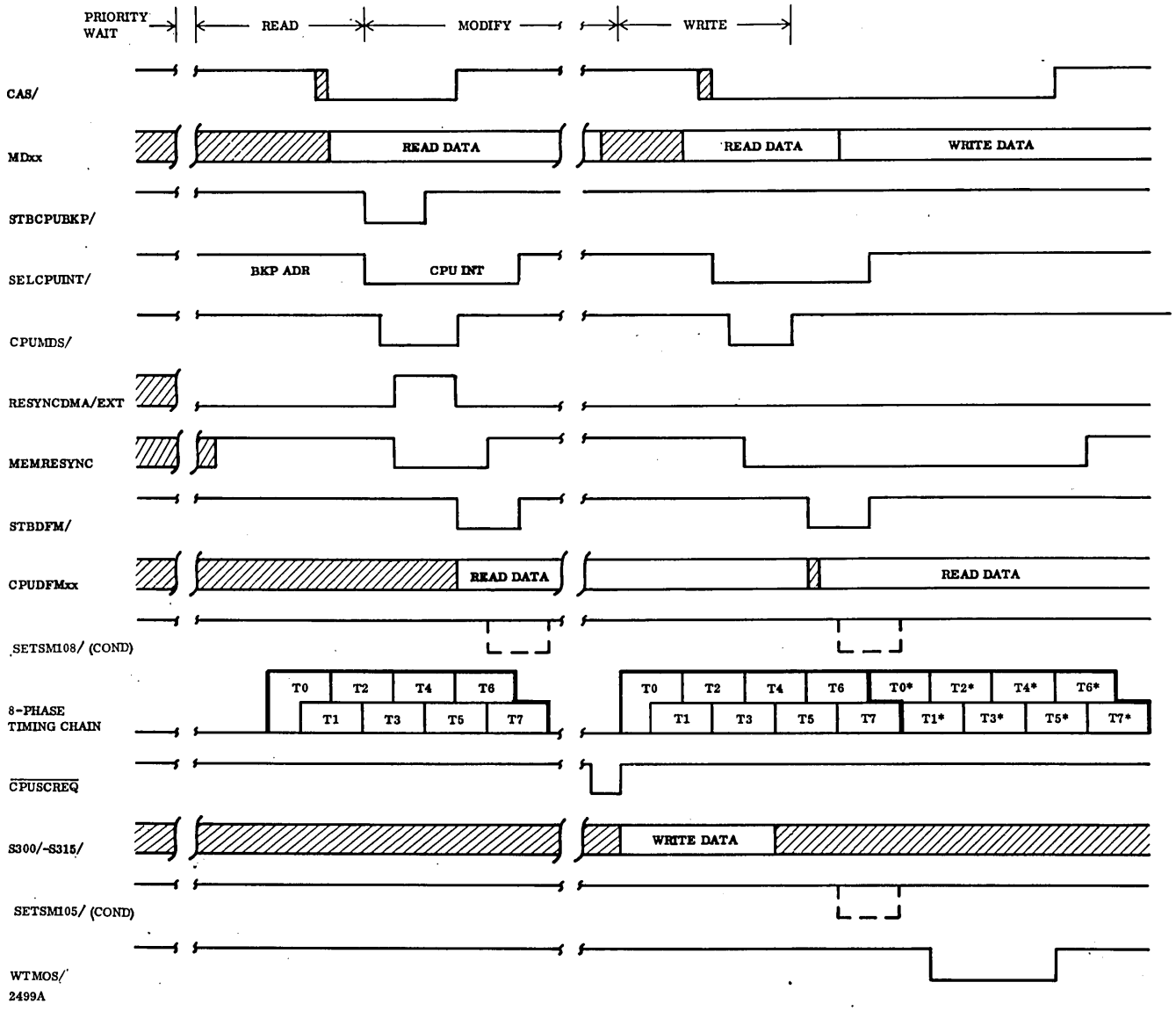


Figure 4-14. CPU Split Cycle Read/Modify/Write (Protected) (Sheet 2 of 2)

TABLE 4-14. CPU READ/MODIFY/WRITE SPLIT CYCLE (PROTECTED)

Signal	Description
CPUBKPWRT	CPU breakpoint write control line becomes true whenever the CPUSC/ is false or when the $\overline{\text{CPUWRT}}$ condition is false.
CPUREQ/	CPU memory request. Must stay true until leading edge of first CPUMDS/ signal.
CPUSC/	CPU split cycle control line. Must stay true from 25 nanoseconds following CPUREQ/ until the first CPUMDS/. Must return false before 10 nanoseconds following change of CPUWRITE to a true level.
CPUWRITE	Read/write control line. Must be false until first CPUMDS/. Change of state to true level initiates write portion of cycle.
STBCPUDFM/	First pulse strobes breakpoint address into CPU output register if previous CPU cycle was a read cycle. Second and third pulses strobe read data into CPU output register.
STRMOSADR	Begins read memory cycle after acceptance of request. Second pulse begins write cycle following initiation by write line.
ENMOSWRTD/	First high portion enables E64KNx-x/ output to memory array for read data. At the start of STRMOSADR in the write portion a true level enables write data input for parity generation. Returns false at T2 time to enable read data from memory, and returns true to enable write data at T6+50 nanoseconds.
E64KNx-x/	Transmit enables (0-1 through 1-4) place read data to MDxx lines.
MOSMAxx/	Row and column address lines (06 through 00) to memory arrays.
RAS64Kx-x/	Row address strobes (0-1 through 1-4).
STRCONT	Timing pulse used to set status latches.
T0-T7, T0*-T7*	Eight-phase timing chain. Run once for read portion of split cycle; run twice during write portion. T0*-T7*, the write section of the write cycle, begins at end of T6.
STCP/EXD/	Strobes write data from S300/ through S315/ into CPU write data storage register.
BLKAB/	Inhibits change of memory address by CPU until split cycle is complete.
$\overline{\text{CPUSCHLD}}$	Generates resync and inhibit signals to block other requests from priority circuits until split cycle is complete.
RAE/$\overline{\text{CAE}}$	Row address enabled to MOSMAxx/ lines when true, column address when false.
$\overline{\text{SCCLRDMA}}$	Resync pulse at T1 time to priority circuits to clear any waiting DMA request.
CAS/	Column address strobe to memory array. Shaded portion shows early strobe used with fast arrays.
MDxx	Memory data lines 15 through 00 to memory arrays. Contain read data from first T2 time until write line goes true; read data again from the second T2 time until T6+50 nanoseconds, then write data.
STBCPUBKP/	If previous cycle was a read cycle, leading edge of pulse indicates breakpoint address is available on CPUDFMxx lines until trailing edge.
SELCPUINT/	Selects internal read data to CPU data output register when true, breakpoint address when false. Data is strobed into register by STBCPUDFM/.

TABLE 4-14. CPU READ/MODIFY/WRITE SPLIT CYCLE (PROTECTED) (Contd)

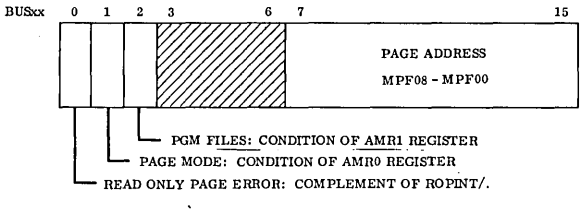
Signal	Description
CPUMDS/	Memory data strobe. First pulse indicates read data available within 5 nanoseconds after trailing edge on CPUDFMxx lines, and CPUWRITE may go true any time thereafter to begin write cycle. Second pulse indicates control lines from CPU are no longer needed, and that read data from write portion of cycle available is 5 nanoseconds after trailing edge.
RESYNCDMA/EXT	Resync pulse during read portion to clear waiting DMA or higher priority requests.
MEMRESYNC	Resync pulses to priority select circuits.
STBDFM/	Timing signals to data interface.
CPUDFMxx	CPU data output lines 16 through 01 from data interface. Contain read data 5 nanoseconds after trailing edges of CPUDMS/.
SETSM108/	Conditional parity error signal. Goes true at T6+50 nanoseconds time if a parity error is detected on read and write portions.
$\overline{\text{CPUSCREQ}}$	Internal memory request generated to begin write portion of split cycle when write line goes true.
S300/ - S315/	Write data lines from CPU. Must contain write data within 50 nanoseconds after change of write line to true state until following CPUMDS/ signal.
SETSM105/	Conditional protect fault signal. Goes true at T6+50 nanoseconds during write portion if a protect fault is detected.
WTMOS/	Write enable line to memory arrays. True from T2* until T6* time during write cycle.

Loading the Memory Page File

The memory page file is loaded with two sequential micro commands. The first, GATE/58 (line MIR28' true while the GATEUBDS/ line is brought true), places the memory page file in write mode. Next, a standard CPU write operation is commanded, with the memory page file address contained in AB00 through AB05 (U/L select) and the memory page file data in AB07 through AB15. See figure 4-15 for the memory page file loading and address format. While the memory page file is being loaded, the LMS executes a dummy read cycle from a location in which the memory page file loading code is interpreted as an absolute address.

Memory Page File Status

The contents of the memory page file may be checked by two sequential micro commands, GATE/58 followed by READ, which enables the following output of the BUSxx lines:



Page Mode Selection

The page mode is a latched condition that is set with a GATE/56 or GATE/57 command (GATEUBDS/, MIR29' true) with MIR30' true, setting the AMR0 register. The page mode remains in force for all subsequent CPU requests until a GATE/54 command is given (MIR30' false) or a master clear signal clearing the AMR0 register. The page mode status has no effect on requests from the DMA or external ports.

On the same command, the AMR1 register is loaded with the level on the MIR31' line. A true level selects the upper half or set zero, a false level selects the lower half or set one.

Page Mode Read/Write

Once page mode has been selected, a CPU memory cycle must be preceded in the normal manner by a CPU memory address loading consisting of a GATEAB command, loading the contents of the address register with the contents of lines S300/ through S315/. Lines S300/ through S304/ contain the 5-bit memory page file address, while lines S305/ through S315/ contain the word address within the selected 2K-word page.

It is the output of this address register that is compared with the breakpoint address (see Protection, below) to determine an in- or out-of-bounds condition. Thus in page mode the bounds comparison is made against the logical address from the CPU rather than the physical memory address.

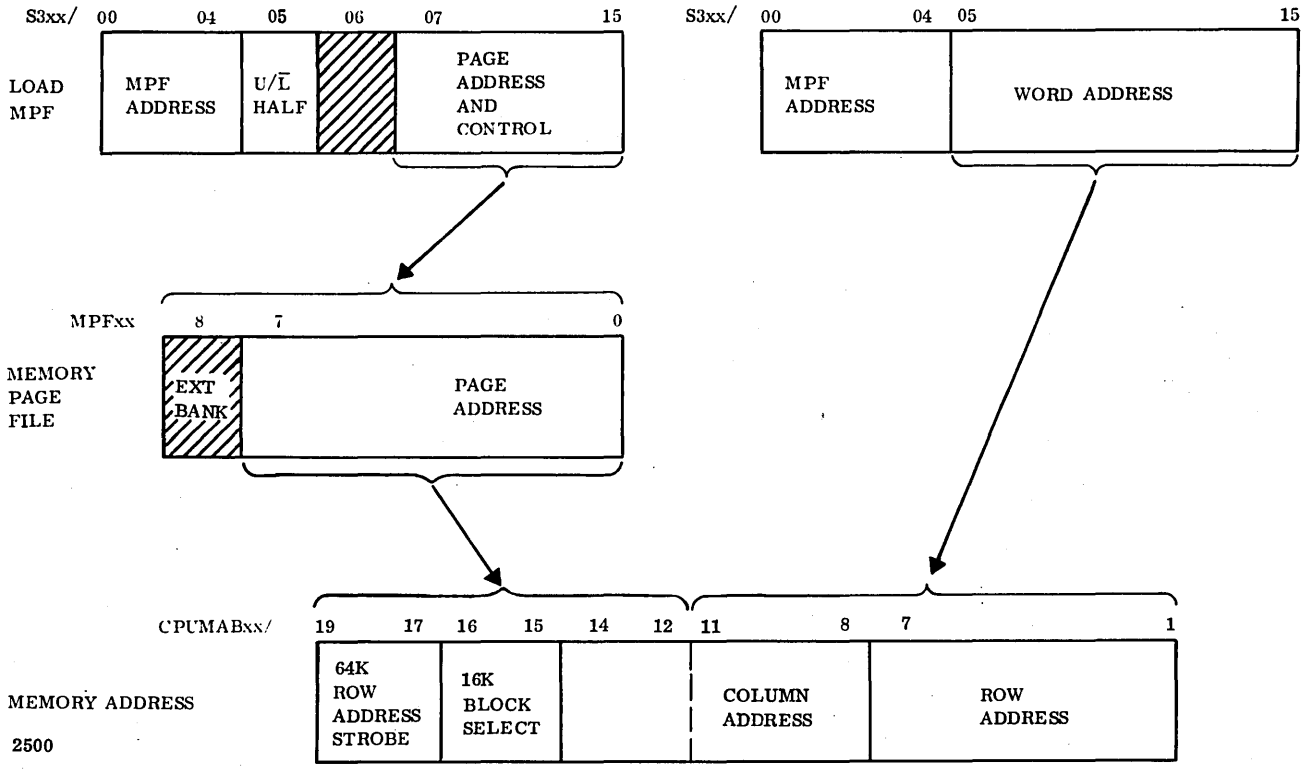


Figure 4-15. Page Mode Address Formats

MULTILEVEL INDIRECT MODE

If multilevel indirect addressing is selected by the CPU (MULTIND line true), the most significant bit of the CPU address register and both bounds registers is forced false. The following conditions would then apply:

- Absolute mode - The CPU maximum addressable physical memory is the least significant 32K words of the internal bank, as only a 15-bit address is being utilized. Note that since the breakpoint comparison addresses are also affected, even if the inbounds segment is in the second 32K of memory a duplicate in-bounds segment is established in the first 32K when in multilevel indirect mode.

- Page mode - The addressable pages are limited to the least significant 16 set zero pages and 16 set one pages, thus limiting the immediate addressing space to 32K words. However, each page may be located anywhere in the possible 1M words of memory.

PROTECTION

Memory protection is a method of identifying certain words or groups of words as being protected from ordinary write operations. The method used is a protect bit stored in the most significant bit position (bit 17) of each 18-bit memory word. When the protect bit is true, the word is considered a protected location. Read operations are not affected by the condition of the protect bit, but provide the status of the protect bit over the lines MEMPROTBT/ to the CPU port, DMAMPB/ to the DMA port, or PRF/MPB to the external port. (An exception to this is that on a CPU in-bound condition, the MEMPROTBT/ line is false regardless of the status of the protect bit.) Timing for the memory protect bit output is the same as for read data output for the type of cycle requested.

Protect Bit Write

To change the condition of the protect bit, the write protect bit line from the CPU must be true and a write cycle must be requested. The protect bit then assumes the level of S306. The condition of the protect bit remains unchanged through subsequent write operations until the write protect bit function is again employed. (The ENPRTSYS line false is an exception; see Protect System Disable below.) A protect bit write operation is aborted if a parity error exists.

Write Operations to Protected Location

To write to a protected location requires a true program protect line during the write operation. The condition of the protect bit remains unchanged unless a protect bit write operation is requested.

Port	Program Protect Line
CPU	CPUPROTECT/
DMA	DMAPROT/
External	PRF/MPB

Writing to a protected location without a true protect line may only be accomplished when the ENPRTSYS line is false, which disables the normal protection system, and on a CPU in-bounds condition (see below).

Protect Faults

When write operations to a protected location are attempted and the conditions for a successful write operation are not met, the write operation is aborted. The write cycle is completed, but the write enable line WTMOS/ to the memory arrays is inhibited.

A protect fault signal is also generated to the requesting port to indicate the aborted operation. Timing for protect fault signals for the various types of write cycles are shown in the timing diagrams earlier in this section.

Port	Protect Fault Line
CPU	SETSM105/
DMA	DMAPF/
External	CH0/PF

CPU Bounds Address Operation

The CPU may establish up to 64K of memory as an unprotected area. The bounds addresses are loaded into the upper or lower bounds registers by the GATEUBDS/ or GATELBDS/ signals, respectively, which load the selected register with the contents of multipurpose lines S300/ through S315/. This operation may be made at any time except when a CPU memory request is active.

When an absolute address is loaded into the CPU address register by the GATEAB/ signal, that address is compared with the addresses in the bounds registers. When the address is lower than the upper bounds address and higher than the lower bounds address, the location is considered to be in bounds.

On a write cycle, an in-bounds condition has exactly the same effect as if the program protect line were true. There is no effect upon a read cycle.

In page mode, the bounds area refers to a location within the memory page file rather than a physical memory location. Therefore, the protected area may be scattered throughout the physical memory. In general, a bounds area should not be established (set upper bounds to zero) if absolute and page mode addressing are to be intermixed.

Protect System Disable

The ENPRTSYS line may be brought false (by clearing the FCR protect switch bit 8) to cause the entire memory to be unprotected. Write requests from the CPU, DMA, or external ports are executed regardless of the state of the protect bit. When protection is disabled, a shortened write cycle is employed on full-word write requests, and the protect bit is set true in each memory location referenced. There is no effect on a read operation.

If the external bank is referenced, the write operation is still executed regardless of the state of the protect bit, but the short cycle and the setting of the protect bit does not take place unless the ENPRTSYS line is also false on the external bank. In general, however, the ENPRTSYS lines to the two banks are at the same level.

ERROR DETECTION (PARITY AND ECC)

Error detection in the LMS is accomplished by a parity bit stored with each data word. The parity bit is set to produce an odd parity (odd number of ones) in the 18-bit memory word. The use of a parity bit detects single-bit errors, which are the most common.

Word Parity

A word parity bit is generated for each 16-bit write data word received by the data interface to generate an odd parity. This parity bit has no effect on memory operation, but is fed back to the requesting port so that the requesting device can determine if the data word was received correctly. Timing for the word parity output is the same as for the read data output for the type of cycle requested.

Port	Word Parity Output Line
CPU	EVENPAR
DMA	DMAMPL/
External	SC/MPL

Parity Error

The parity bit generated to be stored in memory bit 16 differs from the word parity bit in that the parity includes the protect bit. Therefore, the memory parity bit is the same as the word parity bit if the protect bit is false, and is the complement of the word parity bit if the protect bit is true.

A parity error is detected by regenerating a word parity bit from the 16 bits in the data word read from memory. Parity is then determined by three bits: word parity, memory parity (bit 16), and the protect bit (bit 17). An even parity among these three bits passes the test; an odd parity indicates an error.

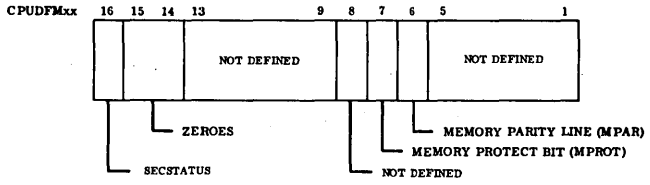
Parity error signals are available to the requesting device on both read and normal write cycles. Timing for the various cycles is shown in the appropriate timing diagram earlier in this section.

Port	Parity Error Output Line
CPU	SETSM108/
DMA	DMAMPE/
External	CH1/PE

On normal write cycles, the write portion of the cycle is aborted if a parity error is detected on the read portion unless the protect line is true or protection is otherwise disabled.

Status

The status of some internal bank signals may be referenced by the CPU at any time with a GATE STATUS (GATE/68) command (GATELBDS/ and MIR28' true). This command places status information on the BUSxx outputs according to the following format:



Status information remains selected until a GATEAB/ command is received. Status information is not available from the external bank. SECSTATUS is cleared after the completion of every GATE STATUS command.

DUAL CPU OPERATION

A CPU may be connected to each CPU port in a dual bank system. Although both ports may operate independently, certain control lines should be tied together on the two ports and controlled by one of the two CPUs, designated as the master. These lines are:

- MC/ - Master clear; both banks should be cleared simultaneously
- ENPRTSYS - Enable protect system

ADDRESS INTERFACE

The address interface provides timing and control for the LMS, handling and decoding of memory addresses, and priority selection of incoming memory cycle requests. Refer to the simplified block diagram of figure 4-16.

A delay line oscillator generates an on-board eight-phase timing chain that produces the primary timing signals to the control logic.

A power-on master clear/Master Clear from the CPU resets many of the latches in the control logic to provide a fresh start when necessary without disturbing memory. Data stored in memory is volatile only during power on/off without battery backup.

The control and timing logic receives control information from the CPU, DMA, and external inputs, synchronizes subsequent LMS operation to the eight-phase timing chain, and transmits status control signals back to the requesting device.

The address interface also contains a refresh timer and address counter, refreshing all row addresses in the memory array approximately every 2 milliseconds.

Priority select circuits operate on the memory request signals from the CPU, the four DMA inputs, the external memory port, and the refresh timer. The highest priority active request is determined, and the appropriate port is selected to provide the memory address, data, and read/write request.

The read/write logic on the address interface produces the control signals for the arrays.

Address signals from the three ports (DMA, external, CPU) and the refresh address counter appear at the inputs to the memory address register/selector. One of the four inputs is selected by the priority select circuits. The selected address is then fed to the internal bank selector. If the external memory bank is requested by bit 20 of the CPU or DMA memory address, the priority is determined between DMA and CPU and the address is routed to the external memory bus output.

If the internal bank is selected, the address is routed to the address decoding circuits. These circuits interpret the 19-bit address into the array board, block, row, and column information needed by the arrays.

PORT SIGNALS AND TIMING

CPU Port

Activity at the CPU port begins with the loading of the least significant 11 bits of the memory address from CPU data lines S305/ through S315/ into the CPU address register, coincident with the GATEAB/ signal from the CPU. (The most significant bits are simultaneously loaded on the data interface and are available to the address interface over lines CPUMAB20/ through CPUMAB12/.) The CPU then initiates a memory cycle request with a true level on CPUREQ/. Within 25 nanoseconds following the leading edge of this signal, these control lines from the CPU must be at their proper value for the instruction:

CPUWRITE	True for WRITE, false for READ
CHAR0 and CHAR1	Both true for word read or write
CPUSC/	True for split cycle operation
CPUWRTPB	True to write protect bit
CPUPROTECT/	True for program protect

All of these lines, including CPUREQ/, must remain stable until the CPUMDS/ line goes true, signaling the CPU that all information from the CPU has been acted upon.

The CPUMDS/ line goes true for approximately 100 nanoseconds after the CPU request has been honored and the memory array addressed.

Several memory cycles may lapse between CPUREQ/ and CPUMDS/ under heavily loaded conditions, due to the low priority of the CPU. If the address is for the external bank, the priority selection for the CPU in both the internal LMS and the external LMS must be satisfied before the requested cycle can be completed.

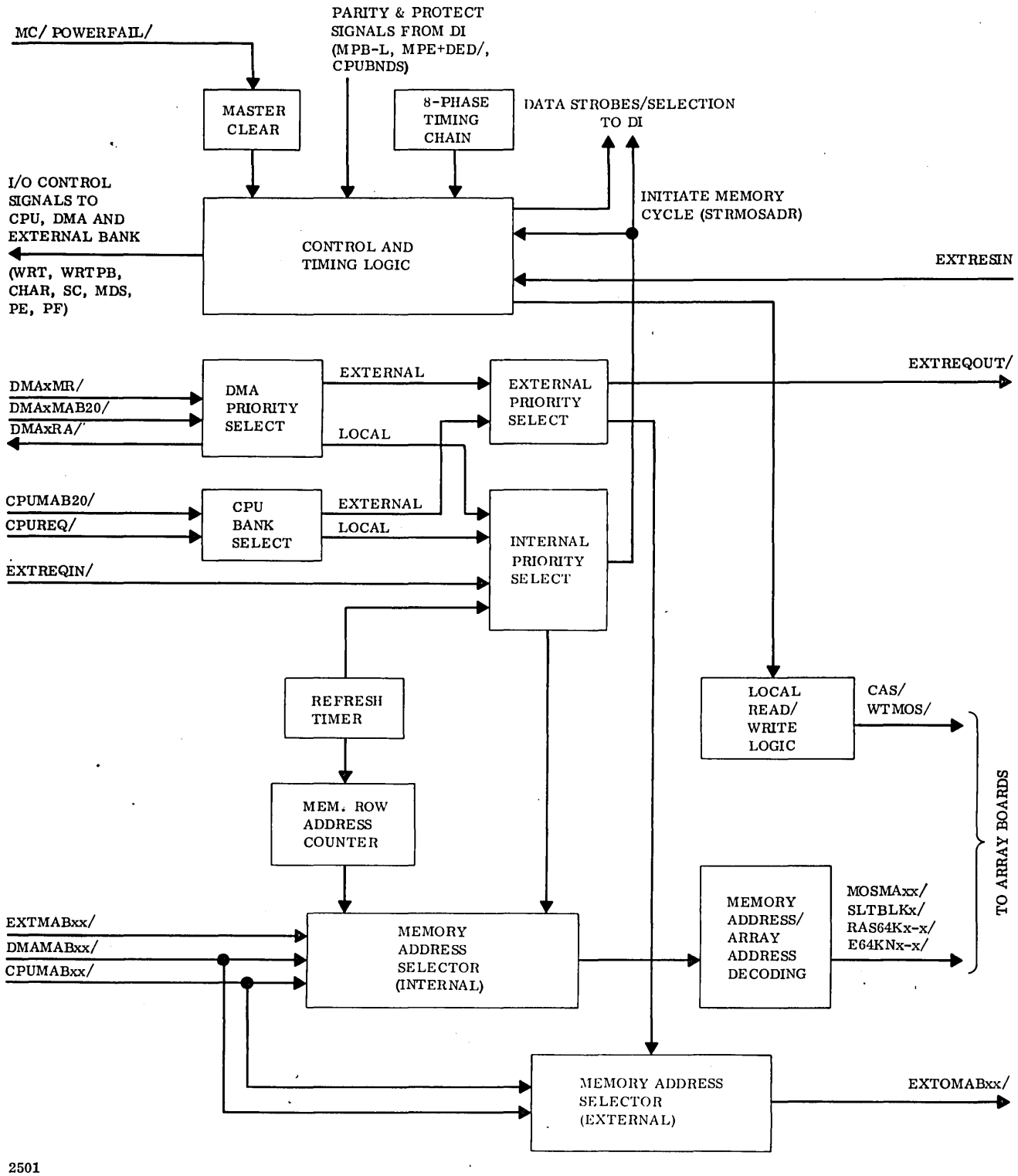


Figure 4-16. Address interface Simplified Block Diagram

Following the CPUMDS/ output, the results of a parity test failure (SETSM108/ true) and, on a write cycle, the protect fault indication (SETSM105/ true) appear on the appropriate lines to the CPU as approximately 100-nanosecond wide pulses. Read data and memory parity is also available from the data interface CPU port following the CPUMDS/ output.

When a CPU split cycle request has been made, the first CPUMDS/ output signals the CPU that the read portion of the cycle has been executed. The beginning of the write portion occurs when the CPU WRITE line changes from a false to a true state, after which a second CPUMDS/ signal is generated by the address interface to signal completion of the entire cycle.

The CPU port also includes input lines MC/ (master clear), which resets the latches in the LMS, and POWERFAIL/, which holds the LMS latches in a reset condition (with the exception of the refresh control logic) until power is restored to the CPU.

The line ENPRTSYS, when forced false by the CPU, overrides the usual protection methods and the LMS executes a full write cycle regardless of the state of the protect bit or DMA or CPU program protect lines.

Three clock lines from the CPU to the LMS (T0/, T2ODDCLOCK/, and T3MEM/) speed the CPU memory cycle by providing a resync window for the CPUREQ/ signal to LMS so that a new request can be acknowledged in the shortest possible time.

DMA Ports

Up to four peripherals may have direct memory access (DMA). Each has its own memory request line, labeled DMA1MR/ through DMA4MR/. The cycle begins with one or more of these request lines going true. Coincident with the request, the DMA device must indicate whether the internal or external bank is to be selected. The appropriate bank select line, DMA1MAB20/ through DMA4MAB20/, must be false for internal and true for external selection. If the external bank is selected but no external bank is connected, the DMA memory address error line DMAMAE/ is forced true, and the reference is made to the same address in the internal bank with write inhibited.

The request with highest priority (DMA1MR/ is highest) is accepted and the address interface responds with an appropriate request accept signal (DMA1RA/ to DMA4RA/ true).

Within 50 nanoseconds after the DMARA/ leading edge, the appropriate DMA device must place the proper signal levels on the common DMA control lines to execute the desired type of memory cycle:

DMAWRITE	True for WRITE, false for READ
DMAPROT/	True for program protect
DMAMAB19/ through DMAMAB01/	Memory address lines

All of these lines must remain stable as long as the DMARA/ line stays true.

The DMAMDS/ line goes true for approximately 100 nanoseconds after the DMA request has been honored and an array addressed. Following the DMAMDS/ output, the results of a parity test failure (DMAMPE/ true) and, on a write cycle, the protect fault indication (DMAPF/ true) appear on the appropriate lines to the DMA as approximately 100-nanosecond-wide pulses. Read data and memory parity is also available from the data interface DMA port following the DMAMDS/ output.

The line ENPRTSYS, when false, also affects DMA operation by overriding the usual protection method, allowing execution of a full write cycle regardless of the state of the protect bit or the DMA program protect line (DMAPROT/).

External Port

The external port is dual unidirectional and is used for output and input:

- To communicate CPU or DMA read/write commands to the external bank (external request out)
- To receive and act upon read/write commands from the external bank has its own DMA or CPU requests (external request in)

The two operations may be processed simultaneously.

A DMA or CPU external bank request is initiated as described above for DMA and CPU port timing. External bank selection is made by the appropriate DMAMAB20/ line or CPUMAB20/ line in a true state. The write, memory address, split cycle, character, protect bit, and program protect lines must all meet the same requirements as for an internal bank request.

After priority selection between DMA and CPU, the EXTREQOUT/ line to the external bank goes true, and the appropriate DMA or CPU memory address bus lines are gated onto lines EXTOMAB19/ through EXTOMAB01/, as well as the following control lines:

DMA	CPU	Gated to External Output Line
DMAWRITE	CPUWRITE	EXTWRTOUT
DMAPROT/	CPUPROTECT/	MPB/PRF

The external bank receives EXTREQOUT/ at its EXTREQIN/ input and, after the possible processing of a refresh request or the completion of a split cycle operation, processes the entered request based on these input lines of address and control information from the local requesting bank:

Internal Bank Outputs	External Bank Inputs
EXTOMAB19/ - EXTOMAB01/	EXTIMAB19/ - EXTIMAB01/
EXTWRTOUT	EXTWRTIN
MPL/SC	SC/MPL
PF/CH0	CH0/PF
PE/CH1	CH1/PE
MPB/PRF	PRF/MPB

Following the acceptance of the external request, the LMS completes the read or write cycle and sends EXTRESOUT/ back to the requesting LMS, followed by this status information:

External Bank Outputs	Internal Bank Inputs
SC/MPL	MPL/SC
CH0/PF	PF/CH0
CH1/PE	PE/CH1
PRF/MPB	MPB/PRF

The EXTRESOUT/ signal is received by the internal bank on the EXTRESIN/ line. The address interface in turn generates a CPUMDS/ or DMAMDS/ signal to the requesting port, followed by the appropriate conditional parity error or protect fault signals.

REFRESH TIMER

The memory circuits used in the arrays require a refresh signal at least every 2 milliseconds. In the LMS the refresh oscillator is set for approximately 15.6 microseconds. At each timer cycle one count is added to a seven-stage binary counter and the refresh request latch is set, requesting a memory refresh cycle at the input to the priority logic. The output of the seven stages form lines REFMAB07/ through REFMAB01/. Since refreshing is made an entire row at a time, the column addresses formed by MAB14 through MAB08 are not needed (see Memory Address Bus/Array Address Decoding below), nor are the block selects and transmit enables usually formed by MAB19, MAB18, MAB17, MAB16, and MAB15.

Therefore, 1/128 of the memory is refreshed at each timer cycle, representing one row out of 128 in each memory chip on each of the row address strobes. With 128 cycles at 15.6 microseconds per cycle, an entire bank is refreshed each 2 milliseconds.

PRIORITY SELECTION

Priority selection among internal bank requests takes place in the following order:

1. Refresh request
2. External request.
3. DMA request, with DMA1 having highest priority and DMA4 the lowest
4. CPU request

If a CPU or DMA request addresses the external bank, the priority is handled separately, as discussed below.

Refresh Priority

Refresh requests are made directly from the refresh timer approximately every 15.6 microseconds and are given highest priority. When the refresh select latch is set, the request is honored immediately after the completion of the existing cycle. At that time, the eight-phase timing chain is started, the refresh address is selected, and all RAS/ lines to the bank are pulsed, which refreshes the dynamic RAMs. The refresh select latch is then cleared.

External Priority

Requests from the external bank appear as an EXTREQIN/ signal and set the external request latch. At the conclusion of the current memory cycle, if a refresh request is not active and a CPU read/modify/write cycle is not in process, the eight-phase timing chain is started, the EXTIMAB/ address is selected and the EXTREQACT/ line to the data interface is forced true so that external data is processed.

DMA Priority

Requests from the four DMA subports appear on lines DMAxMR/. The first step in DMA priority selection is to determine which of two or more simultaneous DMA requests is to be processed first. DMA1 has the highest priority, DMA4 the lowest. The one selected is sent a request accept signal (DMAxRA/) so that the selected DMA device can place its data, memory address, and control information on the common DMA lines. At the same time, the appropriate DMAxMAB20/ line is checked to see whether the internal or external bank is being selected.

If the DMA device has chosen the external bank, the DMA external select latch is set, an EXTREQOUT/ signal is sent to the external bank (assuming no CPU external request is in process), the DMAMABxx/ address lines are gated to the EXTOMABxx/ lines to the external bank, and the ENDMWRTDO/ line (enable DMA write data out) to the data interface is pulled true to gate the DMA data to the external bank.

If the DMA chooses the internal bank, the DMA internal latch is set. At the conclusion of the current memory cycle, if there is no existing refresh or external request waiting, the eight-phase timing chain is started, the DMAMABxx/ address is selected, and the DMAREQACT/ line to the data interface is pulled true so that DMA data is processed.

An exception to the priority scheme occurs following a DMA memory cycle. If both a CPU and DMA request are waiting, the CPU is processed first, its latch having been set before the DMA latch has been cleared and reset with the new request.

CPU Priority

At the bottom of the priority list is the CPU memory request. Since CPU requests normally make up the vast majority of incoming requests, they would monopolize the LMS if given any higher priority.

The CPU request (CPUREQ/ true) is honored when:

- No higher priority request is active, or
- On an internal request, a DMA cycle has just been completed, and no refresh or external request is active.

If the CPU chooses the internal bank (CPUMAB20/ false), the eight-phase timing chain is started and the CPUMABxx/ address is selected. The data interface interprets the lack of an EXTREQACT/ or DMAREQACT/ to mean that CPU information is to be processed.

If the CPU chooses the external bank, the CPU external select latch is set if no external data request is active, an EXTREQOUT/ signal is sent to the external bank, the CPUMABxx/ address is gated to the EXTOMABxx/ lines to the external bank, and the ENCPWRTDO/ (enable CPU write data out) line to the data interface is pulled true to gate CPU data to the external bank.

ADDRESS DECODING

Address Register/Selector

The four address inputs from the refresh counter, the CPU, the DMA address bus, and the external-in address bus are applied to the inputs of a one-of-four selector. In addition to the 19 address lines, the following control lines are selected by the same selector:

1. Split cycle
2. Program protect
3. Character 0
4. Character 1
5. Write
6. Write protect bit

Refresh Addresses

Address lines REFMAB07/ through REFMAB01/ are supplied by the refresh counter. The remaining inputs are unused.

CPU Addresses

Data lines S305/ through S315/, the 11 least significant bits of the address, are latched by the GATEAB/ line from the CPU. The outputs of the latch are lines CPUMAB11/ through CPUMAB01/, respectively. Lines CPUMAB19/ through CPUMAB12/ are supplied by the data interface. CPUMAB20/, also supplied by the data interface, is not fed to the selector, but to the bank selector. It selects the internal bank if false, the external bank if true.

DMA Addresses

DMA address lines DMAMAB20/ through DMAMAB01/ are supplied by the common DMA address bus to the selector.

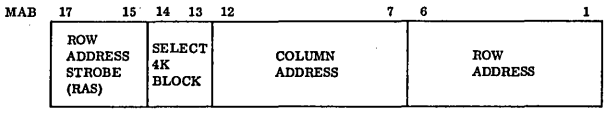
External Addresses

The address lines from the external bank, EXTIMAB19/ through EXTIMAB01/, are available to the address selector from the external bank I/O cable.

Selection among the four inputs is determined by the priority logic.

Array Address Decoding

The array organization is explained in detail under Memory Arrays below. The meaning of the 19-bit address is shown below:



The least significant seven bits of the address (MAB07 through MAB01) are strobed into the row address register, and the next significant seven bits (MAB14 through MAB08) into the column address register, by the same signal generated at the end of the priority decoding that starts the eight-phase timing chain (STRMOSADR).

The seven row address bits are sent to the array along with a row address strobe (RAS) on lines MOSMA06/ through MOSMA00/. This address is stored in the array. Next, the seven column address bits are sent along the same MOSMA06/ through MOSMA00/ lines with the column address strobe (CAS/). The 14 bits select a single 18-bit word from one 16K word block.

The next most significant address bits, MAB16 and MAB15, are decoded to select one of four 16K memory blocks, by activating one of lines SLTBLK0/ through SLTBLK3/.

The most significant three bits, MAB19 through MAB17, are decoded to select one of eight row address strobes, each one directed to a set of four 16K memory blocks. Thus, the address bus addresses eight 64K-word memory sections for a total of 512K words.

In the refresh mode, the four block select lines are all true, and the eight row address strobes are tied together, thus selecting a row in the memory chips on each and every array on each cycle.

Bank Selection and Memory Address Error

External bank selection is made by the CPU by a true level on line CPUMAB20/ at the time of a CPU memory request. If no external bank is connected, a read request is made from the same address in the internal bank, and a write request is aborted.

DMA requests for the external bank are made by a true level on the appropriate DMAxMAB20/ line from the same device that is selected by the priority select circuit. If no external bank is connected, a read request results in a read cycle from the same address in the internal bank and a write request is aborted. In addition, control line DMAMAE/ to the common DMA bus is set true as a warning to the DMA device.

CONTROL LOGIC AND TIMING

Eight-Phase Timing Chain

The eight-phase, overlapping timing chain consists of two 4-bit counters driven by a delay line oscillator. The oscillator is controlled by taps on the delay line, which set the frequency at approximately and 12.5 MHz. The timing chain diagram is shown in figure 5-8, along with the abbreviated representation used in the read/write cycle timing diagram.

The timing chain provides complementary outputs for each clock signal.

The oscillator is started with the signal STRMOSADR after the priority logic has selected the appropriate incoming request. This in turn clocks the timing chain, which normally ends with the completion of the T7 signal. In automatic split cycle write, used in character write and split cycle protected mode (ENPRTSYS enabled), the cycle is run two times, with the leading edge of the T0 pulse coincident with the trailing edge of the T6 pulse.

I/O Time Delay Selection

Time delays are incorporated in the priority select logic and read/write timing. An 80-nanosecond delay is incorporated in the DMA priority select circuits to allow for priority selection and to provide time for the selected DMA to respond to the DMAxRA/ signal. An additional 50-nanosecond delay is incorporated after priority selection before the STRMOSADR signal is generated to allow time for the data, address, and control lines to switch from CPU selection to external, DMA, or refresh select.

If the LMS is not busy, a window is generated by CPU clock signals T0/ and T3MEM/ to allow synchronizing a CPU request with respect to a DMA, external, or refresh request.

Read/Write Timing

The memory cycle begins with the STRMOSADR signal generated by the priority logic. At this time the selected incoming address lines are clocked into the row and column address registers, the decoded block select address register, and the decoders for the row address strobe and the transmit enables. After a 30-nanosecond delay, the output of the row address strobe decoder is enabled by the ENRAS line, completing the selection of a 16K block in an array.

Twenty nanoseconds later (or 70 nanoseconds for a CPU full cycle write), the eight-phase timing chain is started and the STRCONT signal is activated, which clocks on the DMAREQACT/ or EXTREQACT/ lines, locks up incoming control lines, if appropriate, and the STCP/EXD/ strobe to the data selectors in the data interface. The determination of full or split cycle operation is made at this time. (The remainder of the section assumes full cycle operation.)

At T1 time, the column address from lines MAB14 through MAB08 is placed on lines MOSMA06/ through MOSMA00/. A CPUMDS/ signal is sent to the CPU if it is a CPU full write cycle.

At T2 time, the column address strobe (CAS/) is sent to the array, the read/write function is determined, and either the WTMOS/ signal is sent to the array for a write cycle or one of the E64KN0-1/ through E64KN1-4/ lines is sent to the array for a read cycle.

At T3 time, the signal STBCPUBKP/ is sent to the CPU on a CPU cycle to indicate that breakpoint address is available on the CPUDFMxx lines. The CPUMDS/ signal is sent to the CPU for a CPU read or split cycle.

At T4 time, the RESYNCDMA/EXT line is energized on all but a CPU cycle, resetting the input latches for refresh, DMA, and external priority selection. On external requests, the external resume signal EXTRESOUT/ is sent to the external bank.

At T6 time the DMAMDS/ signal is sent to the DMA devices and the STBDMADFM/ signal is sent to the data interface on a DMA cycle, the STBCPUDFM/ signal is sent on a CPU cycle, or the ENEXTRDO/ is sent on an external cycle.

After a 50-nanosecond delay following T6, the RESYNCDMA line resets the DMA priority select latches and the following conditional signals are activated as appropriate:

SETSM105/ (CPU protect fault)
DMAPF/
DMAMPE/
SETSM108/ (CPU parity error)

At T7 time, MEMRESYNC goes positive, resetting priority latches. On the trailing edge of T7, MEMHALT goes high, turning off the oscillator and allowing the next memory request to start a new cycle.

Split Cycle Operation

Two types of split cycle operations are encountered - one is generated by a split cycle request (read/modify/write), and the second is an automatic split cycle generated by a character write request or whenever the ENPRTSYS line is true.

Read/modify/write split cycle operation is selected by the CPUSC/ or SC/MPL (external) split cycle request lines. At T0 time, the following lines are brought low:

- CPUSCHLD - On a CPU split cycle request, locks out higher priority requests until the split cycle is complete.
- CPU/EXTSC - On a CPU or external split cycle request, locks out a new DMA request.
- EXTSC - On an external split cycle request, locks out an internal CPU request and inhibits the window given a CPU request by the CPU clock.

At T1 time on a CPU split cycle the line SCCLRDMA/ clears out any higher priority request that may have already been latched in the priority circuits.

The read portion of the split cycle continues to completion. As the write line from the CPU port goes high, a new request is generated by the split cycle logic to begin the write cycle:

- On a CPU split cycle request, the line CPUSCREQ/ is brought true, starting a new cycle.
- On an external split cycle request, the cycle is completed and the EXTRESOUT/ signal is returned to the external bank. The external bank then generates a new external request, beginning the second half of the split cycle.

A forced, or automatic, split cycle occurs when a character write or protect bit write is requested or the ENPRTSYS line is true during a write request. Here, the normal WTMOS/ signal to the array is inhibited and a read cycle is made. The eight-phase timing chain is restarted after T6 time, continuing on so that a second cycle is begun immediately. At T2 time the second time around, the WTMOS/ line is brought true, causing a write cycle to be accomplished. The usual split cycle request inhibits are not needed, as the MEMHALT line does not go true until the second cycle.

In the character write or protect bit write mode, the purpose of the automatic split cycle is to read the contents of the selected memory location into the data interface, so that the character or word (protect bit write) that is not to be changed by the write operation is restored as the other character or protect bit is written.

In the ENPRTSYS (enable protect system) mode, the read portion of the cycle permits the reading of the protect and parity bits. The lines MPE+DED/ and MPBL from the data interface abort the write portion of the cycle if the program protect input line is true and the protect bit is false by inhibiting the WTMOS/ output to the array.

CPU or DMA External Bank Timing

CPU or DMA read or write requests specifying the external bank are handled asynchronously by the internal address interface. Requests are handled without delay, except for priority selection between DMA external and CPU external requests. The EXTREQOUT/ is set true, the proper address is gated to the EXTOMABxx/ lines, and either the ENCPWRTDO/ (for the CPU) or ENDMWRTDO/ (for the DMA) line to the data interface is set true for the gating of write data to the external bank.

As the EXTRESIN/ line goes true after the completion of the cycle by the external bank, the STRCPUBKP/ line to the CPU goes true to a CPU cycle. After a slight delay, either the STBCPUDFM/ or STBDMADFM/ line goes true as well as CPUMDS/ or DMAMDS/, followed by the appropriate parity or protect fault conditional signals.

This completes the cycle, except for read/modify/write split cycle requests, which generate a second external bank request when the appropriate write line goes true.

ADDRESS INTERFACE INTERNAL SIGNALS

Referenced signals contained entirely within the address interface are described in table 4-15.

DATA INTERFACE

The data interface provides the read and write data handling for the LMS, generates and detects parity signals, and controls CPU addressing and bounds comparison. Timing for the data interface is supplied by the address interface. Note the simplified block diagram of figure 4-17.

Incoming CPU addresses are processed by the data interface. Page or absolute mode is determined by commands from the CPU micro instruction register, and the data interface supplies the most significant eight bits of the 19-bit CPU memory address plus the bank select bit to the address interface. These bits are either selected from the incoming address in absolute mode or from the page file in page mode. The data interface also loads and stores bounds addresses used to establish unprotected areas available to the CPU and makes the comparison against those bounds with incoming CPU addresses.

Incoming write data from the CPU or DMA devices is either directed to the external bank or stored for use in the internal bank. Write data from the external bank is stored for internal bank use. On command from the address interface priority circuits, the appropriate write data and protect bit is selected, the parity bit is generated, and the resulting 18-bit word is sent to the selected array.

On a read cycle or the read portion of a write cycle from the internal bank, the data interface receives the 18-bit word from the array and the parity bit is checked. A parity error is flagged and a signal is sent to the address interface so an error signal may be sent to the requesting device. The read data word is directed to the three data output ports and is also latched for use in a possible restore or character write operation.

The CPU and DMA data output registers, on command from the address interface, select either this internal read data or read data from the external bank. The CPU output port also has provisions for providing breakpoint addresses, page file content, or status over its two sets of output lines.

For the remainder of this section, refer to the data interface block diagram, figure 5-5.

CPU ADDRESS CONTROL

CPU address information is received by the data interface over the 16-bit multipurpose path S300/ through S315/. Operation on this data is controlled by the three address control lines from the CPU (GATEAB/, GATEUBDS/, GATELBDS/) and the condition of the 4-bit micro instruction register (MIR28' through MIR31'), which is continually updated by clock pulse T0'/ from the CPU. Control is also affected by the state of the 2-bit address memory register (AMR0, AMR1).

TABLE 4-15. ADDRESS INTERFACE INTERNAL SIGNALS

Signal	Description
CAE	Column address enable. Sends MAB bits 14 through 08 to array.
CPUACT	Indicates a DMA or higher priority request is not active.
CPUACT	Pulse coincident with start of CPU memory cycle. Not a complement of CPUACT.
CPUADR20	Selects external bank for a CPU request.
CPUCHARO	Input CHARO inhibited by a high CPUWRTPB input.
CPUEXSCHLD	Inhibits a new DMS external request.
CPUEXSCRQ	CPU external split cycle request.
CPU/EXTSC	Inhibits new DMA internal request.
CPUEXTSEL	Enables CPU external control circuits.
CPUEXTSEL	Resets CPU request latch. Not a complement of CPUEXTSEL.
CPUMABxx	CPU memory address bits 11 through 01. Latched from CPU data lines S305/ through S315/.
CPUMAE	CPU memory address error.
CPUPROT	CPU program protect line, enabled by inputs CPUPROTECT/, CPUNBNDS/, or ENPRTSYS.
CPUREQ	CPU request latch set. Not a complement of input CPUREQ/.
CPUREQL	CPU internal request received.
CPUSC	Complement of input CPUSC/.
CPUSCHLD	Inhibits the RESYNDMA/EXT signal.
CPUSCREQ	Initiates write portion of split cycle CPU request.
CPUWRT	Input CPUWRITE inhibited by CPUMAE or EMPF.
CPUWRTHLD	Delays the start of an unprotected CPU full write cycle.
DMAACT	DMA or higher priority request active.
DMAACT'	Pulse just preceding start of DMA or higher priority cycle.
EXTSC	Inhibits new CPU internal request.
DMAEXTSEL	Enables DMA external control circuits.
DMALREQ	New DMA request received.
DMAPROT	Input DMAPROT/ inhibited by ENPRTSYS.
DMAREQACT	Complement of output DMAREQACT/.
DMAWRT	DMAWRITE inhibited by DMAMAE/.
EMPF	Complement of input EMPF.
ENCPUADR	Enables CPU and refresh memory addresses.
ENDMAADR	Enables DMA and external memory addresses, complement of ENCPUADR.

TABLE 4-15. ADDRESS INTERFACE INTERNAL SIGNALS (Contd)

Signal	Description
ENEXTSTAT	Enables external outputs CH0/PF and CH1/PE.
$\overline{\text{ENMOS250}}$	Complement of input ENMOS250.
$\overline{\text{ENMOSRDD}}$	Enables output of transmit enable selector D8.
$\overline{\text{ENPRTSYS}}$	Complement of input ENPRTSYS.
$\overline{\text{ENRAS}}$	Enables outputs of row address strobe selectors.
EXTREQACT	Complement of output EXTREQACT/.
EXTRESYNC	Resets the external request latches.
$\overline{\text{EXTRESYNDMA}}$	Brings RESYNCDMA high following a DMA external cycle.
$\overline{\text{FULWRT}}$	Unprotected word write (single cycle) control line.
$\overline{\text{MC/POMC}}$	Master clear/power-on master clear. Resets latches.
MEMHALT	Eight-phase clock stopped.
MEMRESYNC	Timing for DMA or higher priority start of cycle.
MPE+DEDFF	Parity error line to abort a write operation.
$\overline{\text{MPE+DEDFF}}$	MPE+DEDFF inhibited by $\overline{\text{FULWRT}}$.
$\overline{\text{POMC}}$, $\overline{\text{POMC}}'$	Power on master clear control line.
PROTFAULT	Derived from protect inputs and MPBL line to form PF outputs.
$\overline{\text{RAE}}$	Row address enable; complement of $\overline{\text{CAE}}$.
$\overline{\text{REFMABxx}}$	Refresh memory address (07 through 01)
$\overline{\text{REFRESHREQ}}$	Memory cycle request from refresh timer.
RESYNCDMA, $\overline{\text{RESYNCDMA}}$	Reset DMA priority and internal/external select latches.
$\overline{\text{RESYNCEXT}}$	Inhibits DMA external request.
$\overline{\text{RESYNCINT}}$	Inhibits DMA or higher priority select latch.
RESYNDMA/EXT	Resets DMA, external, and refresh priority latches.
$\overline{\text{SCCLRDMA}}$	Clears DMA or higher priority select latch during CPU split cycle.
SCFF	Enables split cycle logic.
$\overline{\text{SELREFRESH}}$ $\overline{\text{SELREFRESH}}'$	Selects refresh memory addresses.
$\overline{\text{SETDMAMAE}}$	Sets DMA memory address error flip-flop.
$\overline{\text{STBCPUEXT}}$	Pulls output STBCPUDFM/ true following a CPU external cycle.
$\overline{\text{STBDMAEXT}}$	Pulls output STBDMADFM/ true following a DMA external cycle.

TABLE 4-15. ADDRESS INTERFACE INTERNAL SIGNALS (Contd)

Signal	Description
STCPUPEEXT	Pulls output SETSM108/ true on CPU external parity error.
$\overline{\text{STCPUPFEXT}}$	Pulls output SETSM105/ true on CPU external protect fault.
$\overline{\text{STDMAPEEXT}}$	Pulls output DMAMPE/ true on DMA external parity error.
$\overline{\text{STDMAPFEXT}}$	Pulls output DMAPF/ true on DMA external protect fault.
STRCONT	Clocks flip-flops producing EXTREQACT, DMAREQACT, and CPUACT outputs.
STRCPBPEXT	Pulls output STBCPUBKP/ true following a CPU external cycle.
STRMOSADR	Starts eight-phase clock and strobes memory addresses into address registers and decoders.
T2ODDCLK	Complement of input T2ODDCLK/.
$\overline{\text{WRITE}}$	Clears split cycle latches.
WRITESC	Enables automatic split cycle logic.
$\overline{\text{WRTABORT}}$	Inhibits WTMOS/ output on protect fault or parity error.

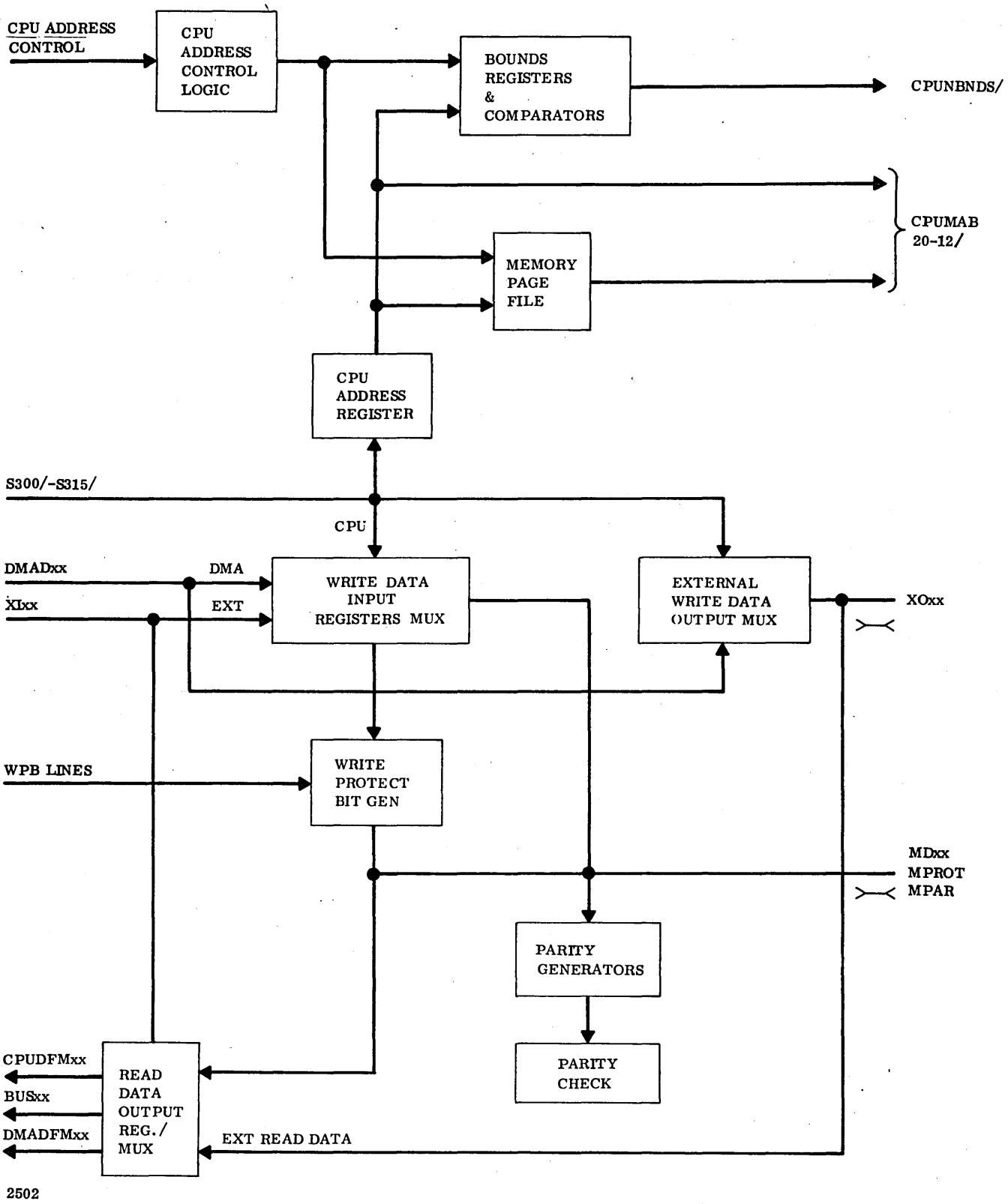


Figure 4-17. Data Interface Simplified Block Diagram

In general, the least significant 11 bits of the CPU address are used directly by the address interface over a parallel connection of lines S305/ through S315/. The most significant eight bits of the 19-bit address, plus the bank select bit, are controlled by the data interface and are sent to the address interface over CPUMAB19/ through CPUMAB12/ plus CPUMAB20/ for the bank select bit.

The CPU address control section also includes a 16-bit CPU address register (AB register), 16-bit upper and lower bounds registers, two 16-bit comparators, a 64-word by 9-bit RAM (memory page file), and a 9-bit CPU address mux.

The CPU address control commands are shown in table 4-16. Note that each of the commands loads the AB register with the data on S300/ through S315/, but this action is inhibited by a true BLKAB/ signal from the address interface, which occurs during a CPU read/modify/write split cycle.

AB Register

This 16-bit register is loaded from S300/ through S315/ (LSB) during any of the commands shown in table 4-16. The register output AB00 through AB15 (LSB), is available to the CPU as a breakpoint address at the beginning of each CPU read or normal write cycle.

Bounds Registers and Comparators

Two 16-bit registers store the upper and lower bounds addresses of the unprotected memory area available to the CPU. The comparators continuously compare AB00 through AB15 with the contents of the upper and lower bounds registers. When the AB00 through AB15 address is lower than the upper bounds address and higher than the lower bounds address, CPUNBNDS/ goes true, which has the same effect as a true CPUPROTECT/ line during a CPU write operation.

Memory Page File (MPF)

This 64-word by 9-bit RAM contains a file of page addresses that form the most significant eight bits of the CPU address when in page mode, plus the bank select bit. The outputs of the memory page file are (see figure 4-15):

MPF07 - MPF00	Most significant eight memory address bits
MPF08	Bank select bit

The five least significant memory page file address bits are provided by lines AB00 through AB04 (LSB). The most significant bit is supplied by the AMR1 register for memory cycles, and by AB05 when the memory page file is enabled by a load/read memory page file command.

TABLE 4-16. CPU ADDRESS CONTROL COMMANDS

Command	Condition	Effect
Load AB	GATEAB/ true	Loads S300/ through S315/ into AB register on leading edge. Clears load/read memory page file flip-flop and status flip-flop.
Load Lower Bounds	GATELBDS/ true MIR28' false	Loads S300/ through S315/ into AB register on leading edge; then loads contents of AB register into lower bounds register on trailing edge.
Read Status	GATELBDS/ true MIR28' true	Loads S300/ through S315/ into AB register and sets enable status flip-flop (to enable status lines to CPU data output register). CPU read cycle necessary to place status levels on output lines.
Load Upper Bounds	GATEUBDS/ true MIR28' false MIR29' false	Loads S300/ through S315/ into AB register on leading edge; then loads contents of AB register into upper bounds register on trailing edge.
Load AMR	GATEUBDS/ true MIR29' true	Loads S300/ through S315/ into AB register and loads address memory register. AMR0 assumes level of MIR30', selecting page mode if true, absolute mode if false. AMR1 assumes level of MIR31', providing the most significant address bit to the memory page file on a page mode memory cycle.
Load/Read MPF	GATEUBDS/ true MIR28' true	Loads S300/ through S315/ into AB register and enables memory page file for loading or statusing. Selects memory page file status lines to CPU data output register, which are placed on the output lines on a subsequent CPU read or write cycle. A CPU write cycle loads the memory page file with a new page address, but does not affect the addressed location in memory.
Load/Read MPF/AMR	GATEUBDS/ true MIR28' true MIR29' true	Combination of load/read memory page file and load address memory register.

CPU Address Mux

The CPU address mux selects the nine most significant bits of the CPU address from either the AB register or the memory page file, depending upon the condition of AMR0. The outputs are sent to the address interface board.

Output	Source: AMR0 False (Absolute Mode)	AMR0 True (Page Mode)
CPUMAB12/ (LSB)	AB04	MPF00
CPUMAB13/	AB03	MPF01
CPUMAB14/	AB02	MPF02
CPUMAB15/	AB01	MPF03
CPUMAB16/	AB00	MPF04
CPUMAB17/	0	MPF05
CPUMAB18/	0	MPF06
CPUMAB19/	0	MPF07
CPUMAB20/ (bank select)	0	MPF08

PORT SIGNALS AND TIMING

CPU Port

Data on the multipurpose path S300/ through S315/ is considered an address when address control commands are being executed, but it is assumed to be write data when a CPU memory request (CPUREQ/) is received.

When a CPU internal request is received by the address interface, the first activity of the data interface CPU port is a STBCPUDFM/ signal, which gates the ABxx lines from the AB register to the CPUDFMxx outputs to supply the breakpoint address to the CPU. After the request has been accepted, the STCP/EXD/ signal strobes write data from S300/ through S315/ into the CPU write data register. ENMOSWRTD/ true enables the CPU write data register to the MDxx lines to generate word parity. After ENMOSWRTD/ line goes false, read data appears on the MDxx lines, and is available to the CPU port. The second STBCPUDFM/ signal strobes this read data, the protect bit, and the write data word parity to the CPUDFMxx outputs. On a read cycle, a true ENMEMBUS/ line from the CPU enables these same outputs to the BUSxx lines. On a write cycle, the ENMOSWRTD/ lines goes true again, and the latched write data is placed on the MDxx lines for writing into memory.

For a CPU external write cycle, write data is not latched but is gated on the XOxx lines by the ENCPWRTDO/ signal to the external bank. Read data is received over the same XOxx lines and is strobed into the CPU data output register as in a CPU internal cycle.

DMA Ports

After a DMA internal request has been selected and accepted, data from incoming lines DMAD16/ through DMAD01/ is strobed into the DMA write data register by the STCP/EXD/ signal. The true ENMOSWRTD/ and DMAREQACT/ lines from the address interface place the latched DMA write data on the MDxx lines to generate word parity. After the ENMOSWRTD/ line goes false, read data appears on the MDxx lines and is available to the DMA data output port. The signal STBDMADFM/ strobes this read data, the protect bit, and the write data word parity to the DMADFMxx outputs. On a write cycle, the ENMOSWRTD/ line again goes true, and the latched DMA write data is placed on the MDxx lines for writing into memory.

For a DMA external write cycle, write data is not latched but is gated to the XOxx lines by the ENDMWRTD/ signal to the external bank. Read data is received over the same XOxx lines and is strobed into the DMA data output register by the STBDMADFM/ signal as in a DMA internal cycle.

External Port

After an external request has been accepted by the external bank, data from incoming lines XI16/ through XI01/ is strobed into the external write data register by the STCP/EXD/ signal. The true ENMOSWRTD/ and EXTREQACT/ lines from the address interface place the latched write data on the MDxx lines to generate word parity. After ENMOSWRTD/ goes false, read data appears on the MDxx lines and is gated to the PRF/MPB line. Write word parity is gated to the SC/MPL line by the same signal. On a write cycle, the ENMOSWRTD/ line again returns true and the latched external write data is placed on the MDxx lines for writing into memory.

PROTECT BIT GENERATION AND DETECTION

A protect bit is written into memory on each write cycle. The source of this protect bit is determined by the following truth table:

ENPRTSYS	DMAREQACT/	Write Protect Bit Line Source	EXTREQACT/	Protect Bit
X	True	False	False	S306/ (CPU)
X	False	X	X	LSECMPB
False	X	X	X	HIGH

Thus if the write protect bit line is true during a CPU write cycle, the protect bit sent to bit 17 of the addressed memory word is the level of bit 10 of the CPU write data line. If the write protect bit line is false, the protect bit previously in memory is restored. The protect bit line to the arrays is the bidirectional line MPROT. If only the protect bit is to be written, the requesting device must bring both character lines false, which restores the 16-bit data word during the write cycle.

If the ENPRTSYS line from the CPU is false, the protect bit is set true on every write cycle regardless of the condition of the protect bit write line or bit 10 of the incoming write data.

On a read cycle or the read portion of a write cycle, the condition of the protect bit is received over the MPROT line from the array, and is made available to the requesting port over lines MEMPROTBT/ to the CPU, DMAMPB/ to the DMA devices, and PRF/MPB to the external bank. The protect bit is also latched by the STBDFM/ signal and is available as LSECMPB for restoration to memory and as MPBL to the address interface for the detection of protect faults.

PARITY GENERATION AND DETECTION

As write data is strobed in on a write cycle, even parity is detected and the WORD PARITY line is latched true if an even number of ones is present in the 16-bit write word. This latched level is available to the requesting port at the end of the read portion of a write cycle over the lines EVENPAR to the CPU, DMAMPL/ to the DMA devices, or SC/MPL to the external bank. This latched parity line has no other effect on operation.

Overall parity is generated during the read portion of a write cycle or during a read cycle from the 17-bit data word and protect bit read from memory. It is compared to the read parity bit on line MPAR from bit 16 of the addressed memory word. If the comparison is true then parity is correct and no error signal is sent. If the two parity signals are complementary, the MPE+DED line to the address interface flags a parity error.

Parity is generated again from the write data word at the beginning of the write portion of a write cycle, this time including the protect bit. This parity level is placed on the MPAR line for writing into bit 16 of the addressed memory location.

CHARACTER WRITE OR RESTORE CONTROL

Character write or restore operation is controlled by the CHR0/ and CHR1/ lines from the address interface, which are derived from the character 0 and 1 lines from the requesting device.

The write data mux is split into two independently controlled segments, so that the most significant and least significant eight bits may be selected from one of the three write data in registers (write) or from the latched read data (restore).

On a word write operation both character lines are true and both halves of the write word are derived from the data in the register selected by lines EXTREQACT/ and DMAREQACT/ (the CPU is assumed if both are false).

On a character write operation, one of the character lines is true and the other false. If CHR1/ is true, the least significant eight data bits are derived from the least significant eight bits of the incoming write data, while the most significant eight bits are from the latched read data LSEC15 through LSEC08. If CHR0/ is true, the least significant eight bits are from the latched read data LSEC07 through LSEC00, while the most significant eight data bits are sourced from the most significant eight bits of the selected incoming write data. If both character lines are false, the data word consists entirely of the latched read data.

The protect bit is unaffected by a character write or restore operation, and may only be changed under control of the protect bit write line. The parity bit is regenerated on each word write or character write operation.

DATA INTERFACE INTERNAL SIGNALS

Referenced signals contained entirely within the data interface board are described in table 4-17.

LARGE MEMORY ARRAYS (LMAS)

The LMAs contain the dynamic memory ICs and the buffers for data, address, and control signals. The arrays have a 128K-word by 18-bit storage capacity.

LMAs are physically placed in slots X, Y, Z, and AC of the CYBER 18 mainframe to configure a macro memory for the system. Placement of the arrays is discussed below.

Circuit descriptions, a functional block diagram, and backplane wiring notes are contained in section five.

MEMORY CONFIGURATION

Total memory using LMAs is 524,288 words of 18 bits each, expandable to 1,048,576 words by the use of an external bank.

To configure a memory, arrays are chosen to meet the desired total storage capacity, and are placed in slots X, Y, Z, and AC in the mainframe beginning with slot X.

Possible configurations are shown in table 4-18. The configurations adhere to the mandatory rule that an LMS may not be in a slot following a blank slot.

	X	Y	Z	AC
Incorrect:	128K	--	128K	--
Correct:	128K	128K	--	--

Once memory is configured, array boards may be interchanged for troubleshooting purposes.

MEMORY ORGANIZATION AND ADDRESSING

The basic element of the LMS is the 16K block, consisting of eighteen 16,384-bit dynamic RAM chips, organized as 16,384 words by 18 bits. Control and addressing inputs are paralleled to the 18 chips within the block, while each chip receives and provides a single data bit of the 18-bit data word.

Addressing a word within the 16K block requires a 14-bit address. In an LMA the address is loaded in two stages over the seven address lines MOSMA06 through MOSMA00. The first stage occurs when the RAS/ signal (row address strobe) is received by the selected 16K block. The signal strobes the seven address bits on the MOSMAxx/ lines into on-chip row address registers. The second seven bits are then placed on the MOSMAxx/ lines and strobed into the column address registers in memory chips by the CAS/ signal (column address strobe). The terms row and column refer to the 128 by 128-bit arrangement of storage cells within each 16,384-bit chip.

A 64K memory section contains four 16K blocks. A section constitutes half of a 128K array. To select a 16K block within a section, one of the four lines SLTBLK0/ through SLTBLK3/ is brought true by the address interface. The signal RAS64Kx-x/ gates the active SLTBLKx/ line to provide the appropriate RAS signal to the selected block. Refer to figure 4-18.

TABLE 4-17. DATA INTERFACE INTERNAL SIGNALS

Signal	Description
ABxx	CPU address bits 00 through 15. Latched from lines S300/ through S315/.
$\overline{\text{ABxx}}$	Complements of AB07 through AB15; used to load page addresses.
$\overline{\text{BPxx}}$	Bit positions 03 through 22.
CPDFMxx	Outputs 16 through 01 from CPU data output mux.
$\overline{\text{CPUxx}}$	Latched CPU write data bits 00 through 15 from lines S300/ through S315/.
$\overline{\text{DMALOCAL}}$	Same as input DMAREQACT/.
$\overline{\text{ECPUL}}$	Enables lower half of CPU write data to memory array.
$\overline{\text{ECPUU}}$	Enables upper half of CPU write data to memory array.
$\overline{\text{EDMAL}}$	Enables lower half of DMA write data to memory array.
$\overline{\text{EDMAU}}$	Enables upper half of DMA write data to memory array.
EECC	Complement of $\overline{\text{ENABLE LOCAL STATUS}}$; trailing edge clears status latch.
$\overline{\text{EMPF}}$	Complement of output EMPF (Enable Memory Page File).
$\overline{\text{ENABLE LOCAL STATUS}}$	Selects status inputs to CPU data output multiplexer.
$\overline{\text{ENABLE LOCAL MPF STATUS}}$	Selects MPF status inputs to CPU operand data outputs. Complement of output EMPF.
$\overline{\text{EPROT/PAR}}$	Enables protect and parity bit outputs to array. Same as input ENMOSWRTD/.
$\overline{\text{ESECL}}$	Enables lower half of read data to memory array (restore).
$\overline{\text{ESECU}}$	Enables upper half of read data to memory array (restore).
$\overline{\text{EXL}}$	Enables lower half of external write data to memory array.
$\overline{\text{EXU}}$	Enables upper half of external write data to memory array.
$\overline{\text{LDMAxx}}$	Latched DMA write data bits 16 through 01.
LOADAB	Loads contents of S300/ through S315/ into CPU address register.
$\overline{\text{LOADLBDS}}$	Trailing edge loads contents of CPU address register into lower bounds register.
$\overline{\text{LOADUBDS}}$	Trailing edge loads contents of CPU address register into upper bounds register.
LOAD WRITE DATA	Complement of input STCP/EXD/.
LSEC	Latched value of the SEC line, indicating a single bit error.
LSECxx	Latched read data from memory, bits 15 through 00.
LSECMPB	Latched value of last protect bit read. Same as output MPBL.
MPFxx	Outputs 08 through 00 from the memory page file.
MR/	Same as input MC/.
$\overline{\text{MULTIND}}$	Complement of input MULTIND.

TABLE 4-17. DATA INTERFACE INTERNAL SIGNALS (Contd)

Signal	Description
PBMUXA PBMUXB	Protect bit multiplexer select lines.
PGM FILES	Selects upper half of memory page file.
REF+CPU	Status line active during refresh or CPU memory cycles.
SEC	Single error flag.
SECxx	Read data bits 15 through 00.
SECMPB	Read memory protect bit.
WORD PARITY	Output of parity generator.
$\overline{\text{WRITEMPF}}$	Sets memory page file to write mode.
WTMPL	Internal bank parity line; latched value of WORD PARITY.
XMPB	Protect bit from external bank.
XMPL	Parity bit from external bank.
XRDxx	Latched read data bits 16 through 01 (LSB) from external bank.
XWDxx	Latched write data bits 16 through 01 (LSB) from external bank.

TABLE 4-18. MEMORY CONFIGURATION

Memory Bank Capacity (Words)	CPU Slot			
	X	Y	Z	AC
131,072	128K	-	-	
262,144	128K	128K	-	
393,216	128K	128K	128K	
524,288	128K	128K	128K	128K

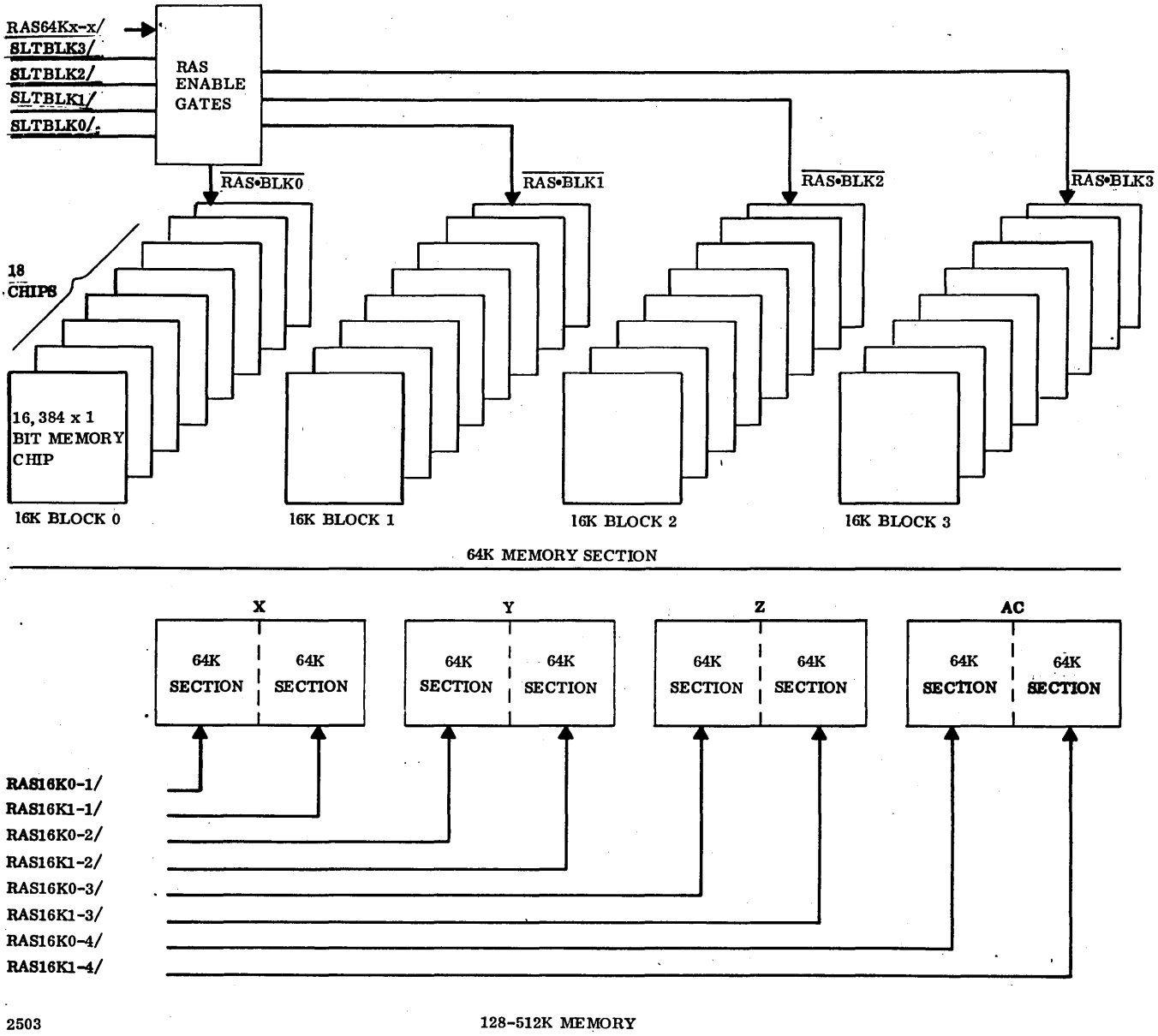


Figure 4-18. LMS Array Organization

Selection of a 64K section is made by one of the eight row address strobes from the address interface, RAS64K0-1/ through RAS64K1-4/. At slot X through AC of the CYBER 18 mainframe, the RAS64K0-x/ line is available to the array. When true this line selects the first 64K memory section, which is the lower half of a 128K array. The next sequential row address strobe select line RAS64K1-x/ is picked up on the LMA, selecting the upper half (64K section).

Read data from the arrays is selected in the same manner by the E64KNx-x/ lines from the address interface. On a read cycle, or the read portion of a write cycle, one of these lines is true, enabling the read data buffers of a single 64K section to the bidirectional memory data lines MD15 through MD00, MPAR, and MPROT.

LMA SIGNALS AND TIMING

Timing for the arrays is supplied by the address interface timing circuits, which are somewhat slower than the capabilities of the memory chips. The access time of 225 nanoseconds for an LMA is measured from the time the row address strobe signal is available to the LMA until read data is available at the LMA read data output lines. Actual system access time is somewhat greater (slower).

Activity begins at the array with the receipt of address and block select information on lines MOSMAxx/ and SLTBLKx/, following acceptance of a memory request by the LMS. The cycle begins when the selected RAS64Kx-x/ signal arrives and is routed to the appropriate 64K memory section through the backplane wiring. There it gates one true SLTBLKx/ line to form a row address strobe (RAS/) to a specific 16K block, strobing the row address on the MOSMAxx/ lines into the on-chip registers. After the row address is stored on the chip, the MOSMAxx/ lines contain column address information from the address interface and the column address strobe (CAS/) signal is received (approximately 90 nanoseconds after the row address strobe), strobing column address information into the selected chip.

At this time, on a read cycle or the read portion of a write cycle, the associated E64KNx-x/ line is brought true, which enables read data outputs from the selected block to the MDxx, MPAR, and MPROT lines. However, valid data is not present until each chip has reached its access time. At T6 time of the eight-phase timing chain, when all chips in the section should have reached access time, data is strobed from the MDxx line to a data register. At T7 time, the enabling E64Kx-x/ signal is removed.

On a write cycle, write line WTMOS/ is brought true at T2 or T2* time depending upon the type of cycle, strobing write data from the MDxx, MPAR, and MPROT lines into the memory chips of the selected 16K block. The write line and row address strobe are held true for the time required by the chips (until T5 or T5* time) to achieve storage.

Approximately every 15.6 microseconds a refresh operation takes place to ensure that stored data remains intact. All of the RAS64K0-x/ lines and all of the RAS64K1-x/ lines are enabled, along with all of the SLTBLKx/ lines. Thus on refresh cycles all of the 16K blocks are enabled. A single row address is given, but no column address is necessary as the chips are refreshed a row at a time. Thus, 128 cycles are necessary on each set of row address strobes for a complete refresh, which occurs each 2 milliseconds.

DIAGRAMS 5

This section contains the interconnecting, block, and logic diagrams for the large memory subsystem. The block diagrams are related to the theory of operation in section 4 and the logic diagrams in this section.

INTERCONNECTING DIAGRAMS

These diagrams (figures 5-1 through 5-3) show all connections between the address interface, data interface, memory arrays, and other system components. Signal names and logic board pin numbers are included. The diagrams are useful for troubleshooting to the board level.

FUNCTIONAL BLOCK DIAGRAMS

These diagrams (figures 5-4 through 5-6) show data flow, address paths, and major control signals in the array and interface PWAs. All data and address paths are highlighted. Each block of the diagrams contains a number in the upper right corner that corresponds to the logic diagram sheet number. These block diagrams supplement the logic diagrams and are also intended to be used for training maintenance personnel.

LOGIC DIAGRAMS

These diagrams (figures 5-7 through 5-9) show all logic circuit paths, including chip type numbers, PWA location references, and chip pin numbers.

DIAGRAMS 5

This section contains the interconnecting, block, and logic diagrams for the large memory subsystem. The block diagrams are related to the theory of operation in section 4 and the logic diagrams in this section.

INTERCONNECTING DIAGRAMS

These diagrams (figures 5-1 through 5-3) show all connections between the address interface, data interface, memory arrays, and other system components. Signal names and logic board pin numbers are included. The diagrams are useful for troubleshooting to the board level.

FUNCTIONAL BLOCK DIAGRAMS

These diagrams (figures 5-4 through 5-6) show data flow, address paths, and major control signals in the array and interface PWAs. All data and address paths are highlighted. Each block of the diagrams contains a number in the upper right corner that corresponds to the logic diagram sheet number. These block diagrams supplement the logic diagrams and are also intended to be used for training maintenance personnel.

LOGIC DIAGRAMS

These diagrams (figures 5-7 through 5-9) show all logic circuit paths, including chip type numbers, PWA location references, and chip pin numbers.

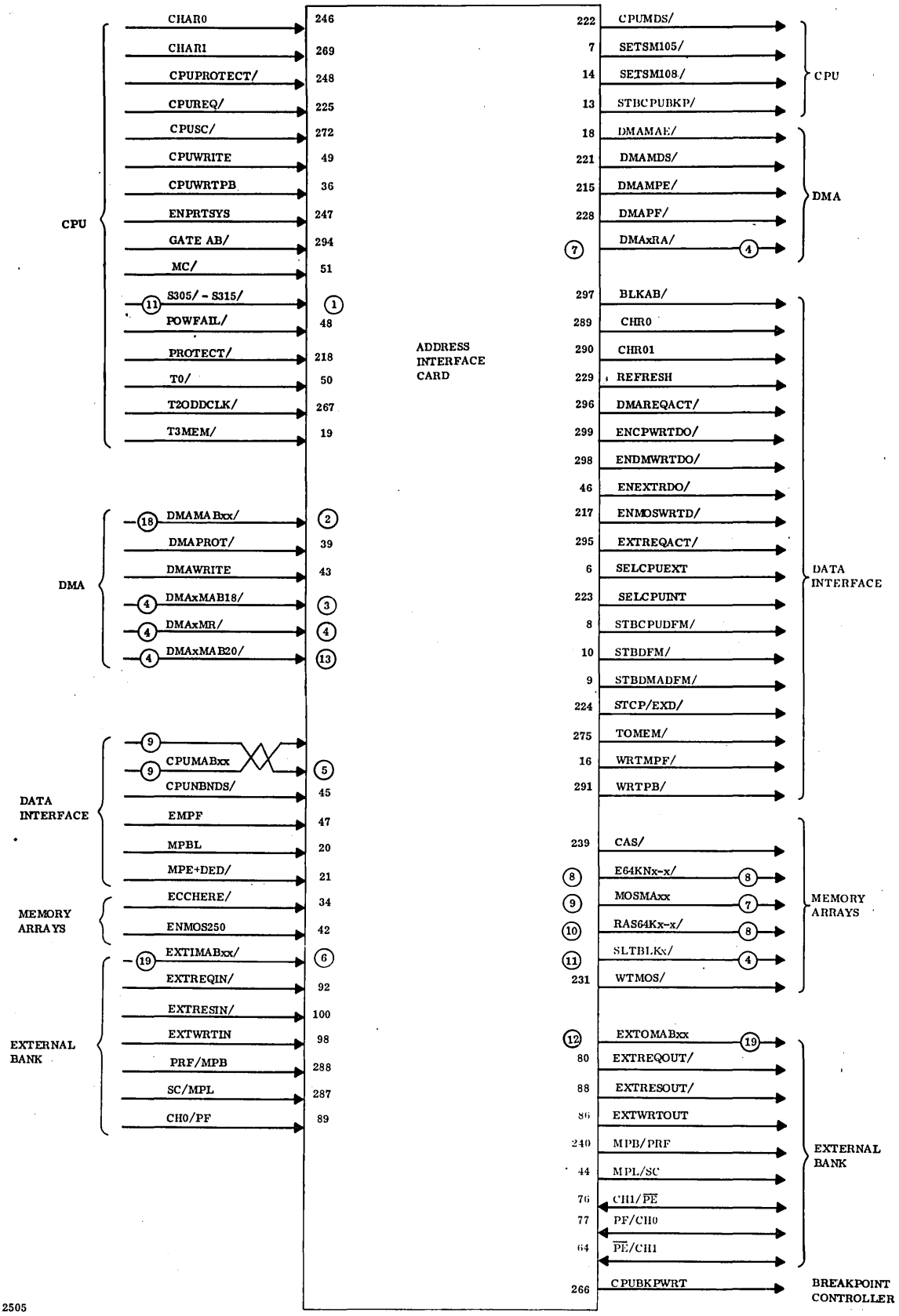


Figure 5-1. Address Interface Signals (Sheet 1 of 2)

① S305/ - S315/

S305/	41
S306/	40
S307/	38
S308/	23
S309/	28
S310/	24
S311/	22
S312/	29
S313/	12
S314/	15
S315/	5

② DMAMABxx/

DMAMAB01/	37
DMAMAB02/	253
DMAMAB03/	255
DMAMAB04/	256
DMAMAB05/	263
DMAMAB06/	35
DMAMAB07/	277
DMAMAB08/	52
DMAMAB09/	252
DMAMAB10/	261
DMAMAB11/	262
DMAMAB12/	274
DMAMAB13/	292
DMAMAB14/	281
DMAMAB15/	282
DMAMAB16/	271
DMAMAB17/	284
DMAMAB19/	230

③ DMAxMAB18/

DMA1MAB18/	210
DMA2MAB18/	205
DMA3MAB18/	248
DMA4MAB18/	211

④ DMAxMR/

DMA1MR/	11
DMA2MR/	214
DMA3MR/	212
DMA4MR/	213

⑤

CPUMABxx/	SIG SLW	GND STRIP WX††
CPUMAB12/	268	68
CPUMAB13/	286	86
CPUMAB14/	280	80
CPUMAB15/	293	93
CPUMAB16/	285	85
CPUMAB17/	283	83
CPUMAB18/	17	17
CPUMAB19/	249	49
CPUMAB20/	218	18

⑥ EXTIMABxx/

EXTIMAB01/	65
EXTIMAB02/	66
EXTIMAB03/	67
EXTIMAB04/	68
EXTIMAB05/	69
EXTIMAB06/	70
EXTIMAB07/	71
EXTIMAB08/	72
EXTIMAB09/	73
EXTIMAB10/	74
EXTIMAB11/	75
EXTIMAB12/	90
EXTIMAB13/	91
EXTIMAB14/	93
EXTIMAB15/	94
EXTIMAB16/	95
EXTIMAB17/	96
EXTIMAB18/	99
EXTIMAB19/	97

⑦ DMAxRA/

DMA1RA/	227
DMA2RA/	219
DMA3RA/	220
DMA4RA/	216

⑧ E64KNx-x /

E64KN0-1/	208
E64KN1-1/	226
E64KN0-2/	207
E64KN1-2/	26
E64KN0-3/	206
E64KN1-3/	25
E64KN0-4/	209
E64KN1-4/	27

⑨ MOSMAxx/

MOSMA00/	258
MOSMA01/	257
MOSMA02/	260
MOSMA03/	259
MOSMA04/	265
MOSMA05/	264
MOSMA06/	237
MOSMA07/	279†

⑩ RAS64Kx-x/

RAS64K0-1/	236
RAS64K1-1/	244
RAS64K0-2/	32
RAS64K1-2/	30
RAS64K0-3/	33
RAS64K1-3/	233
RAS64K0-4/	232
RAS64K1-4/	31

⑪ SLTBLKx /

SLTBLK0/	238
SLTBLK1/	235
SLTBLK2/	241
SLTBLK3/	242

⑫ EXTOMABxx/

EXTOMAB01/	53
EXTOMAB02/	54
EXTOMAB03/	55
EXTOMAB04/	56
EXTOMAB05/	57
EXTOMAB06/	58
EXTOMAB07/	59
EXTOMAB08/	60
EXTOMAB09/	61
EXTOMAB10/	62
EXTOMAB11/	63
EXTOMAB12/	78
EXTOMAB13/	79
EXTOMAB14/	81
EXTOMAB15/	82
EXTOMAB16/	83
EXTOMAB17/	84
EXTOMAB18/	87
EXTOMAB19/	85

⑬ DMAxMAB20/

DMA1MAB20/	270
DMA2MAB20/	273
DMA3MAB20/	276
DMA4MAB20/	278

†FOR FUTURE USE ONLY

††TWISTED PAIR, BLUE WIRE TO GROUND. POSITION WX REFERS TO GROUND STRIP LOCATED BETWEEN SLOTS W AND X. PIN DESIGNATION NUMBERS ARE 4-50 AND 53-102.

2505A

Figure 5-1. Address Interface Signals (Sheet 2 of 2)

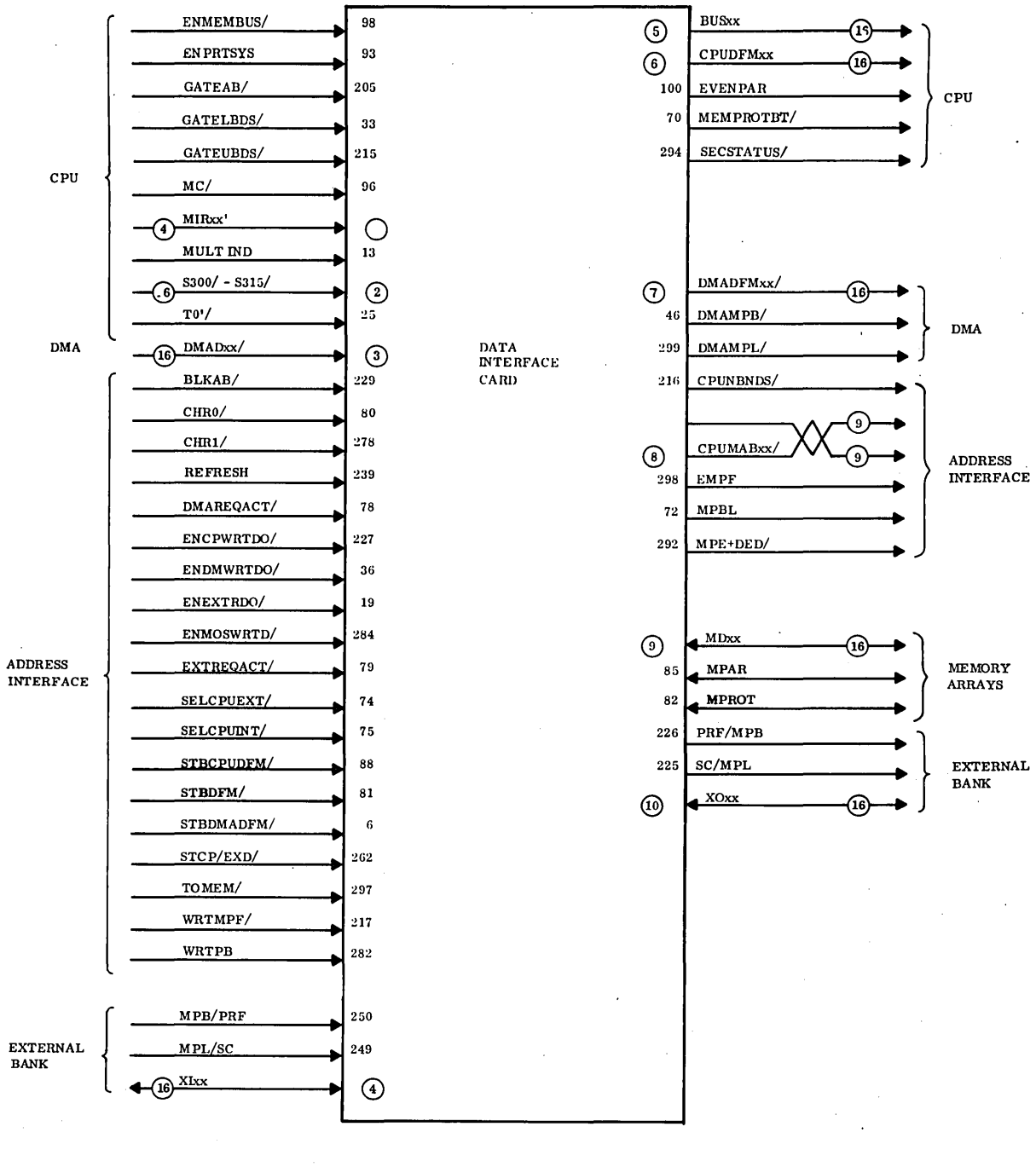


Figure 5-2. Data Interface Signals (Sheet 1 of 2)

① MIRxx

MIR28'	9
MIR29'	300
MIR30'	7
MIR31'	11

② S300/ - S315/

S300/	54
S301/	56
S302/	55
S303/	53
S304/	39
S305/	41
S306/	40
S307/	38
S308/	23
S309	28
S310/	24
S311/	22
S312/	29
S313/	35
S314/	15
S315/	27

③ DMADxx/

DMAD01/	76
DMAD02/	261
DMAD03/	31
DMAD04/	228
DMAD05/	253
DMAD06/	252
DMAD07/	42
DMAD08/	258
DMAD09/	257
DMAD10/	37
DMAD11/	45
DMAD12/	60
DMAD13/	62
DMAD14/	61
DMAD15/	59
DMAD16/	52

④ XIxx

XI01	209
XI02	208
XI03	207
XI04	206
XI05	211
XI06	213
XI07	218
XI08	214
XI09	212
XI10	210
XI11	220
XI12	222
XI13	224
XI14	223
XI15	221
XI16	219

⑤ BUSxx

BUS00	92
BUS01	94
BUS02	95
BUS03	91
BUS04	295
BUS05	43
BUS06	44
BUS07	57
BUS08	84
BUS09	97
BUS10	64
BUS11	296
BUS12	89
BUS13	99
BUS14	10
BUS15	283

⑥ CPUDFMxx

CPUDFM01	71
CPUDFM02	73
CPUDFM03	66
CPUDFM04	77
CPUDFM05	63
CPUDFM06	65
CPUDFM07	267
CPUDFM08	69
CPUDFM09	86
CPUDFM10	291
CPUDFM11	260
CPUDFM12	58
CPUDFM13	259
CPUDFM14	266
CPUDFM15	275
CPUDFM16	276

⑦ DMADFMxx

DMADFM01	264
DMADFM02	241
DMADFM03	5
DMADFM04	47
DMADFM05	14
DMADFM06	32
DMADFM07	17
DMADFM08	30
DMADFM09	16
DMADFM10	8
DMADFM11	18
DMADFM12	265
DMADFM13	12
DMADFM14	67
DMADFM15	21
DMADFM16	34

⑧ CPUMABxx

CPUMABxx	SIG SLV	GND STRIP VW†
CPUMAB12/	50	50
CPUMAB13/	48	48
CPUMAB14/	49	49
CPUMAB15/	51	50
CPUMAB16/	270	70
CPUMAB17/	26	26
CPUMAB18/	68	68
CPUMAB19/	240	40
CPUMAB20/	279	79

⑨ ECCxx

ECC00	87
ECC01	285
ECC02	288
ECC03	90
ECC04	289

⑩ MDxx

MD00	255
MD01	268
MD02	269
MD03	256
MD04	280
MD05	277
MD06	271
MD07	263
MD08	272
MD09	274
MD10	281
MD11	273
MD12	293
MD13	290
MD14	286
MD15	287

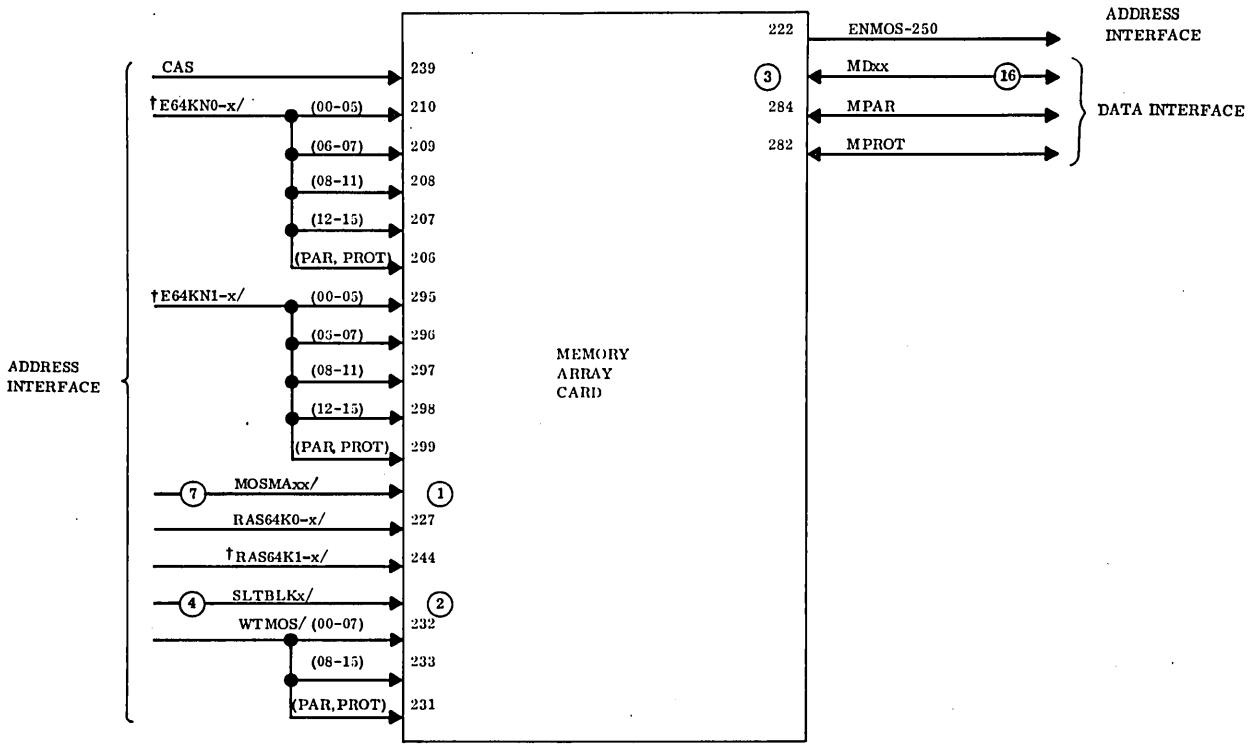
⑪ XOxx

XO01	233
XO02	232
XO03	231
XO04	230
XO05	235
XO06	237
XO07	242
XO08	238
XO09	236
XO10	234
XO11	244
XO12	246
XO13	248
XO14	247
XO15	245
XO16	243

†TWISTED PAIR, BLUE WIRE TO GROUND. POSITION VW REFERS TO STRIP LOCATED BETWEEN SLOTS V AND W. PIN DESIGN NUMBERS ARE 4-50 AND 53-102.

2506A

Figure 5-2. Data Interface Signals (Sheet 2 of 2)



1 MOSMAxx/

MOSMA00/	258
MOSMA01/	257
MOSMA02/	260
MOSMA03/	259
MOSMA04/	265
MOSMA05/	264
MOSMA06/	237

2 SLTBLKx/

SLTBLK0/	238
SLTBLK1/	235
SLTBLK2/	241
SLTBLK3/	242

3 MDxx

MD00	255
MD01	268
MD02	269
MD03	256
MD04	280
MD05	277
MD06	271
MD07	263
MD08	272
MD09	274
MD10	281
MD11	273
MD12	293
MD13	290
MD14	286
MD15	287

†THE FOLLOWING PINS ARE STRAPPED TOGETHER ON THE BACKPLANE:

E64KN0-x/: 206, 207, 208, 209, 210
E64KN1-x/: 295, 296, 297, 298, 299
RAS64K1-x/: 227, 228, 229, 230
WTMOS/: 231, 232, 233

2507

Figure 5-3. Large Memory Array Signals

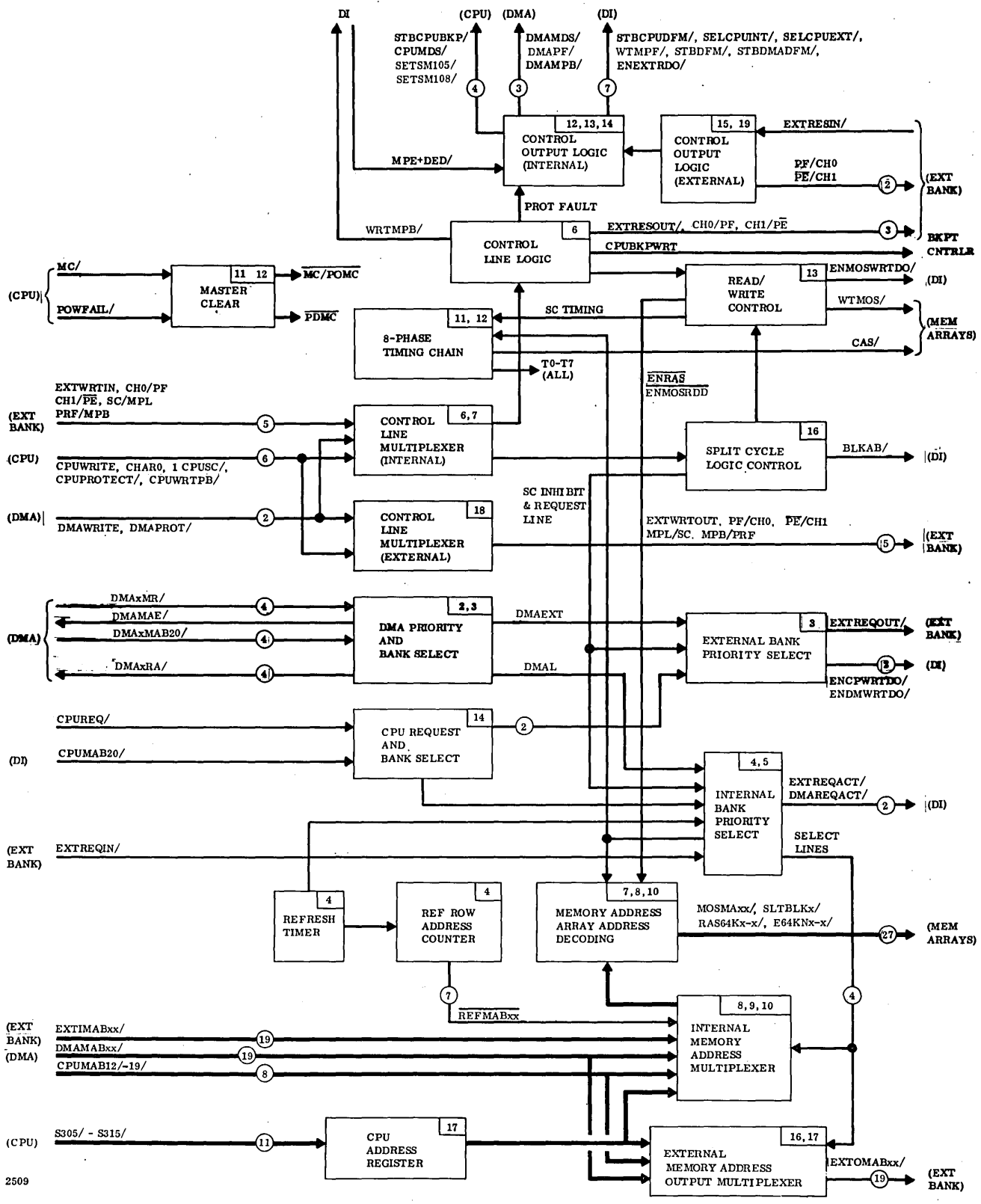


Figure 5-4. Address Interface Block Diagram

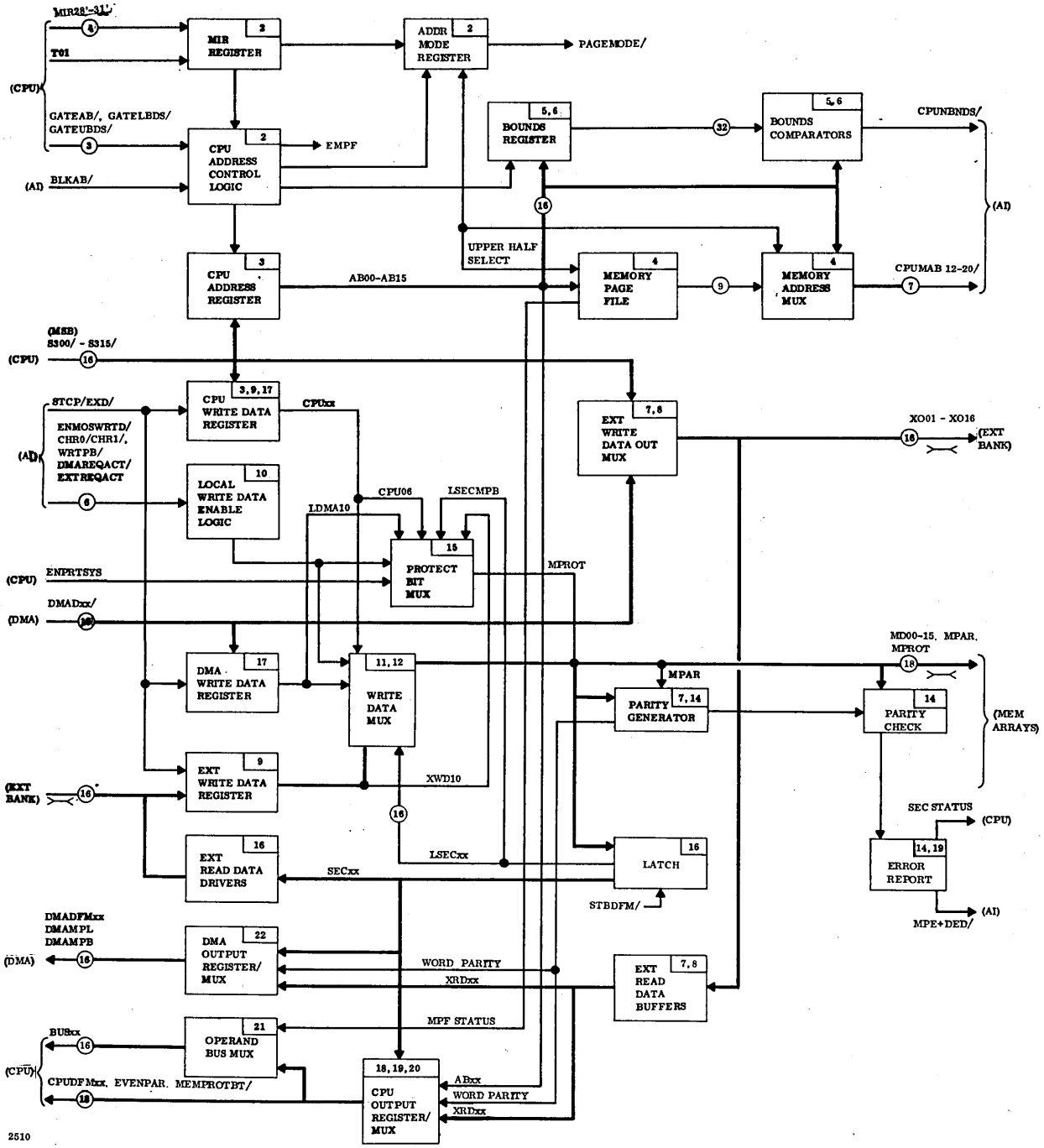
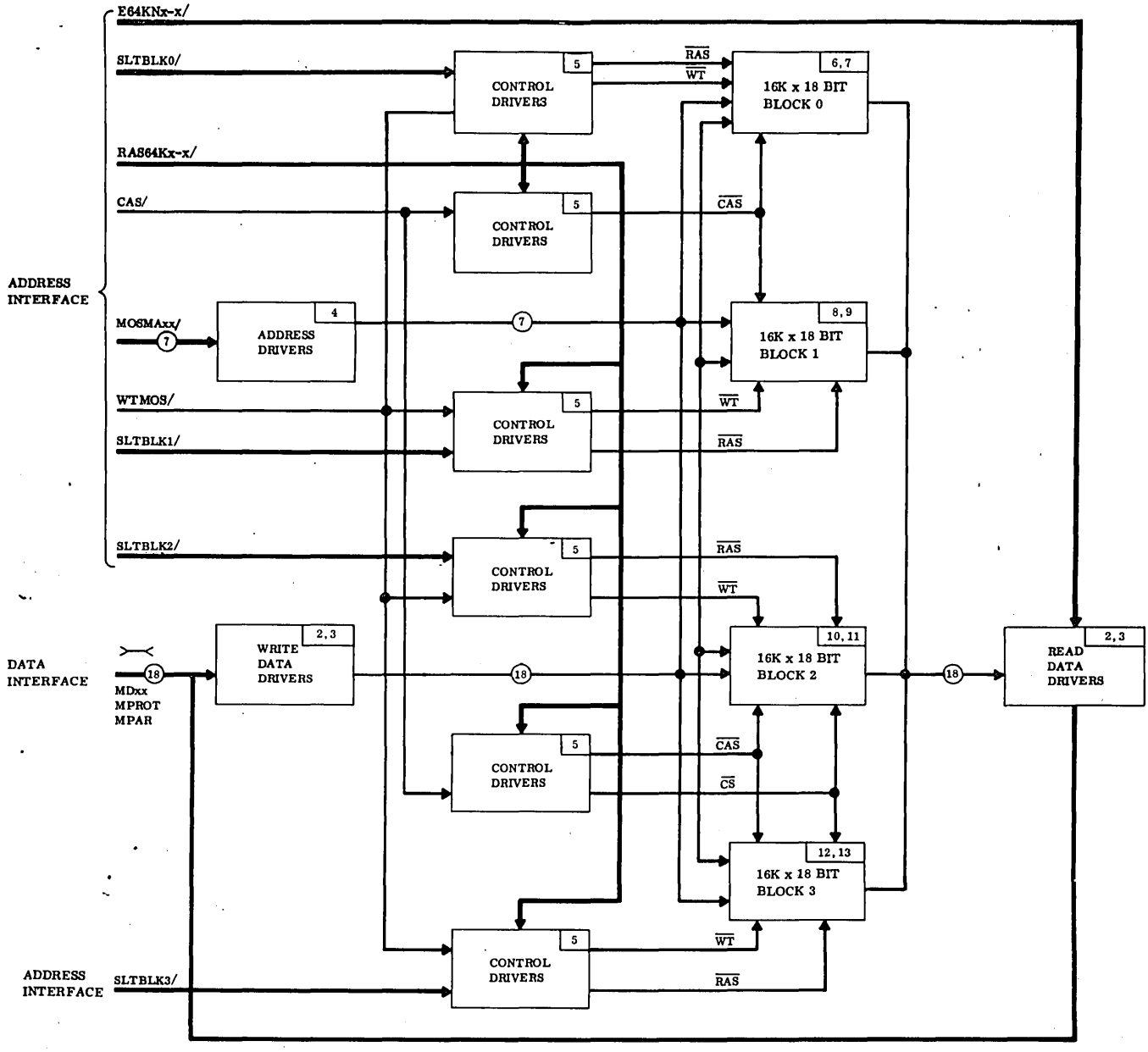
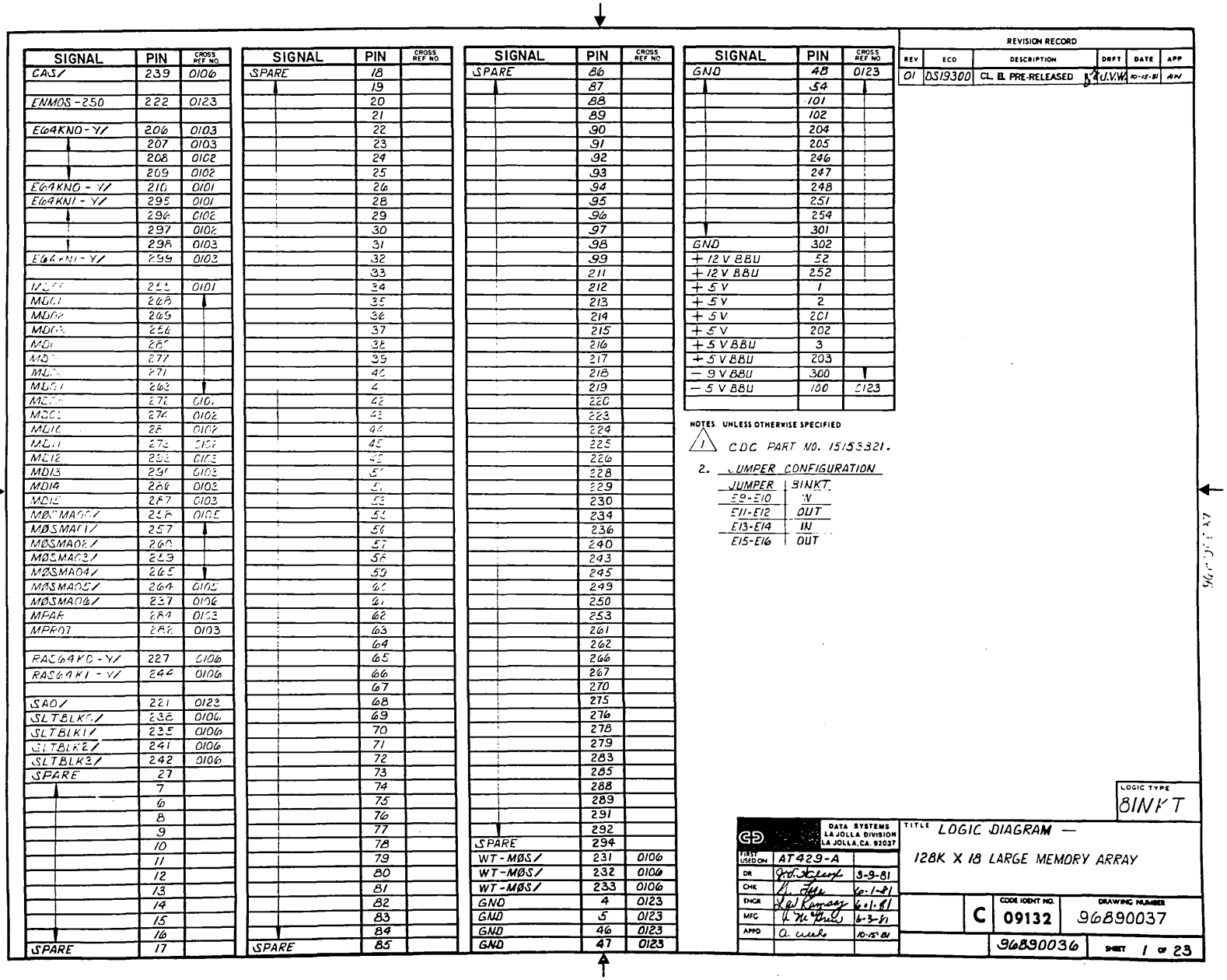


Figure 5-5. Data Interface Block Diagram



2511

Figure 5-6. One Half of Memory Array Block Diagram



SIGNAL	PIN	CROSS REF NO
CAS/	239	0106
ENMOS-250	222	0123
EG4KNO-Y/	206	0103
	207	0103
	208	0102
	209	0102
EG4KNO-Y/	210	0101
EG4KN1-Y/	295	0101
	296	0102
	297	0102
	298	0103
EG4KN1-Y/	299	0103
MD0/	265	0101
MD1/	268	
MD2/	269	
MD3/	256	
MD4/	280	
MD5/	277	
MD6/	271	
MD7/	263	
MD8/	272	0101
MD9/	274	0102
MD10/	28	0102
MD11/	278	0103
MD12	283	0103
MD13	291	0103
MD14	266	0102
MD15	287	0103
MØSMA00/	226	0105
MØSMA01/	257	
MØSMA02/	260	
MØSMA03/	259	
MØSMA04/	265	
MØSMA05/	264	0105
MØSMA06/	237	0106
MPAR	289	0103
MPR01	282	0103
RAS64KO-Y/	227	0106
RAS64K1-Y/	244	0106
SAO/	221	0123
SLTBLK0/	238	0106
SLTBLK1/	235	0106
SLTBLK2/	241	0106
SLTBLK3/	242	0106
SPARE	27	
	7	
	6	
	8	
	9	
	10	
	11	
	12	
	13	
	14	
	15	
	16	
SPARE	17	

SIGNAL	PIN	CROSS REF NO
SPARE	18	
	19	
	20	
	21	
	22	
	23	
	24	
	25	
	26	
	28	
	29	
	30	
	31	
	32	
	33	
	34	
	35	
	36	
	37	
	38	
	39	
	40	
	4	
	42	
	43	
	44	
	45	
	46	
	5	
	51	
	53	
	55	
	54	
	57	
	56	
	59	
	61	
	62	
	63	
	64	
	65	
	66	
	67	
	68	
	69	
	70	
	71	
	72	
	73	
	74	
	75	
	76	
	77	
	78	
	79	
	80	
	81	
	82	
	83	
	84	
SPARE	85	

SIGNAL	PIN	CROSS REF NO
SPARE	86	
	87	
	88	
	89	
	90	
	91	
	92	
	93	
	94	
	95	
	96	
	97	
	98	
	99	
	211	
	212	
	213	
	214	
	215	
	216	
	217	
	218	
	219	
	220	
	223	
	224	
	225	
	226	
	228	
	229	
	230	
	234	
	236	
	240	
	243	
	245	
	249	
	250	
	253	
	261	
	262	
	266	
	267	
	270	
	275	
	276	
	278	
	279	
	283	
	285	
	288	
	289	
	291	
	292	
SPARE	294	
WT-MØS/	231	0106
WT-MØS/	232	0106
WT-MØS/	233	0106
GND	4	0123
GND	5	0123
GND	46	0123
GND	47	0123

SIGNAL	PIN	CROSS REF NO
GND	48	0123
	54	
	101	
	102	
	204	
	205	
	246	
	247	
	248	
	251	
	254	
	301	
GND	302	
+12 V BBU	52	
+12 V BBU	252	
+5 V	1	
+5 V	2	
+5 V	201	
+5 V	202	
+5 V BBU	3	
+5 V BBU	203	
−9 V BBU	300	
−5 V BBU	100	0123

REVISION RECORD

REV	ECO	DESCRIPTION	DRFT	DATE	APP
01	DS19300	CL. B. PRE-RELEASED	U.V.W.	10-15-81	AM

NOTES UNLESS OTHERWISE SPECIFIED

1. CDC PART NO. 15153321.
2. JUMPER CONFIGURATION

JUMPER	BINKT
E9-E10	IN
E11-E12	OUT
E13-E14	IN
E15-E16	OUT

DATA SYSTEMS LA JOLLA DIVISION LA JOLLA, CA. 92037

FIRST USED ON	AT429-A	
DR		3-9-81
CHK		6-1-81
ENGR		6-1-81
MFG		6-3-81
APPD		10-15-81

LOGIC TYPE 8INKT

TITLE LOGIC DIAGRAM —
128K X 18 LARGE MEMORY ARRAY

C | CODE IDENT NO. 09132 | DRAWING NUMBER 96890037

96890036 | SHEET 1 OF 23

Figure 5-7. Large Memory Array Logic Diagram (Sheet 1 of 23)

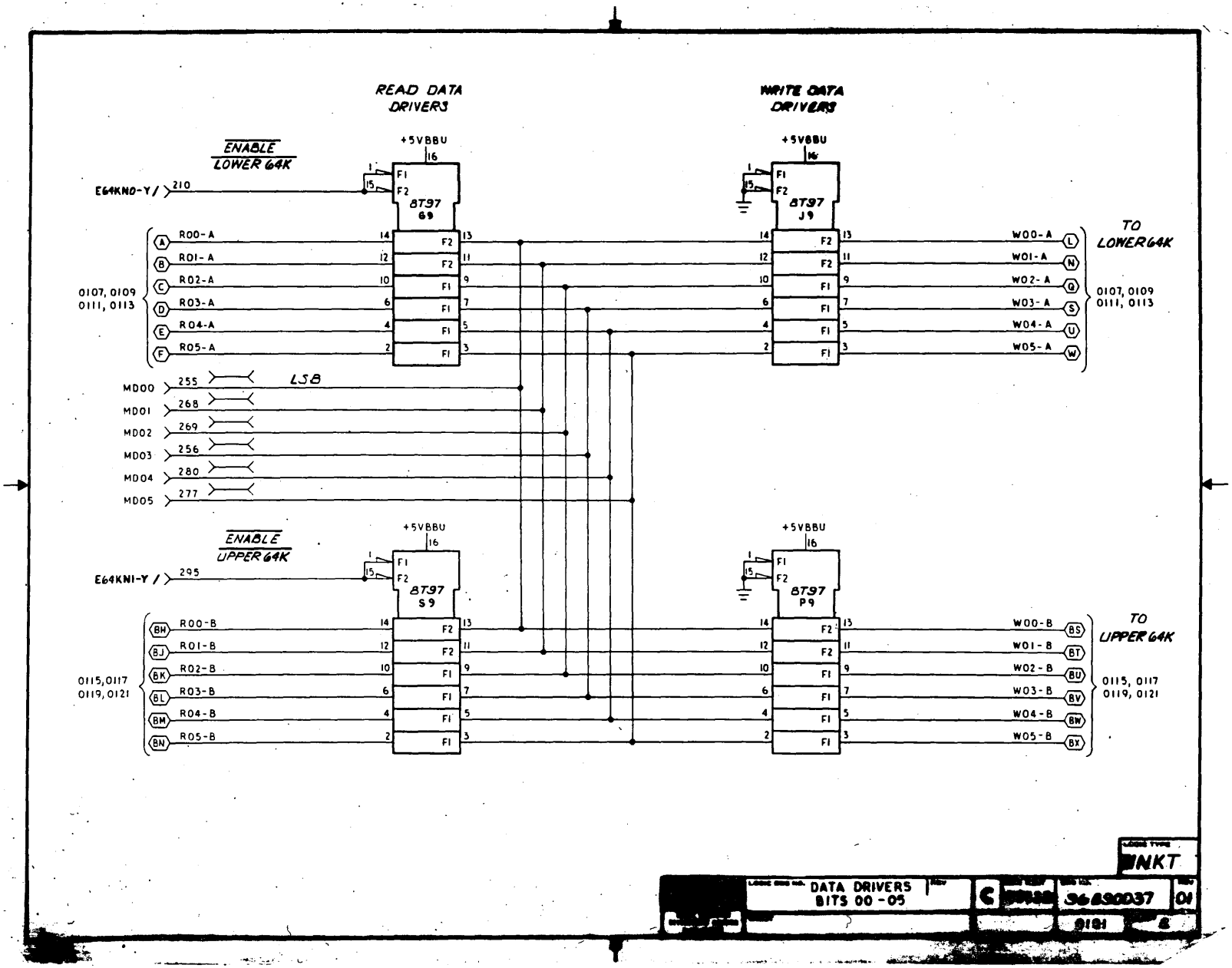


Figure 5-7. Large Memory Array Logic Diagram (Sheet 2 of 23)

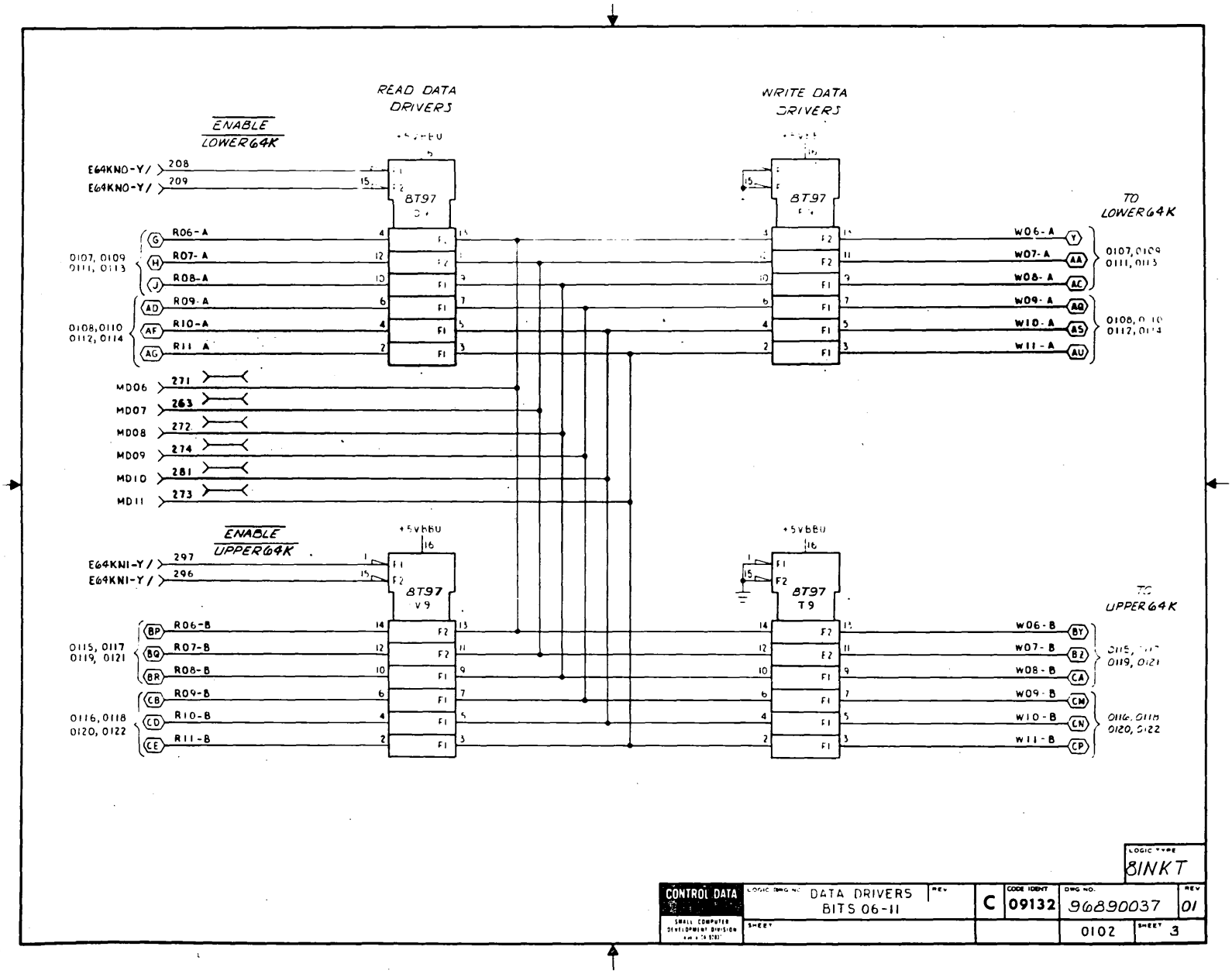


Figure 5-7. Large Memory Array Logic Diagram (Sheet 3 of 23)

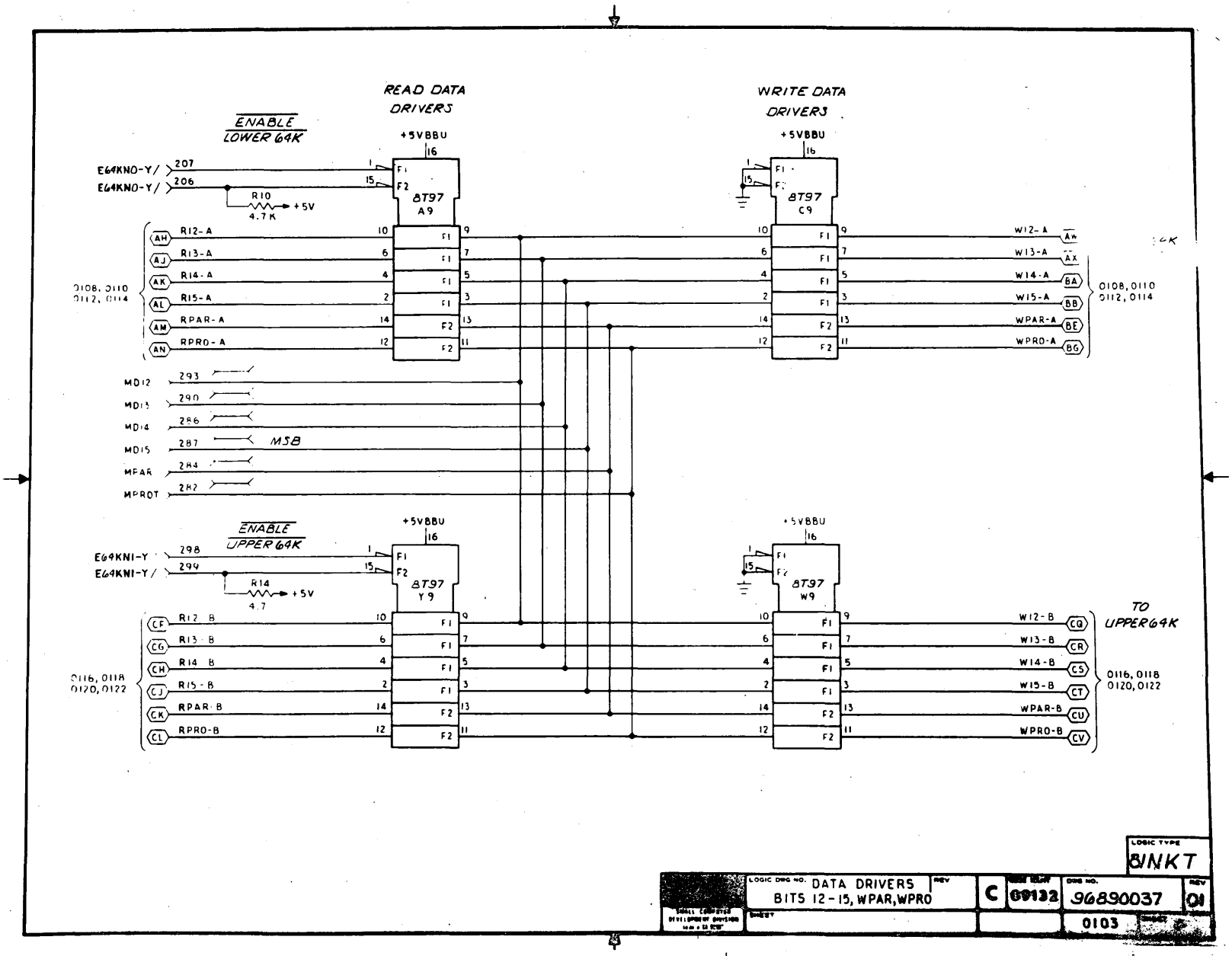


Figure 5-7. Large Memory Array Logic Diagram (Sheet 4 of 23)

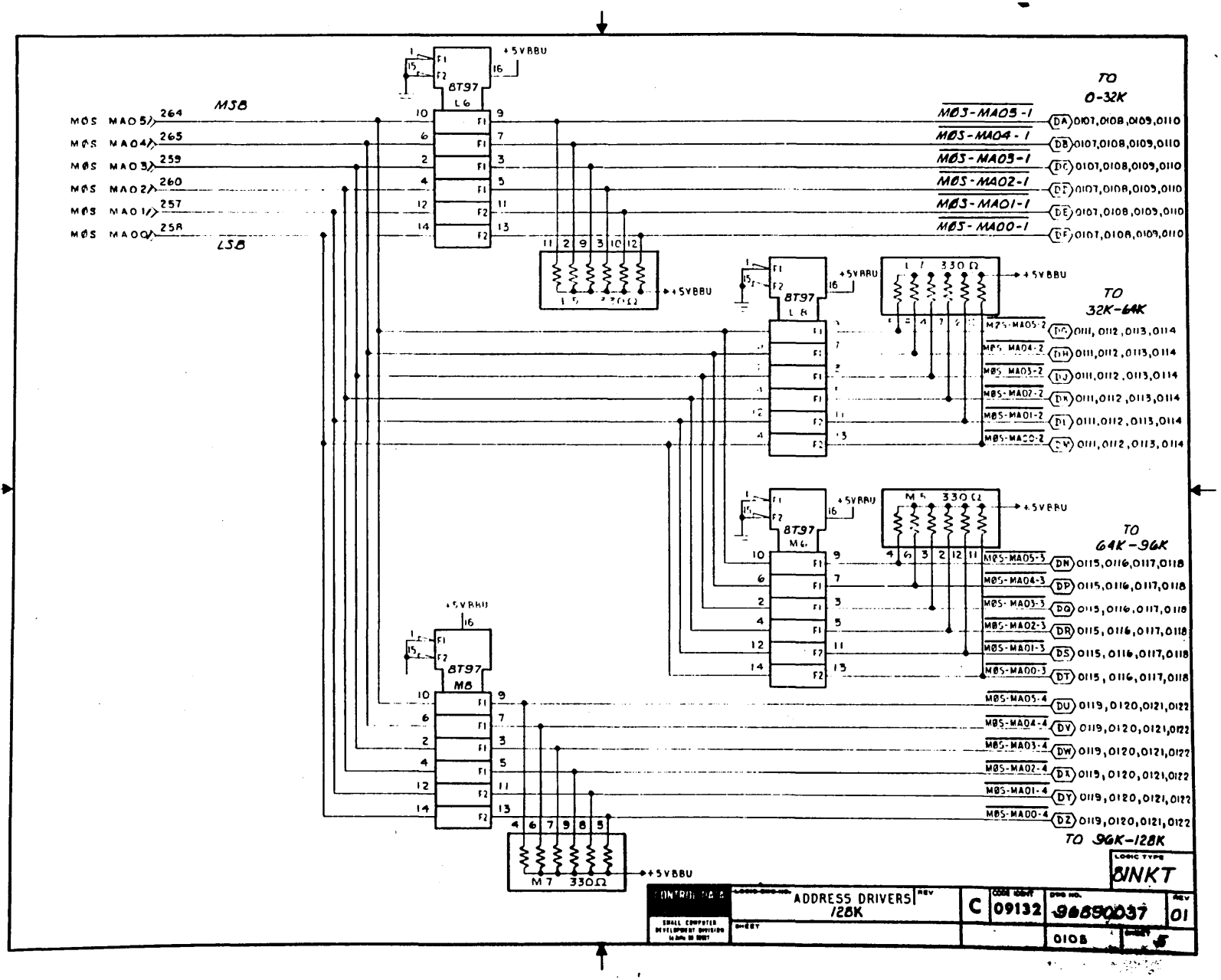


Figure 5-7. Large Memory Array Logic Diagram (Sheet 5 of 23)

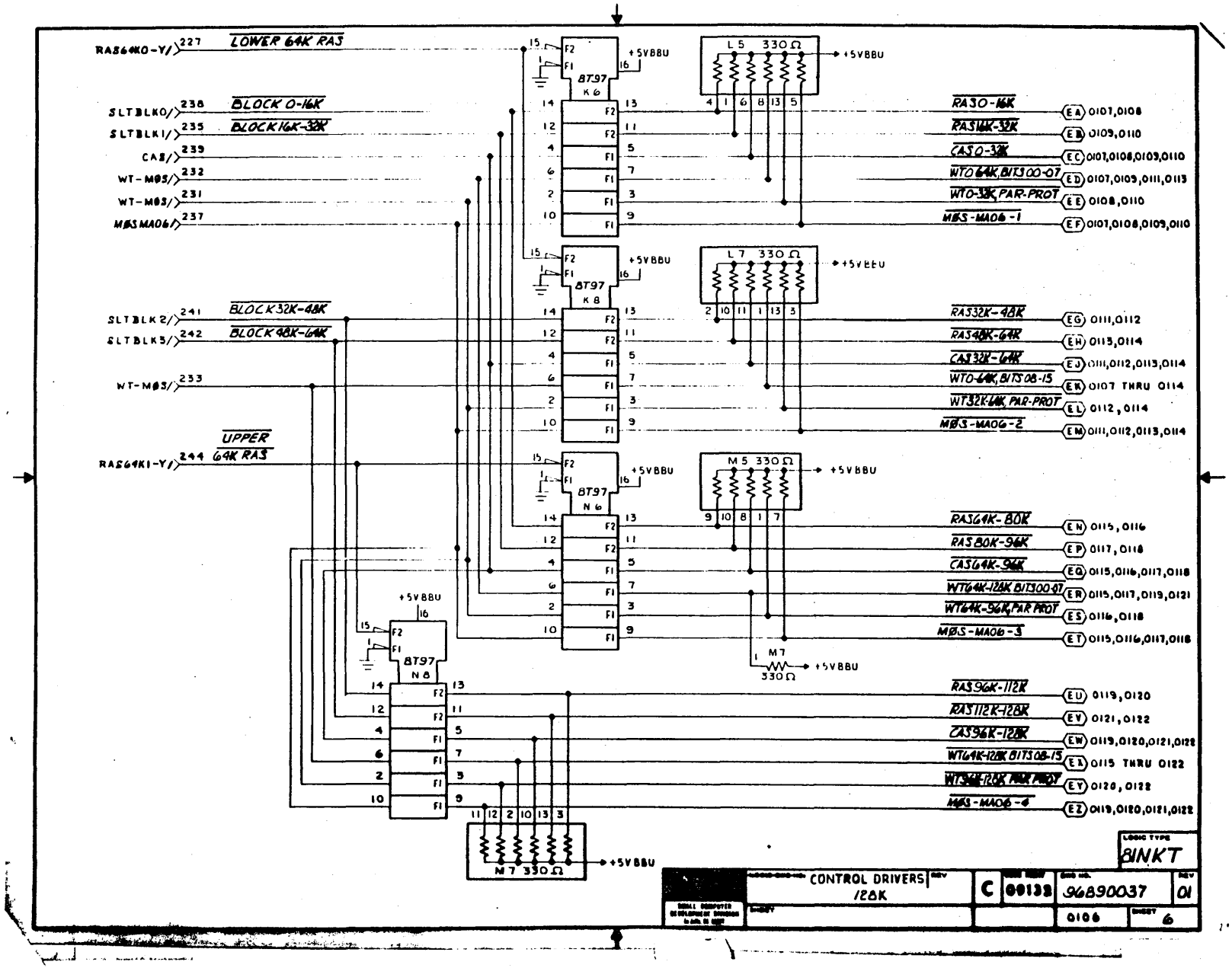


Figure 5-7. Large Memory Array Logic Diagram (Sheet 6 of 23)

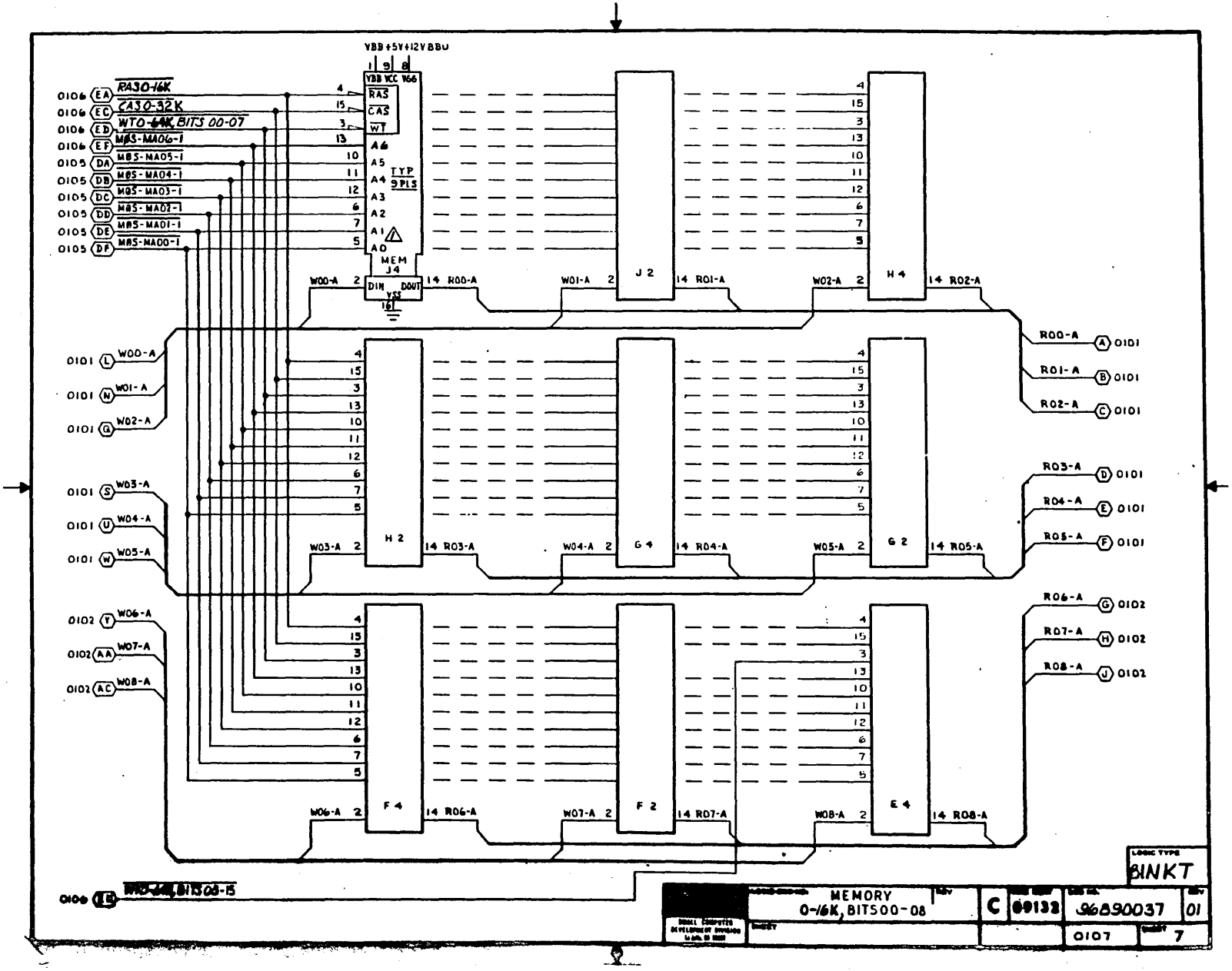


Figure 5-7. Large Memory Array Logic Diagram (Sheet 7 of 23)

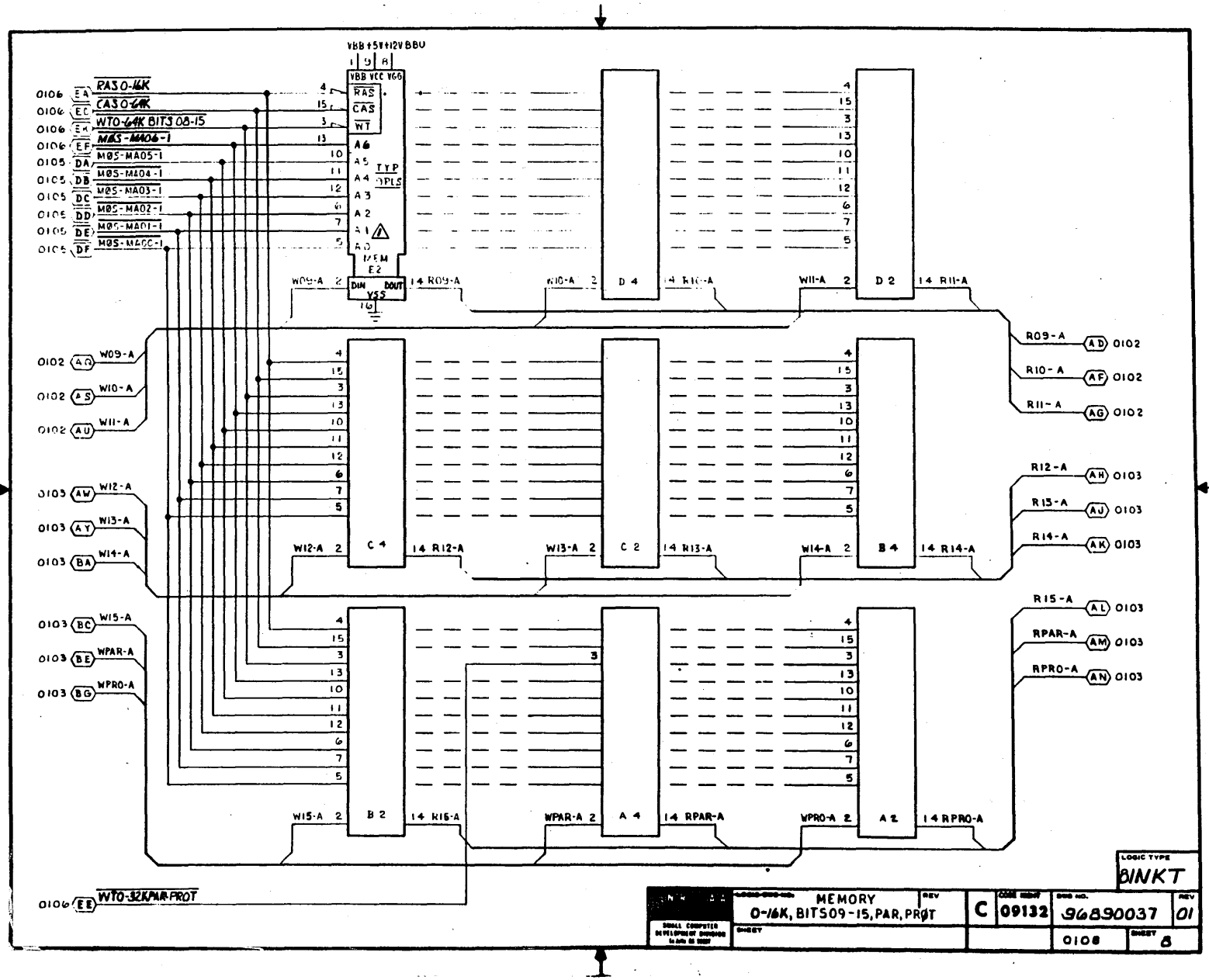


Figure 5-7. Large Memory Array Logic Diagram (Sheet 8 of 23)

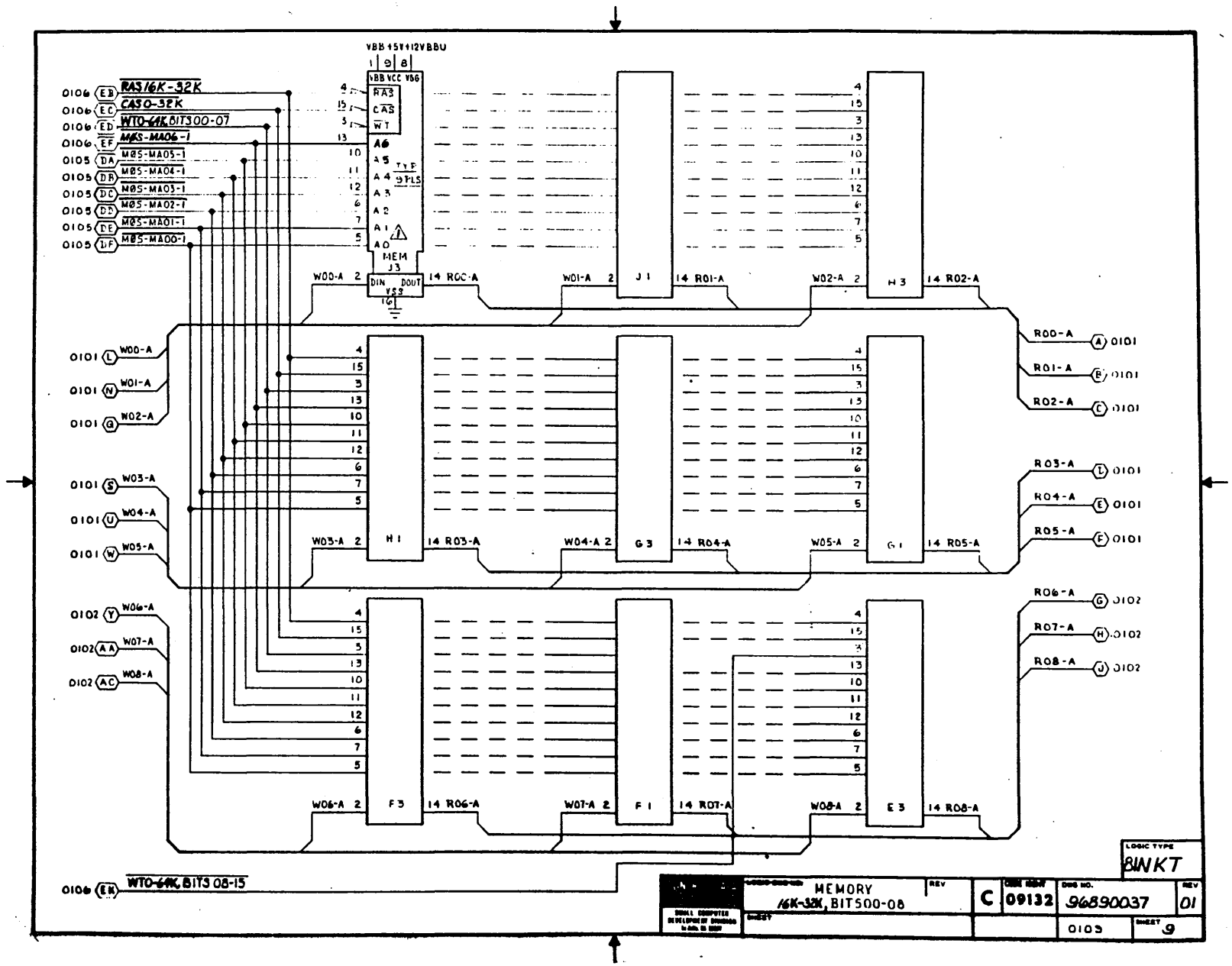


Figure 5-7. Large Memory Array Logic Diagram (Sheet 9 of 23)

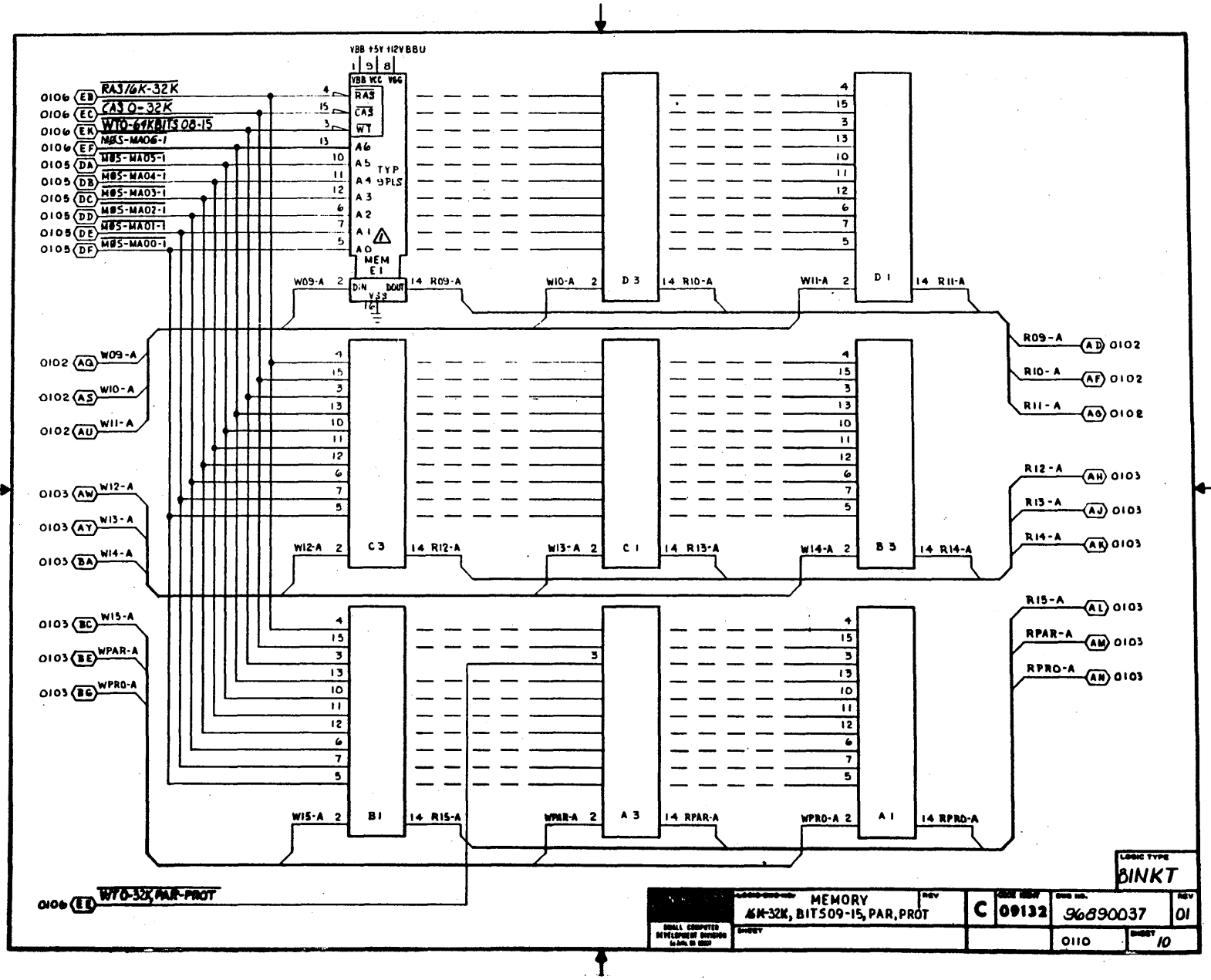


Figure 5-7. Large Memory Array Logic Diagram (Sheet 10 of 23)

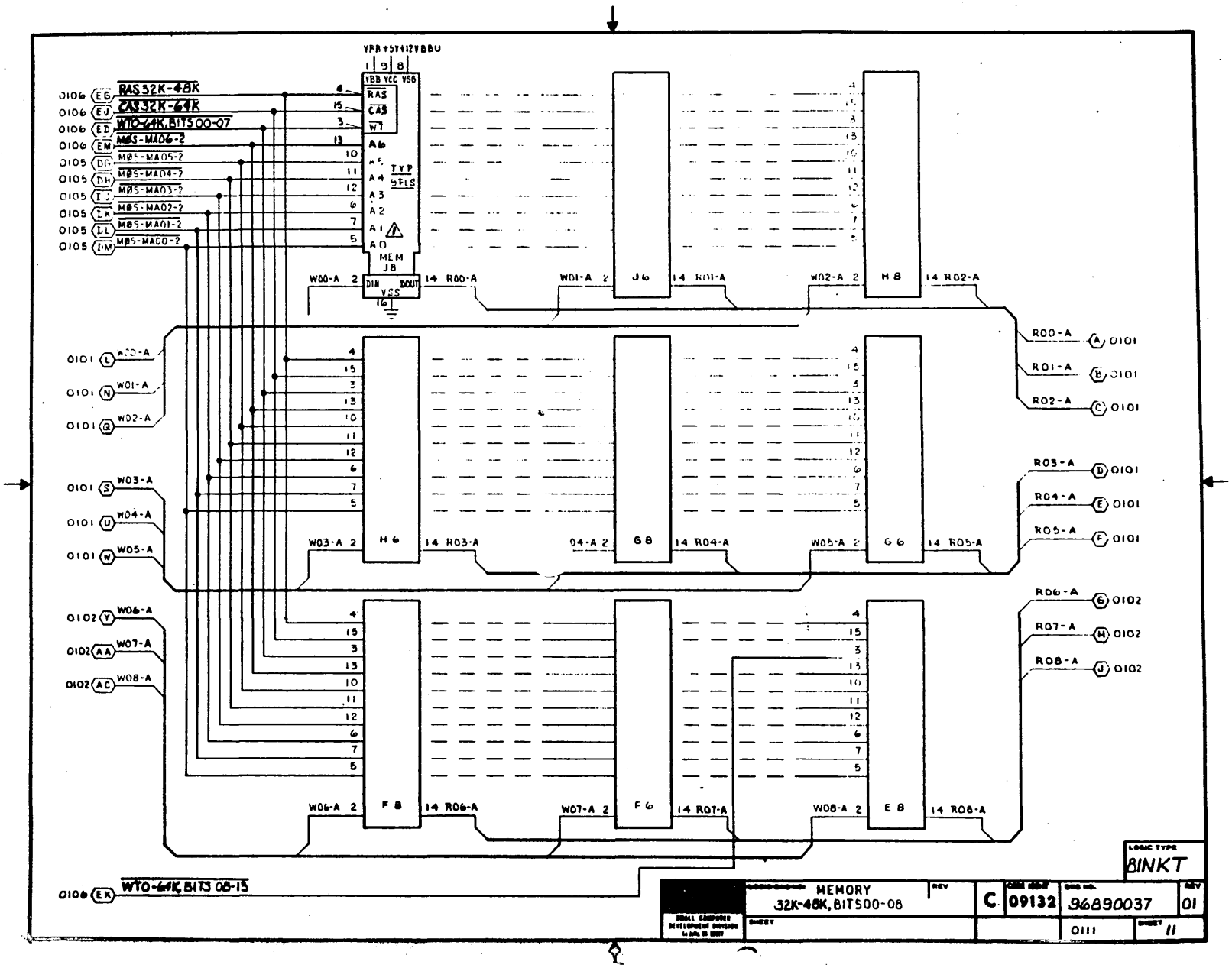


Figure 5-7. Large Memory Array Logic Diagram (Sheet 11 of 23)

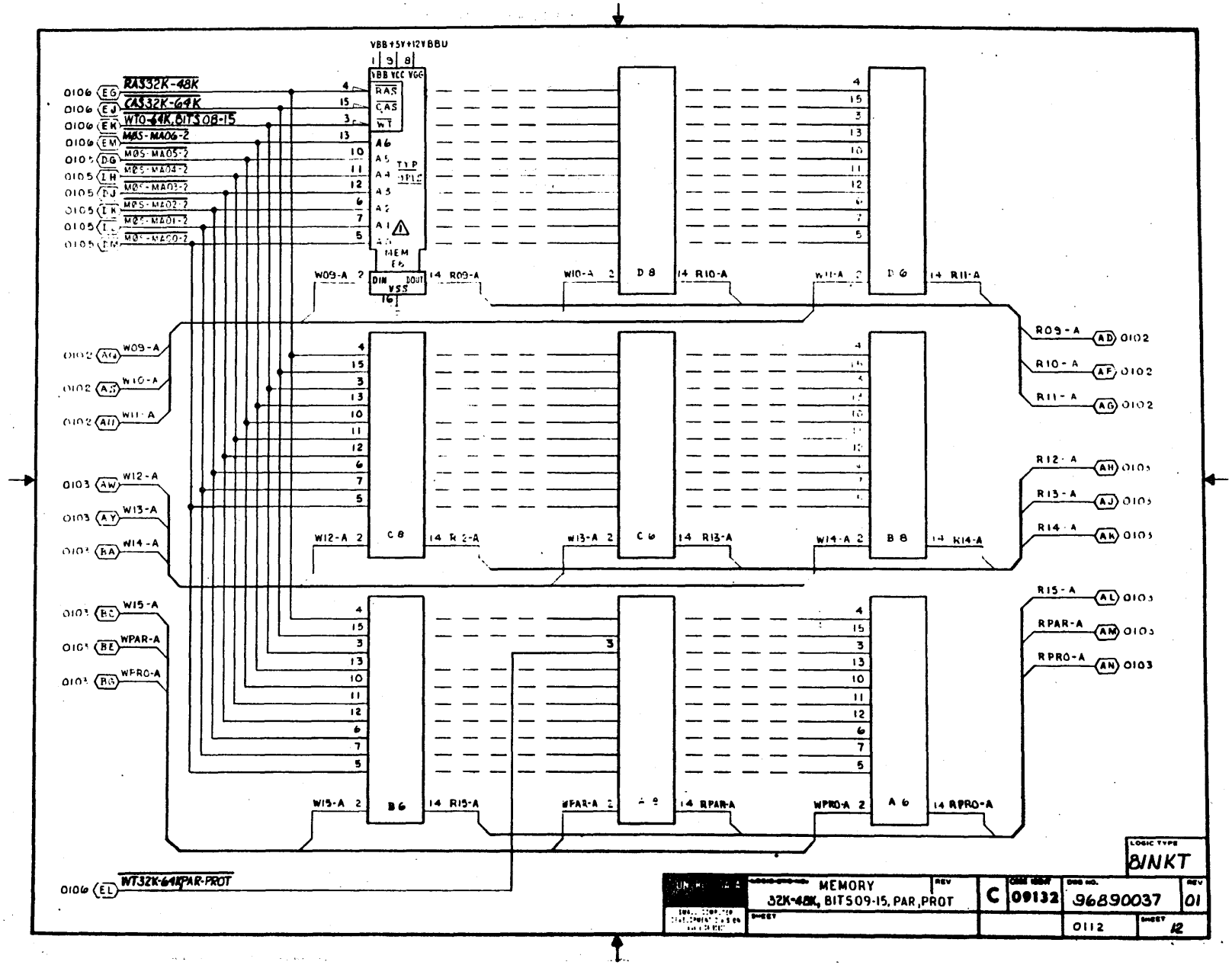


Figure 5-7. Large Memory Array Logic Diagram (Sheet 12 of 23)

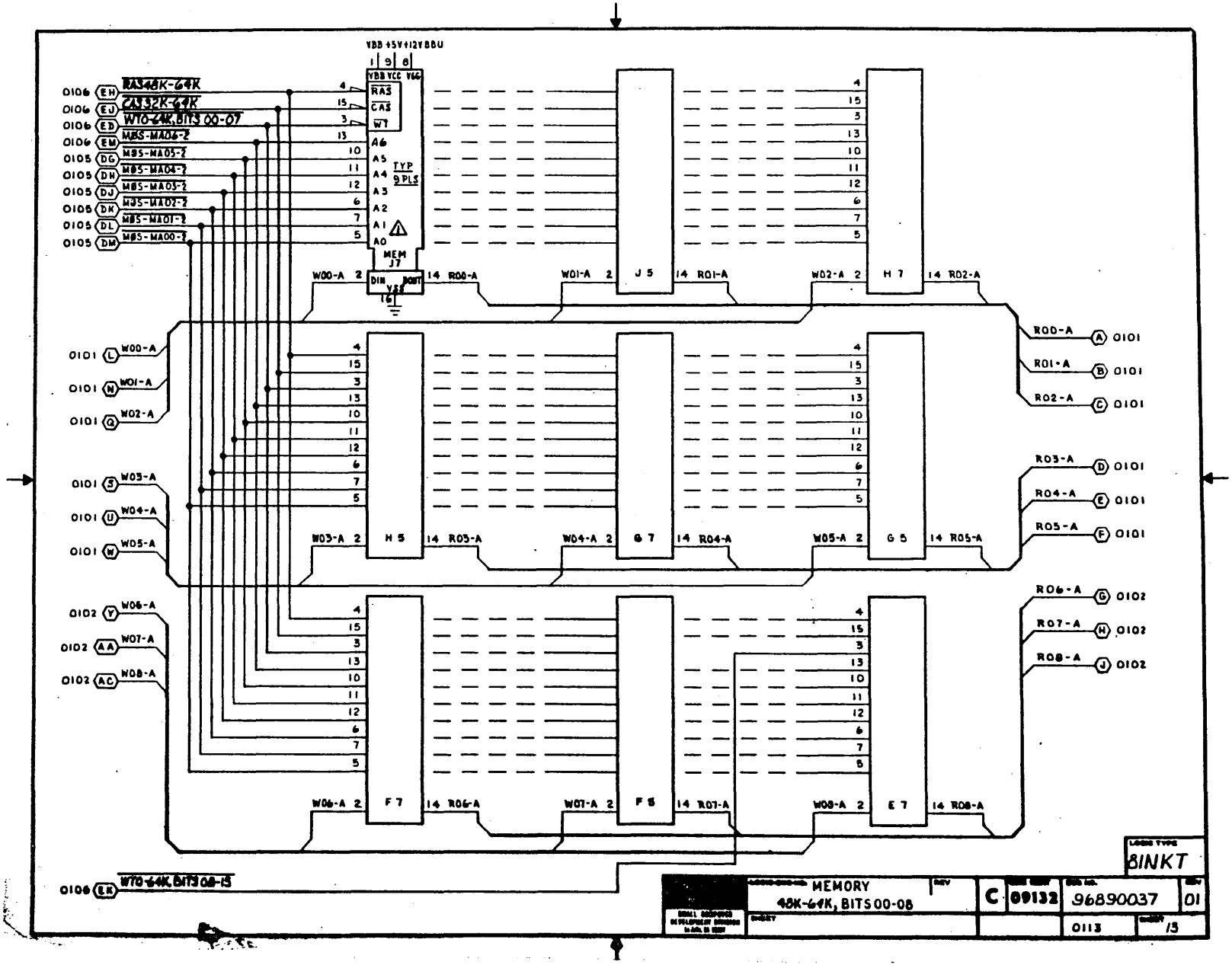


Figure 5-7. Large Memory Array Logic Diagram (Sheet 13 of 23)

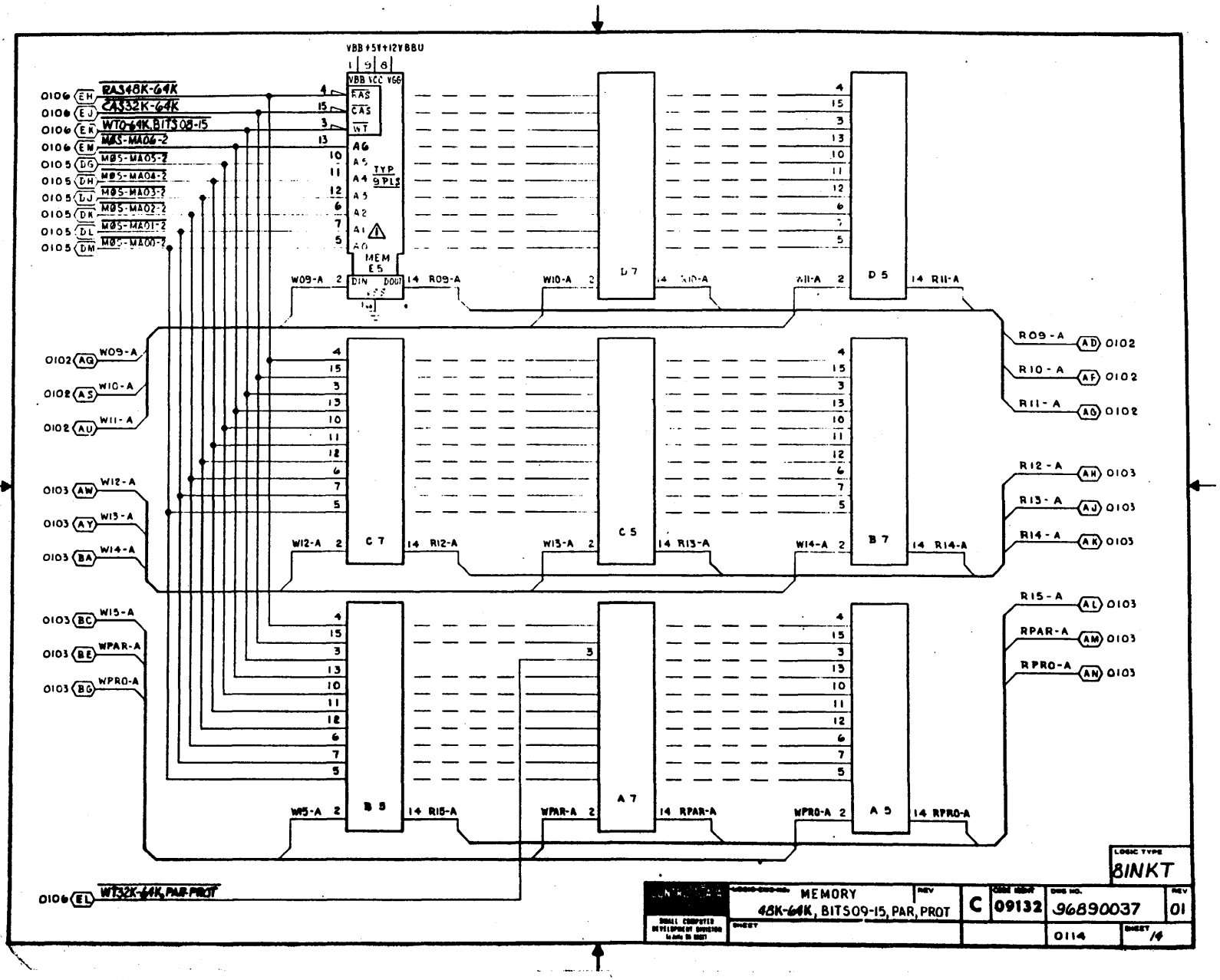


Figure 5-7. Large Memory Array Logic Diagram (Sheet 14 of 23)

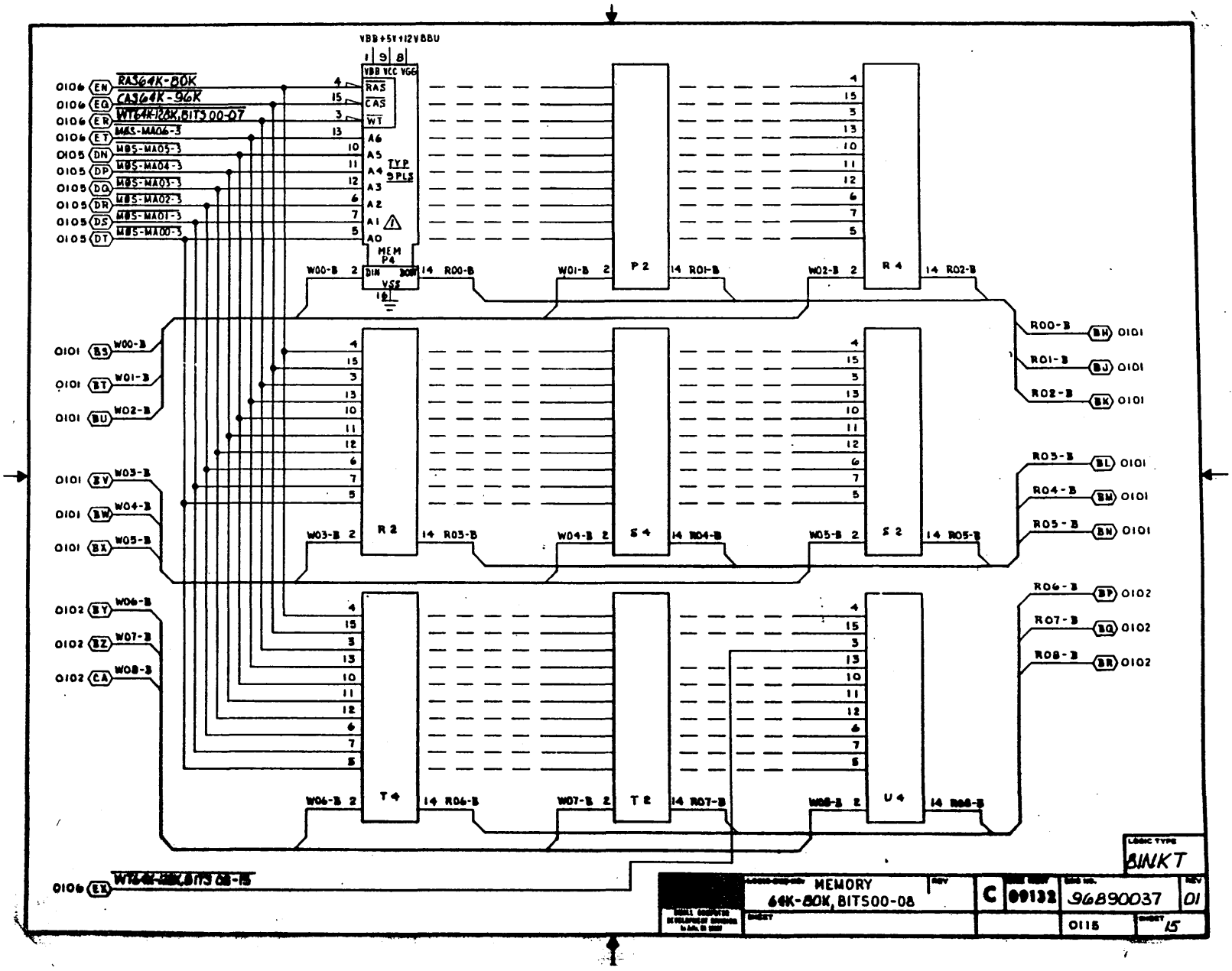


Figure 5-7. Large Memory Array Logic Diagram (Sheet 15 of 23)

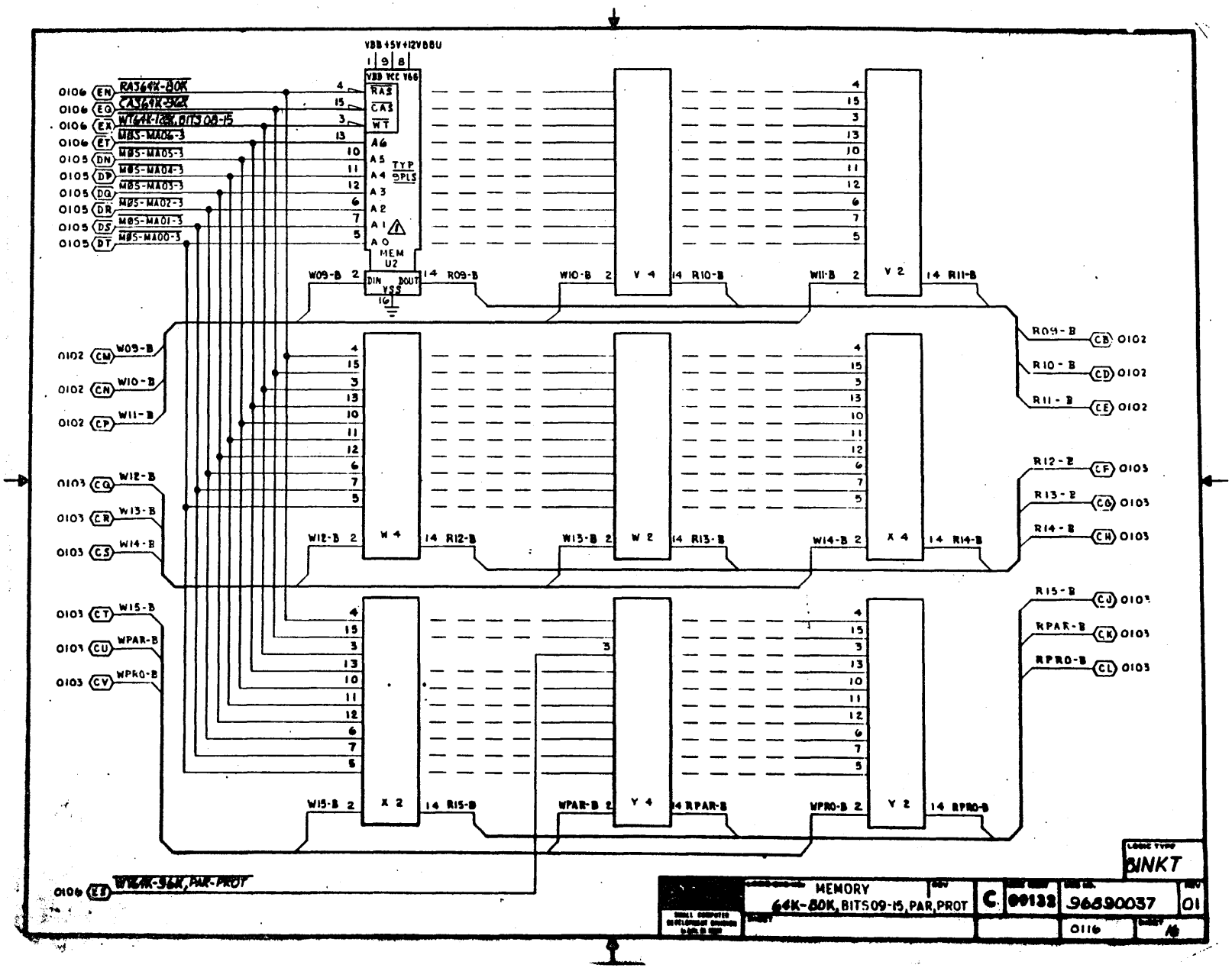


Figure 5-7. Large Memory Array Logic Diagram (Sheet 16 of 23)

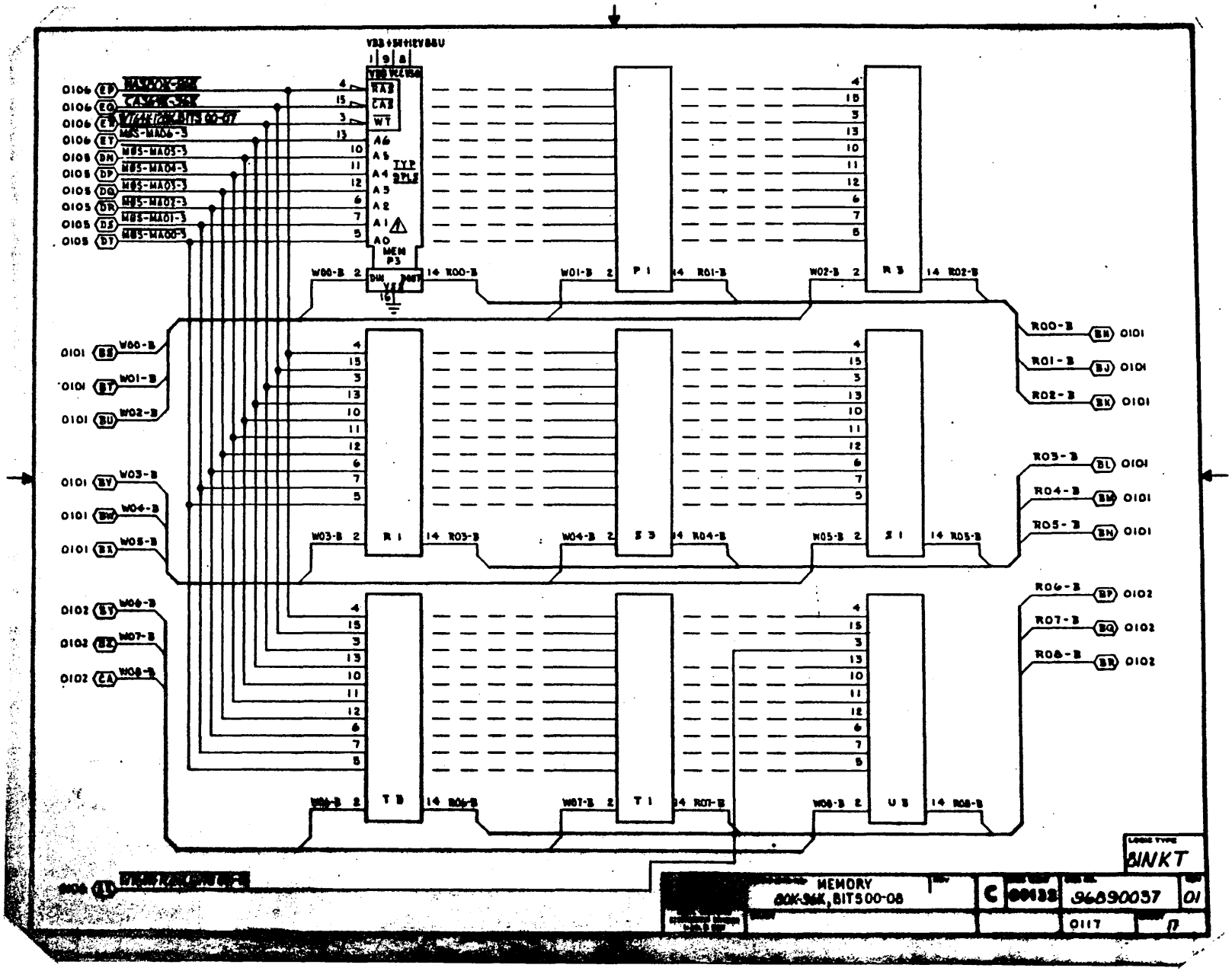


Figure 5-7. Large Memory Array Logic Diagram (Sheet 17 of 23)

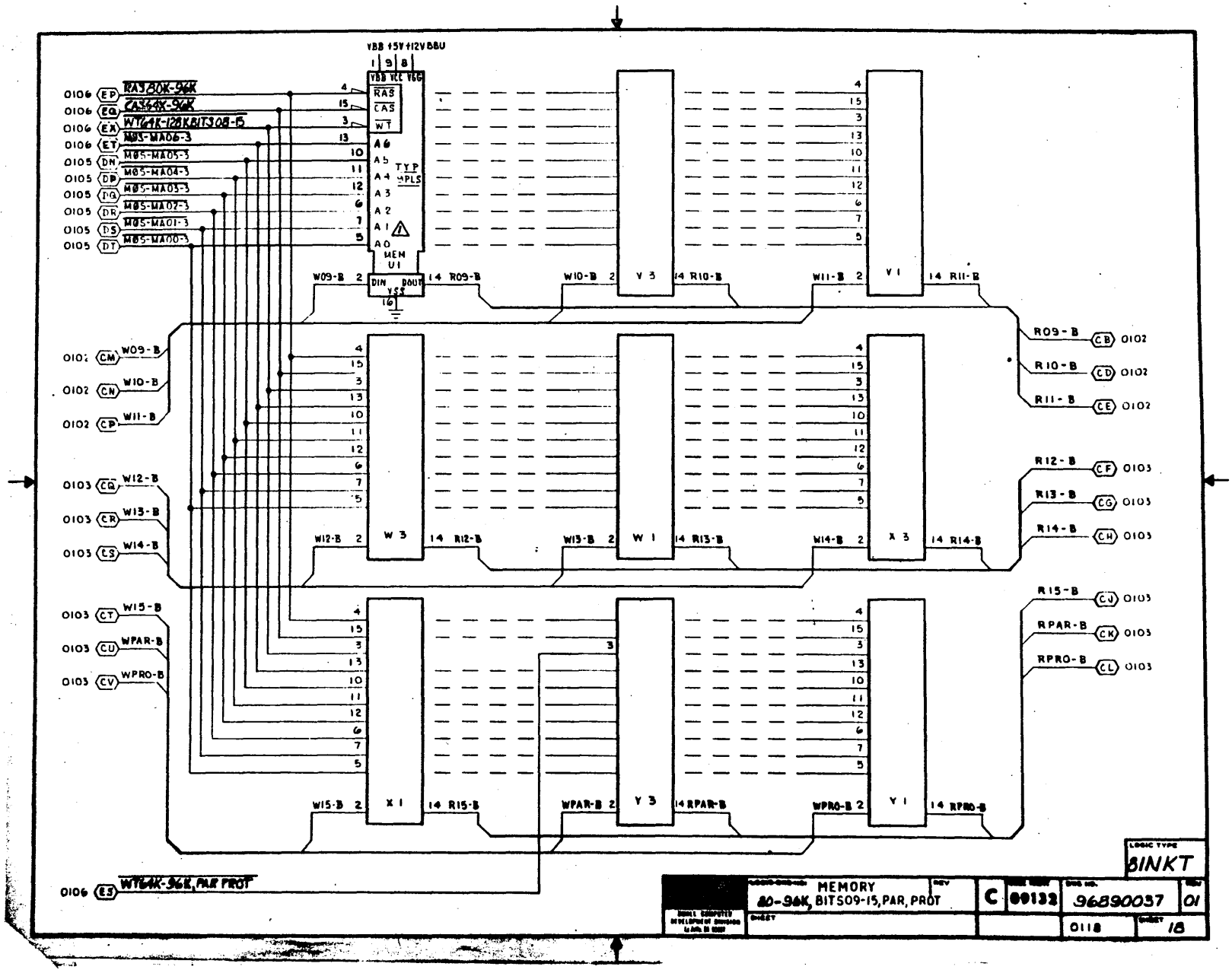


Figure 5-7. Large Memory Array Logic Diagram (Sheet 18 of 23)

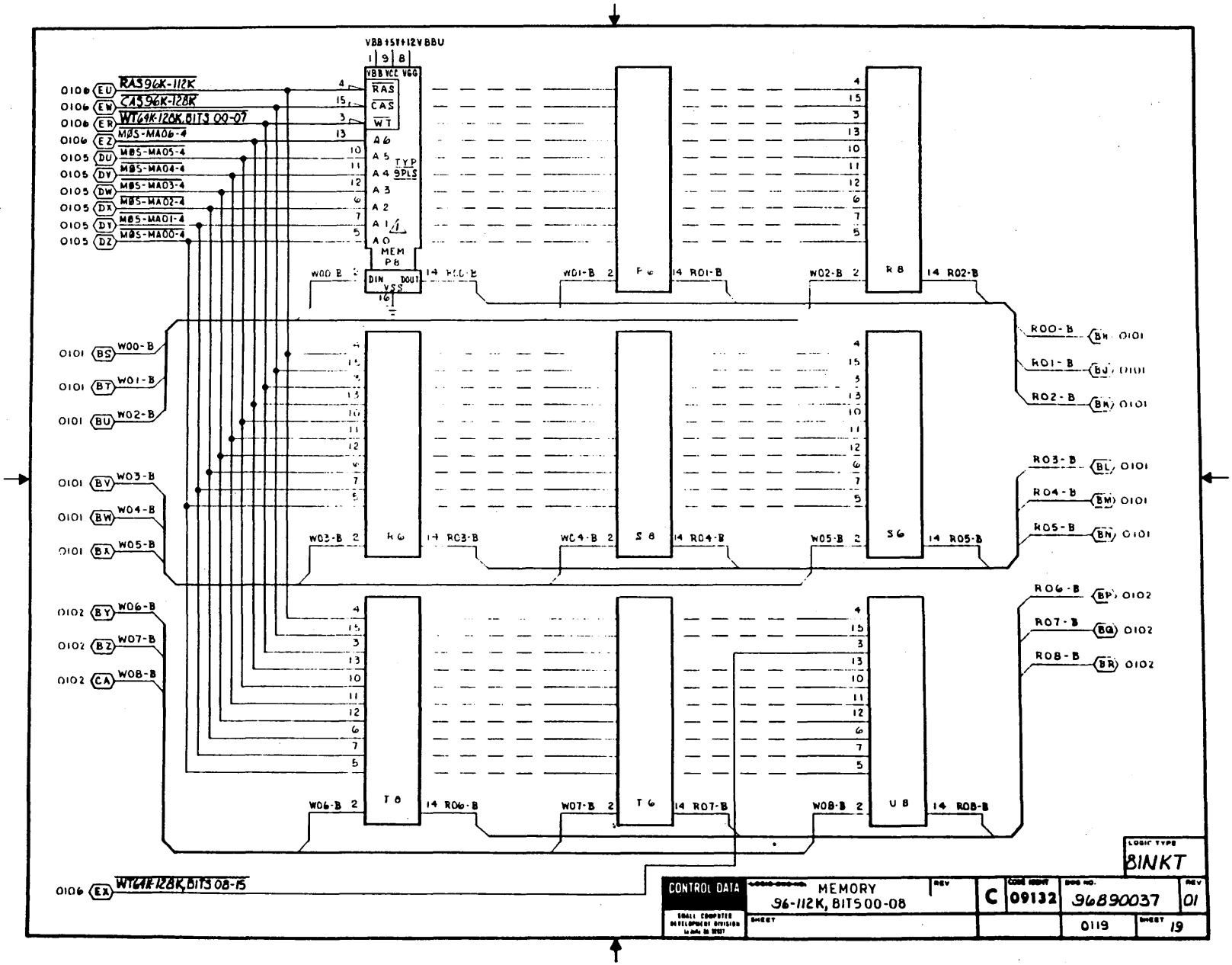


Figure 5-7. Large Memory Array Logic Diagram (Sheet 19 of 23)

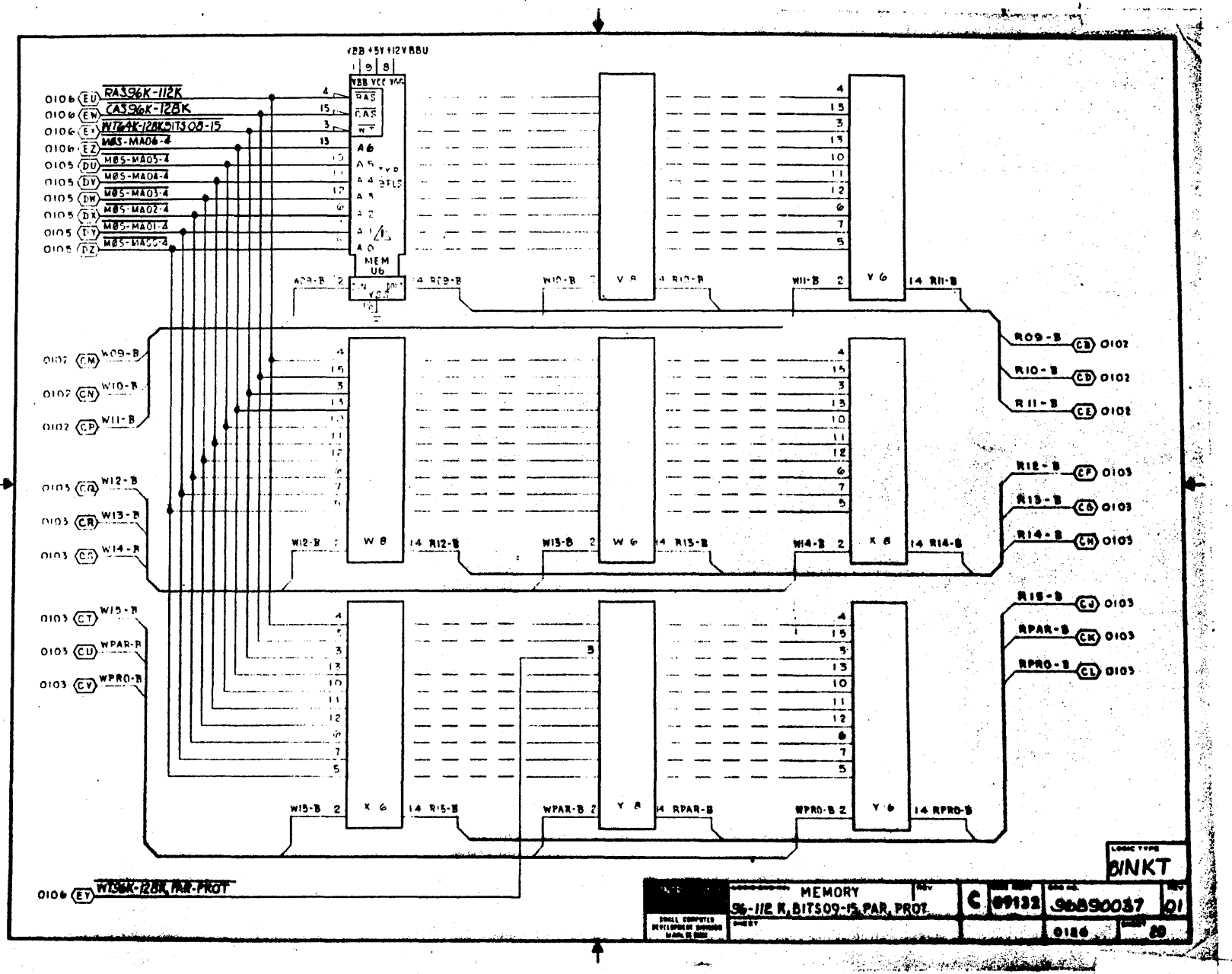


Figure 5-7. Large Memory Array Logic Diagram (Sheet 20 of 23)

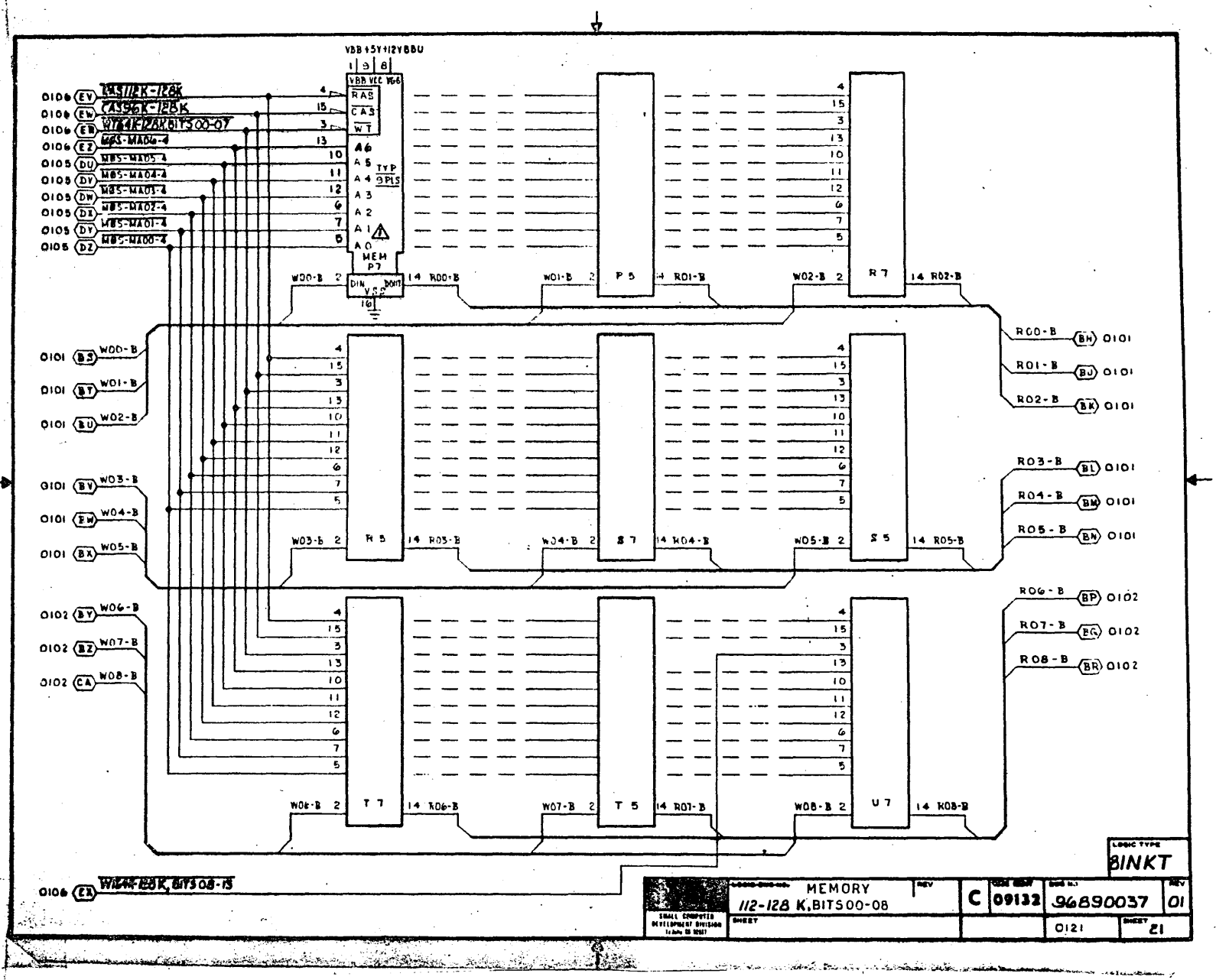


Figure 5-7. Large Memory Array Logic Diagram (Sheet 21 of 23)

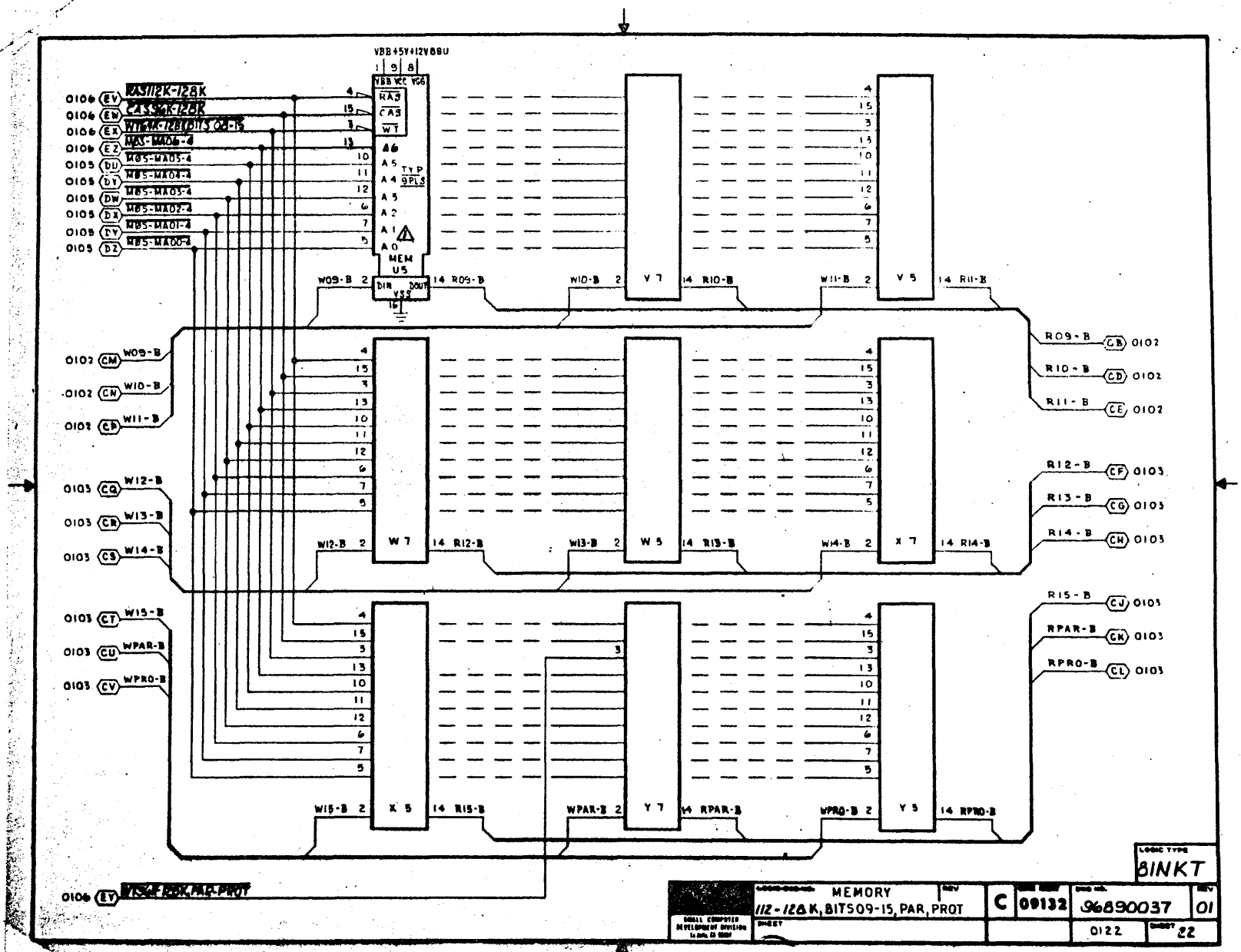


Figure 5-7. Large Memory Array Logic Diagram (Sheet 22 of 23)

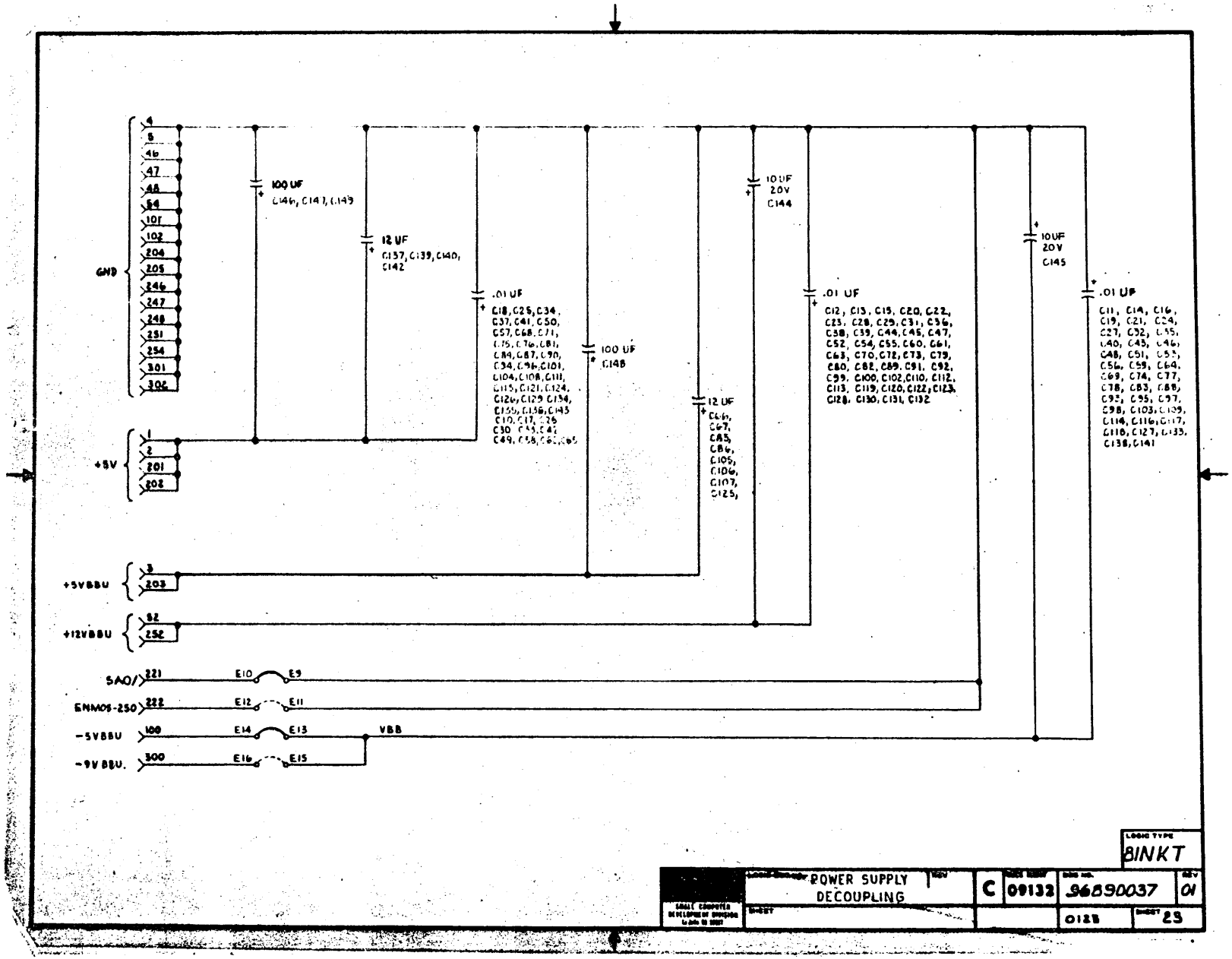
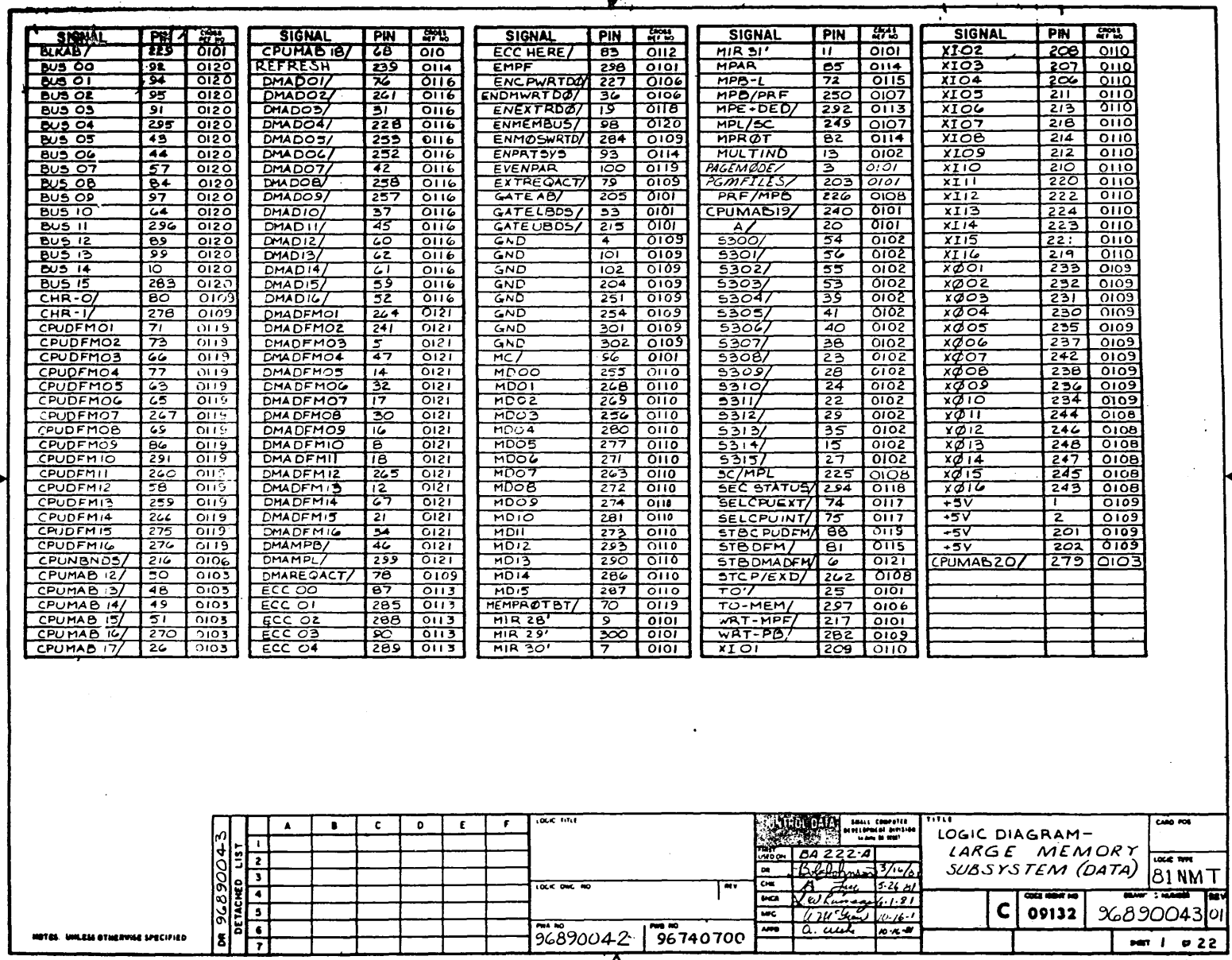


Figure 5-7. Large Memory Array Logic Diagram (Sheet 23 of 23)



SIGNAL	PIN	CROSS REF NO
BLKAB/	229	0101
BUS 00	98	0120
BUS 01	94	0120
BUS 02	95	0120
BUS 03	91	0120
BUS 04	295	0120
BUS 05	43	0120
BUS 06	44	0120
BUS 07	57	0120
BUS 08	84	0120
BUS 09	97	0120
BUS 10	64	0120
BUS 11	296	0120
BUS 12	89	0120
BUS 13	99	0120
BUS 14	10	0120
BUS 15	283	0120
CHR-0/	80	0109
CHR-1/	278	0109
CPUDFM01	71	0119
CPUDFM02	73	0119
CPUDFM03	66	0119
CPUDFM04	77	0119
CPUDFM05	63	0119
CPUDFM06	65	0119
CPUDFM07	267	0119
CPUDFM08	69	0119
CPUDFM09	86	0119
CPUDFM10	291	0119
CPUDFM11	260	0119
CPUDFM12	58	0119
CPUDFM13	259	0119
CPUDFM14	266	0119
CPUDFM15	275	0119
CPUDFM16	276	0119
CPUNBND5/	216	0106
CPUMAB 12/	50	0103
CPUMAB 13/	48	0103
CPUMAB 14/	49	0103
CPUMAB 15/	51	0103
CPUMAB 16/	270	0103
CPUMAB 17/	26	0103

SIGNAL	PIN	CROSS REF NO
CPUMAB 18/	68	010
REFRESH	239	0114
DMADO1/	76	0116
DMADO2/	261	0116
DMADO3/	31	0116
DMADO4/	228	0116
DMADO5/	255	0116
DMADO6/	252	0116
DMADO7/	42	0116
DMADO8/	258	0116
DMADO9/	257	0116
DMADI0/	37	0116
DMADI1/	45	0116
DMADI2/	60	0116
DMADI3/	62	0116
DMADI4/	61	0116
DMADI5/	59	0116
DMADI6/	52	0116
DMADFM01	244	0121
DMADFM02	241	0121
DMADFM03	5	0121
DMADFM04	47	0121
DMADFM05	14	0121
DMADFM06	32	0121
DMADFM07	17	0121
DMADFM08	30	0121
DMADFM09	16	0121
DMADFM10	8	0121
DMADFM11	18	0121
DMADFM12	265	0121
DMADFM13	12	0121
DMADFM14	67	0121
DMADFM15	21	0121
DMADFM16	54	0121
DMAMPB/	46	0121
DMAMPL/	299	0121
DMAREQACT/	78	0109
ECC 00	87	0113
ECC 01	285	0113
ECC 02	288	0113
ECC 03	90	0113
ECC 04	289	0113

SIGNAL	PIN	CROSS REF NO
ECC HERE/	85	0112
EMPF	298	0101
ENCPWRTDØ/	227	0106
ENDMWRTDØ/	36	0106
ENEXTRDØ/	19	0118
ENMEMBUS/	98	0120
ENMØSWRTD/	284	0109
ENPRTSYS	93	0114
EVENPAR	100	0119
EXTREQACT/	79	0109
GATEAB/	205	0101
GATELBDS/	53	0101
GATEUBDS/	215	0101
GND	4	0109
GND	101	0109
GND	102	0109
GND	204	0109
GND	251	0109
GND	254	0109
GND	301	0109
GND	302	0109
MC/	96	0101
MDO0	255	0110
MDO1	268	0110
MDO2	269	0110
MDO3	256	0110
MDO4	280	0110
MDO5	277	0110
MDO6	271	0110
MDO7	263	0110
MDO8	272	0110
MDO9	274	0118
MDI0	281	0110
MDI1	273	0110
MDI2	293	0110
MDI3	290	0110
MDI4	286	0110
MDI5	287	0110
MEMPRØTBT/	70	0119
MIR 28'	9	0101
MIR 29'	300	0101
MIR 30'	7	0101

SIGNAL	PIN	CROSS REF NO
MIR 31'	11	0101
MPAR	85	0114
MPB-L	72	0115
MPB/PRF	250	0107
MPE+DED/	292	0113
MPL/SC	249	0107
MPRØT	82	0114
MULTIND	13	0102
PAGEMØDE/	3	0101
PGMFILES/	203	0101
PRF/MPB	226	0108
CPUMAB19/	240	0101
A/	20	0101
S300/	54	0102
S301/	56	0102
S302/	55	0102
S303/	53	0102
S304/	39	0102
S305/	41	0102
S306/	40	0102
S307/	38	0102
S308/	23	0102
S309/	28	0102
S310/	24	0102
S311/	22	0102
S312/	29	0102
S313/	35	0102
S314/	15	0102
S315/	27	0102
SC/MPL	225	0108
SEC STATUS/	294	0118
SELCPUEXT/	74	0117
SELCPUINT/	75	0117
STBCPUDFM/	88	0119
STBDFM/	81	0115
STBDMADFM/	6	0121
STCP/EXD/	262	0108
TO'/	25	0101
TO-MEM/	297	0106
WRT-MPF/	217	0101
WRT-PB/	282	0109
XI01	209	0110

SIGNAL	PIN	CROSS REF NO
XI02	208	0110
XI03	207	0110
XI04	206	0110
XI05	211	0110
XI06	213	0110
XI07	218	0110
XI08	214	0110
XI09	212	0110
XI10	210	0110
XI11	220	0110
XI12	222	0110
XI13	224	0110
XI14	223	0110
XI15	22:	0110
XI16	219	0110
XØØ1	233	0109
XØ02	232	0109
XØ03	231	0109
XØ04	230	0109
XØ05	235	0109
XØ06	237	0109
XØ07	242	0109
XØ08	238	0109
XØ09	236	0109
XØ10	234	0109
XØ11	244	0108
XØ12	246	0108
XØ13	248	0108
XØ14	247	0108
XØ15	245	0108
XØ16	243	0108
+5V	1	0109
+5V	2	0109
+5V	201	0109
+5V	202	0109
CPUMAB20/	279	0103

NOTES: UNLESS OTHERWISE SPECIFIED

DR 96890043 DETACHED LIST

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						
7						

LOGIC TITLE

LOGIC DWG NO | REV

PWA NO 96890042 | PWB NO 96740700

XEROX DATA SYSTEMS — SMALL COMPUTER DEVELOPMENT DIVISION

DESIGNED	BA 222-A	
DR		3/16/81
CHK		5-26-81
ENGR		6-1-81
MFG		10-16-81
APPD		

TITLE: LOGIC DIAGRAM- LARGE MEMORY SUBSYSTEM (DATA)

CARD POS

LOGIC TYPE 81NMT

CODE IDENT NO	DRAWING NUMBER	REV
C 09132	96890043	01

SHEET 1 OF 22

Figure 5-8. LMS Data Logic Diagram (Sheet 1 of 23)

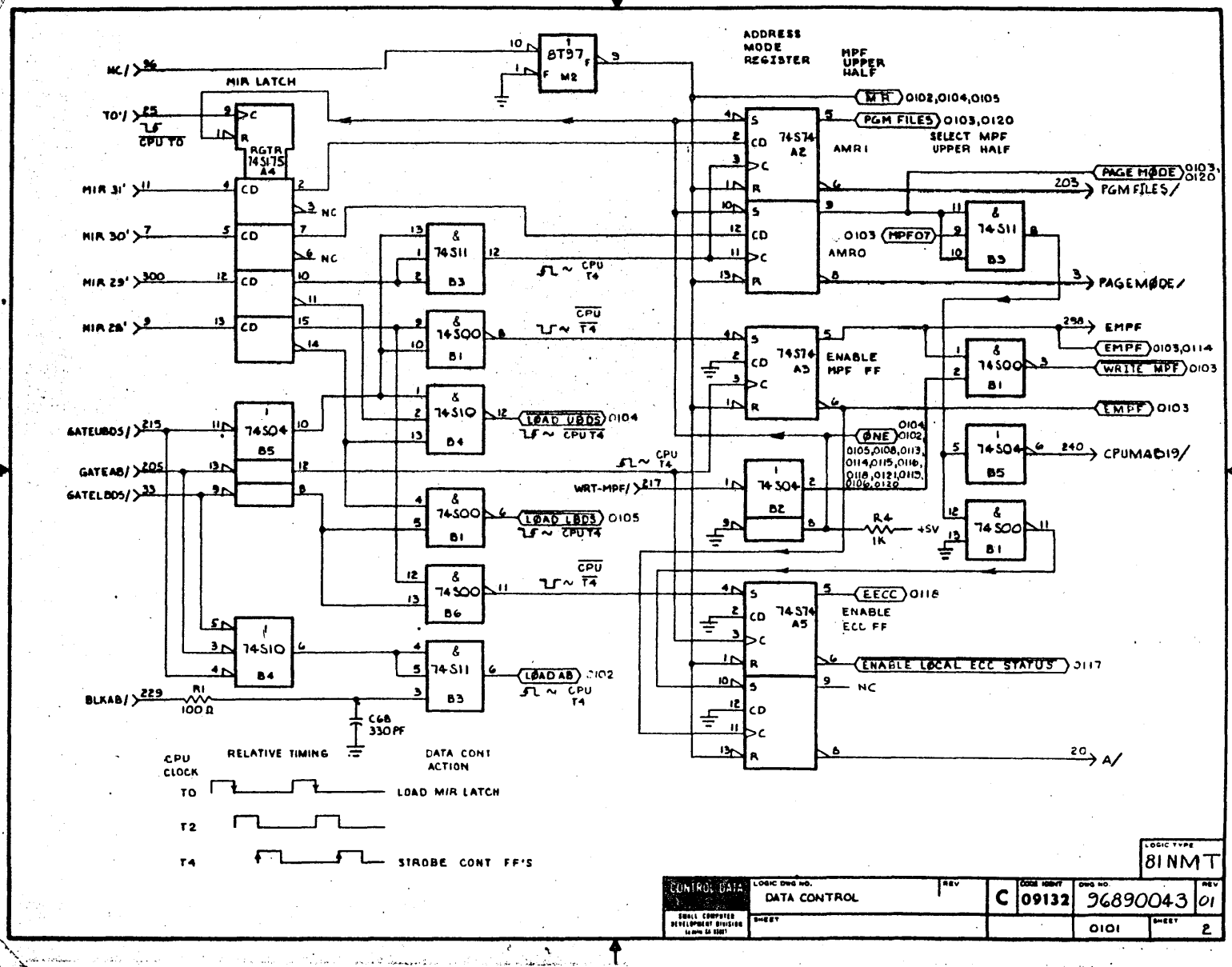


Figure 5-8. LMS Data Logic Diagram (Sheet 2 of 23)

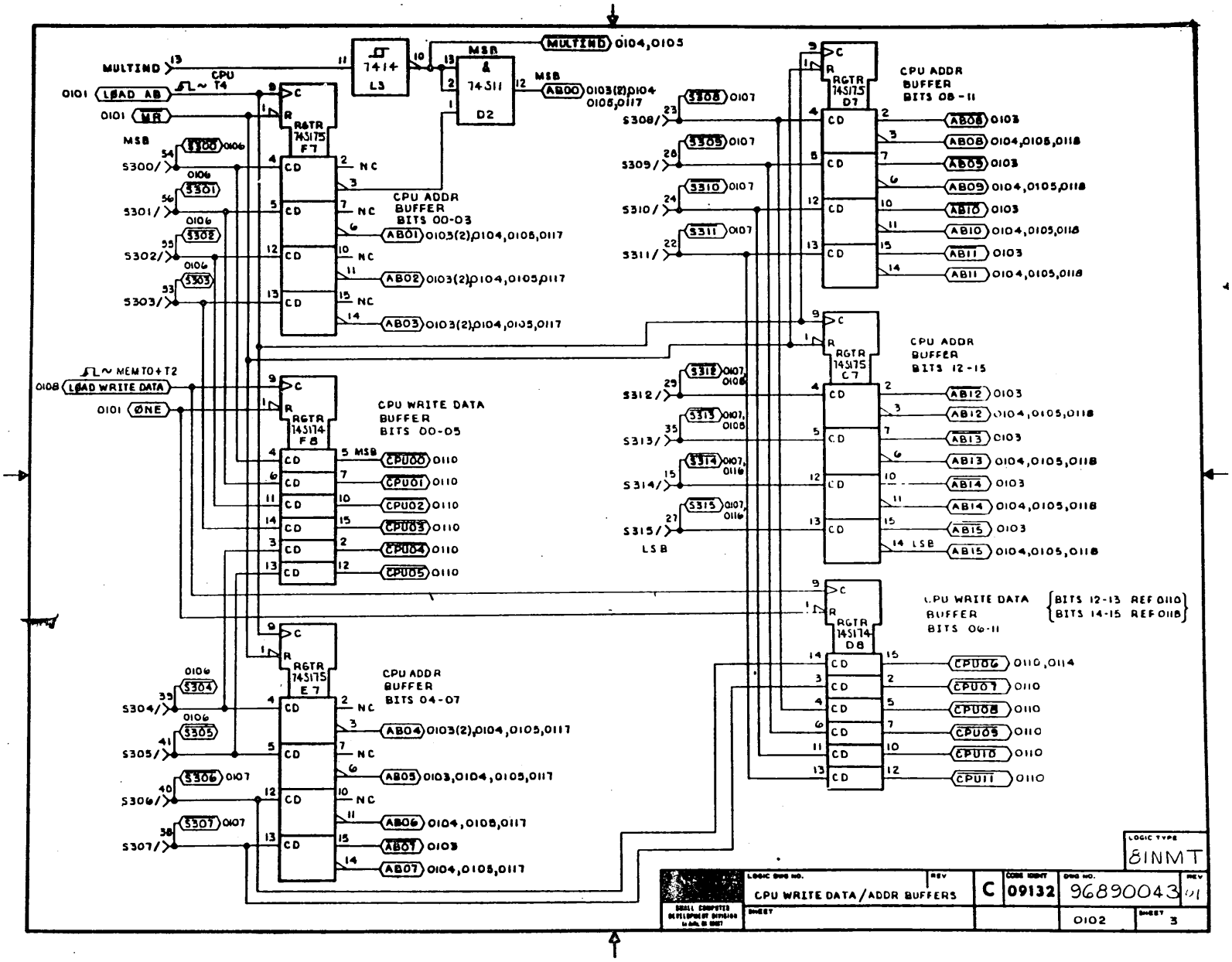


Figure 5-8. LMS Data Logic Diagram (Sheet 3 of 23)

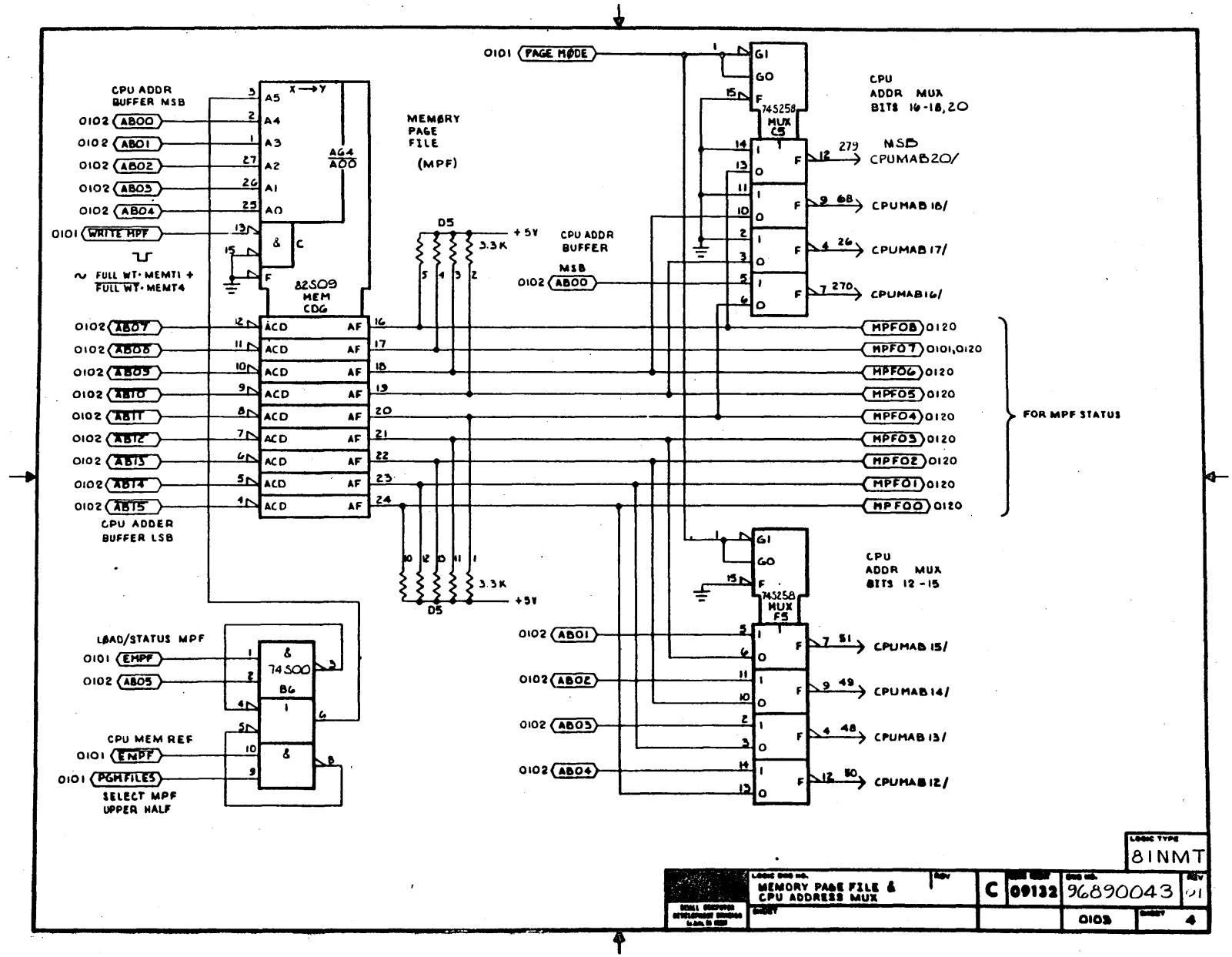


Figure 5-8. LMS Data Logic Diagram (Sheet 4 of 23)

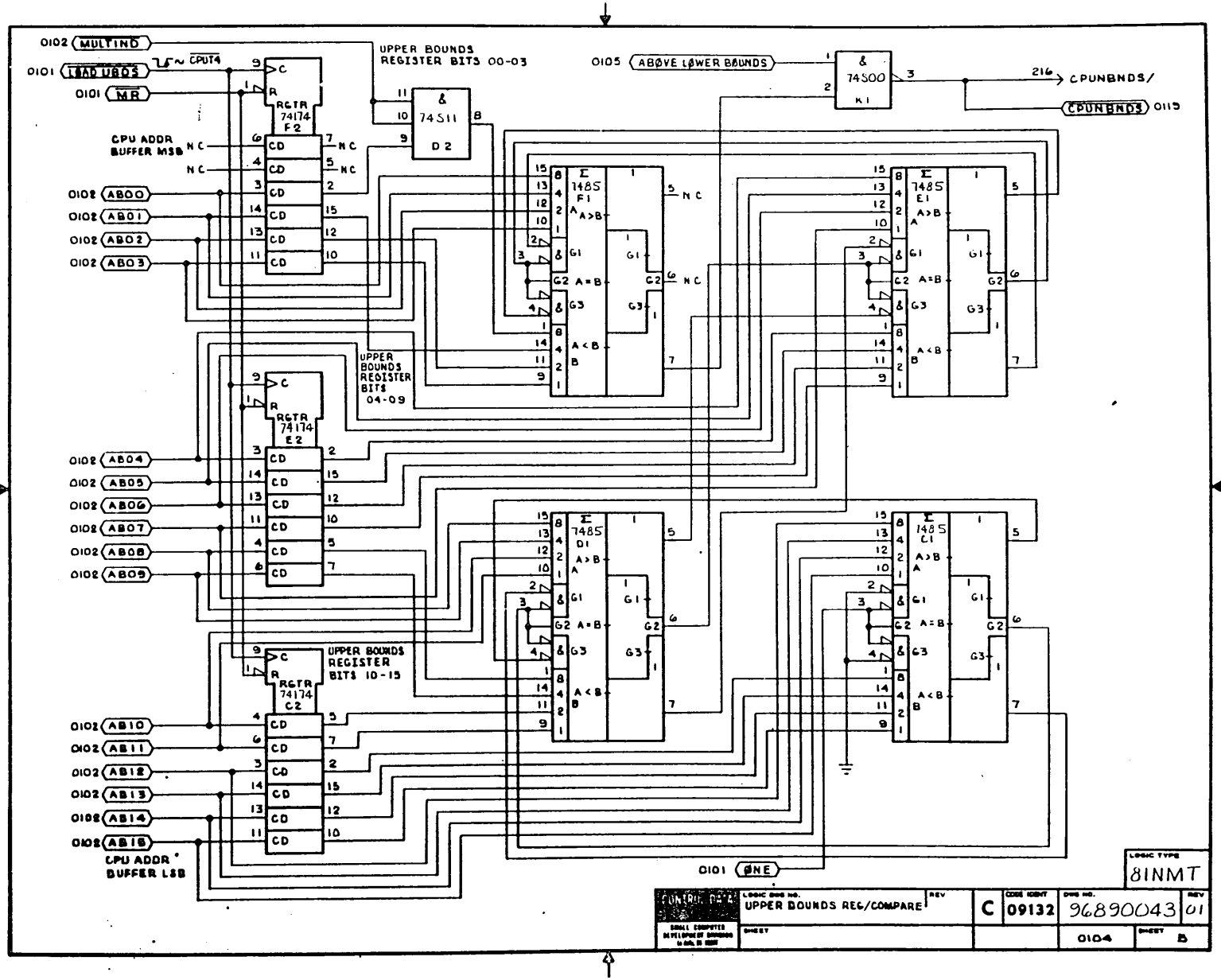


Figure 5-8. LMS Data Logic Diagram (Sheet 5 of 23)

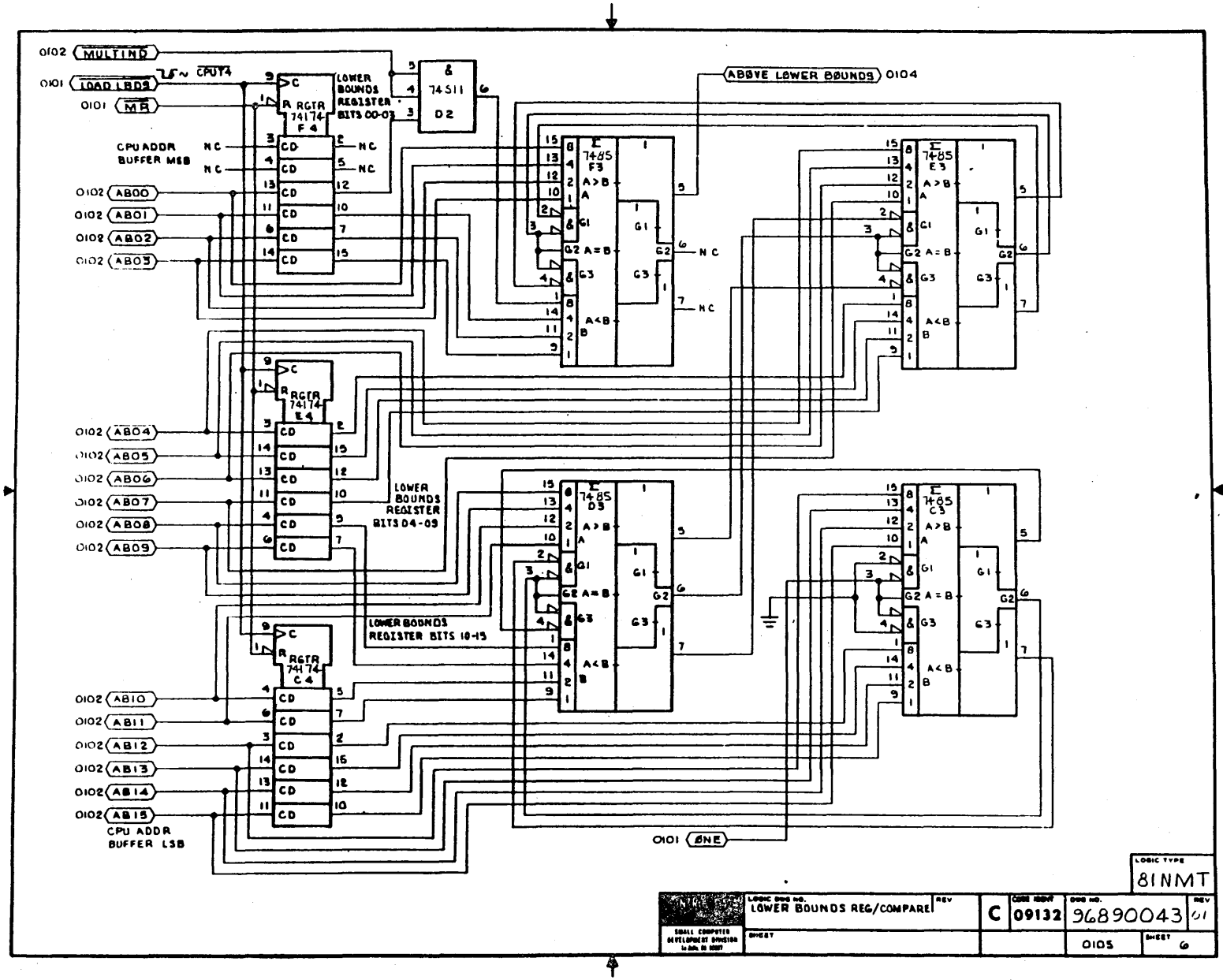


Figure 5-8. LMS Data Logic Diagram (Sheet 6 of 23)

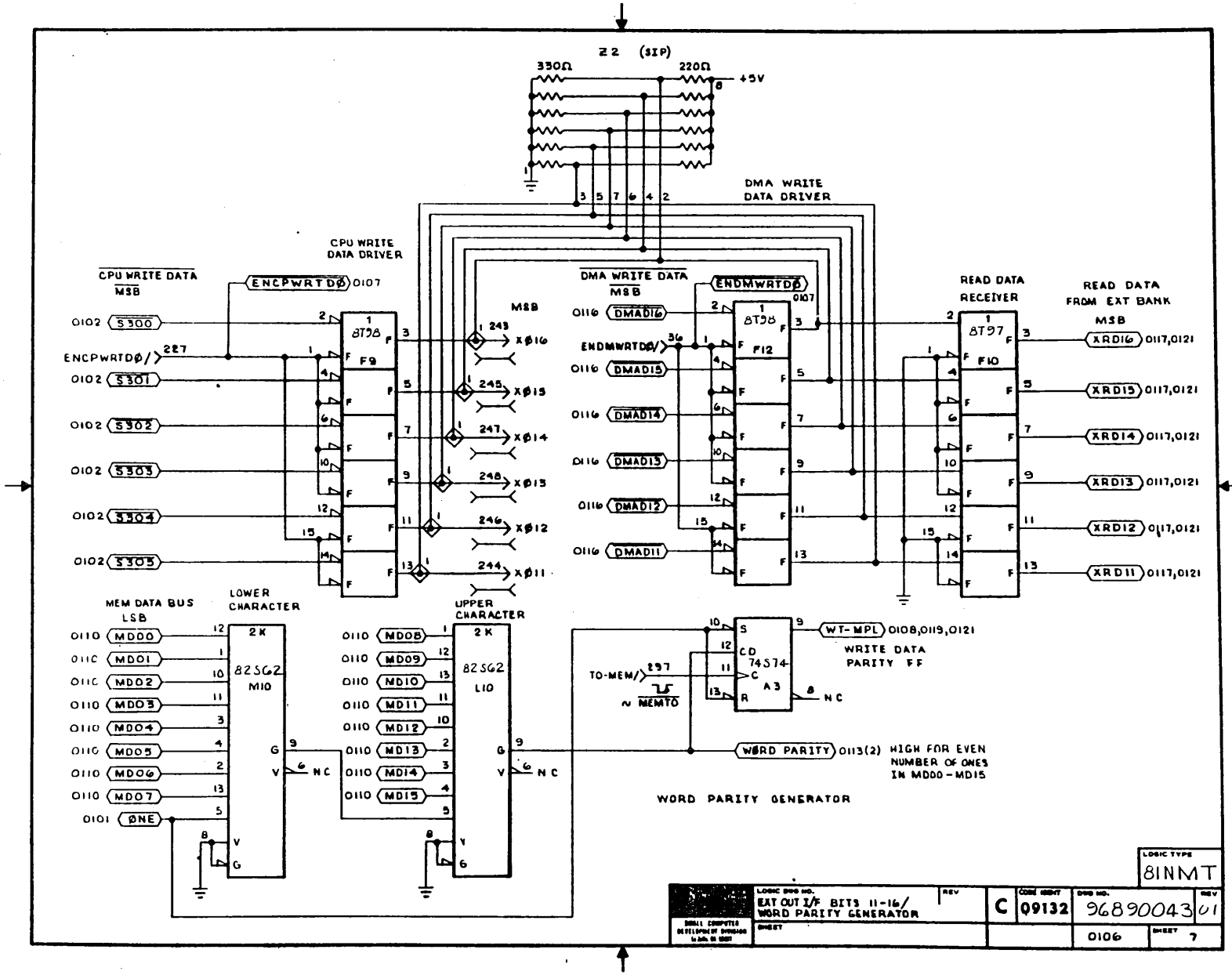


Figure 5-8. LMS Data Logic Diagram (Sheet 7 of 23)

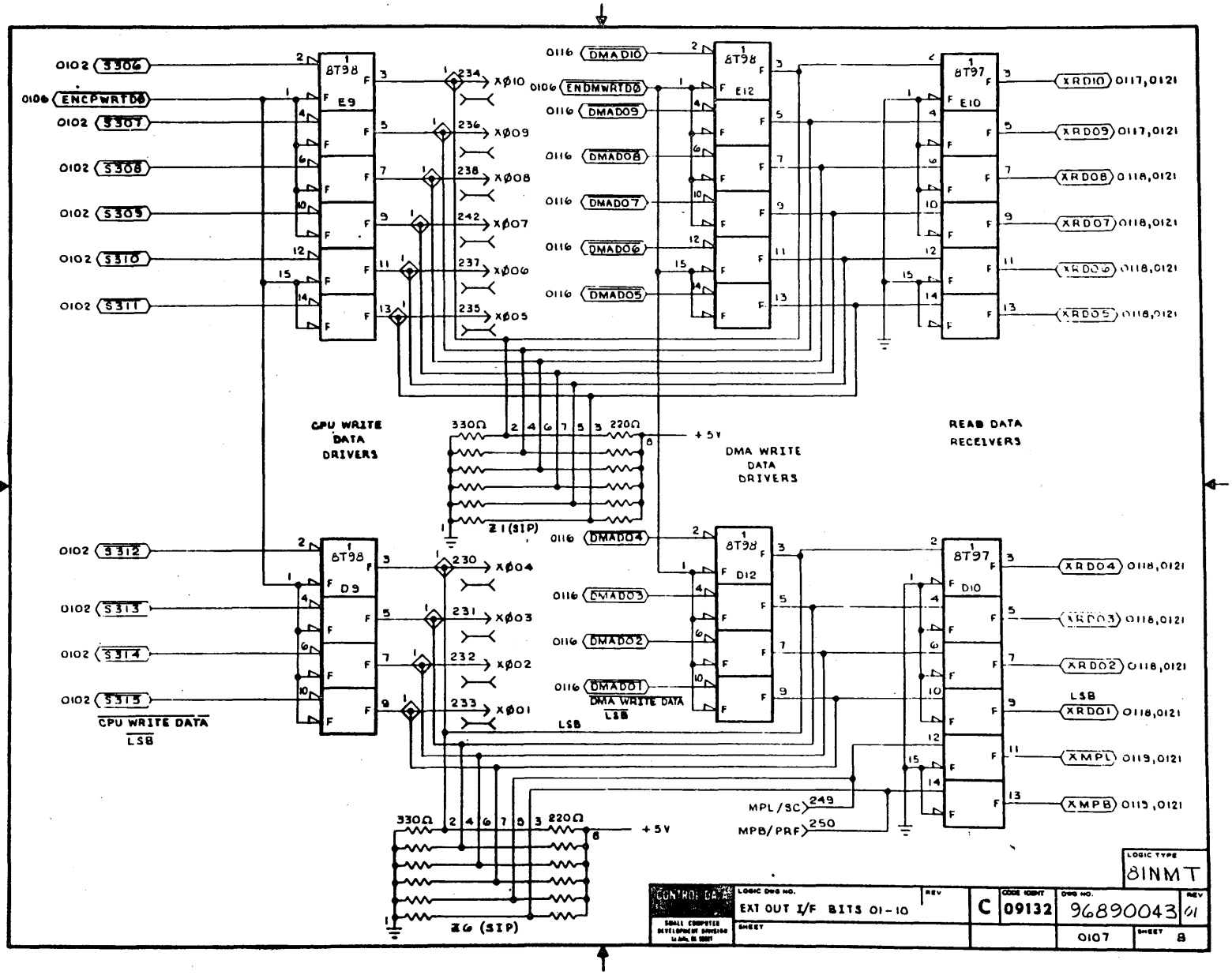


Figure 5-8. LMS Data Logic Diagram (Sheet 8 of 23)

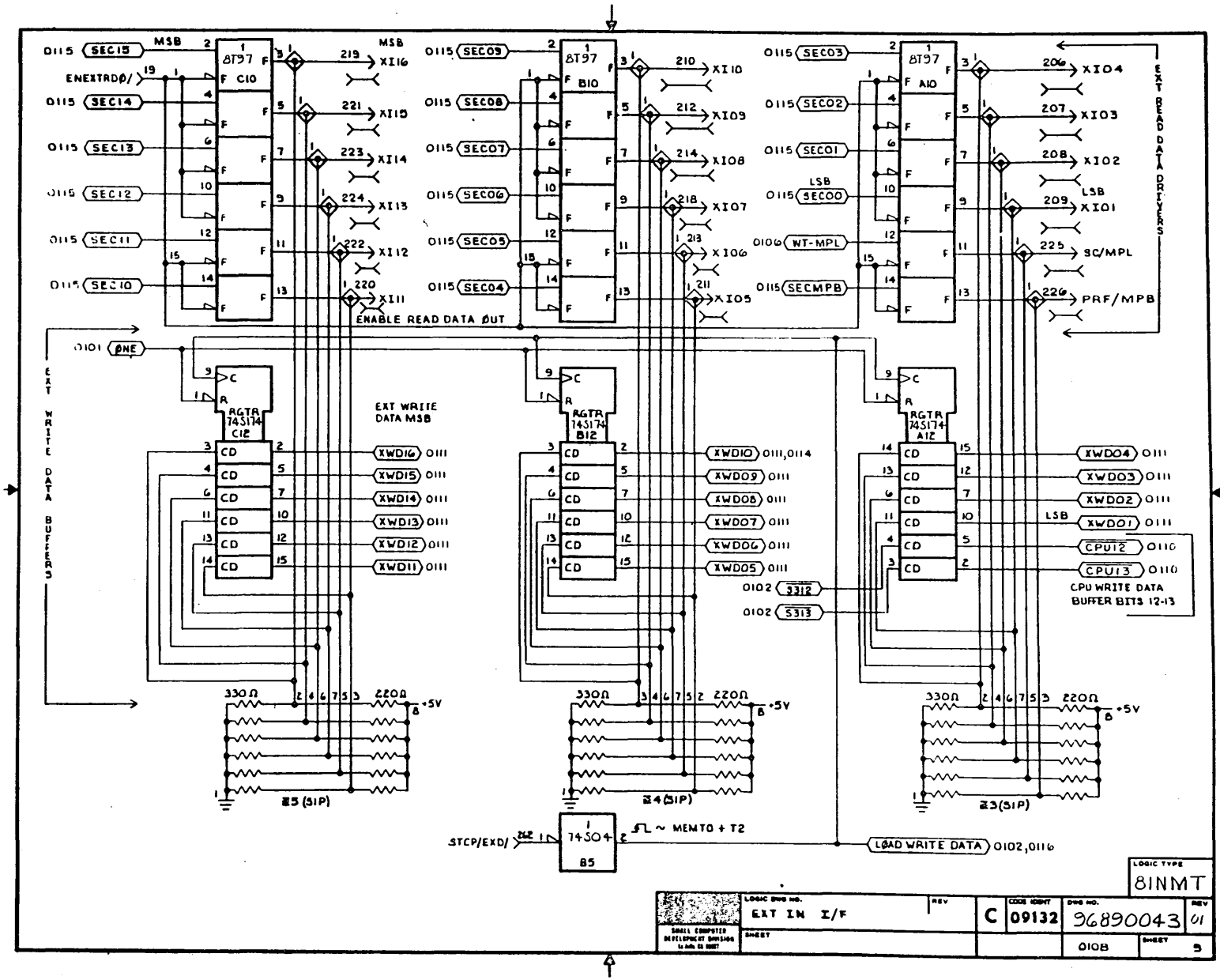


Figure 5-8. LMS Data Logic Diagram (Sheet 9 of 23)

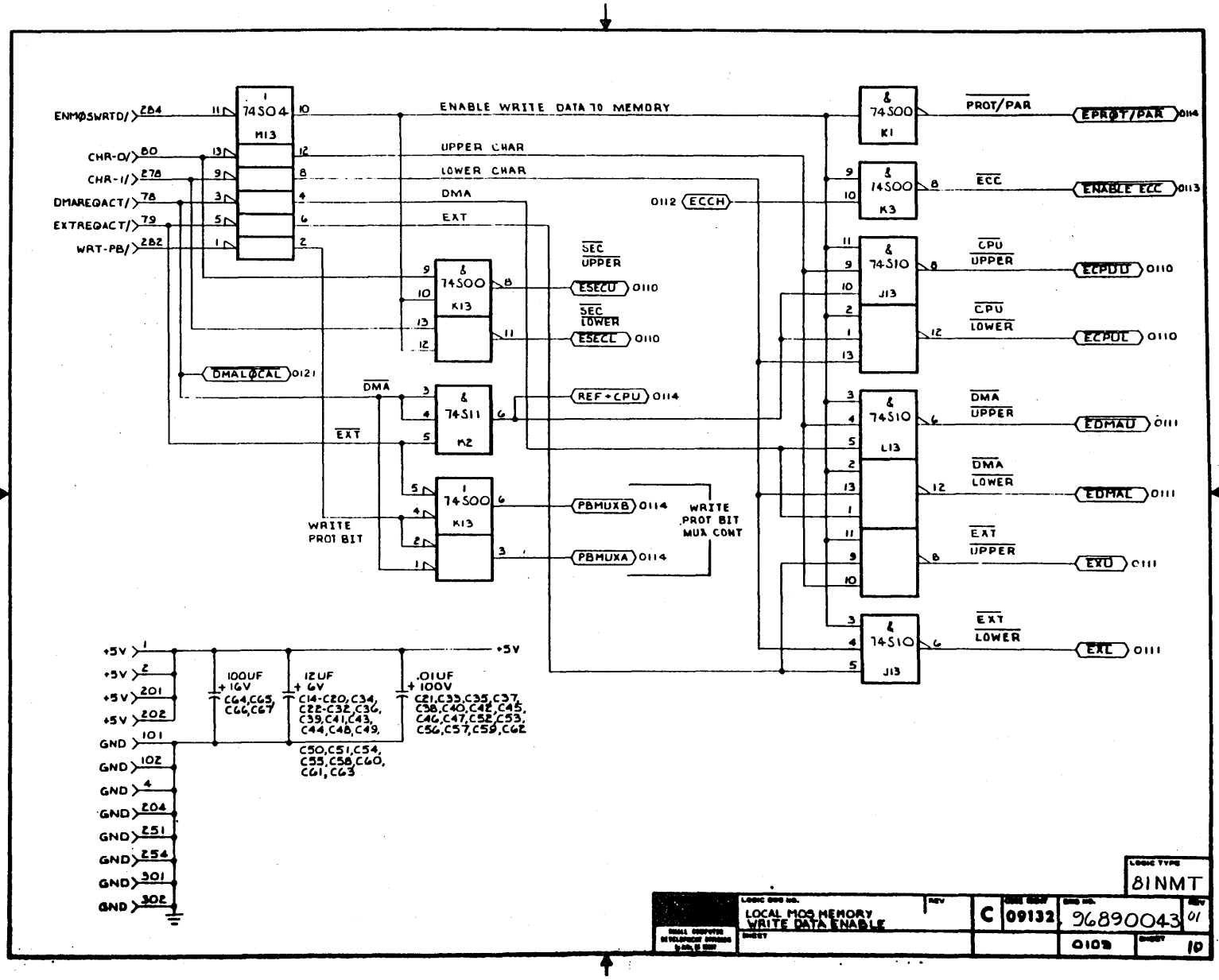


Figure 5-8. LMS Data Logic Diagram (Sheet 10 of 23)

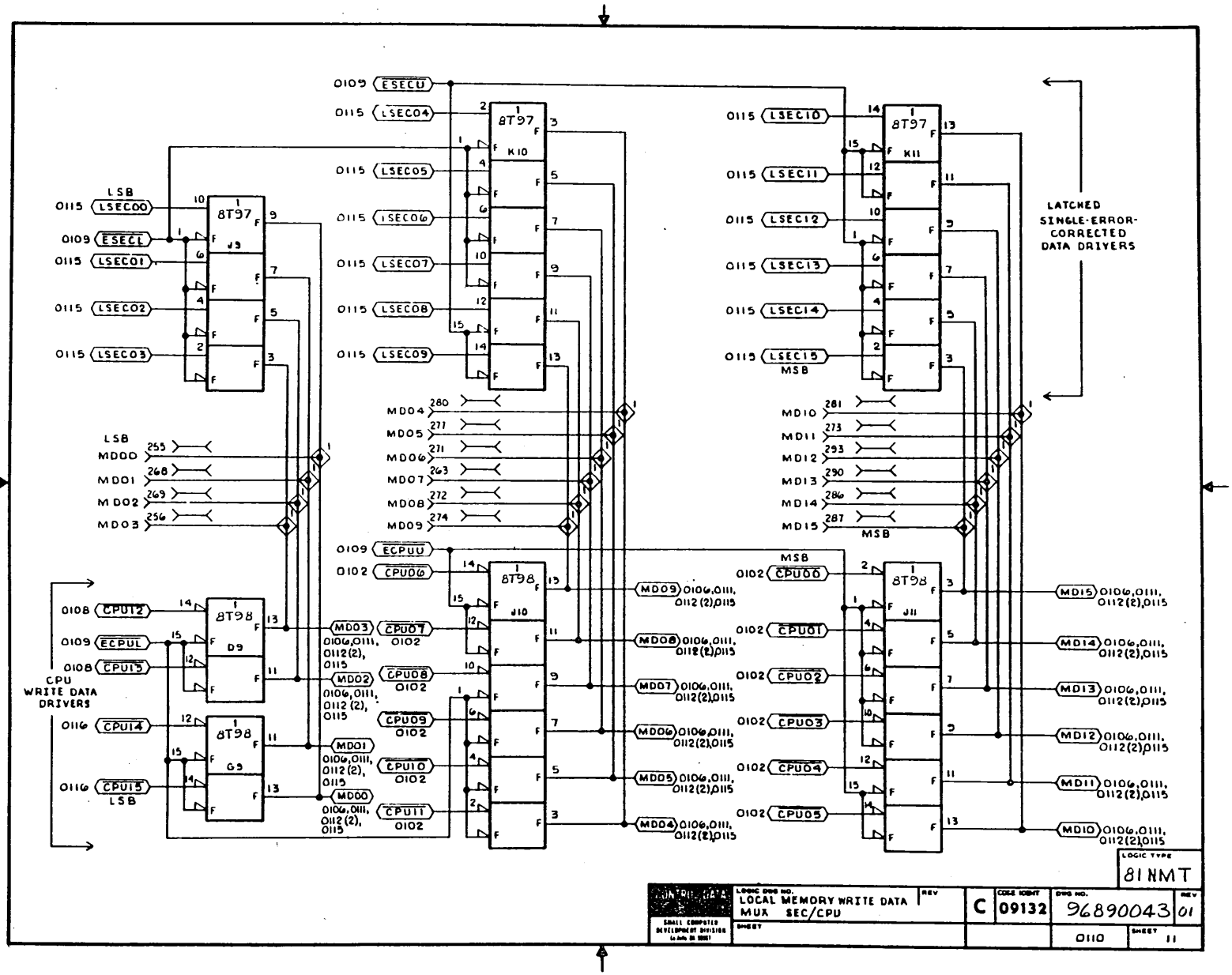


Figure 5-8. LMS Data Logic Diagram (Sheet 11 of 23)

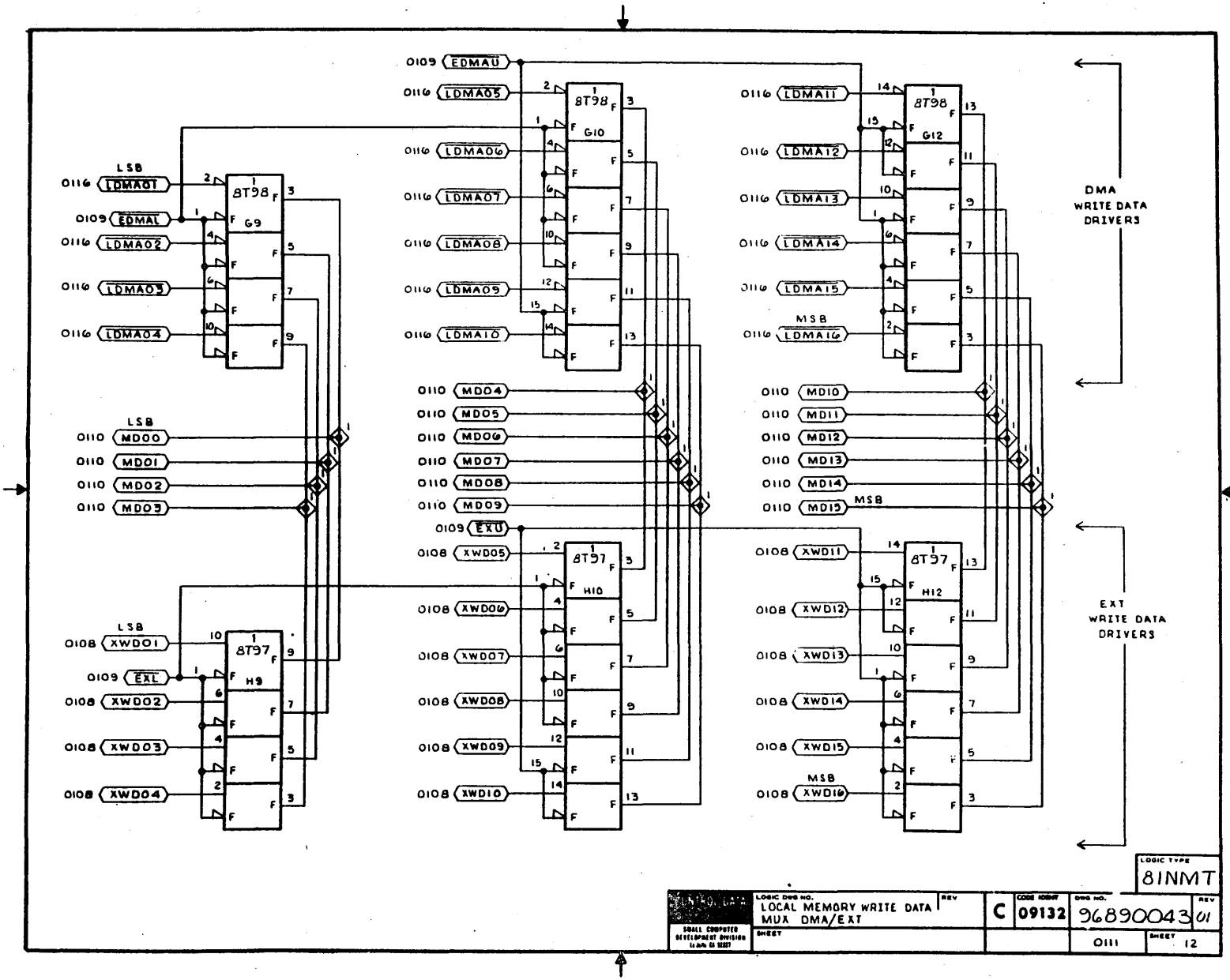
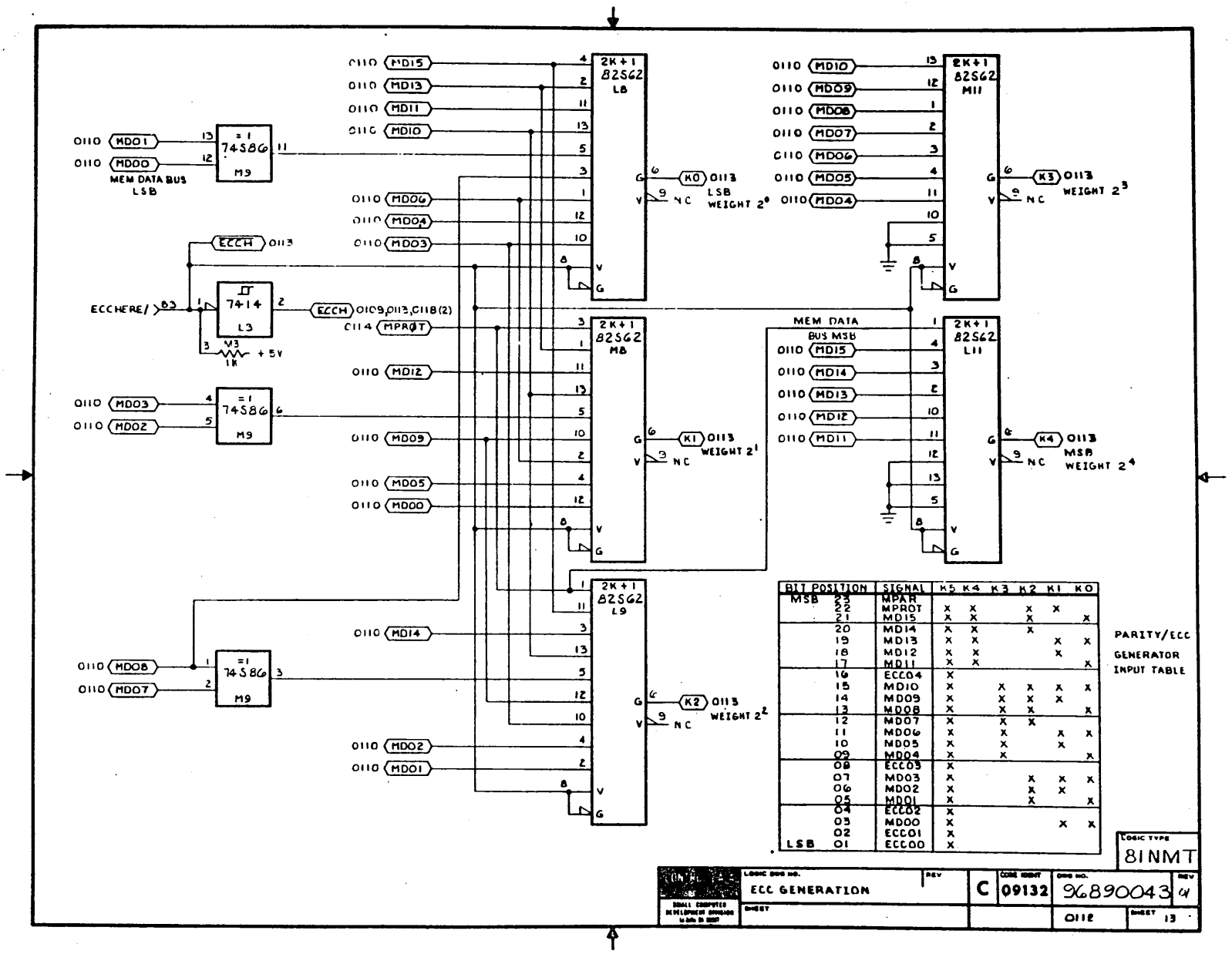


Figure 5-8. LMS Data Logic Diagram (Sheet 12 of 23)



BIT POSITION	SIGNAL	K5	K4	K3	K2	K1	K0
MSB 23	MPAR						
22	MPROT	X	X		X	X	
21	MD15	X	X		X		X
20	MD14	X	X		X		
19	MD13	X	X			X	X
18	MD12	X	X			X	
17	MD11	X	X				X
16	ECC04	X					
15	MD10	X		X	X	X	X
14	MD09	X		X	X	X	
13	MD08	X		X	X		X
12	MD07	X		X	X		
11	MD06	X		X		X	X
10	MD05	X		X		X	
09	MD04	X		X			X
08	ECC03	X					
07	MD03	X			X	X	X
06	MD02	X			X	X	
05	MD01	X			X		X
04	ECC02	X					
03	MD00	X				X	X
02	ECC01	X					
LSB 01	ECC00	X					

Figure 5-8. LMS Data Logic Diagram (Sheet 13 of 23)

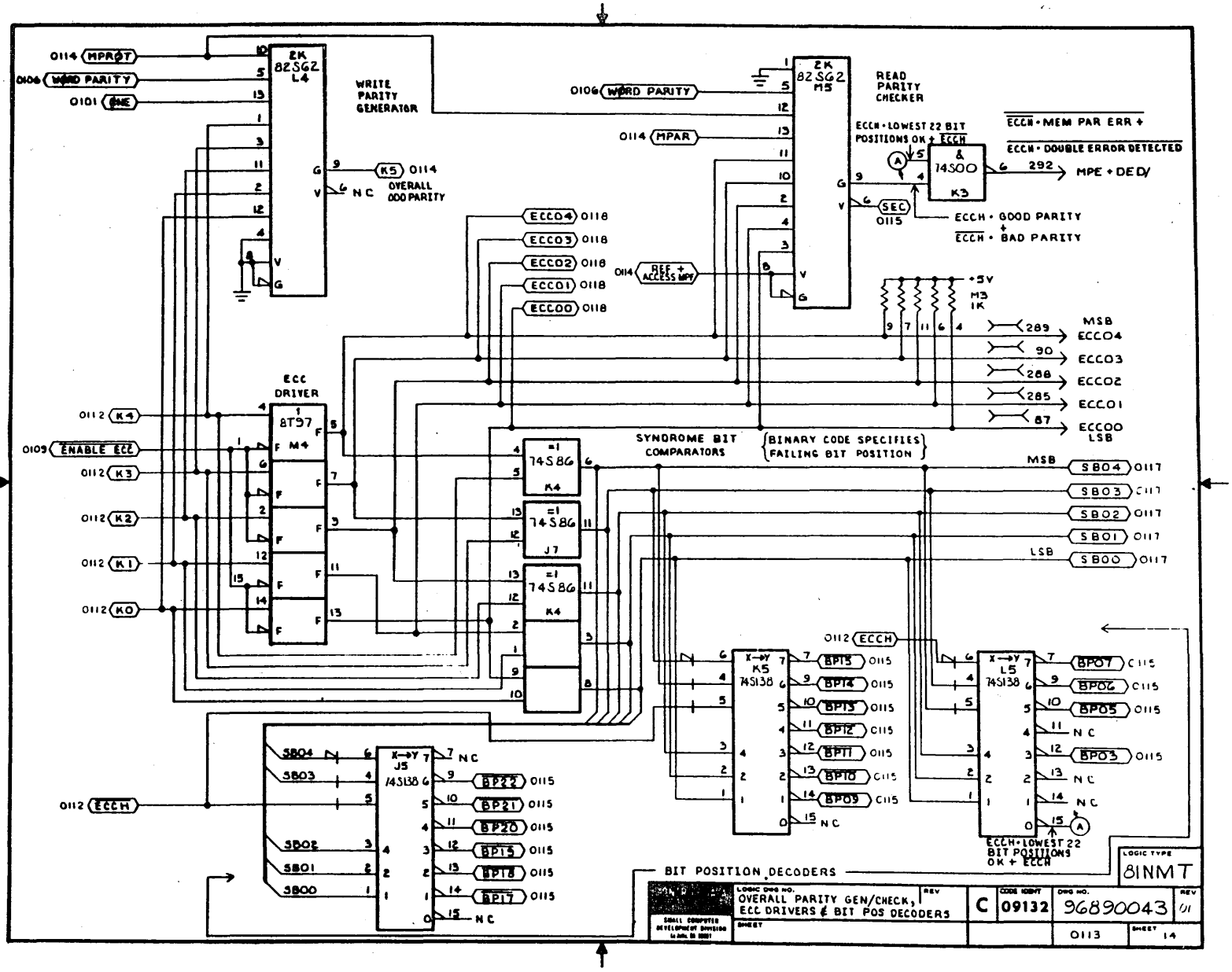


Figure 5-8. LMS Data Logic Diagram (Sheet 14 of 23)

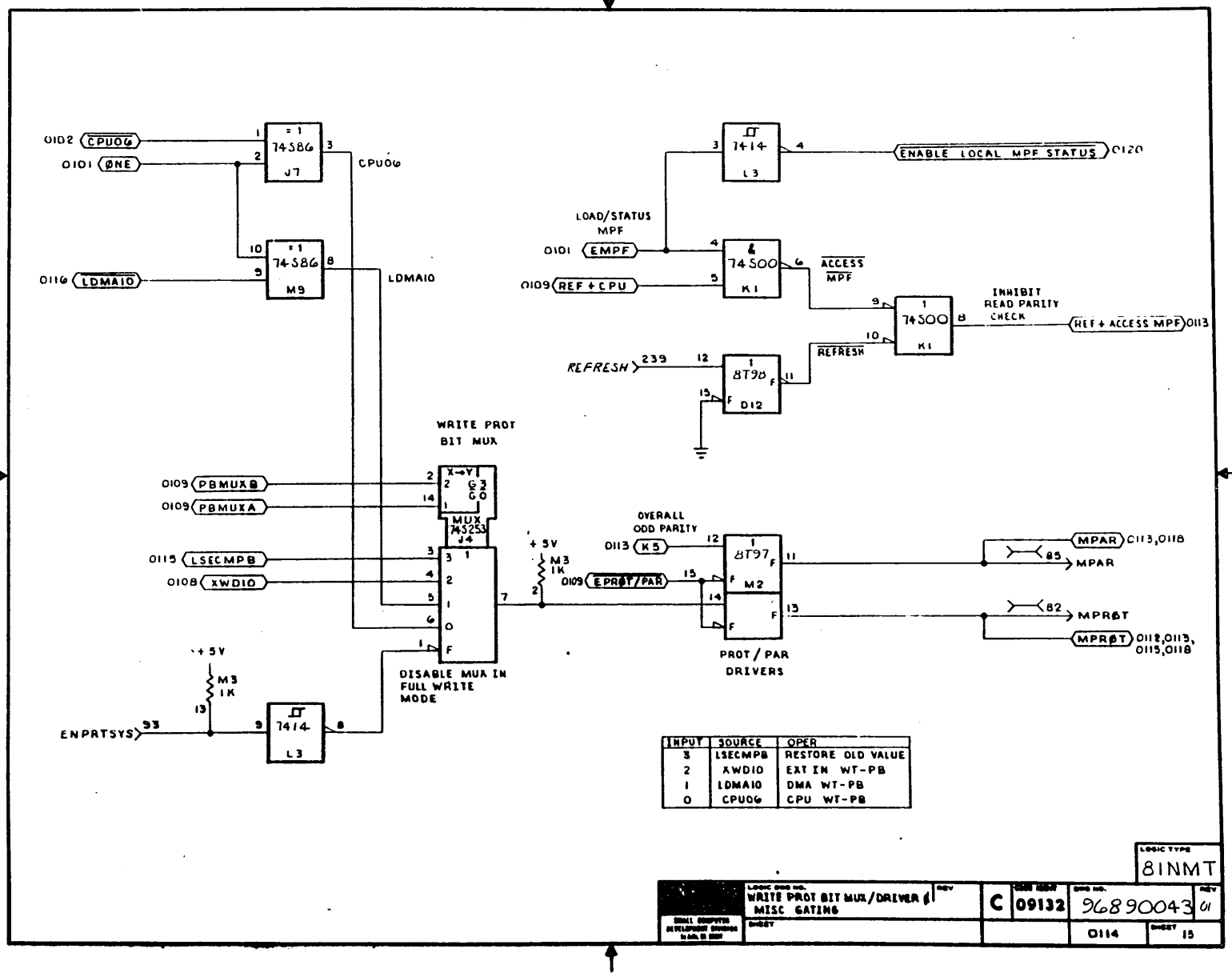


Figure 5-8. LMS Data Logic Diagram (Sheet 15 of 23)

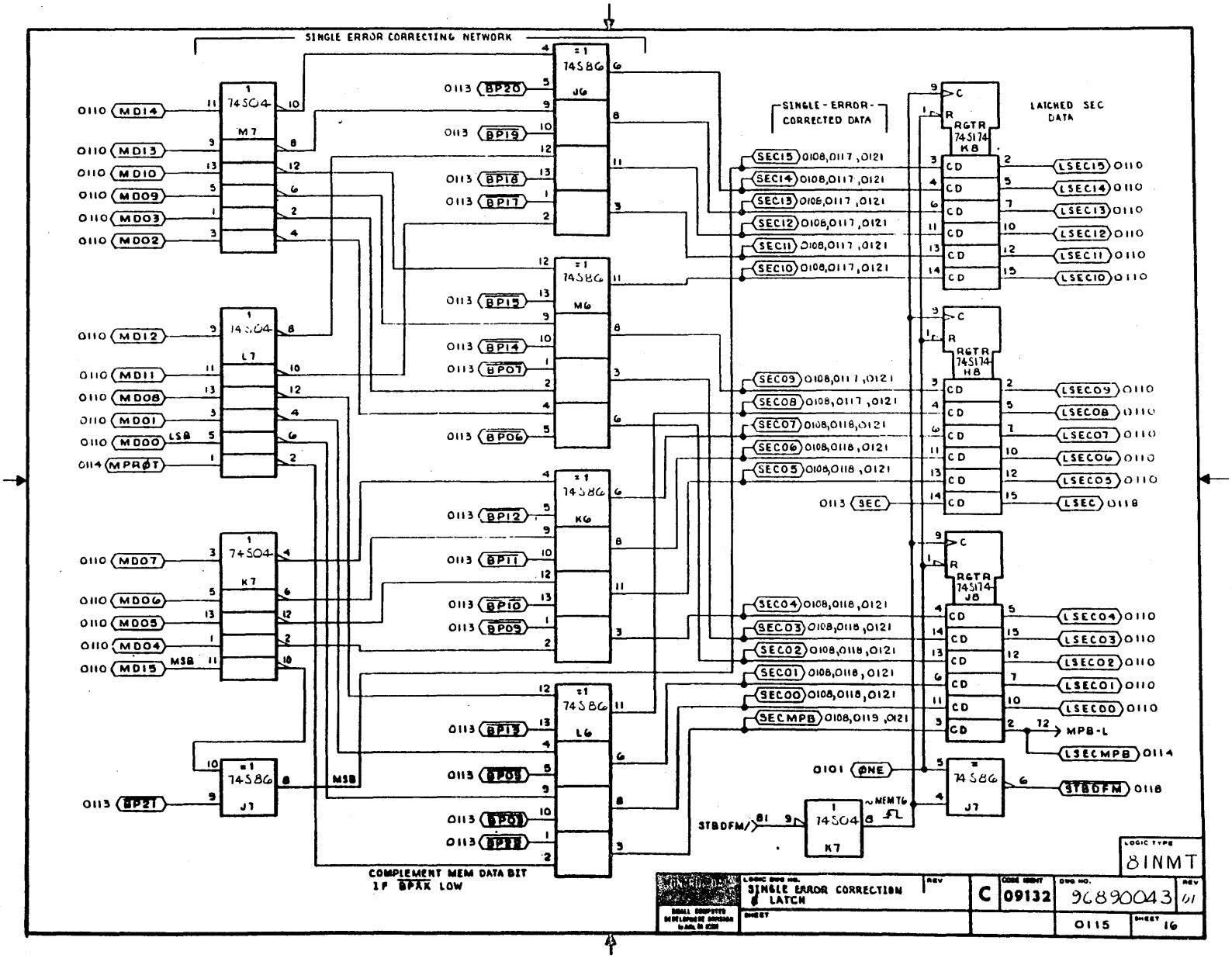


Figure 5-8. LMS Data Logic Diagram (Sheet 16 of 23)

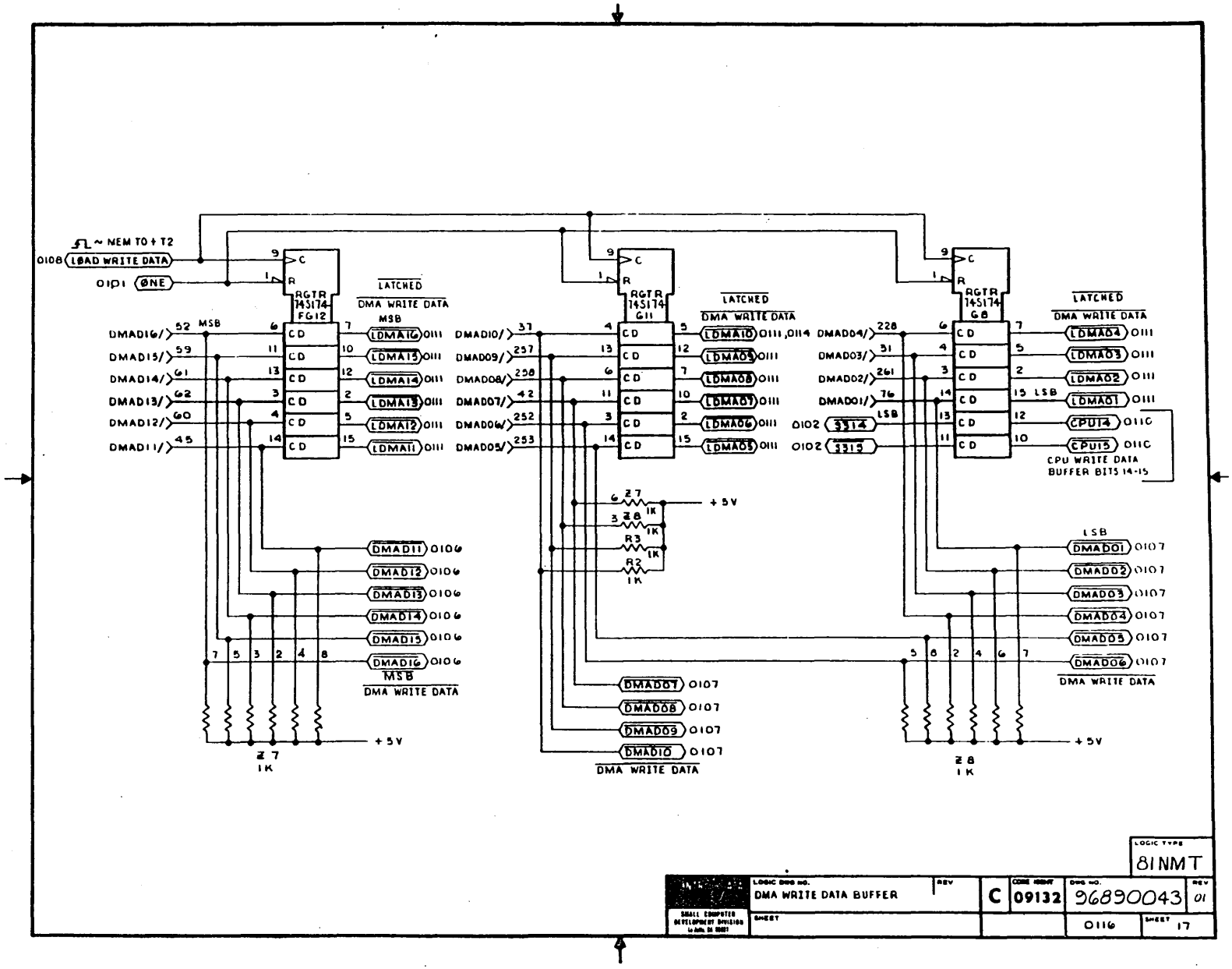


Figure 5-8. LMS Data Logic Diagram (Sheet 17 of 23)

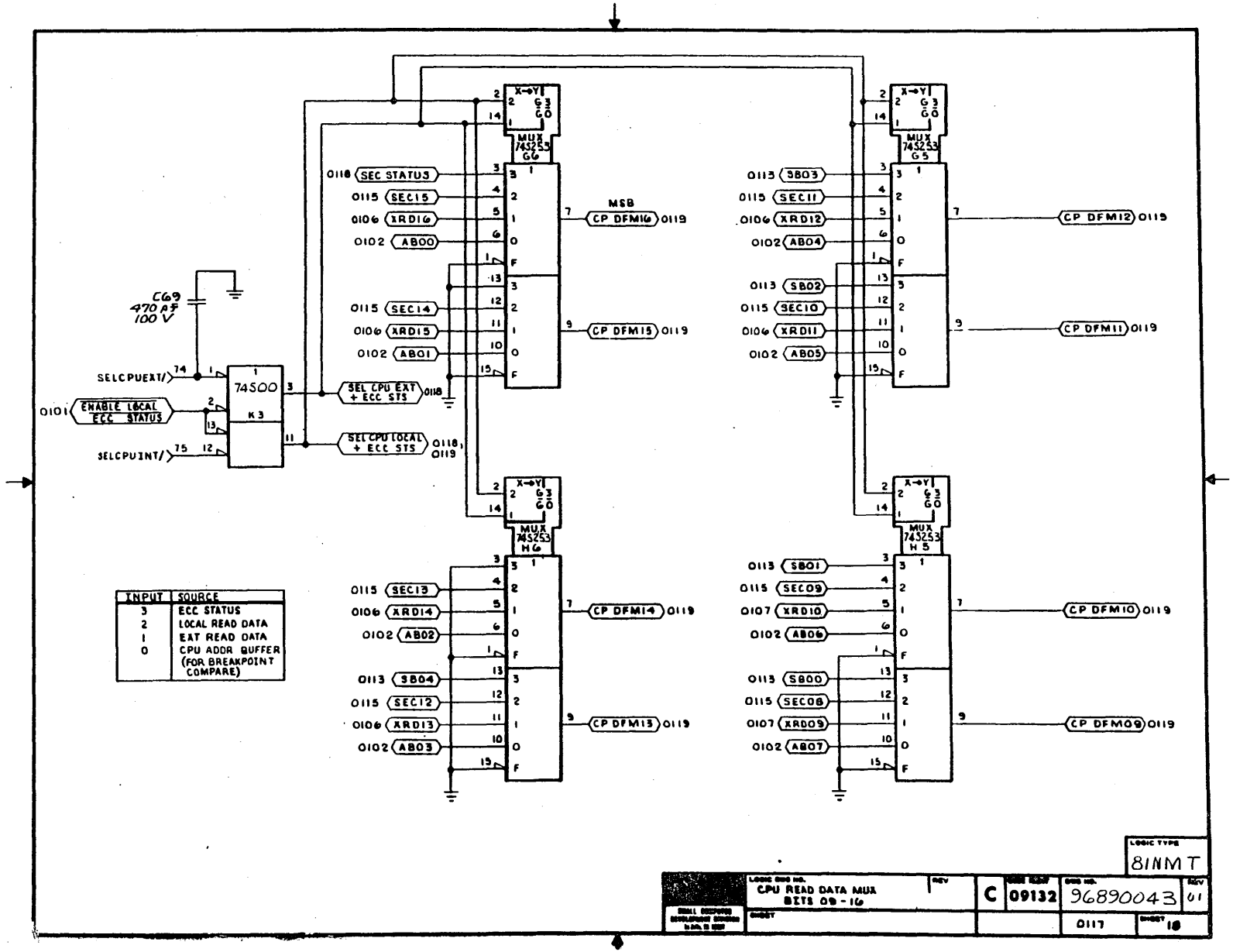


Figure 5-8. LMS Data Logic Diagram (Sheet 18 of 23)

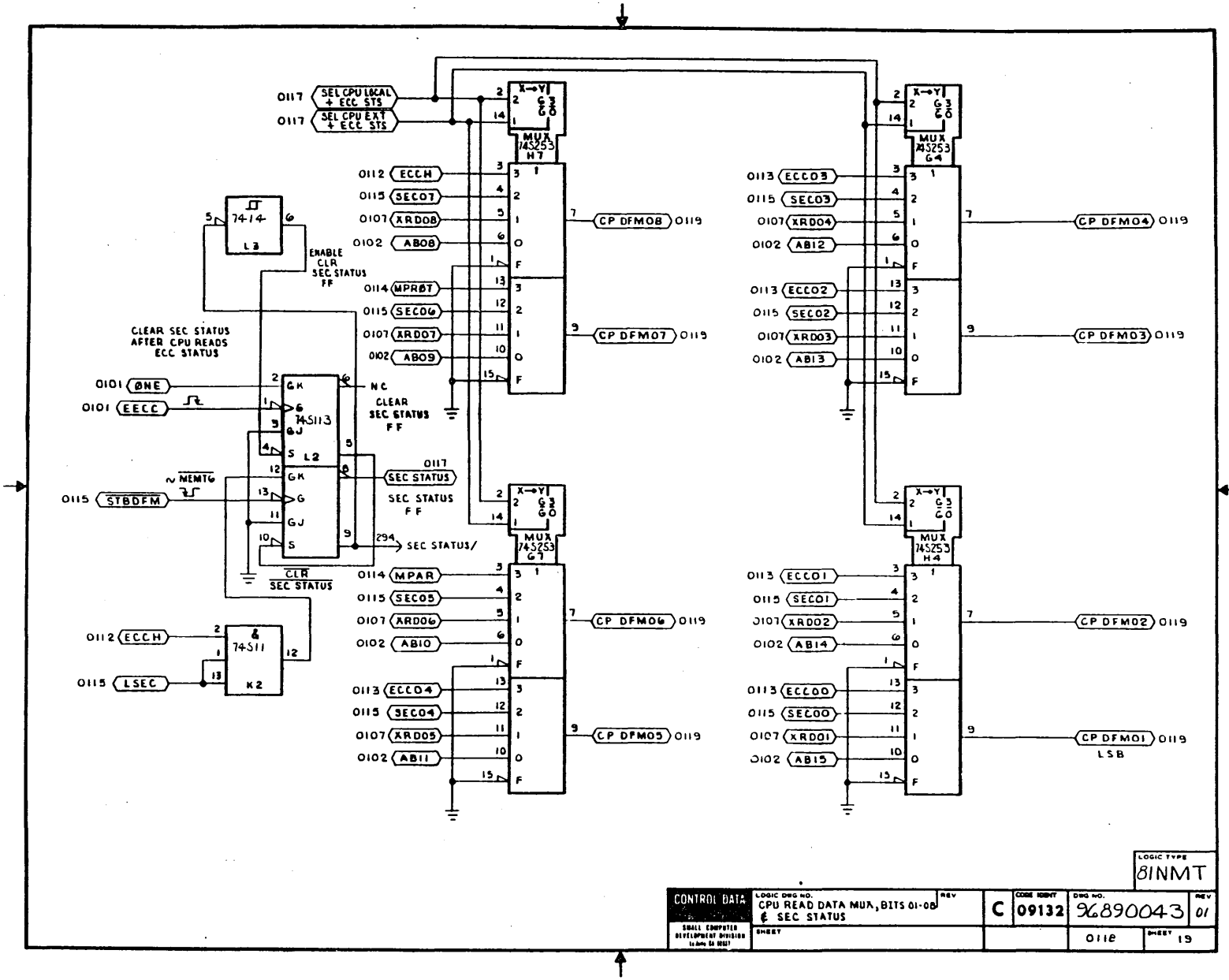


Figure 5-8. LMS Data Logic Diagram (Sheet 19 of 23)

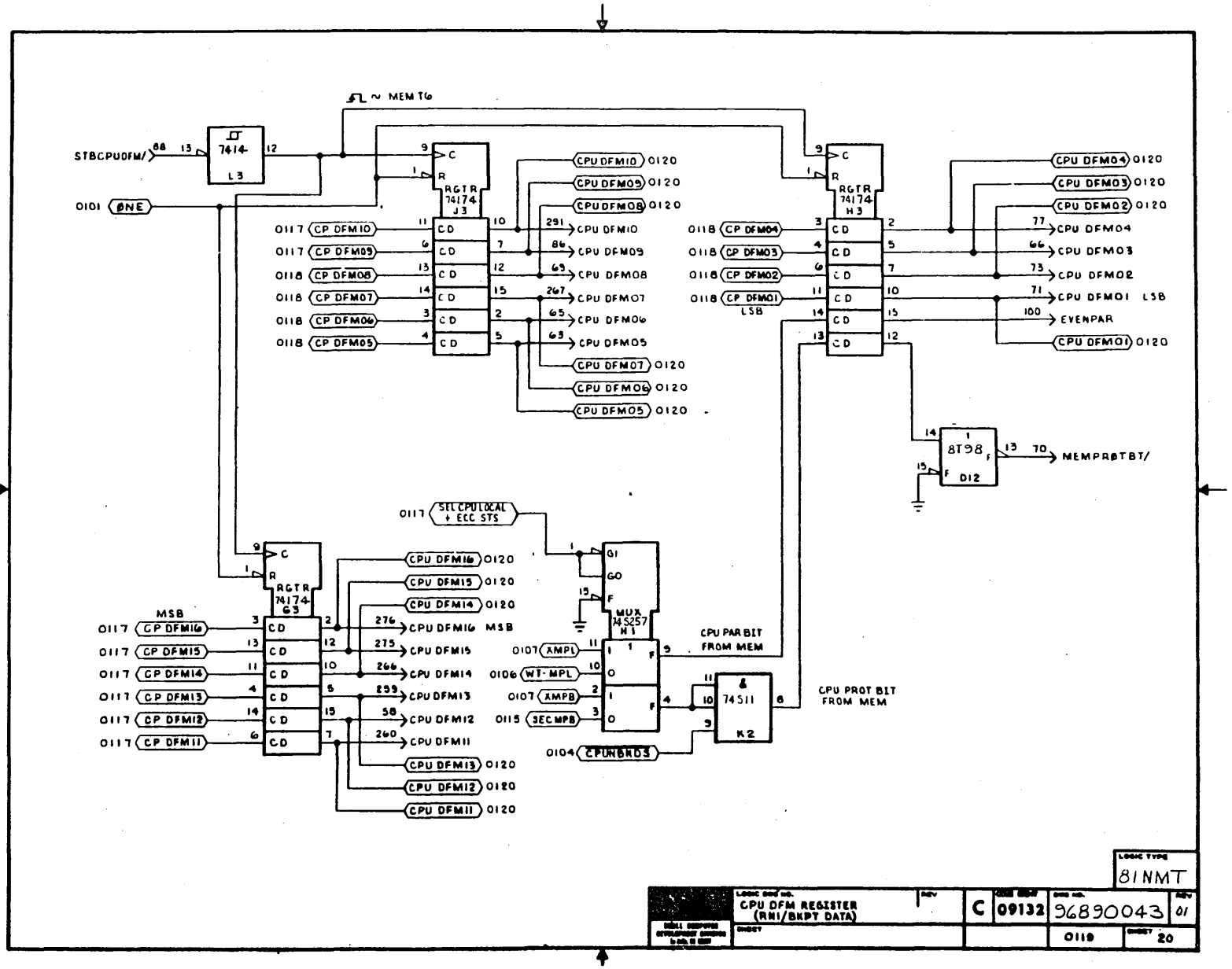


Figure 5-8. LMS Data Logic Diagram (Sheet 20 of 23)

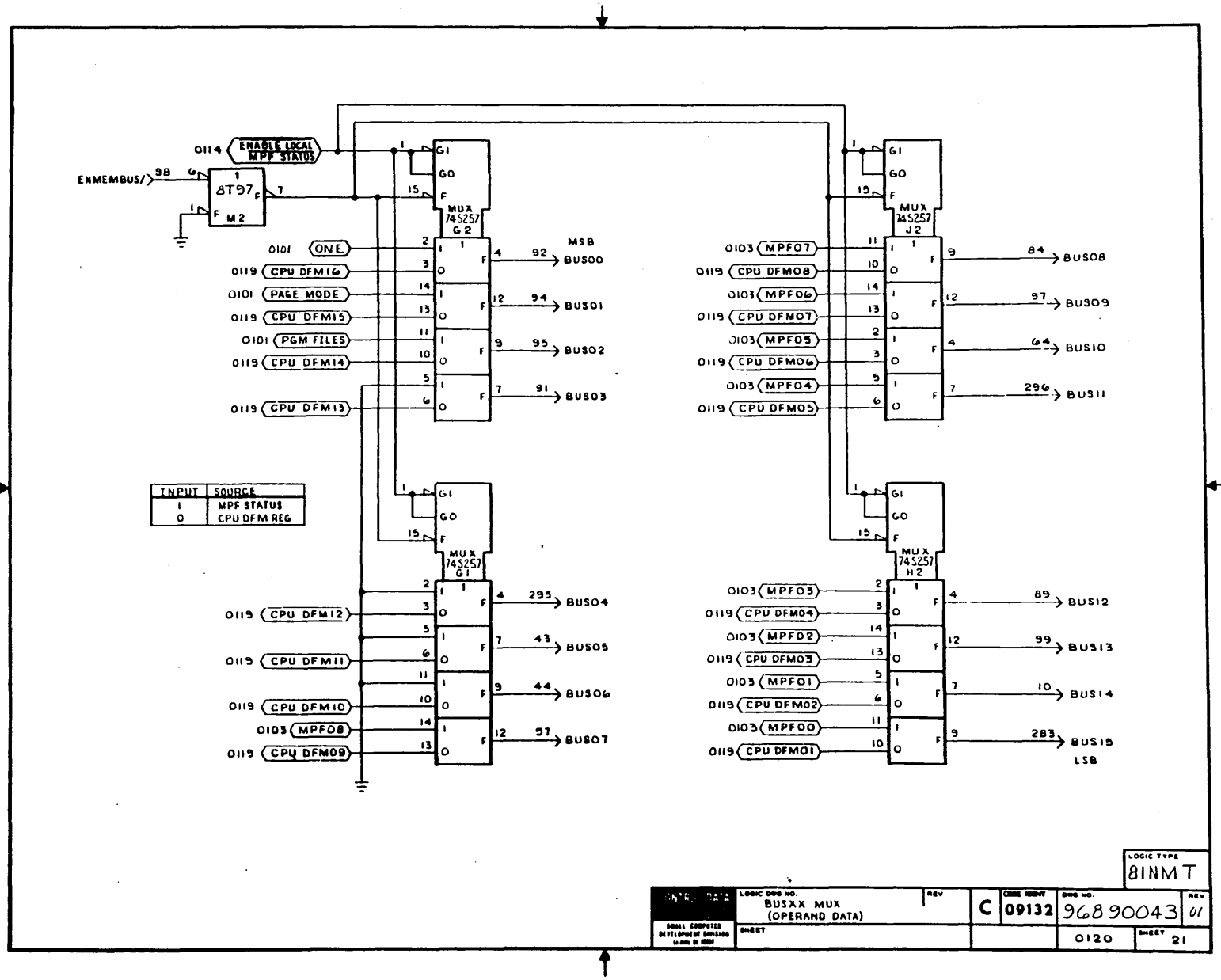


Figure 5-8. LMS Data Logic Diagram (Sheet 21 of 23)

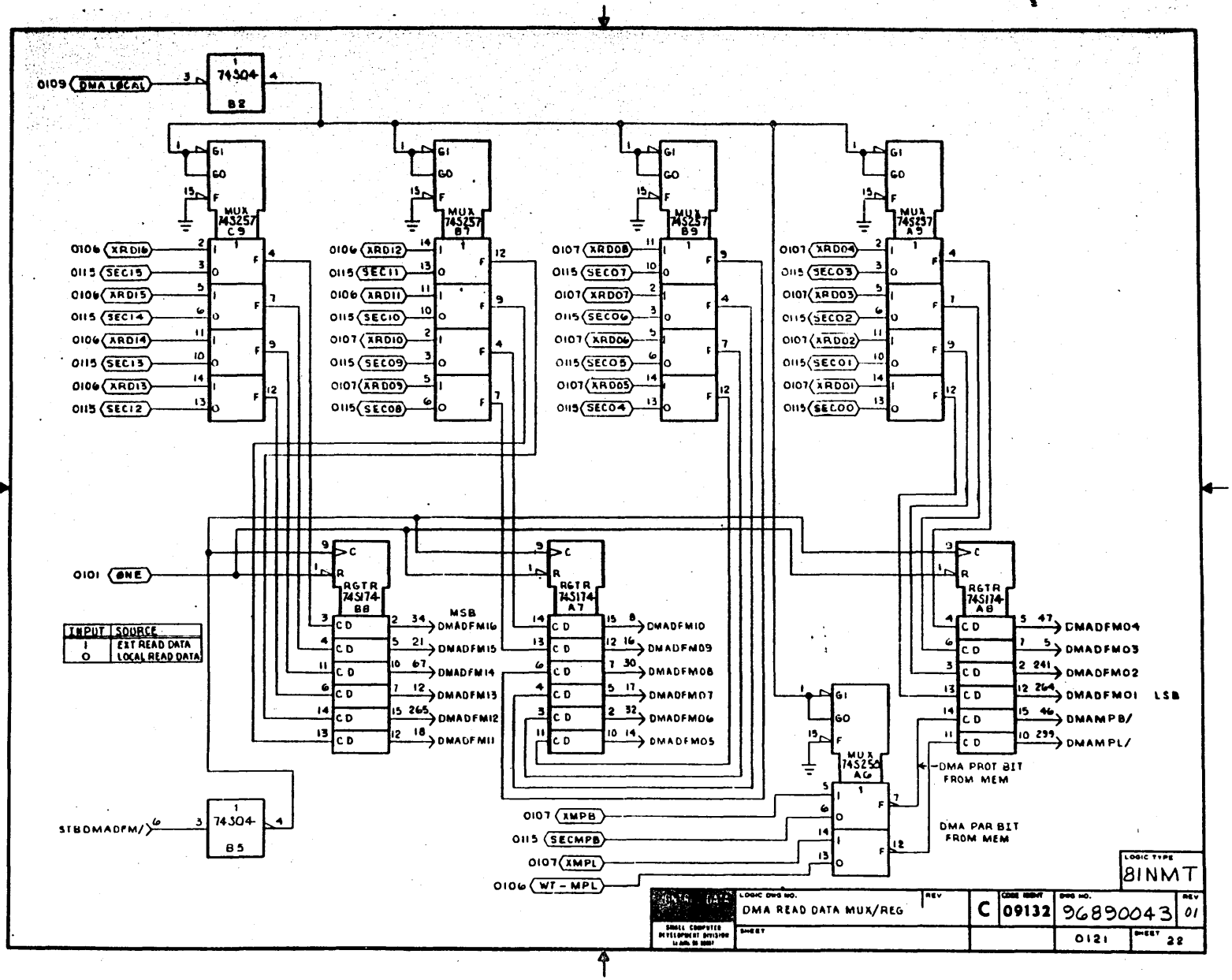


Figure 5-8. LMS Data Logic Diagram (Sheet 22 of 23)

DWN	Bob Johnson	5/16/81	CONTROL DATA CORPORATION DATA SYSTEMS LA JOLLA DIVISION LA JOLLA CA. 92037	TITLE LOGIC- LARGE MEMORY SUBSYSTEM (DATA) 8INMT	PREFIX DR	DOCUMENT NO. 96890043	REV. 01
CHKD	B. Fagan	5/26/81					
ENG	[illegible]	6/4/81					
MFG	[illegible]	10-16-81		FIRST USED ON BA 222-A		SHEET 1 OF 1	
APPR	A. Wiebe	10-16-81	CODE IDENT 09132				

SHEET REVISION STATUS

REVISION RECORD

REV	ECO	DESCRIPTION	DRFT	DATE	CHKD	APP
01	DS19333	CL. B. PRE-RELEASED	[signature]	10-16-81	[signature]	AW

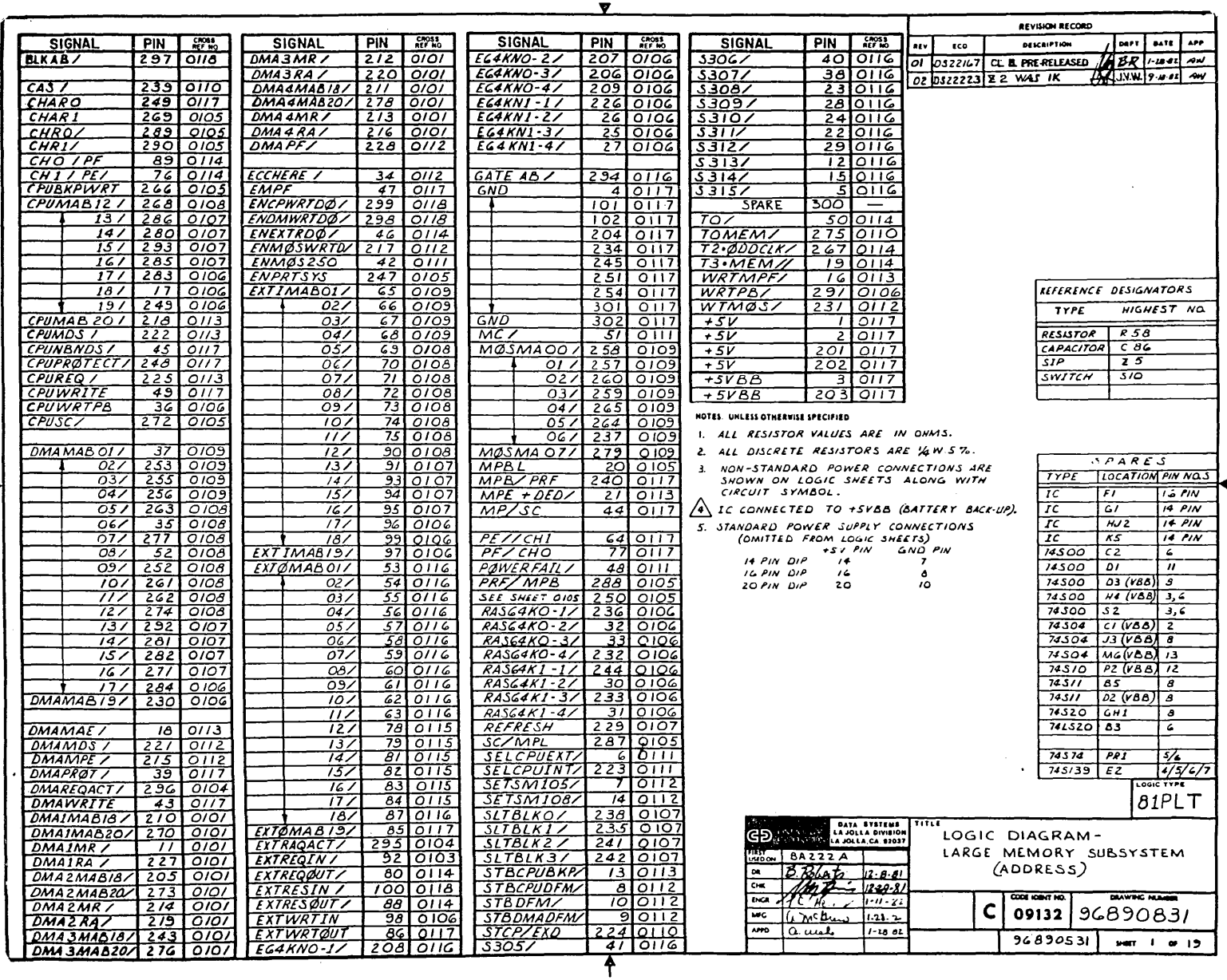
NOTES:

DETACHED LISTS

AA3180 REV. 6/71

PRINTED IN U.S.A.

Figure 5-8. LMS Data Logic Diagram (Sheet 23 of 23)



SIGNAL	PIN	CROSS REF NO
BLKAB/	297	0118
CA3/	239	0110
CHARØ	249	0117
CHAR1	269	0105
CHRØ/	289	0105
CHR1/	290	0105
CHØ/PF	89	0114
CH1/PE/	76	0114
CPUBKPWRT	266	0105
CPUMAB12/	268	0108
13/	286	0107
14/	280	0107
15/	293	0107
16/	285	0107
17/	283	0106
18/	17	0106
19/	249	0106
CPUMAB20/	218	0113
CPUMDS/	222	0113
CPUNBNDS/	45	0117
CPUPRØTECT/	248	0117
CPUREQ/	225	0113
CPUWRITE	49	0117
CPUWRTPB	36	0106
CPUSC/	272	0105
DMAMAB01/	37	0109
02/	253	0109
03/	255	0109
04/	256	0109
05/	263	0108
06/	35	0108
07/	277	0108
08/	52	0108
09/	252	0108
10/	261	0108
11/	262	0108
12/	274	0108
13/	292	0107
14/	281	0107
15/	282	0107
16/	271	0107
17/	284	0106
DMAMAB19/	230	0106
DMAMAE/	18	0113
DMAMDS/	221	0112
DMAMPE/	215	0112
DMAPRØT/	39	0117
DMAREQACT/	296	0104
DMAWRITE	43	0117
DMA1MAB18/	210	0101
DMA1MAB20/	270	0101
DMA1MR/	11	0101
DMA1RA/	227	0101
DMA2MAB18/	205	0101
DMA2MAB20/	273	0101
DMA2MR/	214	0101
DMA2RA/	219	0101
DMA3MAB18/	243	0101
DMA3MAB20/	276	0101

SIGNAL	PIN	CROSS REF NO
DMA3MR/	212	0101
DMA3RA/	220	0101
DMA4MAB18/	211	0101
DMA4MAB20/	278	0101
DMA4MR/	213	0101
DMA4RA/	216	0101
DMAPF/	228	0112
ECCHERE/	34	0112
EMPF	47	0117
ENCPWRTDØ/	299	0118
ENDMWRTDØ/	298	0118
ENEXTRDØ/	46	0114
ENMØSWRTD/	217	0112
ENMØS250	42	0111
ENPRTSYS	247	0105
EXTIMAB01/	65	0109
02/	66	0109
03/	67	0109
04/	68	0109
05/	69	0108
06/	70	0108
07/	71	0108
08/	72	0108
09/	73	0108
10/	74	0108
11/	75	0108
12/	90	0108
13/	91	0107
14/	93	0107
15/	94	0107
16/	95	0107
17/	96	0106
18/	99	0106
EXTIMAB19/	97	0106
EXTØMAB01/	53	0116
02/	54	0116
03/	55	0116
04/	56	0116
05/	57	0116
06/	58	0116
07/	59	0116
08/	60	0116
09/	61	0116
10/	62	0116
11/	63	0116
12/	78	0115
13/	79	0115
14/	81	0115
15/	82	0115
16/	83	0115
17/	84	0115
18/	87	0116
EXTØMAB19/	85	0117
EXTRAQACT/	295	0104
EXTREQIN/	92	0103
EXTREQØUT/	80	0114
EXTRESIN/	100	0118
EXTRESØUT/	88	0114
EXTWRTIN	98	0106
EXTWRTØUT	86	0117
EG4KNO-1/	208	0116

SIGNAL	PIN	CROSS REF NO
EG4KNO-2/	207	0106
EG4KNO-3/	206	0106
EG4KNO-4/	209	0106
EG4KN1-1/	226	0106
EG4KN1-2/	26	0106
EG4KN1-3/	25	0106
EG4KN1-4/	27	0106
GATE AB/	294	0116
GND	4	0117
	101	0117
	102	0117
	204	0117
	234	0117
	245	0117
	251	0117
	254	0117
	301	0117
GND	302	0117
MC/	51	0111
MØSMA00/	258	0109
01/	257	0109
02/	260	0109
03/	259	0109
04/	265	0109
05/	264	0109
06/	237	0109
MØSMA07/	279	0109
MPBL	20	0105
MPB/PRF	240	0117
MPE+DED/	21	0113
MP/SC	44	0117
PE//CH1	64	0117
PF/CHØ	77	0117
PØWERFAIL/	48	0111
PRF/MPB	288	0105
SEE SHEET 0105	250	0105
RASG4KO-1/	236	0106
RASG4KO-2/	32	0106
RASG4KO-3/	33	0106
RASG4KO-4/	232	0106
RASG4K1-1/	244	0106
RASG4K1-2/	30	0106
RASG4K1-3/	233	0106
RASG4K1-4/	31	0106
REFRESH	229	0107
SC/MPL	287	0105
SELCPUEXT/	6	0111
SELCPUINT/	223	0111
SETSM105/	7	0112
SETSM108/	14	0112
SLTBLKØ/	238	0107
SLTBLK1/	235	0107
SLTBLK2/	241	0107
SLTBLK3/	242	0107
STBCPUBKP/	13	0113
STBCPUDFM/	8	0112
STBDFM/	10	0112
STBDMADFM/	9	0112
STCP/EXD	224	0110
S305/	41	0116

SIGNAL	PIN	CROSS REF NO
S306/	40	0116
S307/	38	0116
S308/	23	0116
S309/	28	0116
S310/	24	0116
S311/	22	0116
S312/	29	0116
S313/	12	0116
S314/	15	0116
S315/	5	0116
SPARE	300	—
TØ/	50	0114
TØMEM/	275	0110
T2·ØDDCLK/	267	0114
T3·MEM//	19	0114
WRTMPF/	16	0113
WRTPB/	291	0106
WTMØS/	231	0112
+5V	1	0117
+5V	2	0117
+5V	201	0117
+5V	202	0117
+5VBB	3	0117
+5VBB	203	0117

NOTES UNLESS OTHERWISE SPECIFIED

1. ALL RESISTOR VALUES ARE IN OHMS.
2. ALL DISCRETE RESISTORS ARE ¼ W 5%.
3. NON-STANDARD POWER CONNECTIONS ARE SHOWN ON LOGIC SHEETS ALONG WITH CIRCUIT SYMBOL.
4. IC CONNECTED TO +5VBB (BATTERY BACK-UP).
5. STANDARD POWER SUPPLY CONNECTIONS (OMITTED FROM LOGIC SHEETS)

	+5V PIN	GND PIN
14 PIN DIP	14	7
16 PIN DIP	16	8
20 PIN DIP	20	10

REVISION RECORD

REV	ECO	DESCRIPTION	DEPT	DATE	APP
01	0322167	CL B PRE-RELEASED	BR	1-18-82	MW
02	0322223	Z2 WAS 1K	J.V.W.	9-10-82	MW

REFERENCE DESIGNATORS

TYPE	HIGHEST NO.
RESISTOR	R58
CAPACITOR	C86
SIP	Z5
SWITCH	S10

SPARES

TYPE	LOCATION	PIN NOS
IC	F1	16 PIN
IC	G1	14 PIN
IC	HJ2	14 PIN
IC	K5	14 PIN
74S00	C2	6
74S00	D1	11
74S00	D3 (VBB)	9
74S00	H4 (VBB)	3,6
74S00	S2	3,6
74S04	C1 (VBB)	2
74S04	J3 (VBB)	8
74S04	M6 (VBB)	13
74S10	P2 (VBB)	12
74S11	B5	8
74S11	D2 (VBB)	8
74S20	GH1	8
74LS20	B3	6
74S74	PR1	5/6
74S139	E2	4/5/6/7

LOGIC TYPE 81PLT

DATA SYSTEMS LA JOLLA DIVISION LA JOLLA, CA 92037		
FIRST USED ON	BA222A	
DR	B. Blatz	12-8-81
CHK		12-9-81
ENGR		1-11-82
MFG		1-21-82
APPD		1-28-82

TITLE: LOGIC DIAGRAM - LARGE MEMORY SUBSYSTEM (ADDRESS)

C | CODE IDENT NO. 09132 | DRAWING NUMBER 96890831

96890531 | SHEET 1 OF 19

Figure 5-9. LMS Address Logic Diagram (Sheet 1 of 19)

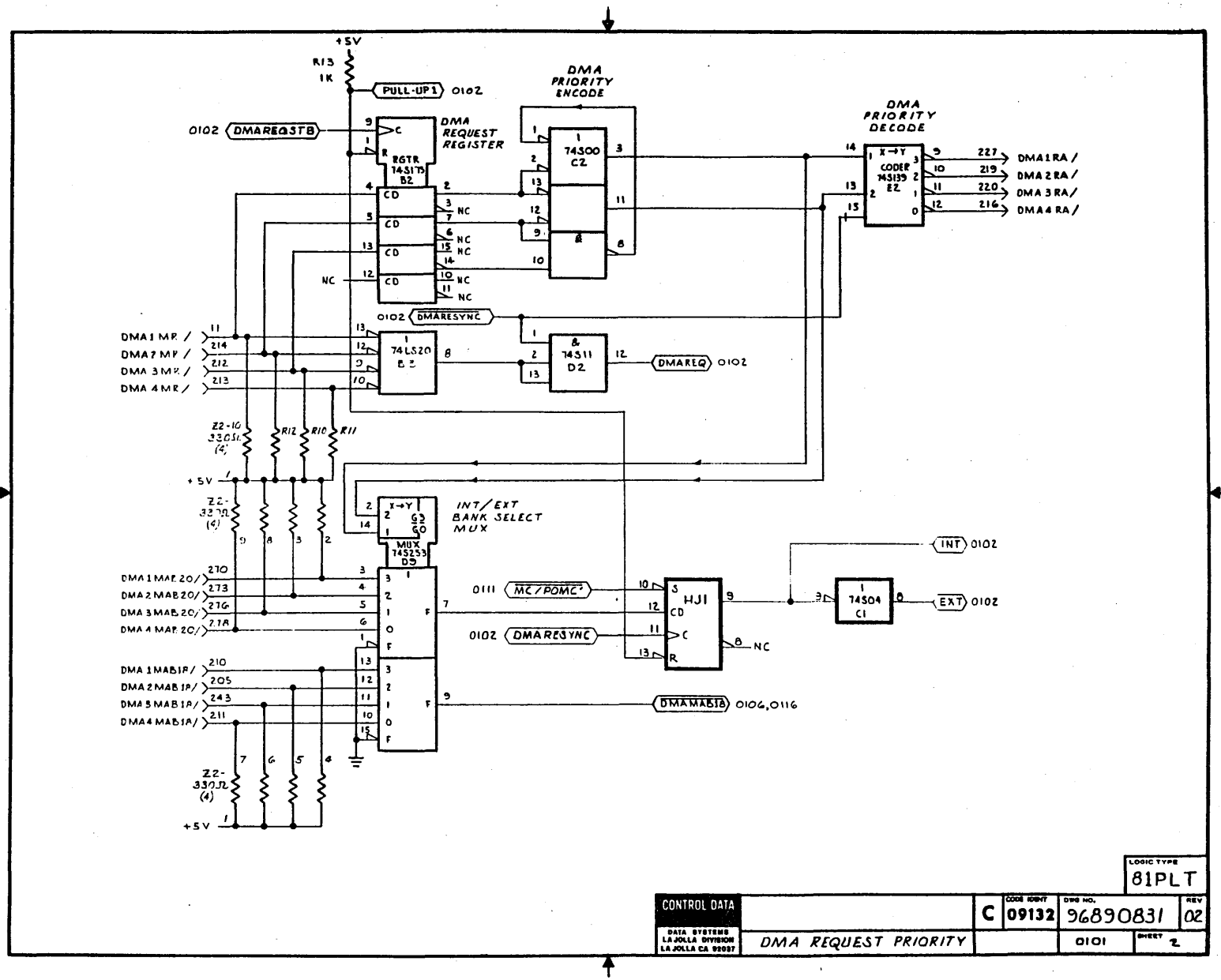


Figure 5-9. LMS Address Logic Diagram (Sheet 2 of 19)

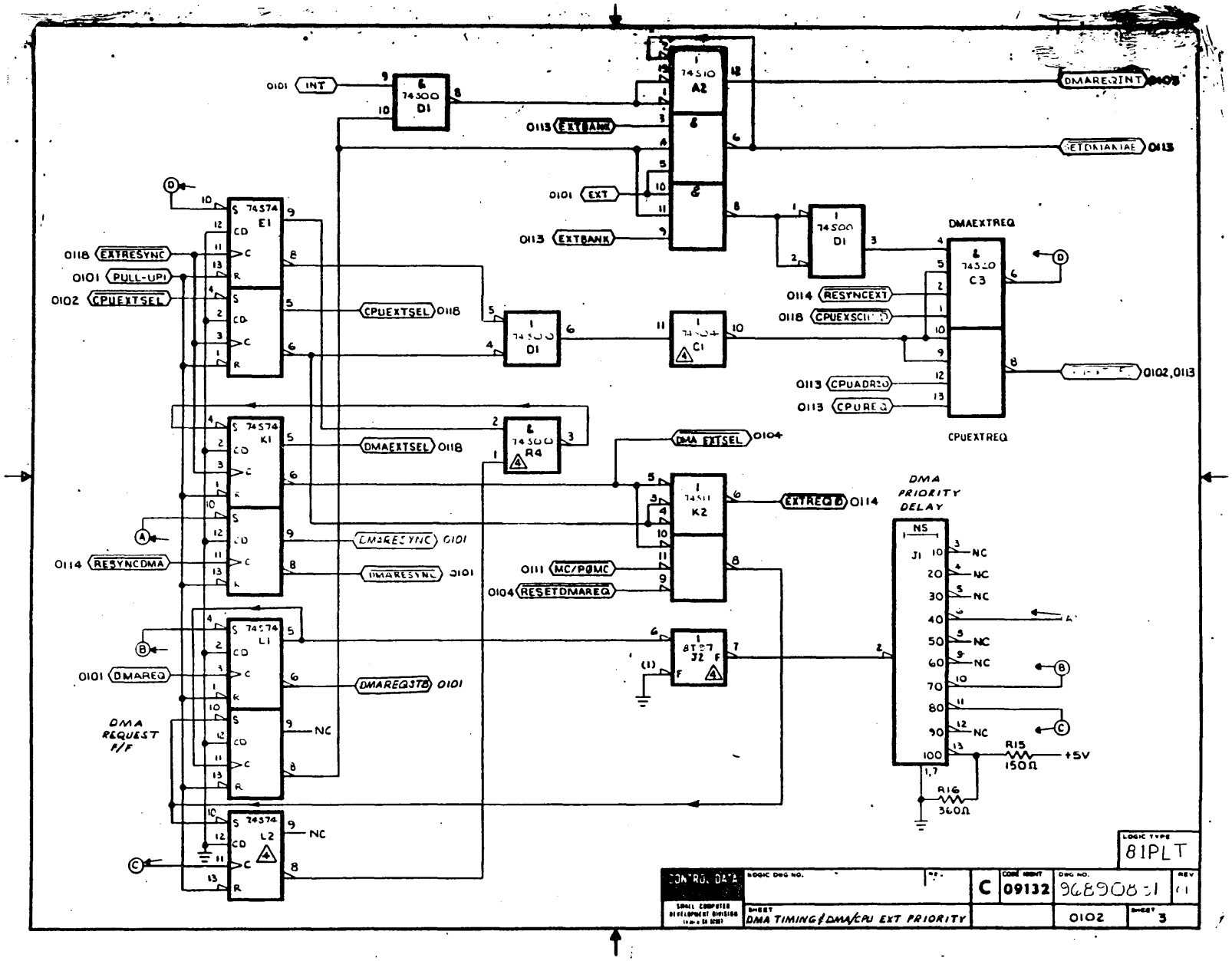


Figure 5-9. LMS Address Logic Diagram (Sheet 3 of 19)

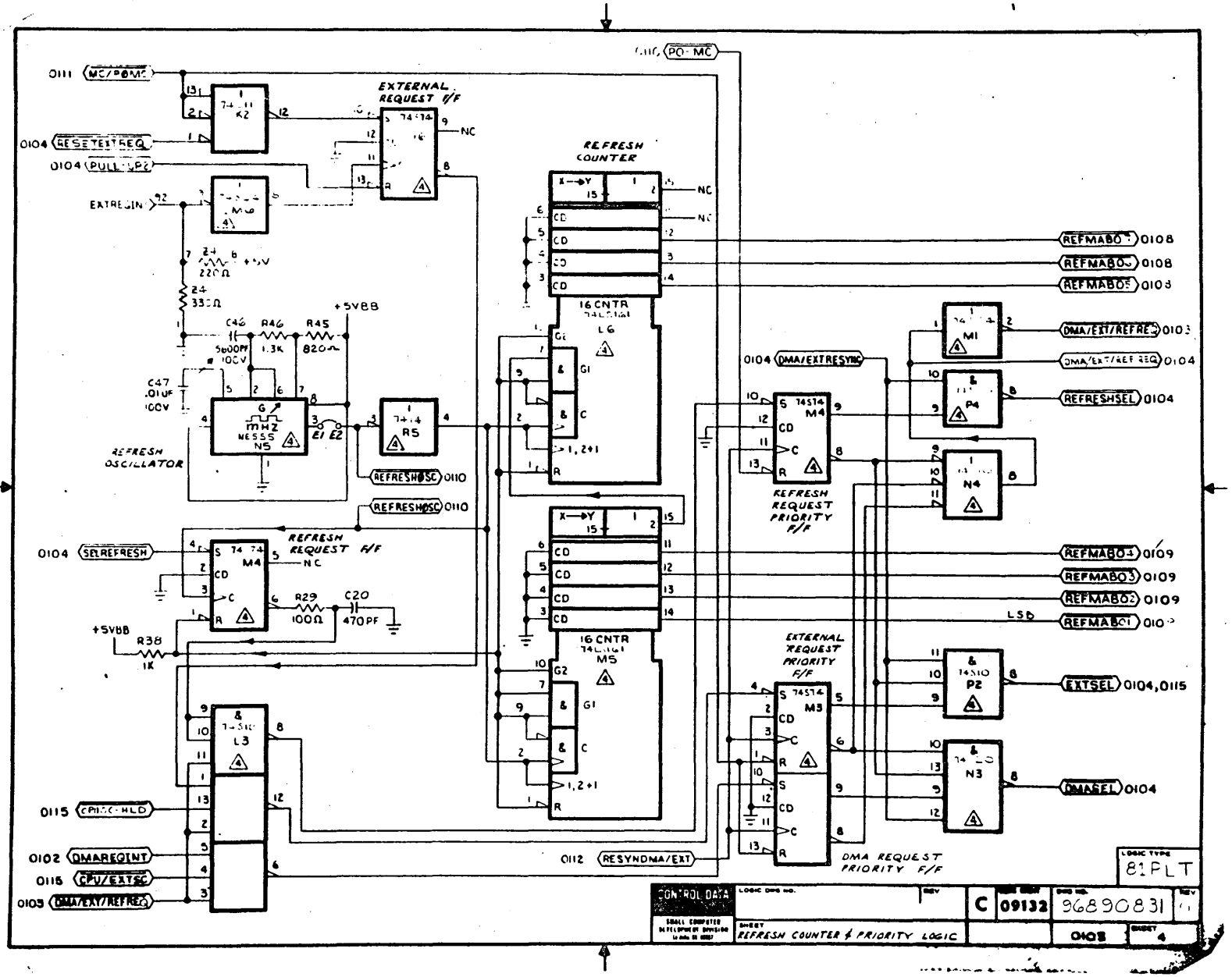


Figure 5-9. LMS Address Logic Diagram (Sheet 4 of 19)

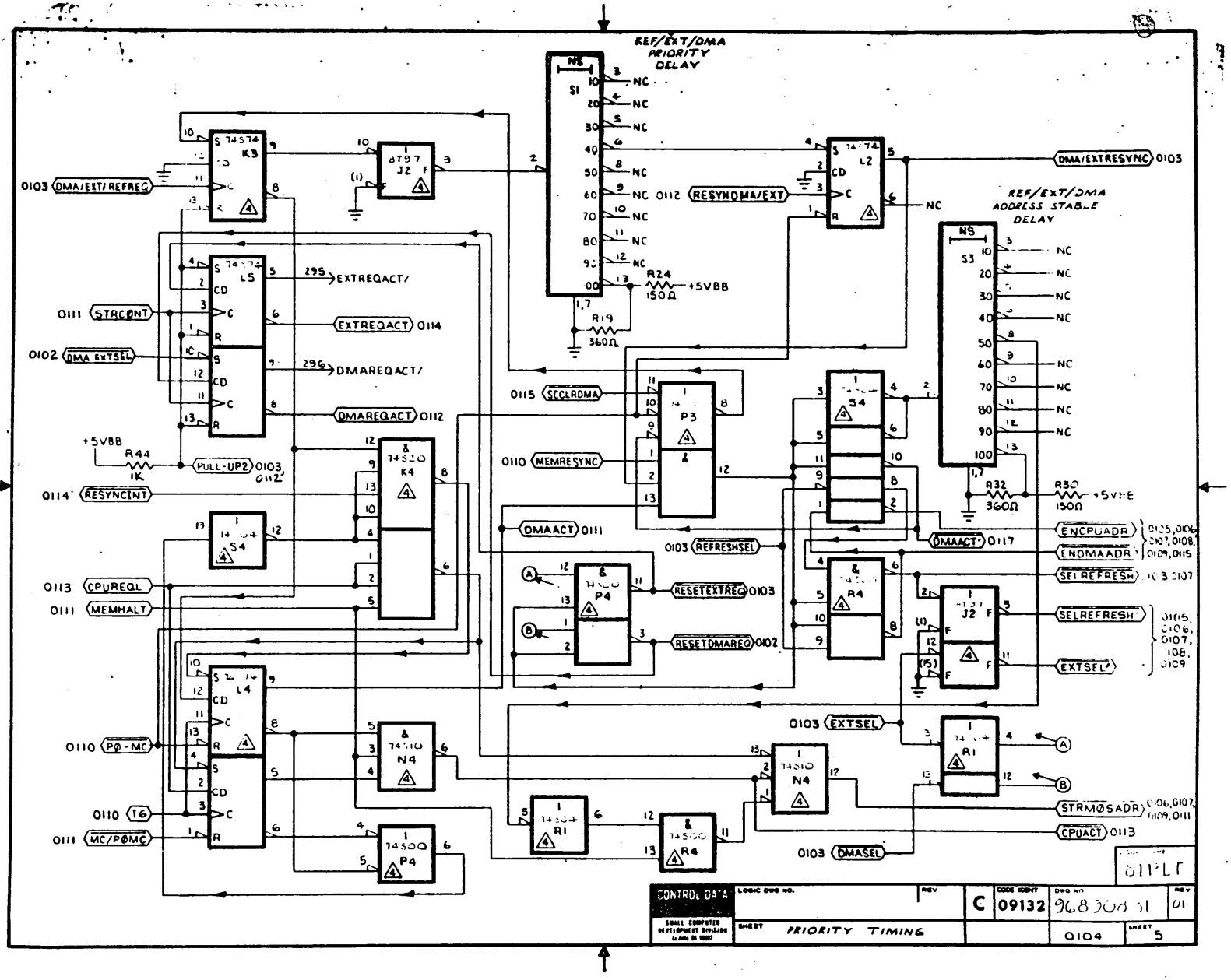


Figure 5-9. LMS Address Logic Diagram (Sheet 5 of 19)

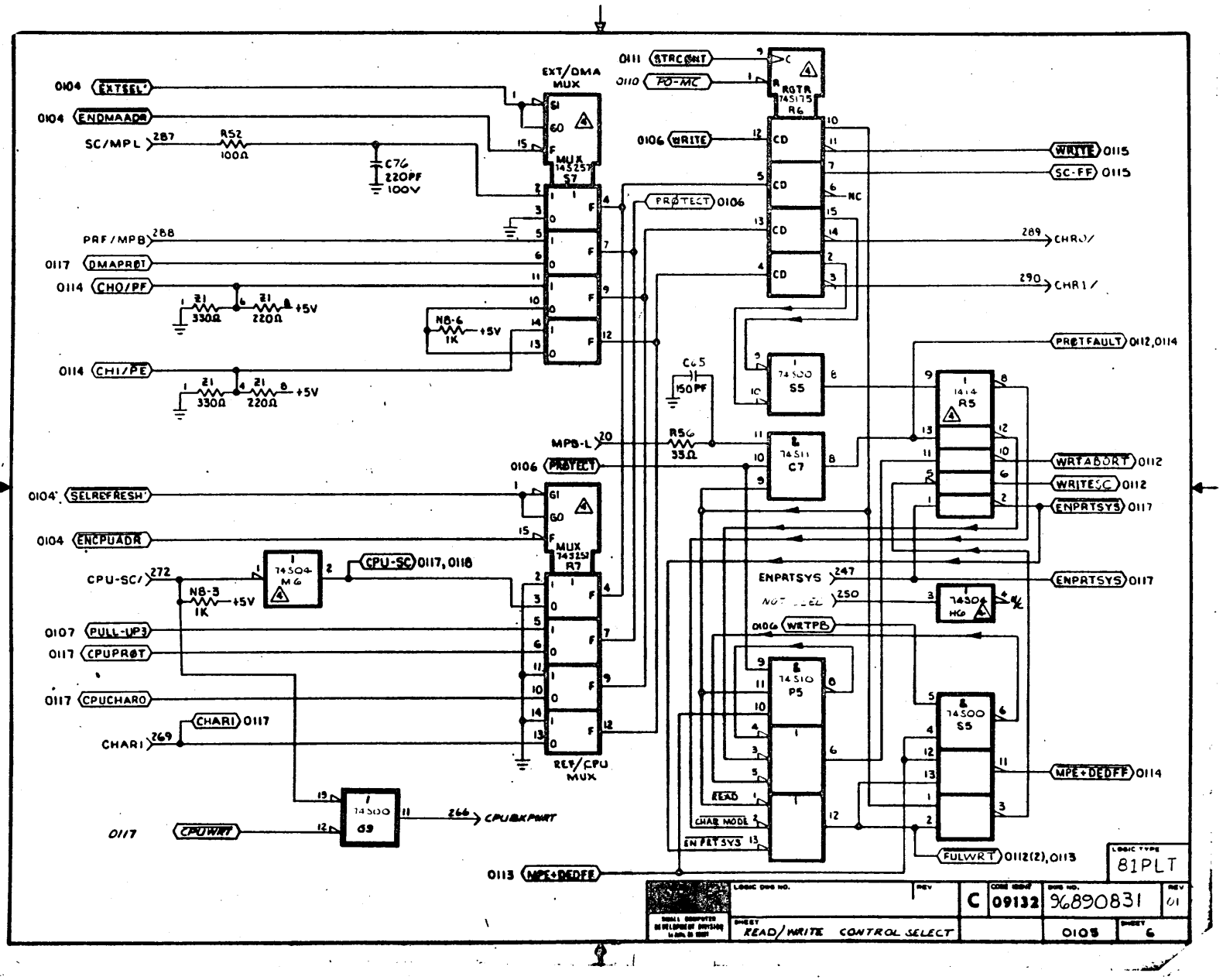


Figure 5-9. LMS Address Logic Diagram (Sheet 6 of 19)

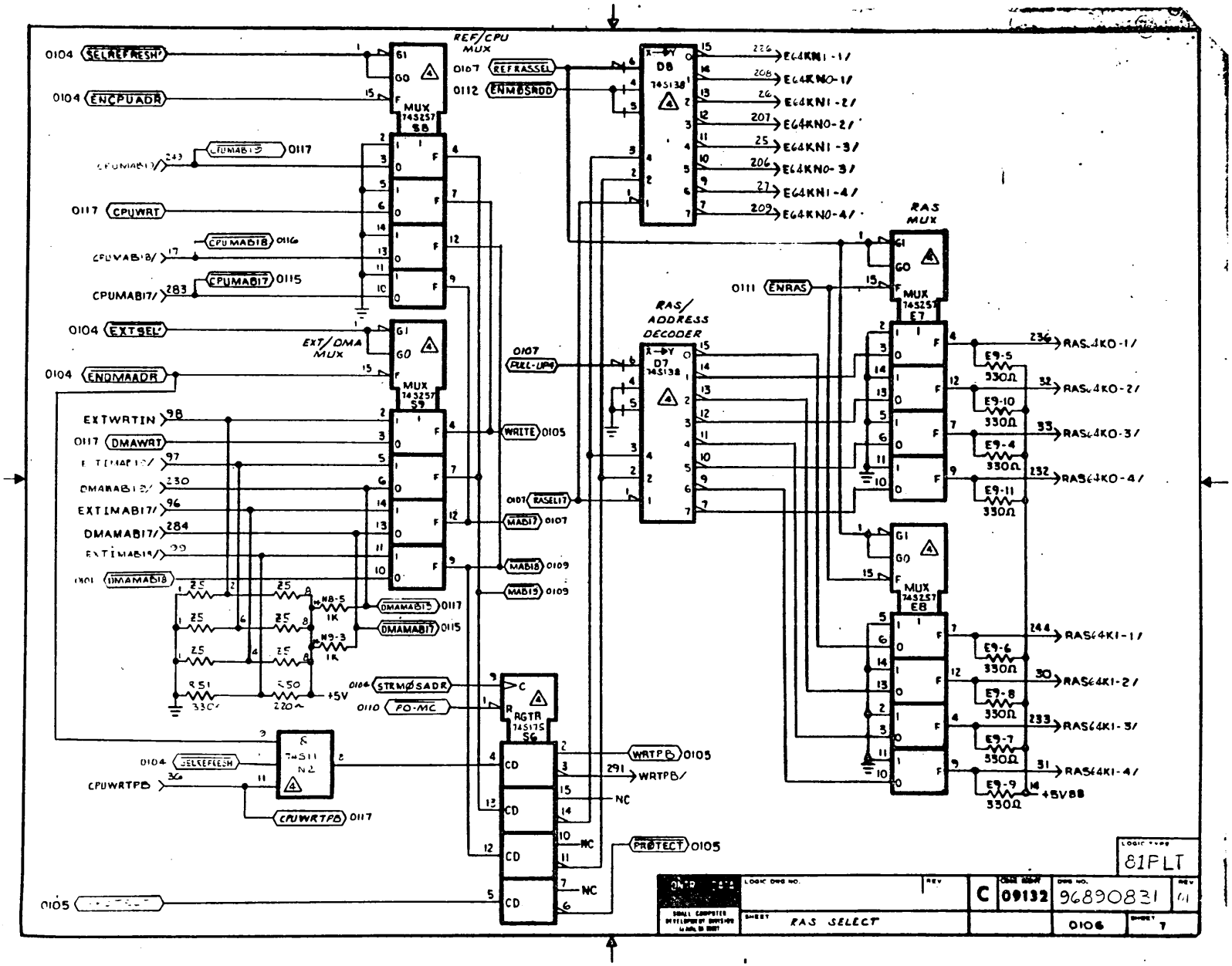


Figure 5-9. LMS Address Logic Diagram (Sheet 7 of 19)

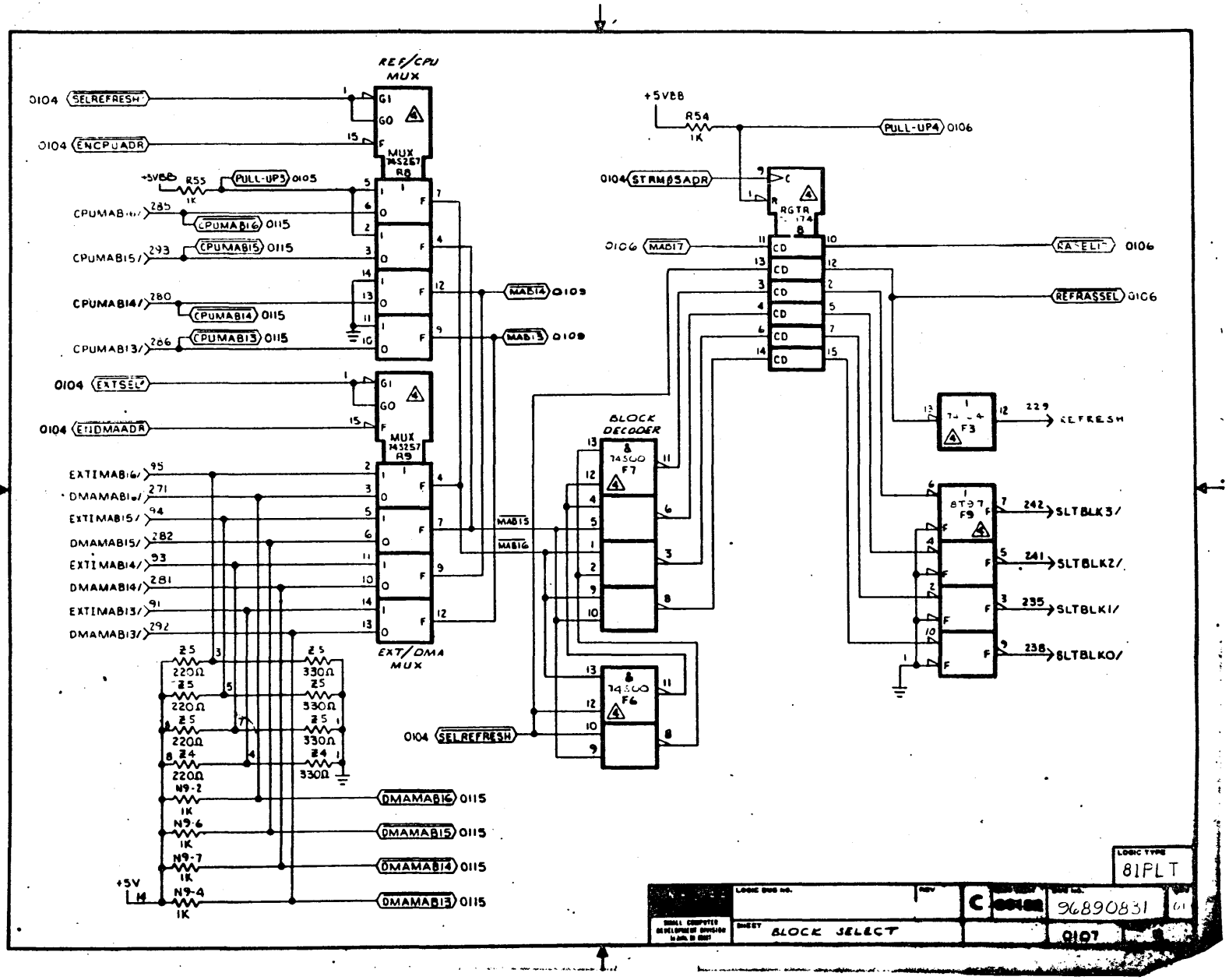


Figure 5-9. LMS Address Logic Diagram (Sheet 8 of 19)

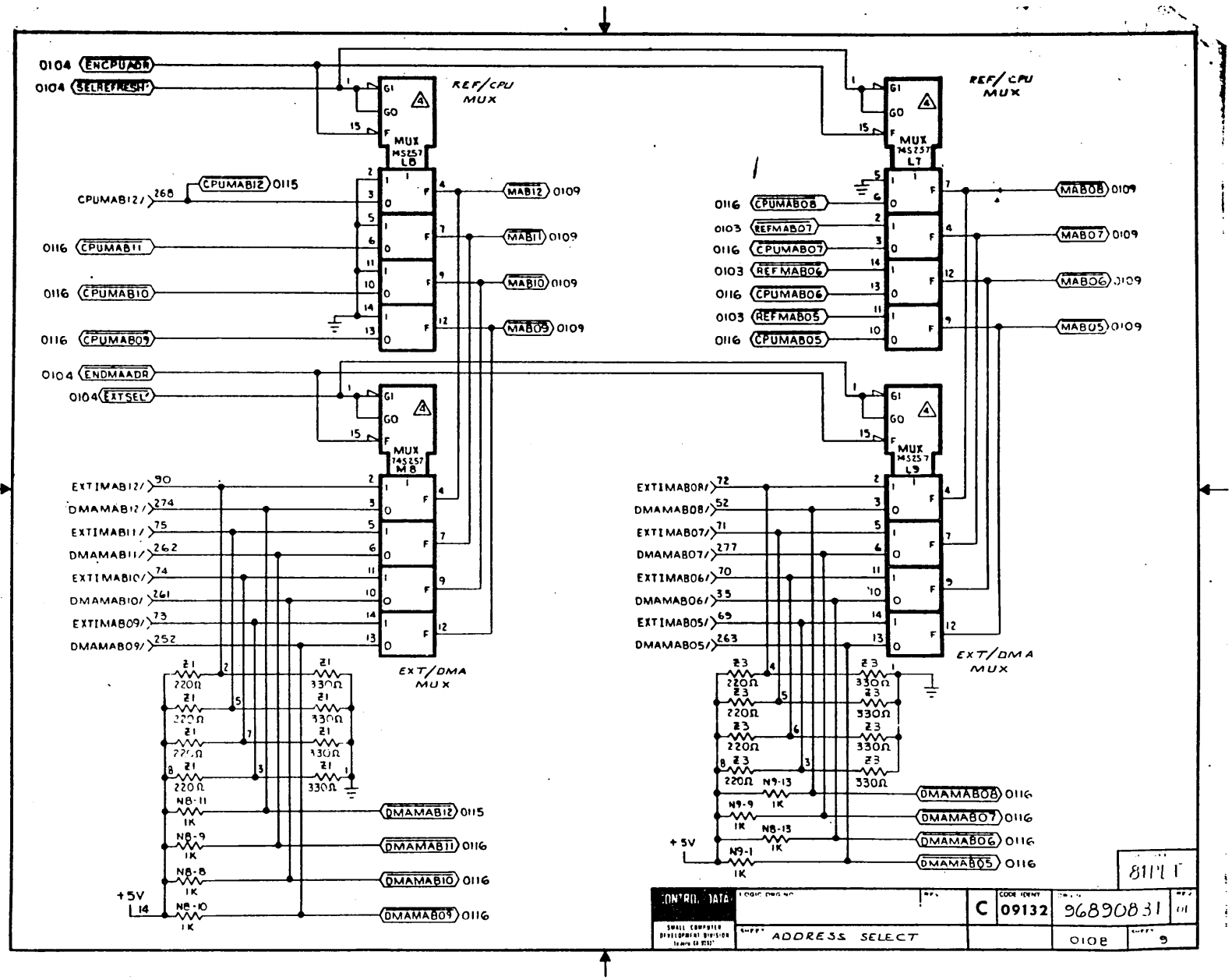


Figure 5-9. LMS Address Logic Diagram (Sheet 9 of 19)

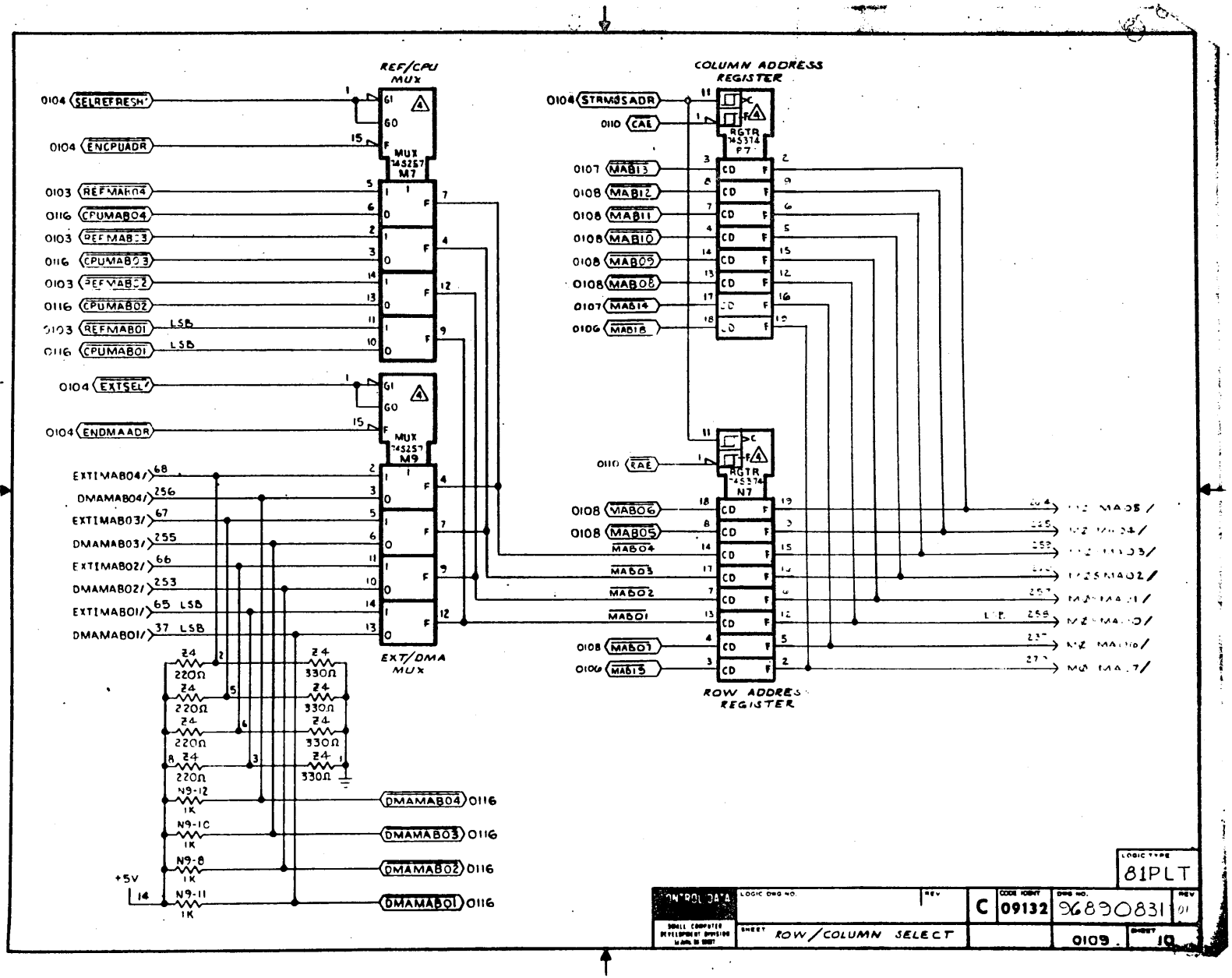


Figure 5-9. LMS Address Logic Diagram (Sheet 10 of 19)

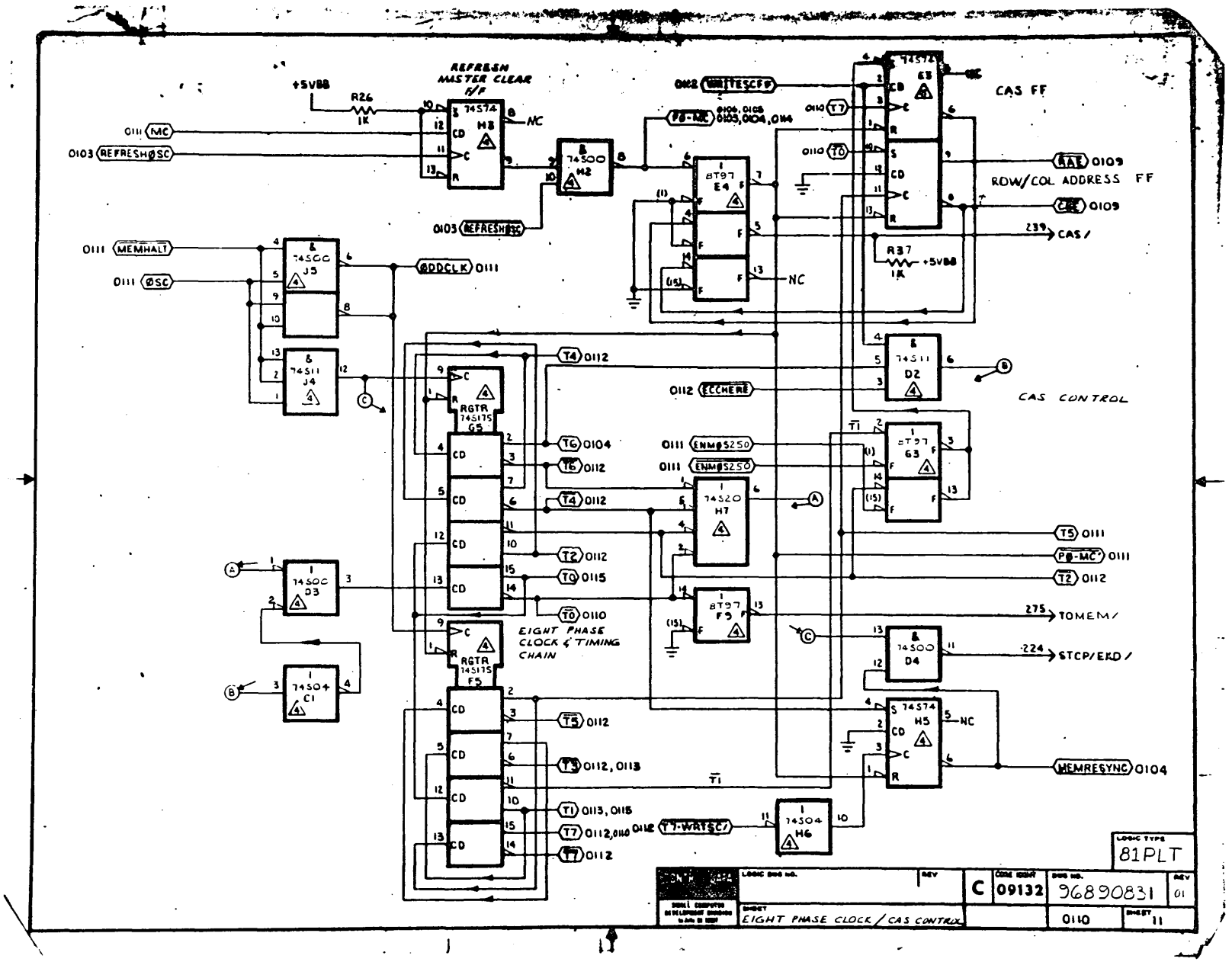


Figure 5-9. LMS Address Logic Diagram (Sheet 11 of 19)

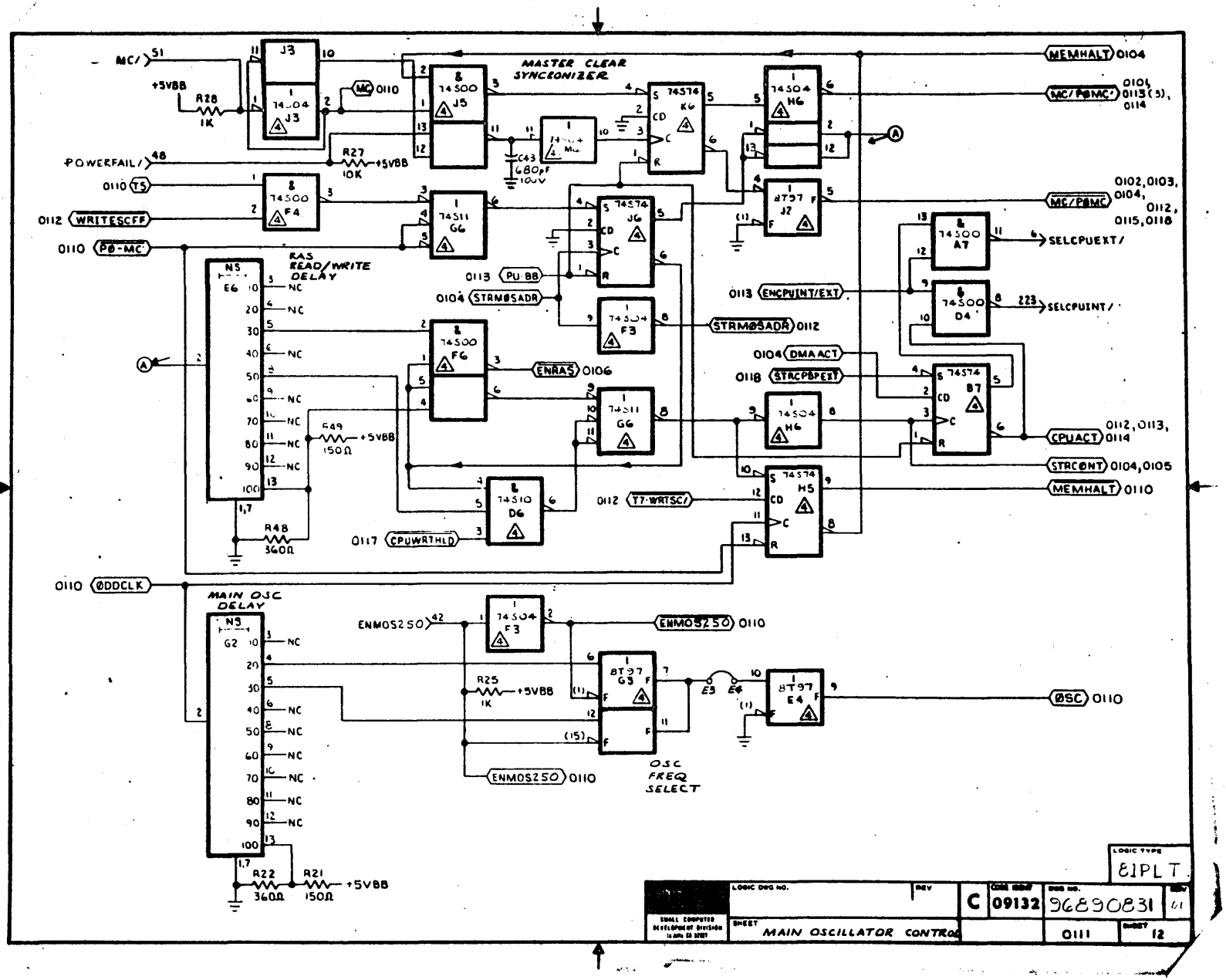


Figure 5-9. LMS Address Logic Diagram (Sheet 12 of 19)

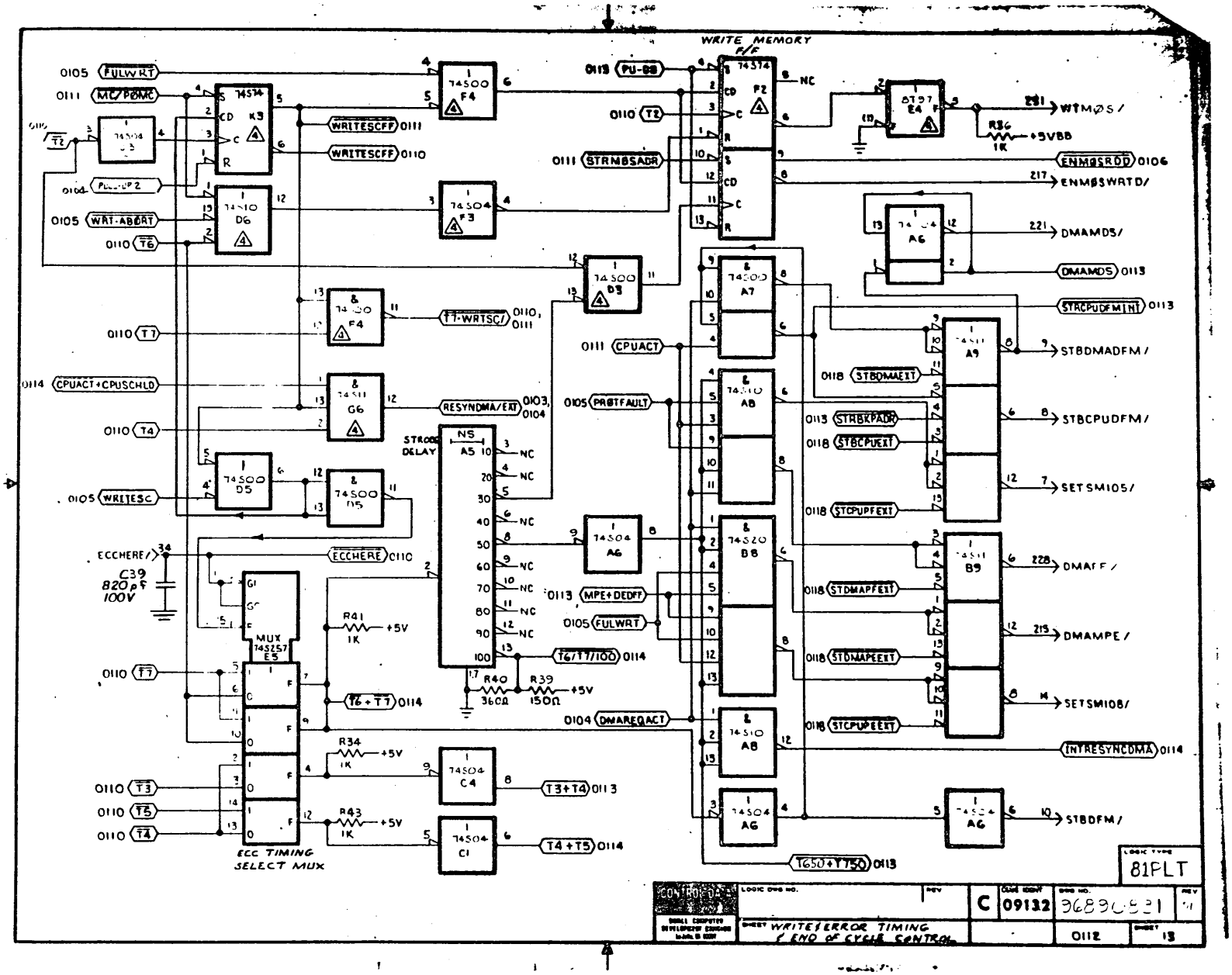


Figure 5-9. LMS Address Logic Diagram (Sheet 13 of 19)

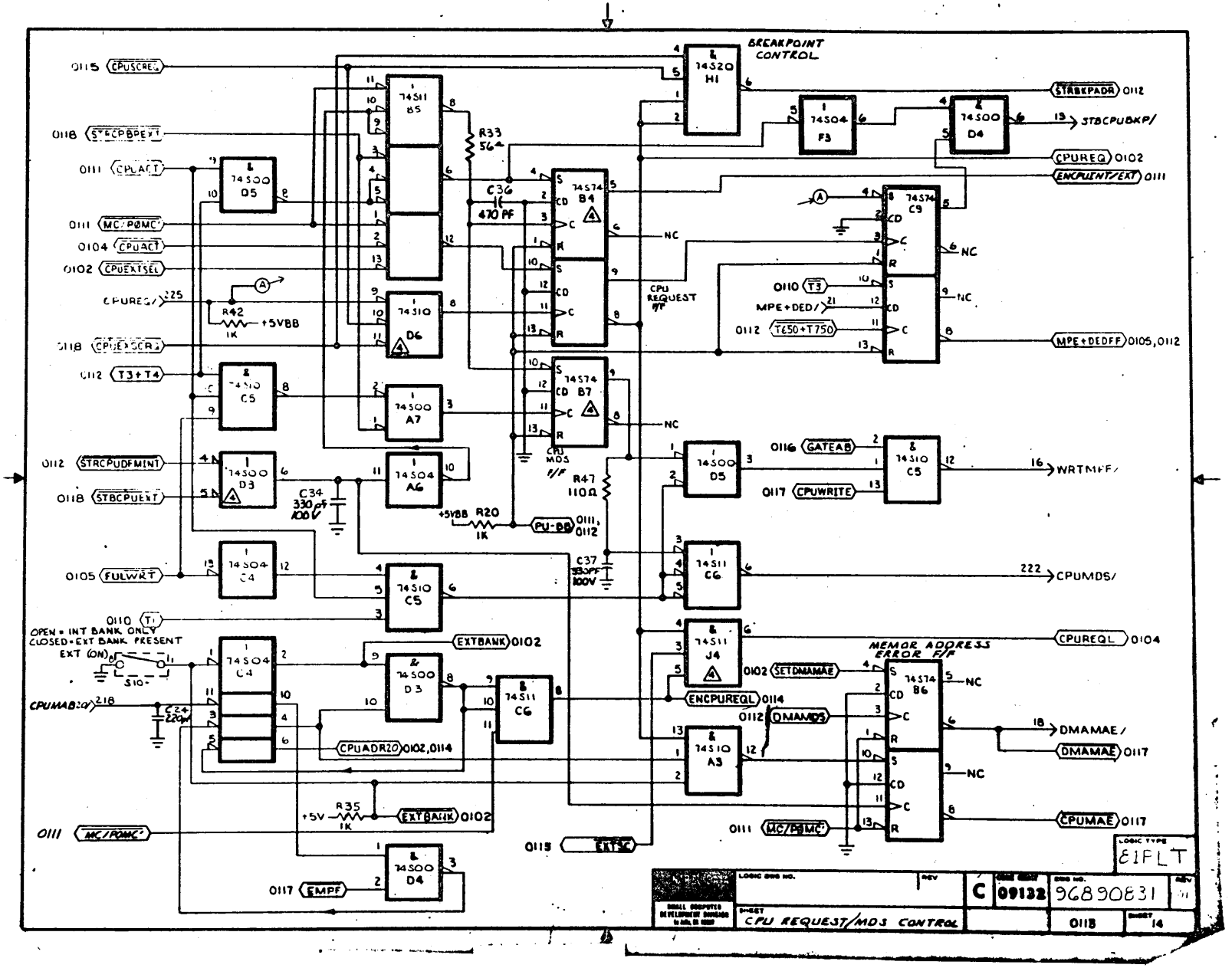


Figure 5-9. LMS Address Logic Diagram (Sheet 14 of 19)

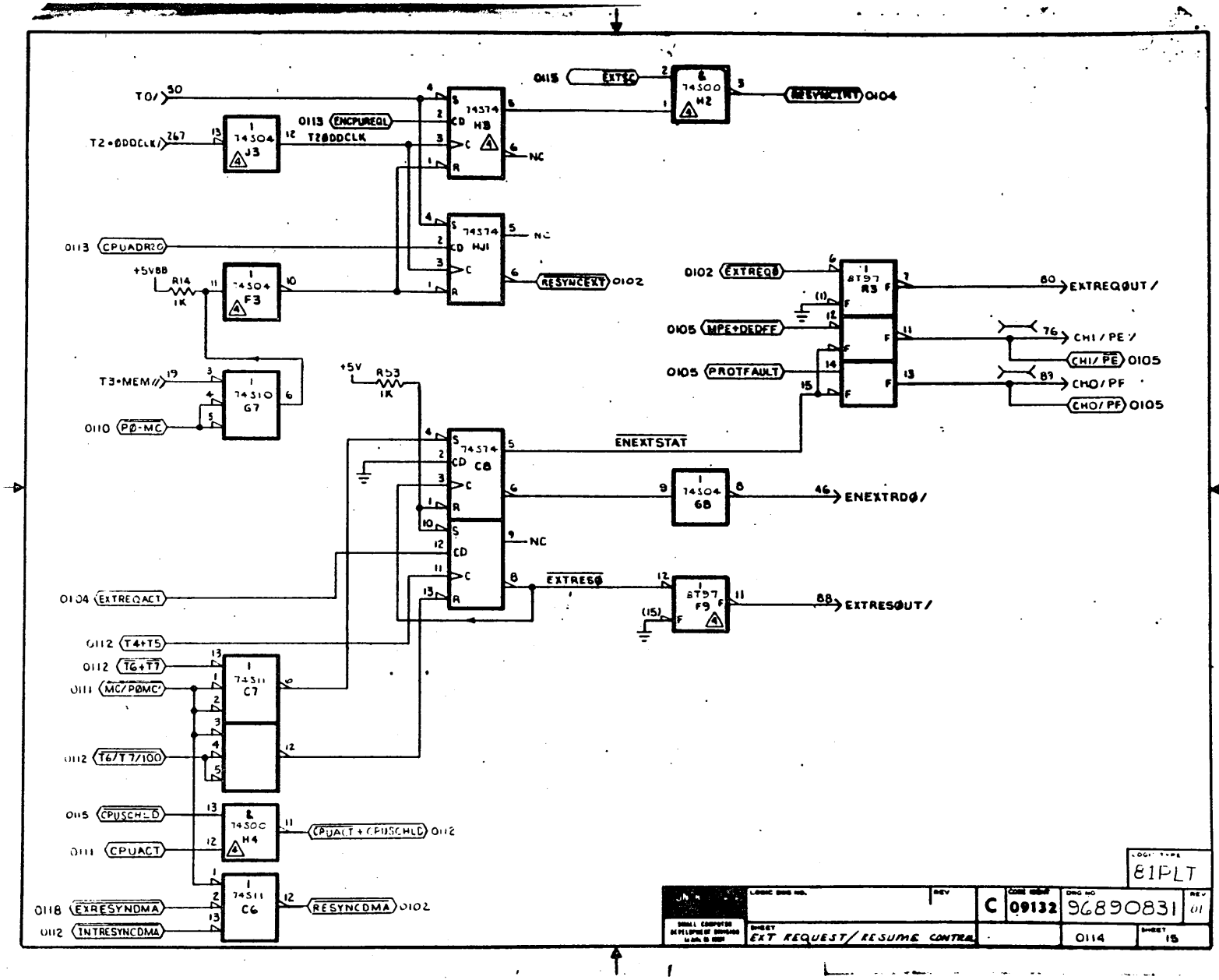


Figure 5-9. LMS Address Logic Diagram (Sheet 15 of 19)

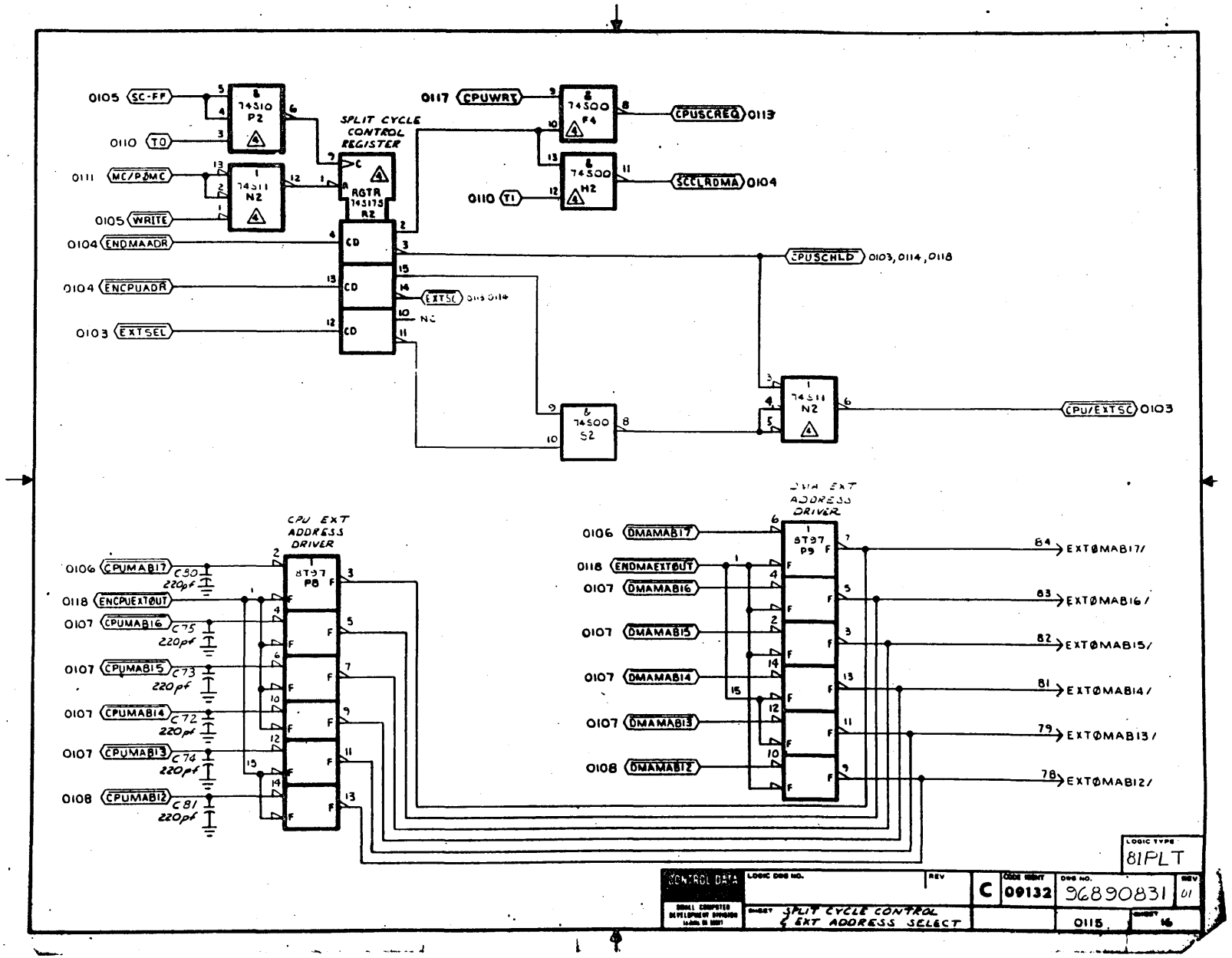


Figure 5-9. LMS Address Logic Diagram (Sheet 16 of 19)

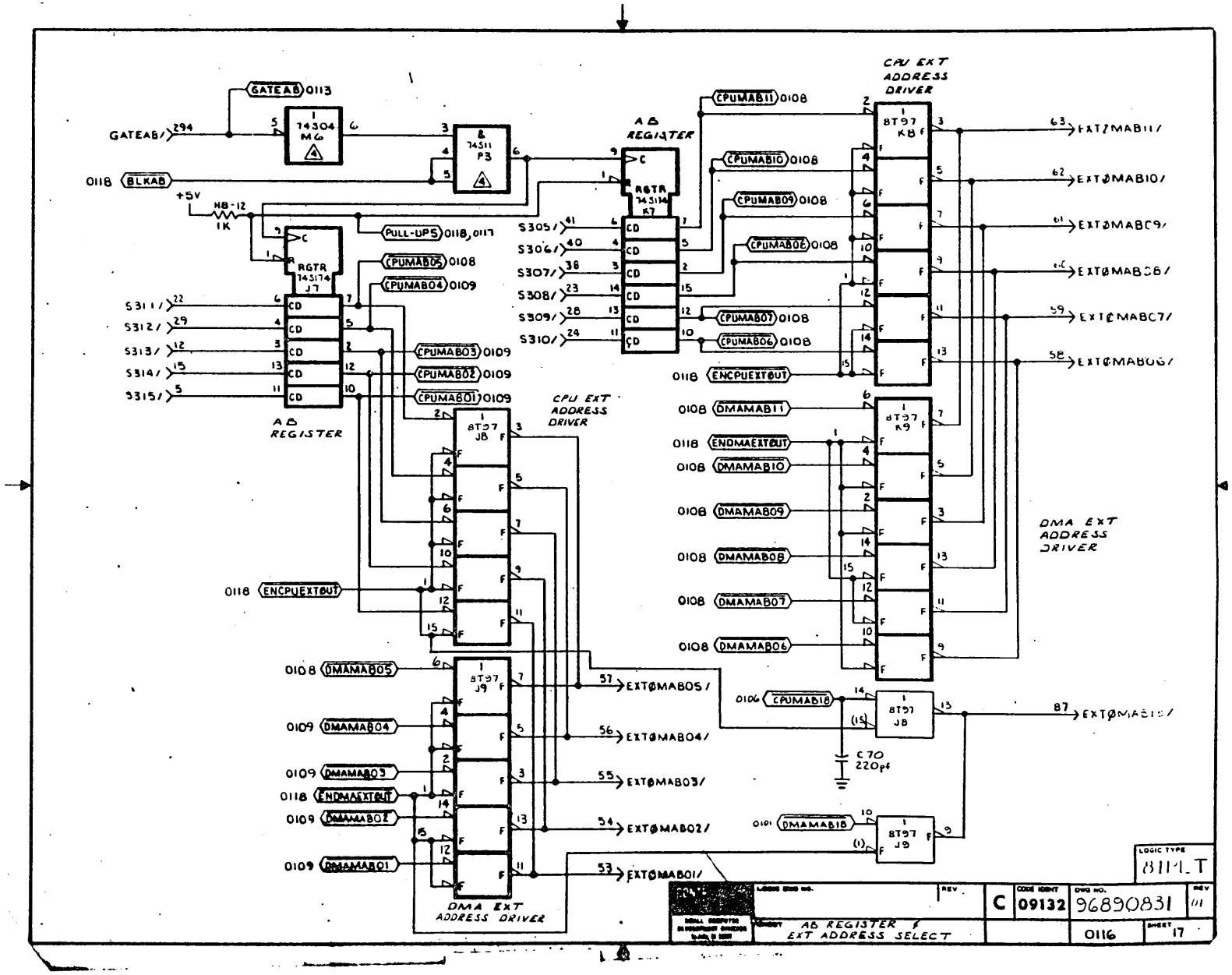


Figure 5-9. LMS Address Logic Diagram (Sheet 17 of 19)

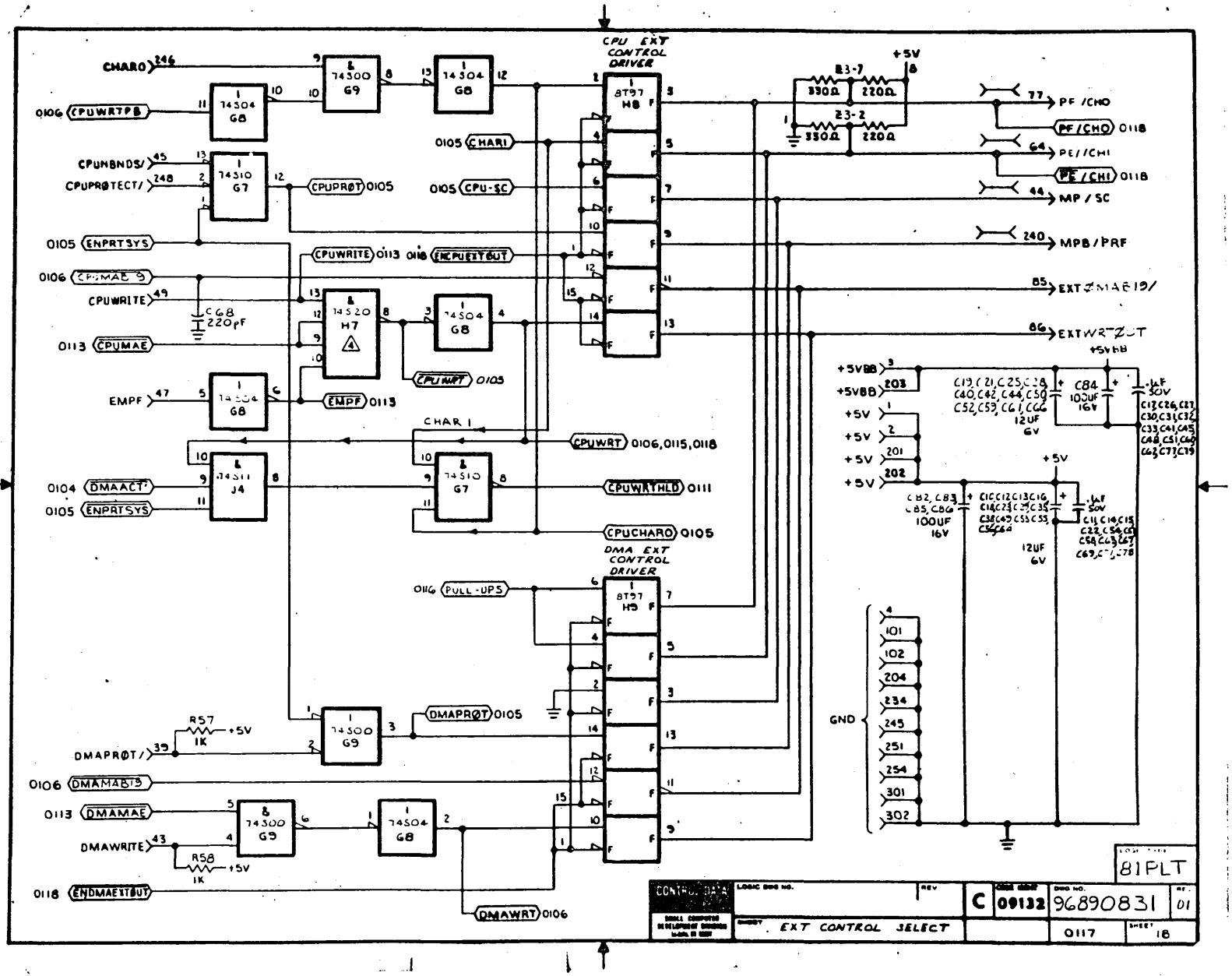


Figure 5-9. LMS Address Logic Diagram (Sheet 18 of 19)

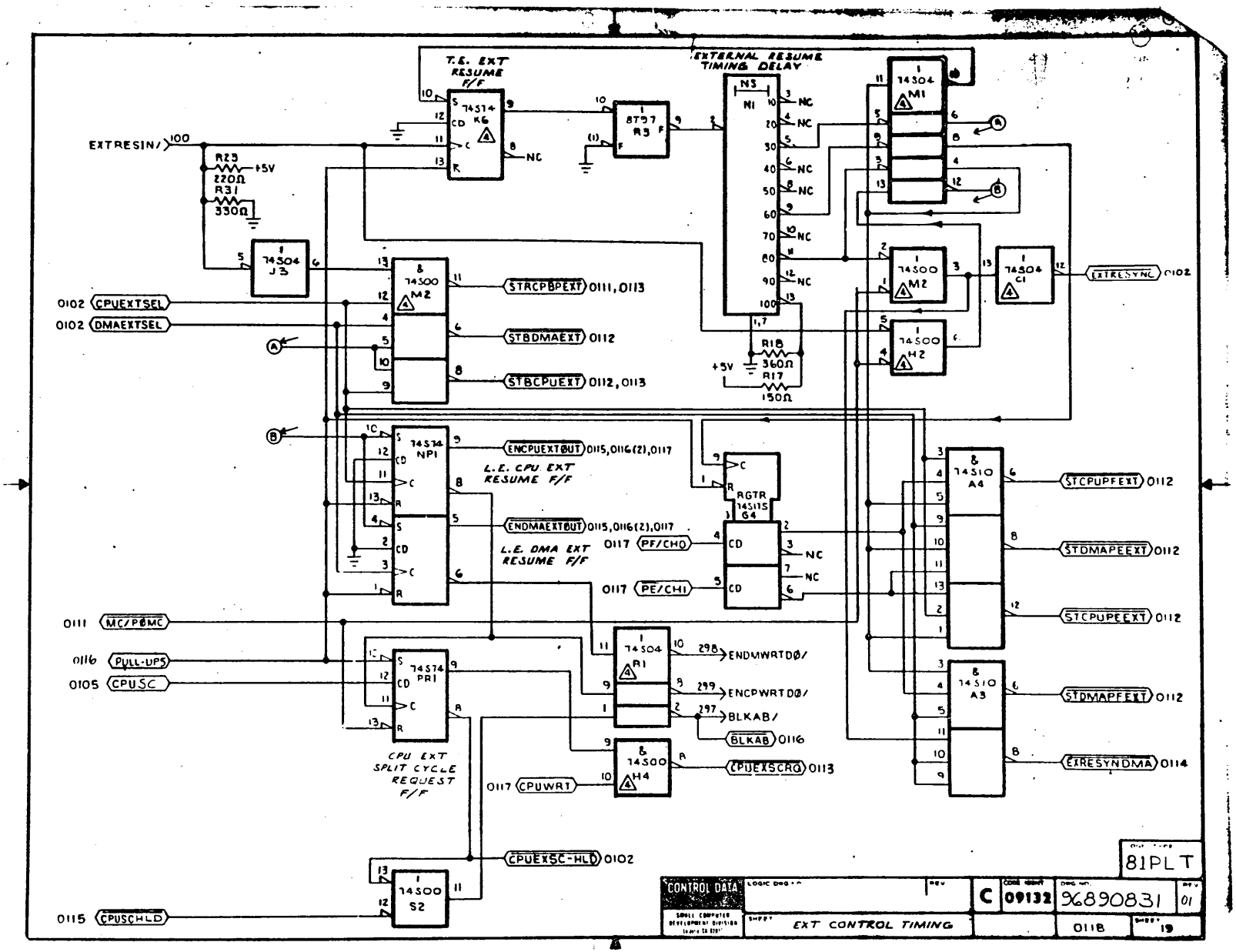


Figure 5-9. LMS Address Logic Diagram (Sheet 19 of 19)

MAINTENANCE 6

This section contains three basic maintenance aids: the Operational Diagnostic System (ODS), the diagnostic decision logic tables (DDLTs), and the procedures applicable to the actions required in the DDLTs.

ODS

The following are precautions and additional information that the customer engineer should remember when performing diagnostic testing:

- Place the console device in the scroll mode.
- Power to the peripherals should be turned off before power is turned off to the computer; peripheral power should be turned on last, after power to the computer has been turned on.
- Printed wiring assemblies may be removed from or installed in the processor with the power on, but a master clear should be performed first and mass memory devices should be stopped before any PWA is removed.
- Actions to replace a PWA reference a CPU slot unless otherwise specified.
- Replacement ACTIONS reference the CPU in which the test is executing (i.e., the CPU under test) unless otherwise specified.
- None of the keys on the console display may be pressed while the diagnostic medium is being loaded.
- When the ESC key is depressed, the computer is placed in panel mode. If the ESC key is accidentally depressed, the condition can be rectified by depressing the @ key.
- The DDLTs presume that all operator inputs are entered correctly. Any incorrect entry may cause the DDLT to direct an incorrect action. If any doubt exists about the accuracy of the operator entry, always repeat the sequence of DDLT steps that led up to an action before taking further steps.
- If an erroneous test entry is made and executed during a level I test, perform the following sequence:

  1. ESC
  2. HG
  3. Master clear
  4. ESC
  5. I @

  This sequence returns the operator to the beginning of the test.

  NOTE

  Do not perform this sequence during a memory test or load test.

For additional information regarding the use of the ODS diagnostic tests, refer to the Operation Diagnostic System (ODS) Version 2 Reference Manual listed in the preface and appendix A of this manual.

For error code typeouts not listed in the DDLTs, refer to the following:

Typeout	Description
Ghost Interrupt	Level I tests. An unexpected interrupt was received.
Lines xxxx	xxxx = A bit mask indicating which line was interrupted.

DESCRIPTION OF DIAGNOSTIC DECISION LOGIC TABLES (DDLTs)

The DDLT tables include the isolation aids for the LMS subsystem. The DDLTs are to be used in conjunction with the procedures contained in this section.

A DDLT is arranged in five sections: assumptions, conditions, responses, actions, and sequence of actions (figure 6-1).

ASSUMPTIONS

The upper section of the DDLT contains the prerequisites for the specific tests to be performed. Read the assumptions and ensure that each assumption is true before proceeding with the examination of the conditions.

Power On						Sheet 1	
ASSUMPTIONS:							
1. Entry is made from system initialization table.							
2. Equipment power cord is connected to ac outlet.							
3. Power switch is set to ON position.							
4. Motor switch is set to OFF.							
5. Keyboard console device is operational.							
CONDITIONS:	1	2	3	4	5	6	RESPONSES
1. Is power indicator illuminated?	Y				N		
2. Are any indicators illuminated?					Y	N	
3. Is cooling fan running?	Y			N			
4. Turn on motor. Did motor start?	Y		N				
5. Are the run parameters correct?	Y	N					
ACTIONS:							SEQUENCE OF ACTIONS
1. Go to sheet 2 of this table.	X						
2. At keyboard enter: Run parameters for procedure Z Go to sheet 2 of this table.		X					
3. Check power toggle switch.						2	
4. Check fuse at rear of cabinet and replace if defective.						1	
5. Replace cooling fan assembly.				2			
6. Check 20-volt power supply and adjust to proper value.			1		2	3	
7. Replace switch board and cable assembly.				1	3	4	
8. Replace lamp in indicator.					1		
9. Replace power supply assembly.			3		4	5	
10. Replace control board.			2	3			
11. Call next level of support.			4	4	5	6	

Figure 6-1. Hypothetical DDLT Example

NOTE

The DDLT is valid only if all assumptions are true.

CONDITIONS

The center left section of the DDLT contains the conditions or tests to be made. They are in the form of questions that can be answered YES or NO. Read the first condition and determine if the response is YES (Y) or NO (N).

CONDITION RESPONSE

The location of the response in the response section determines the column to be evaluated. Continue evaluation under that column (for example in column 1) until an answer to a condition causes a branch to another column, then evaluate that column (for example, column 5).

Continue this procedure until there are no further conditions under the column being evaluated, then proceed to the actions section. If there are no answers for a condition in the column being evaluated, then the condition is not applicable to the evaluation (for example, conditions 3, 4, and 5 in column 5).

ACTIONS

The lowest left quadrant is a list of actions. The actions may direct you to stop, continue troubleshooting a problem, take corrective action, check parameters, or enter parameter changes. Actions are performed according to the sequence specified in the sequence of actions under the column being evaluated.

SEQUENCE OF ACTIONS

The lower right section lists the sequence of actions. These sequences are either a number, indicating sequential order of actions to perform, or an X, for a single action to perform. These actions are performed under the column being evaluated. The action will give the intervening directions to keep troubleshooting moving logically. However, the CE has some general directions that may not be included in the DDLTs. These general directions are:

- Corrective Actions

  Problem corrected - Restore system to operation

  Problem not corrected - Undo attempted correction, then continue to next sequential action.

- Parameter Change Action - Make change, then continue to next sequential action.

- Check Actions

  Problem found - Correct minor problem and follow intervening directions.

  No problem found - Continue to next sequential action.

DDLT EXAMPLE

The DDLT flow in cases A, B, and C, through the example shown in figure 6-1, is based on the following assumed responses to conditions 1 through 5:

Condition	Case A	Case B	Case C
1	YES	NO	YES
2	-	YES	-
3	YES	-	YES
4	YES	-	YES
5	YES	-	NO

Case A

Observe condition 1; the response of Y is found in column 1. Note that condition 2 is not applicable to column 1. Therefore, observe condition 3 for response in this case. Proceed in the same manner through the remaining conditions and then execute action 1, identified by an X under column 1, indicating no sequence other than that single action is performed.

Case B

Observe condition 1; the response of N is found in column 5. Proceed to condition 2 and observe under column 5 the response Y. Note that this is the last applicable condition with actions under column 5; therefore, proceed directly to sequential action 1, column 5. Perform the corresponding action number 8. Restart the test and determine if the same problem exists; if so, continue with the next sequential action 2, which corresponds to action number 6. Continue performing sequential actions and restarting the test after each corrective action is completed until the problem is resolved or other action directed. When a problem is corrected by an action, restore the system to normal operation.

Case C

Observe condition 1; the response Y is found in column 1. Note that condition 2 is not applicable to column 1; therefore, proceed to condition 3 and observe Y in column 1. Condition 4 also appears as Y in column 1. However, in this case condition 5 response is N, located under column 2. Perform action number 2, which is a single action. Note that this is a parameter change action; therefore, do not restart the test unless so directed by the action.

DIAGNOSTIC DECISION LOGIC TABLES

Tables 6-1 through 6-10 are the DDLTs for the LMS subsystem. The removal/replacement and testing procedures referenced in the DDLTs immediately follow these DDLTs.

TABLE 6-1. CUSTOMER INPUT

Sheet 1 of 1

ASSUMPTIONS:

Customer engineer discusses nature of problems with customer.

CONDITIONS:	1	2	3
1. Does customer suspect a particular subsystem or retest after an action?	N	Y	
2. Is a processor suspected of having a failure?		Y	N
ACTIONS:			
1. Select the suspected subsystem from the list below and go to the applicable subsystem manual. Subsystems Card Reader Cartridge Disk Drive Communication Multiplexer Console Display Flexible Disk Drive Keyboard Display Terminal Line Printer (Band) Line Printer (Drum) Magnetic Tape Transport (NRZI) Magnetic Tape Transport (NRZI and Phase-Encoded) Storage Module Drive IOX Communication Line Adapter Eight-Channel CLA, DCCLA, Matrix Printer, etc. Communication Multiplexer			X
2. Select the suspected processor and go to table 6-2.		X	
3. Select processor I (that is, "CPU I") and go to table 6-2.	X		

TABLE 6-2. LDCHK: LOADCHECK DDLT

ASSUMPTIONS:

1. The ODS Reference Manual (96768410), and Appendix A of this manual have been reviewed.
2. All subsystems are powered on.
3. The CPU under test is properly cabled to any and all other subsystems including a second CPU (if applicable).
4. Any and all system CPUs have been MASTER CLEARED.
5. The keyboard is unlocked and the shift key is not locked down.
6. Console and I/O TTY PWA baud rates are the same.
7. The FDD write or initialize enable switch is not ON.
8. The LMS ODS LI, VI diskette is in the FDD with the door closed.
9. The display screen has been cleared (CLEAR key or CNTL X on keyboard).

CONDITIONS:	1	2	3
1. Did the screen clear when CLEAR was pressed?	Y		N
2. Press MASTER CLEAR. Press DEADSTART. After two minutes does the screen display: 123456 DPSR LEVEL XX RELEASED DATE MM/DD/YY	Y	N	
ACTIONS:			
1. Go to next table	X		
2. Go to sheet 2 of this table		X	
3. Check console cabling.			1
4. Run console off-line tests and replace if necessary.			2
5. Replace I/O TTY PWA, slot K.			3
6. Call next level of support.			4

TABLE 6-2. LDCHK: LOADCHECK DDLT

ASSUMPTIONS:

Entry from sheet 1 of this table. An error has occurred.

CONDITIONS:	1	2	3	4
1. Does the screen display:				
a. 123456 or 12345 or 1234	Y	N		
b. 123		Y	N	
c. 12 or 1			Y	N
ACTIONS:				
1. Try another diskette if available	1	1	1	1
2. Remove micromemory PWA(s), if present, slot S and/or T, and replace if LOADCHECK then passes.	2	7	2	2
3. Go to table 6-8.	3			16
4. Go to table 6-9.			13	
5. Replace LMS ADDRESS PWA, slot W.		2	8	4
6. Replace LMS DATA PWA, slot V.		3	9	5
7. Replace PIF PWA, slot U.		15	11	12
8. Replace XFM PWA, slot R.		12	3	3
9. Replace C1 PWA, slot P.		5	5	8
10. Replace C2 PWA, slot N.		6	6	9
11. Replace ALU PWA, slot M.		11	4	7
12. Replace SMI PWA, slot L.		13	7	10
13. Replace I/O TTY PWA, slot K.		14	10	11
14. Call next level of support.	4	16	14	17
15. Replace LMA PWA in slot X.		4	12	6
16. Replace LMA PWA in slot Y.		8		13
17. Replace LMA PWA in slot Z.		9		14
18. Replace LMA PWA in slot AC.		10		15

TABLE 6-3. LMSIN: INSTRUCTION TEST DDLT

Sheet 1 of 2

ASSUMPTIONS:

1. For the remaining testing in tables 6-3 through 6-10 and all subsystems, all tests must be loaded from the same FDD that the Loadcheck DDLT is loaded from. This requires a full set of diagnostics on FDD diskette.

2. After xxxxx? is displayed, the following is entered at the keyboard (if xxxxx is not displayed, go to table 6-8).

   Type LMSIN
   Press CARRIAGE RETURN

   NOTE: xxxxx = test name

3. Observe the console display for the following conditions.

CONDITIONS:	1	2	3	4
1. Does the screen display: LMSIN EXECUTING LMSIN SUSPENDED BOT	Y		N	
2. Is the RUN light lit?			N	Y
3. Does the RUN light remain on for one minute?	Y	N		
ACTIONS:				
1. Enter the following at the keyboard: Type GO Press CARRIAGE RETURN	1			
2. Go to sheet 2 of this table.	2			
3. Try another diskette, if available.		1	1	
4. Go to table 6-9.		4	4	
5. Go to table 6-8.				X
6. Replace LMS ADDRESS PWA, slot W		2	2	
7. Replace LMA PWA in slot X		3	3	

TABLE 6-3. LMSIN: INSTRUCTION TEST DDLT (Contd)

ASSUMPTIONS:

1. System under test has a 1700 Emulator, version D.

CONDITIONS:	1	2	3
1. Is the following displayed on the screen after 10 seconds: LMSIN SECTION 0001 LMSIN SECTION 0002 MULTI-LEVEL INDIRECT SWITCH IS EXPECTED TO BE OFF (ON) LMSIN SECTION 0003 LMSIN SECTION 0004 LMSIN SECTION 0005 LMSIN SECTION 0006 LMSIN SECTION 0007 LMSIN SECTION 0008 LMSIN SECTION 0009 LMSIN SECTION 000A LMSIN SECTION 000B LMSIN COMPLETED 0001 PASSES LMSIN TERMINATED 0000 ERRORS	Y	N	
2. Is RUN indicator illuminated?		Y	N
ACTIONS:			
1. Go to next table.	X		
2. Replace XFM PWA, slot R.		1	
3. Replace ALU, slot M.		2	
4. Replace control 1, slot P.		3	
5. Replace control 2, slot N.		4	
6. Replace SMI, slot L.		5	
7. Replace I/O TTY, slot K.		6	
8. Replace LMS DATA PWA, slot V.		7	
9. Replace LMS ADDRESS PWA, slot W.		8	
10. Replace control panel PWA.			1
11. Go to table 6-9.		14	6
12. Replace LMA PWA in slot X.		9	2
13. Replace LMA PWA in slot Y.		10	3
14. Replace LMA PWA in slot Z.		11	4
15. Replace LMA PWA in slot AC.		12	5
16. Repeat actions 8 and 9 in other CPU.		13	

TABLE 6-4. LMSM1: MEMORY TEST DDLT

Sheet 1 of 5

ASSUMPTIONS:

1. This test must be run to all of available memory prior to running any subsequent tests.
2. The Loadcheck DDLT (table 6-2) and the LMSIN DDLT (table 6-3) run successfully in this processor.
3. Other CPU is MASTER CLEARED (if it exists).
4. After xxxxx? is displayed, the following is keyboard entered (if xxxxx? is not displayed, go to table 6-8):

   Type LMSM1
   Press CARRIAGE RETURN

5. Observe display for following conditions.

CONDITIONS:	1	2	3	4	5	6
1. Does the screen display: LMSM1 EXECUTING LMSM1 SUSPENDED BOT	Y	N				
2. Does the screen display: SET MULTI-LEVEL INDIRECT SWITCH OFF (ESC J40@ GO CR) LMSM1 SUSPENDED SELF			Y	N		
3. Does the screen display: ***INFORMATIVE MESSAGE - ESTIMATED TIME TO TEST nn K *** - xx HOURS yy MINUTES ***PER PASS TO CONTINUE TYPE (GO CR)					Y	N
ACTIONS:						
1. Go to table 6-8.		X				
2. At the keyboard: Type A, ___ (memory size parameter) Press CARRIAGE RETURN Type D, ___ (ending address parameter) Press CARRIAGE RETURN Type 4, 6874 (adds protect test) Press CARRIAGE RETURN Type GO Press CARRIAGE RETURN Return to condition 2 of this sheet. NOTE: The meaning of the above parameters is detailed in appendix A of this manual (under LMSM1).	1					

TABLE 6-4. LMSM1: MEMORY TEST DDLT (CONTD)

	Sheet 1 of 5 (Contd)					
ACTIONS (CONTD)	1	2	3	4	5	6
3. At the keyboard: Press ESC Type J40 @ GO Press CARRIAGE RETURN Return to condition 3 of this sheet.			1			
4. At the keyboard: Type GO Press CARRIAGE RETURN					1	
5. Replace LMS ADDRESS PWA in slot W (CPU I).				1		1
6. Replace LMS DATA PWA in slot V (CPU I).				2		2
7. Replace LMS DATA PWA in slot V (CPU II).						3
8. Replace LMS ADDRESS PWA in slot W (CPU II).						4
9. Call next level of support.				3		5
10. Go to sheet 2 of this table.					2	

TABLE 6-4. LMSM1: MEMORY TEST DDLT (Contd)

ASSUMPTIONS:

1. Entry from sheet 1 of this table.
2. Observe the screen for the following conditions.

CONDITIONS:	1	2	3
1. After several minutes delay (see note), does the screen display the following: LMSM1 INITIALIZATION SECTION INFORMATIVE MSG - TESTING XXXX, XXXX TO XXXX, XXXX LMSM1 SECTION 0001 INFORMATIVE MSG - TESTING XXXX, XXXX TO XXXX, XXXX LMSM1 SECTION 0002 INFORMATIVE MSG - TESTING XXXX, XXXX TO XXXX, XXXX LMSM1 SECTION 0003 INFORMATIVE MSG - TESTING XXXX, XXXX TO XXXX, XXXX LMSM1 SECTION 0005 INFORMATIVE MSG - TESTING XXXX, XXXX TO XXXX, XXXX LMSM1 SECTION 0006 INFORMATIVE MSG - TESTING XXXX, XXXX TO XXXX, XXXX LMSM1 SECTION 0008 INFORMATIVE MSG - TESTING XXXX, XXXX TO XXXX, XXXX LMSM1 SECTION 0007 SET PROTECT SWITCH (ESC J28@ GO CR) LMSM1 SUSPENDED SELF NOTE: INFORMATIVE MSG will be repeated, for each 16K word block in the test area, between each section msg. Test will take approximately 8.0 minutes for each 128K of memory in the test area.	Y	N	
2. Is an error code displayed?		Y	N

ACTIONS:			
1. At the keyboard: Press ESC Type J28@ GO Press CARRIAGE RETURN Go to sheet 3 of this table.	X X		
2. Go to sheet 4 of this table.		X	
3. Repeat Loadcheck DDLT (table 6-2), and repeat this test.			1
4. Call next level of support.			4
5. Replace LMS ADDRESS PWA in slot W (CPU I).			2
6. Replace LMS ADDRESS PWA in slot W (CPU II).			3

TABLE 6-4. LMSM1: MEMORY TEST DDLT (Contd)

Sheet 3 of 5

ASSUMPTIONS:

1. Entry is made from sheet 2 of this table.
2. Observe the screen for the following conditions.

CONDITIONS:	1	2	3	4	5
1. After a delay of approximately one minute, does the screen display the following: CLEAR PROTECT SWITCH (ESC J20@ GO CR) LMSM1 SUSPENDED SELF	Y				N
2. Does the screen display the following: LMSM1 SECTION 0004 INFORMATIVE MSG - TESTING xxxx, xxxx TO xxxx, xxxx		Y		N	
3. After delay of time specified in information message during test initiation does the screen display the following: LMSM1 COMPLETED 0001 PASSES SET MULTI-LEVEL INDIRECT SWITCH ON (ESC J42@ GO CR) LMSM1 SUSPENDED SELF		Y	N		
ACTIONS:					
1. At the keyboard: Press ESC Type J20@ GO Press CARRIAGE RETURN Return to condition 2 of this sheet.	X				
2. At the keyboard: Press ESC Type J42@ GO Press CARRIAGE RETURN		1			
3. Go to next table.		2			
4. Go to sheet 4 of this table			X	X	3
5. Replace LMS ADDRESS PWA, slot W					1
6. Replace LMS DATA PWA, slot V					2

TABLE 6-4. LMSM1: MEMORY TEST DDLT (Contd)

Sheet 4 of 5

ASSUMPTIONS:

1. Entry is made from sheet 2 or 3 of this table.
2. There is an error.

CONDITIONS:	1	2	3	4
Does the screen display the following action codes:				
1. 4180 or 4181 or 4182	Y	N		
2. 4190		Y	N	
3. 4100 through 4117			Y	N
ACTIONS:				
1. Repeat the Loadcheck DDLT (table 6-2) and repeat this test. Run parameters are incorrect.	1			
2. Replace LMS DATA PWA, slot V.	2	1		3
3. Replace LMS ADDRESS PWA, slot W.	3	2		2
4. Replace LMA PWA, slot X.	4	6		7
5. Replace transform PWA, slot R.	6	3		4
6. Replace control 2, slot N.	7	4		5
7. Replace control 1, slot P.	8	5		6
8. Replace I/O TTY, slot K.	9	7		8
9. Replace Panel I/F PWA (if present), slot U.	10	8		9
10. Replace SMI, slot L.	11	9		10
11. Replace ALU, slot M.	12	10		11
12. Run the Loadcheck DDLT (table 6-2) and this test in the alternate CPU if it exists. If it executes properly, rerun load check DDLT (table 6-2) and this test in the CPU and continue with action 3. If it does not execute properly in the alternate CPU, call next level of support.	5			1

TABLE 6-4. LMSM1: MEMORY TEST DDLT (Contd)

Sheet 4 of 5 (Contd)

ACTIONS (CONTD)	1	2	3	4
13. Call next level of support.	16	11		12
14. Replace LMA PWA, slot Y.	13			
15. Replace LMA PWA, slot Z.	14			
16. Replace LMA PWA, slot AC.	15			
17. Go to sheet 5 of this table.			X	

TABLE 6-4. LMSM1: MEMORY TEST DDLT (Contd)

Sheet 5 of 5

ASSUMPTIONS:

1. Entry from sheet 4 of this table.
2. There is an error with a displayed error code of 4100 through 4117.

CONDITIONS:	1	2	3	4	5	6	7	8	9
1. Is the displayed error code: 4100 or 4110 4101 or 4111 4102 or 4112 4103 or 4113 4104 or 4114 4105 or 4115 4106 or 4116 4107 or 4117	Y	N Y	N Y	N Y	N Y	N Y	N Y	N Y	N
ACTIONS:									
1. Replace LMA PWA in slot: X (CPUI) Y (CPUI) Z (CPUI) AC (CPUI)	1	1	1	1					
2. Replace LMA PWA in slot: X (CPUII) Y (CPUII) Z (CPUII) AC (CPUII)	4	4	4	4	1	1	1	1	
3. Replace LMS DATA PWA in slot V (CPU I).	2	2	2	2					
4. Replace LMS ADDRESS PWA in slot W (CPU I).	3	3	3	3					
5. Replace LMS DATA PWA in slot V (CPU II).					2	2	2	2	
6. Replace LMS ADDRESS PWA in slot W (CPU II).					3	3	3	3	
7. Call next level of support.	5	5	5	5	4	4	4	4	X

TABLE 6-5. LMSPG: LMS PAGE FILE TEST DDLT

ASSUMPTIONS:

1. If this test is not to be run, go to next table.
2. The LMSM1 DDLT (table 6-4) runs successfully in this processor.
3. Other processor in system is MASTER CLEARED (if it exists).
4. After xxxxx? is displayed, the following is entered at the keyboard (if xxxxx? is not displayed, go to table 6-8):

   Type LMSPG
   Press CARRIAGE RETURN

   NOTE: xxxxx = test name

5. Observe the console display for the following conditions.

CONDITIONS:	1	2	3	4
1. Does the screen display: LMSPG EXECUTING LMSPG SUSPENDED BOT	Y	N		
2. After approximately 15 seconds delay, does the screen display: LMSPG SECTION 0001 LMSPG SECTION 0002 LMSPG SECTION 0003 LMSPG SECTION 0004 LMSPG SECTION 0005 LMSPG COMPLETED 0001 PASSES LMSPG TERMINATED 0000 ERRORS			Y	N
ACTIONS:				
1. Go to table 6-8.		X		
2. At the keyboard, enter parameters for executing LMSPG: Fill in the spaces below with run parameter A as explained in LMSPG section of appendix A of this manual. Type A, ___ ___ ___ Press CARRIAGE RETURN Type GO Press CARRIAGE RETURN Go to condition 2 of this sheet.	X			
3. Go to sheet 2 of this table.				X
4. Go to next table.			X	

TABLE 6-5. LMSPG: LMS PAGE FILE TEST DDLT (Contd)

ASSUMPTIONS:

1. Entry from sheet 2 of this table. An error has occurred.

CONDITIONS:	1	2	3	4
1. Does the screen display an action code of:				
a. 4801 or 4802 or 4803 or 4804 or 4805	Y	N		
b. 4841 or 4842 or 4843		Y	N	
c. 4844 or 4845 or 4846 or 4847 or 4848 or 4849 or 484A or 484B			Y	N
ACTIONS:				
1. Repeat Loadcheck DDLT (table 6-2) and this test. Run parameter A (action code 4801 or 4802), B (action code 4803), C (action code 4804) is incorrect, or multilevel Indirect Switch not turned off (action code 4805).	1			
2. Replace LMS ADDRESS PWA, slot W.		2	1	
3. Replace LMS DATA PWA, slot V.		1	2	
4. Call next level of support.	2	3	3	X

TABLE 6-6. LMSDU: LMS DUAL-PROCESSOR (DUAL-CPU) DDLT

Sheet 1 of 7

ASSUMPTIONS:

1. If this is not a dual-CPU system, this test is not to be run. Go to table 6-7.
2. The LMSM1 DDLT (table 6-4), Memory Protect DDLT (table 6-5 of HMM 60475001), and Micromemory DDLT (table 6-6 of HMM 60475001) run successfully in both processors.
3. FDD drive for CPU 2 is empty.
4. CPU 1 is selected at the control panel.
5. FDD drive for CPU I contains LI VI LMS ODS diskette.

CONDITIONS:	1	2	3	4	5	6	7
1. Press MASTER CLEAR at the control panel. Press G at the keyboard. Does the screen display G?	Y						N
2. Select CPU II at the control panel. Press G at the keyboard. Does the screen display G?	Y					N	
3. Select CPU I at the control panel. Press G at the keyboard. Does the screen display H0000000X0?	Y				N		
4. Select CPU II at the control panel. Press G at the keyboard. Does the screen display H0000000X0?	Y			N			
5. Press DEADSTART at the control panel. Does DEADSTART indicator remain on?	Y		N				
6. Select CPU I and press MASTER CLEAR at the control panel. Select CPU II at the control panel. Is the DEADSTART indicator off?	Y	N					
ACTIONS:							
1. Go to sheet 2 of this table.	X						
2. Replace I/O-TTY, slot K (CPU 1)		1		1			
3. Replace panel multiplexer PWA				2	2		1
4. Replace I/O-TTY, slot K (CPU 1)			1		1		
5. Call next level of support.		2	2	3	3	3	2
6. Replace LMS DATA IN slot V (CPU II).						1	
7. Replace LMS DATA PWA in slot V (CPU I).						2	

TABLE 6-6. LMSDU: LMS DUAL-PROCESSOR (DUAL-CPU) DDLT (Contd)

ASSUMPTIONS:

1. CPU I is selected at the controlpanel and the MASTER CLEAR and RUN buttons are pressed.

2. After xxxxx? is displayed, the following is entered at the keyboard (if xxxxx? is not displayed, go to table 6-8):

   Type LMSDU
   Press CARRIAGE RETURN

   NOTE: xxxxx = test name

3. Observe the console display for the following conditions.

CONDITIONS:	1	2
1. Does the screen display: LMSDU EXECUTING LMSDU SYSPENDED BOT	Y	N
ACTIONS:		
1. Go to table 6-8.		X
2. At the keyboard: Type GO Press CARRIAGE RETURN	1	
3. Go to sheet 3 of this table	2	

TABLE 6-6. LMSDU: LMS DUAL-PROCESSOR (DUAL-CPU) DDLT (Contd)

ASSUMPTIONS:

1. Entry is made from sheet 2 of this table.

CONDITIONS:	1	2
1. Does the screen display: LMSDU SECTION 0001 LMSDU SECTION 0002 LMSDU SECTION 0003 LMSDU SECTION 0004 LMSDU SECTION 0005 LMSDU SECTION 0006 LMSDU SECTION 0007 LMSDU SECTION 0008 SET PROTECT SWITCHES - PNL SEL - ESC - J28@ - PNL SEL - J28@GO CR LMSDU SUSPENDED SELF	Y	N
ACTIONS:		
1. Press PANEL SELECT at the control panel.	1	
2. At the keyboard: Press ESC Type J28@	2	
3. Press PANEL SELECT at the control panel.	3	
4. At the keyboard: Type J28@GO Press CARRIAGE RETURN	4	
5. Go to sheet 4 of this table.	5	
6. Go to sheet 6 of this table		X

TABLE 6-6. LMSDU: LMS DUAL-PROCESSOR (DUAL-CPU) DDLT (Contd)

ASSUMPTIONS:

1. Entry is made from sheet 3 of this table.

CONDITIONS:	1	2
Does the screen display: RST PROTECT SWITCHES - PNL SEL - ESC - J20@ - PNL SEL - J20@GO CR DUCPU SUSPENDED SELF	Y	N

ACTIONS:		
1. Press PANEL SELECT at the control panel.	1	
2. At the keyboard: Press ESC Type J20@	2	
3. Press PANEL SELECT at the control panel.	3	
4. At the keyboard: Type J20@GO Press CARRIAGE RETURN	4	
5. Go to sheet 5 of this table.	5	
6. Go to sheet 6 of this table.		X

TABLE 6-6. LMSDU: LMS DUAL-PROCESSOR (DUAL-CPU) DDLT (Contd)

ASSUMPTIONS:

1. Entry is made from sheet 4 of this table.

CONDITIONS:	1	2
Does the screen display: LMSDU COMPLETED 0001 PASSES LMSDU TERMINATED 0000 ERRORS	Y	N
ACTIONS:		
1. Go to table 6-7.	X	
2. Repeat this test. Operator error is suspected.		1
3. Replace LMS DATA PWA, slot V (CPU II).		3
4. Call next level of support.		2

TABLE 6-6. LMSDU: LMS DUAL-PROCESSOR (DUAL-CPU) DDLT (Contd)

ASSUMPTIONS:

Entry is made from sheet 3 or 4 of this table. An error has occurred.

CONDITIONS:	1	2	3	4	5	6	7	8	9	10
Does the screen display:										
1. 0D01	Y	N								
2. 0D02		Y	N							
3. 0D03			Y	N						
4. 0D04				Y	N					
5. 0D05					Y	N				
6. 0D06						Y	N			
7. 0D07							Y	N		
8. 0D08								Y	N	
9. 0D09									Y	N
ACTIONS:										
1. Repeat this test from sheet 2. Wrong parameter.	1									
2. Replace SMI, slot L (CPU 1).		1	1	1	2	2		2	1	
3. Replace SMI, slot L (CPU 2).				2	1	1	1	1	2	
4. Replace panel multiplexer PWA.		3		3	3	3		4	4	
5. Examine connectors at each end of panel multiplexer (CPU 1) and panel multiplexer (CPU 2) cable assemblies for loose fit or broken wires.		2						3	3	

TABLE 6-6. LMSDU: LMS DUAL-PROCESSOR (DUAL-CPU) DDLT (Contd)

Sheet 6 of 7 (Contd)

ACTIONS (CONTD)	1	2	3	4	5	6	7	8	9	10
6. Examine connector over backpanel pins 52-76 at slot K, CPU 2, for loose fit or broken wire.			2							
7. Examine interrupt jumper plug over back panel pins 70-81 at slot L, CPU 2, for loose fit or broken wire.							2			
8. Go to sheet 7 of this table.										X
9. Call next level of support.	2	4	3	4	4	4	3	5	5	

TABLE 6-6. LMSDU: LMS DUAL-PROCESSOR (DUAL-CPU) DDLT (Contd)

ASSUMPTIONS:

1. Entry is made from sheet 6 of this table.

CONDITIONS:	1	2	3	4	5	6	7
Does the screen display:							
1. OD0A	Y	N					
2. OD0B		Y	N				
3. OD0C			Y	N			
4. OD0D				Y	N		
5. OD10, OD11, OD14, OD15, or OD16					Y	N	
6. OD12, OD13, OD17, OD18, or OD19						Y	N
ACTIONS:							
1. Replace LMS DATA PWA, CPU 1, slot V.	1	1	11	3	1	1	4
2. Replace LMS ADDRESS PWA, CPU 1, slot W.	2	2	12	4	2	2	5
3. Replace LMS DATA PWA, CPU 2, slot V.	3	11	1	1	3	3	1
4. Replace LMS ADDRESS PWA, CPU 2, slot W.	4	12	2	2	4	4	2
5. Replace transform PWA, CPU 1, slot R.	6	4					13
6. Replace control 1 PWA, CPU 1, slot P.	7	5					14
7. Replace control 2 PWA, CPU 1, slot N.	8	6					15
8. Replace ALU PWA, CPU 1, slot M.	9	7					16
9. Replace SMI PWA, CPU 1, slot L.	10	8					17
10. Replace I/O-TTY PWA, CPU 1, slot K.	11	9					18

TABLE 6-6. LMSDU: LMS DUAL-PROCESSOR (DUAL-CPU) DDLT (Contd)

Sheet 7 of 7 (Contd)

CONDITIONS:	1	2	3	4	5	6	7
11. Replace 2K RAM PWA, CPU 1, slot S.	12	10					19
12. Replace LMA PWA, CPU 1, slot X.	13	3	13	5			20
13. Replace LMA PWA, CPU 2, slot X.	5	13	3	13			3
14. Replace transform PWA, CPU 2, slot R.			4	6			6
15. Replace control 1, PWA, CPU 2, slot P.			5	7			7
16. Replace control 2 PWA, CPU 2, slot N.			6	8			8
17. Replace ALU PWA, CPU 2, slot M.			7	9			9
18. Replace SMI PWA, CPU 2, slot L.			8	10			10
19. Replace I/O-TTY PWA, CPU 2, slot K.			9	11			11
20. Replace 2K RAM PWA, CPU 2, slot S.			10	12			12
21. Replace cable at slot W, pin 77, CPU 2.					5		
22. Replace cable at slot W, pin 53, CPU 2.					6		
23. Replace cable at slot V, pin 240, CPU 2.					7		
24. Replace cable at slot W, pin 77, CPU 1.						5	
25. Replace cable at slot W, pin 53, CPU 1.						6	
26. Replace cable at slot V, pin 240, CPU 1.						7	
27. Call next level of support.	14	14	14	14	8	8	21

TABLE 6-7. CPU TEST COMPLETION DDLT

ASSUMPTIONS:

1. If this is a single-CPU system, run the Memory Protect DDLT (table 6-5 of HMM 60475001) and Micromemory DDLT (table 6-6 of HMM 60475001)(if micromemory is present).
2. If this is a dual-CPU system, return to table 6-1 of this manual and test the second CPU if this has not already been done.
3. When either of the above items has been accomplished, then basic CPU testing is complete.
4. Run the required Level I and/or II I/O Subsystem ODS tests according to system I/O configuration.

CONDITIONS:

None.

ACTIONS:

None.

TABLE 6-8. LOADER FAULT DDLT

ASSUMPTIONS:

1. The load device is a flexible disk, and a loader error has occurred.
2. The following is typed at the keyboard:

   Press ESC
   Type J11G KG

3. Observe the console display for any of the following conditions.

CONDITIONS:	1	2	3	4	5	6	7	8
Does the screen display:								
1. K7FED or K7FF1	Y	N						
2. K7FEF or K7FF3		Y	N					
3. K7FF5			Y	N				
4. K7FF7				Y	N			
5. K7FF9					Y	N		
6. K7FFB						Y	N	
7. K7FFD							Y	N
ACTIONS:								
1. Use another diskette (same program) if available.	1	1	1	1	1	1	1	
2. Verify that the equipment code for the flexible disk drive controller is correct (see the flexible disk drive subsystem manual).	2							
3. Replace I/O TTY, slot K.	3	2	3	4	4	3		
4. Replace flexible disk drive controller, slot E	4	3	2	2	2	2		
5. Replace ALU, slot M.	5	4	5	5	6	4		
6. Replace LMA PWA, slot X.	6		7	6			2	
7. Replace LMS DATA PWA, slot V.	7		8	7			3	
8. Replace LMS ADDRESS PWA, slot W.	8		9	8			4	
9. Replace SMI PWA, slot L	9	5	6	9	5	5		
10. Replace control 2 PWA, slot N.	10	7	10	10	8			
11. Replace control 1 PWA, slot P.	11	8	11	11	7			

TABLE 6-8. LOADER FAULT DDLT

Sheet 1 of 2 (Contd)

ACTIONS (CONTD)	1	2	3	4	5	6	7	8
12. Replace flexible disk drive (refer to the flexible disk drive subsystem manual).	15	6	4	3	3	6		
13. Go to sheet 2 of this table.								X
14. Call next level of support.	16	9	15	15	9	7	8	
15. Replace LMA PWA, slot Y.	12		12	12			5	
16. Replace LMA PWA, slot Z.	13		13	13			6	
17. Replace LMA PWA, slot AC.	14		14	14			7	

TABLE 6-8. LOADER FAULT DDLT (Contd)

ASSUMPTIONS:

1. The load device is a diskette.
2. The following is typed in at the keyboard:

   J14G KG

3. Observe the console display for any of the following conditions.

CONDITIONS: Does the screen display?	1	2	3	4	5	6	7	8
1. K0003	Y	N						
2. K0004		Y	N					
3. K0005			Y	N				
4. K0006				Y	N			
5. K0007					Y	N		
6. K0008 or K000A						Y	N	
7. K0009 or K000B							Y	N
ACTIONS:								
1. Use another diskette (same program) if available.	1	1	1	1	1	1	1	1
2. Replace LMS PWA, slot X.	2	6	7	6	5	6	8	6
3. Replace LMS DATA PWA, slot V.	3	7	8	7	6	7	9	7
4. Replace LMS ADDRESS PWA, slot W.	4	8	9	8	7	8	10	8
5. Replace flexible disk drive controller PWA, slot E.	5	2	2	2	2	2	2	2
6. Replace I/O-TTY controller PWA, slot K.	6	3	3	5	3	3	3	3
7. Replace ALU PWA, slot M.	7	4	4	3	4	5	4	4
8. Replace SMI PWA, slot L.	8	5	10	4	10	4	5	5
9. Replace control 2 PWA, slot N.	9	9	5	9	8	9	6	9
10. Replace control 1 PWA, slot P.	10	10	6	10	9	10	7	10
11. Replace flexible disk drive (refer to the flexible disk drive subsystem manual).	14	14			14		14	14

TABLE 6-8. LOADER FAULT DDLT (Contd)

Sheet 2 of 2 (Contd)

ACTIONS (CONTD)	1	2	3	4	5	6	7	8
12. Go to table 6-9.								15
13. Call next level of support.	15	15	14	14	15	14	15	
14. Replace LMA PWA, slot Y.	11	11	11	11	11	11	11	11
15. Replace LMA PWA, slot Z.	12	12	12	12	12	12	12	12
16. Replace LMA PWA, slot AC.	13	13	13	13	13	13	13	13

TABLE 6-9. I/O FAULT DDLT

Sheet 1 of 1

ASSUMPTIONS:

1. This DDLT satisfies I/O fault isolation and eliminates failure where peripheral controller faults occur in loading the computer.
2. After each replacement-type ACTION (below), return to the beginning of the table from which this table was entered.

CONDITIONS:	1	2	3
1. Is this the first entry into this table from a specific table and action?	Y	N	
2. Have all I/O controller PWAs (except for FDD controller PWA) been removed from CPU (slots AB through J)?		Y	N
ACTIONS:			
1. Replace FDD controller PWA.	X		
2. Call next level of support.		2	
3. Replace FDD (refer to FDD subsystem manual).		1	
4. Remove all I/O controller PWAs (except FDD controller PWA) from CPU (slots AB through J).			X

NOTE: Following Action 4, if calling table and action passes, one I/O controller at a time is reinstalled, (in the slot from which it was removed) and the calling table is repeated until the faulty controller is found and replaced.

PARTS DATA 7

This section contains the spare parts listing, assembly drawings and assembly parts lists associated with the LMS.

DWN	D.V. Styleson	3-9-81	CD DATA SYSTEMS LA JOLLA DIVISION CODE IDENT 09132	TITLE SPL - AT429-A 128K x 18 LARGE MEMORY ARRAY	PREFIX SP	DOCUMENT NO. 96890038	REV. 01
CHKD	[illegible]	6-1-81					
ENG	[illegible]	6-1-81					
MFG	A McBride	6-3-81		FIRST USED ON AT429-A		SHEET 1 OF 2	
APPR	[illegible]	11-15-81					

SHEET REVISION STATUS

REVISION RECORD

REV	ECO	DESCRIPTION	DRFT	DATE	CHKD	APP
01	DS19300	CC. B. PRE-RELEASED	J.V.W.	10-15-81	[illegible]	AW

NOTES:

DETACHED LISTS

AA2150 REV. 8/71

PRINTED IN U.S.A.

Figure 7-1. Large Memory Array Spare Parts List (Sheet 1 of 2)

CONTROL DATA CORPORATION

ASSEMBLY PARTS LIST

SHT. 2

ASSEMBLY NUMBER	REV	CL	DW SZ	ASSEMBLY DESCRIPTION	DESIGN SOURCE	FIRST USAGE	RELEASE DATE	CHANGE ORD. NUMBER	PROCESSING DATE	PAGE NUMBER
96890038	01	B	A	SPL-AT429-A LGE MEM 128 X 18	DS	AT429A	10/09/81		10/09/81	~~1~~ / ~~1~~ MF

FIND NUMBER	DW SZ	PART NUMBER	QUANTITY	UNIT MEAS.	PART DESCRIPTION	IN/OUT STATUS	CHANGE ORD. NUMBER	DATE EFFECTIVE	MAKE/BUY PART TYPE	PN NC	S OR N
1	D	96890036	1.00	PC	PWA-128K X 18 LGE MEM AR 81NKT	IN			AYM4		N

NUMBER OF LINE ITEMS = 1
HIGHEST FIND NUMBER = 1

AA2709-1 REV 2-80

PRINTED IN U.S.A.

SCMD MFG INFO

Figure 7-1. Large Memory Array Spare Parts List (Sheet 2 of 2)

DSGN			CONTROL DATA DATA SYSTEMS LA JOLLA DIVISION LA JOLLA CA 92037 CODE IDENT 09132	TITLE SPL - BA222-A LARGE MEMORY SUBSYSTEM	PREFIX SPL	DOCUMENT NO. 96750269	REV. 02
CHKD							
ENG							
MFG				FIRST U2ZD ON BA222-A		SHEET 1 OF 2	
APPR							

SHEET REVISION STATUS

REVISION RECORD

REV	ECO	DESCRIPTION	DRFT	DATE	CHKD	APP
01	DS19333	CR. B. PRE-RELEASED		10-16-81		AW
02	DS22167	FN 2 WAS 96890530	BR	1-28-82		AW

NOTES:

DETACHED LISTS

AA8100 REV. 8/71 PRINTED IN U.S.A.

CONTROL DATA CORPORATION

ASSEMBLY PARTS LIST

SH. 2

ASSEMBLY NUMBER	REV	CL	DW SZ	ASSEMBLY DESCRIPTION	DESIGN SOURCE	FIRST USAGE	RELEASE DATE	CHANGE ORD. NUMBER	PROCESSING DATE	PAGE NUMBER	
96750269	02	B	A	SPL-BA222-A LARGE MEM SUBSYS	DS	BA222A	10/09/81	022167	01/20/82	~~1~~ 1	MF

FIND NUMBER	DW SZ	PART NUMBER	QUANTITY	UNIT MEAS.	PART DESCRIPTION	IN/OUT STATUS	CHANGE ORD. NUMBER	DATE EFFECTIVE	MANF/SUPPLY PART TYPE	PN HC	S OS N
3	A	88919628	1.00	PC	BACKPANEL MOD-LRG MEM SUBSYS	IN			AYM4		N
1	D	96890042	1.00	PC	PWA-LRG MEM SUBSYS DATA 81NMT	IN			AYM4		N
~~2~~	~~D~~	~~96890530~~	~~1.00~~	~~PC~~	~~PWA-LRG MEM SUBSYS ADDR 81NST~~	~~OUT~~	~~022167~~	~~012082~~	~~AYM4~~		~~N~~
2	D	96890531	1.00	PC	PWA-LRG MEM SUBSYS ADDR 81PLT	IN	022167	012082			N

NUMBER OF LINE ITEMS = 4
HIGHEST FIND NUMBER = 3

AA8700-1 REV 8-68 PRINTED IN U.S.A.

SCMD MFG INFO

Figure 7-2. Large Memory Subsystem Spare Parts List

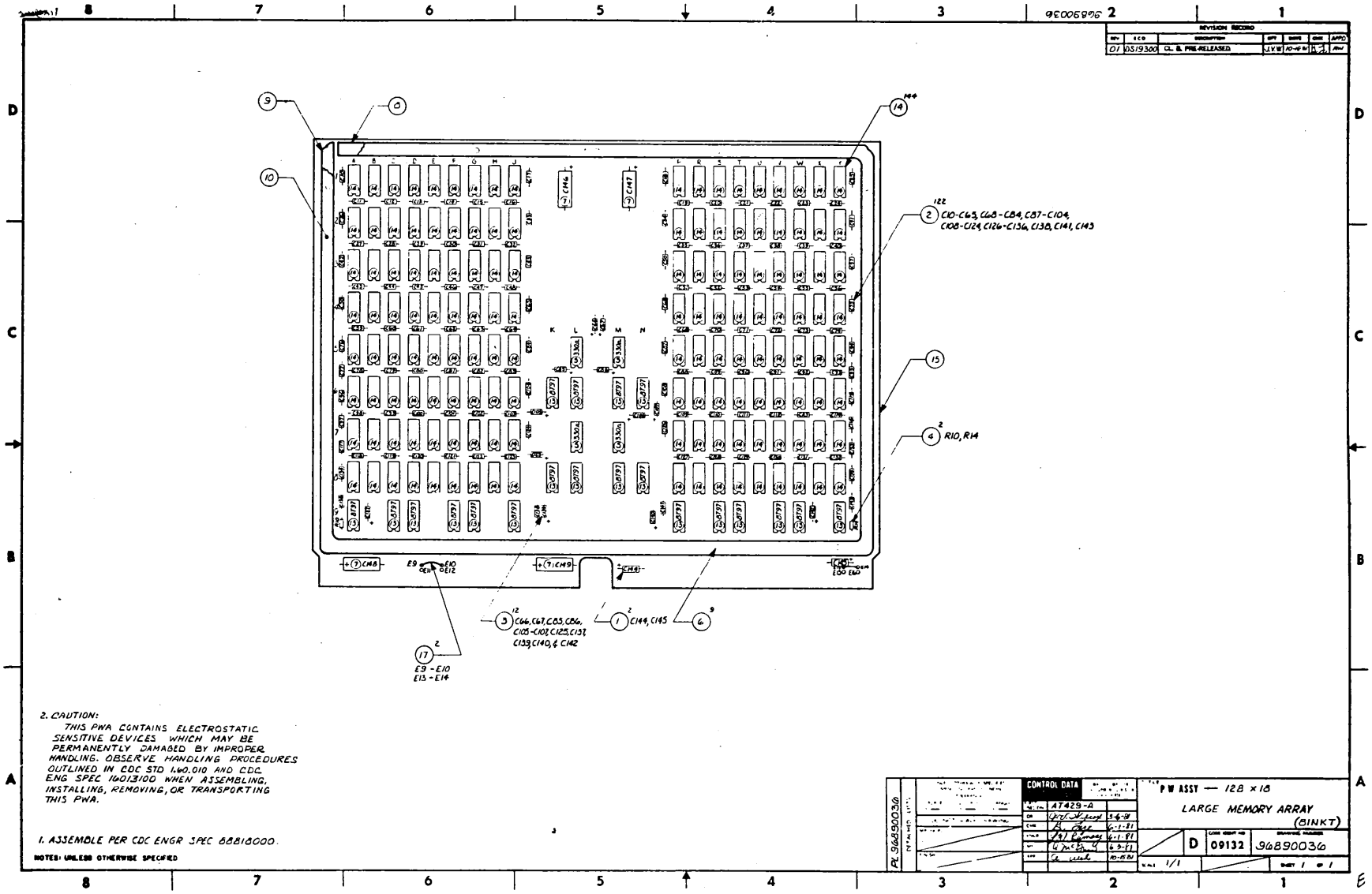


Figure 7-3. Large Memory Array Printed Wiring Assembly (Sheet 1 of 2)

CONTROL DATA CORPORATION

ASSEMBLY PARTS LIST

ASSEMBLY NUMBER	REV	CL	DW SZ	ASSEMBLY DESCRIPTION	DESIGN SOURCE	FIRST USAGE	RELEASE DATE	CHANGE ORD. NUMBER	PROCESSING DATE	PAGE NUMBER
96890036	01	B	D	PWA-128K X 18 LGE MEM AP 81NKT	DS	AT429A	10/09/81		10/09/81	1/ 1 MF

FIND NUMBER	DW SZ	PART NUMBER	QUANTITY	UNIT MEAS.	PART DESCRIPTION	IN/OUT STATUS	CHANGE ORD. NUMBER	DATE EFFECTIVE	MAKE/BUY PART TYPE	PN NC	S OR N
14	A	15153821	144.00	PC	DYNAMIC RAM 16,384 BITS	IN			PPP4		N
1	C	24504382	2.00	PC	CAP,FXD SOLID TANTALUM	IN			PPP4		N
4	C	24563041	2.00	PC	RES FXD COMP 1/8W 4700 OHMS	IN			PPP4		N
5	A	75009915	4.00	PC	RESISTOR PAC, 330 OHMS	IN			PPP4		N
2	A	84996743	122.00	PC	CAP,CER 100V .01 MF	IN			PPP4		N
6	A	88812400	9.00	PC	RIVET-SEMI-TUBULAR,BRS .312 LG	IN			PPP4		N
7	A	88880500	4.00	PC	CAP ELECT-ALUM 16VDC 100UF	IN			PPP4		N
8	C	88896600	1.00	PC	INSULATOR-CARD FRAME,UPPER	IN			PPP4		N
9	C	88896700	1.00	PC	INSULATOR-CARD FRAME,LOWER	IN			PPP4		N
10	C	88896800	1.00	PC	FRAME-CARD (CASTING)	IN			PPP4		N
3	A	88897800	12.00	PC	CAP-FIXED SOLID TANT,6VDC,12UF	IN			PPP4		N
13	A	88918800	20.00	PC	IC 8T97 TTL TRI-STATE HEX BFR	IN			PPP4		N
17	A	88945400	2.00	PC	JUMPER 24AWG TEF INSULATED .50	IN			PPP4		N
15	D	96735900	1.00	PC	PWB 32K X 18 MOS MEMORY ARRAY	IN			PPP4		N
11	C	96890037	REF	PC	LOGIC-128 X 18 LGE MEM 81NKT	IN					N

NUMBER OF LINE ITEMS = 15
HIGHEST FIND NUMBER = 17

AA2709-1 REV 2-80

PRINTED IN U.S.A.

SCMD MFG INFO

Figure 7-3. Large Memory Array Printed Wiring Assembly (Sheet 2 of 2)

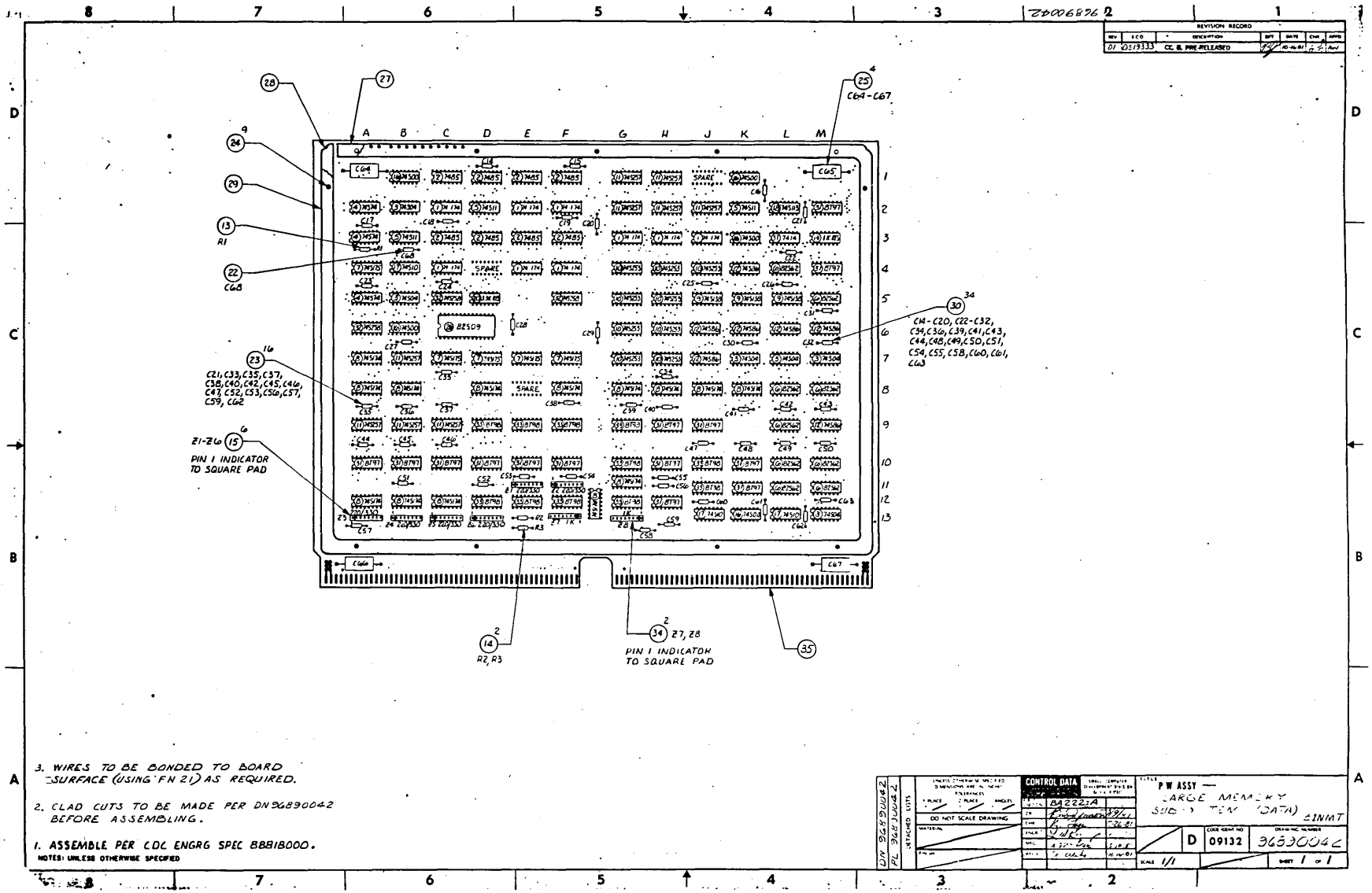


Figure 7-4. LMS Data Printed Wiring Assembly (Sheet 1 of 6)

CONTROL DATA CORPORATION

ASSEMBLY PARTS LIST

ASSEMBLY NUMBER	REV	CL	DW ST	ASSEMBLY DESCRIPTION	DESIGN SOURCE	FIRST USAGE	RELEASE DATE	CHANGE ORD. NUMBER	PROCESSING DATE	PAGE NUMBER
96890042	01	B	D	PWA-LRG MEM SUBSYS DATA 81NMT	DS	CBR 18	10/09/81		10/09/81	1/ 2 MF

FIND NUMBER	DW ST	PART NUMBER	QUANTITY	UNIT MEAS.	PART DESCRIPTION	IN/OUT STATUS	CHANGE ORD. NUMBER	DATE EFFECTIVE	MAKE/BUY PART TYPE	FN HC	S OR N
13	C	24500039	1.00	PC	RES FXD .25W 100 OHMS	IN			PPP4		N
14	C	24500063	3.00	PC	RES FXD .25W 1000 OHMS	IN			PPP4		N
40	C	24563700	1.00	IN	INSULATION SLEEVING HIGH TEMP	IN			PPP3		N
15	A	39452200	6.00	PC	RES NETWORK-VOLTAGE DIVIDR-SIP	IN			PPP4		N
38	C	52629949	56.00	IN	WIRE ELEC 30 GA WHITE	IN			PPP1		N
19	A	75009901	1.00	PC	RESISTOR MODULE 1K OHM 2 0/0	IN			PPP4		N
20	A	75009906	1.00	PC	RESISTOR MODULE 3.3K	IN			PPP4		N
22	A	84996719	1.00	PC	CAP,CER 100V 330 PF	IN			PPP4		N
39	A	84996721	1.00	PC	CAP,CER 100V 470 PF	IN			PPP4		N
23	A	84996743	16.00	PC	CAP,CER 100V .01 MF	IN			PPP4		N
24	A	88812400	9.00	PC	RIVET-SEMI-TUBULAR,BRS .312 LG	IN			PPP4		N
25	A	88880500	4.00	PC	CAP ELECT-ALUM 16VDC 100UF	IN			PPP4		N
2	A	88880900	8.00	PC	IC 7485 TTL 4-BIT MAGN COMPAR	IN			PPP4		N
1	A	88882800	9.00	PC	IC 74174 TTL HEX D F/F W/CLEAR	IN			PPP4		N
11	A	88883300	9.00	PC	IC 74S257 QUAD 2-L D SEL/MUX	IN			PPP4		N
3	A	88883700	6.00	PC	IC 74S04 SCHOTTKY TTL HEX INV	IN			PPP4		N
5	A	88884100	3.00	PC	IC 74S11 SCHOTTKY TTL TPL 3-I	IN			PPP4		N
17	A	88884200	3.00	PC	IC 74S10 SCHOTTKY TTL 3-I NAND	IN			PPP4		N
16	A	88884500	5.00	PC	IC 74S00 SCHOTTKY TTL QUAD 2-I	IN			PPP4		N
18	A	88884600	1.00	PC	IC 74S113 SCHOTTKY TTL DUAL	IN			PPP4		N
9	A	88885100	3.00	PC	IC 74S138 TTL DECODER/MUX	IN			PPP4		N
7	A	88885200	5.00	PC	IC 74S175 TTL QUAD D F/F W/CLR	IN			PPP4		N
26	A	88886200	1.00	PC	IC 82S09 TTL 576 BIT RAM	IN			PPP4		N
37	A	88896100	1.00	PC	IC 7414 TTL HEX SCHMITT TRIGER	IN			PPP4		N
27	C	88896600	1.00	PC	INSULATOR-CARD FRAME,UPPER	IN			PPP4		N
28	C	88896700	1.00	PC	INSULATOR-CARD FRAME,LOWER	IN			PPP4		N
29	C	88896800	1.00	PC	FRAME-CARD (CASTING)	IN			PPP4		N
30	A	88897800	34.00	PC	CAP-FIXED SOLID TANT,6VDC,12UF	IN			PPP4		N
31	A	88918800	14.00	PC	IC 8T97 TTL TRI-STATE HEX BFR	IN			PPP4		N
32	A	88920800	3.00	PC	IC 74S258 TTL QUAD 1-L SEL/MUX	IN			PPP4		N
33	A	88920900	11.00	PC	IC 8T98 TTL TRI-ST HEX BFR/INV	IN			PPP4		N
6	A	88922800	9.00	PC	IC 93S62/82S62 TTL PAR/GEN/CHK	IN			PPP4		N
12	A	88922900	7.00	PC	IC 74S86 QUAD 2IN EXCLUSIVE OR	IN			PPP4		N
4	A	88923000	3.00	PC	IC 74S74 TTL DUAL D EDGE F/F	IN			PPP4		N
8	A	88923100	14.00	PC	IC 74S174 TTL HEX D FLIP/FLOP	IN			PPP4		N
10	A	88924200	9.00	PC	IC 74S253 TTL TRI-ST D 4/1 MUX	IN			PPP4		N

AA3706-1 REV 3-80 PRINTED IN U.S.A.

SCMD MFG INFO

CONTROL DATA CORPORATION

ASSEMBLY PARTS LIST

ASSEMBLY NUMBER	REV	CL	DW ST	ASSEMBLY DESCRIPTION	DESIGN SOURCE	FIRST USAGE	RELEASE DATE	CHANGE ORD. NUMBER	PROCESSING DATE	PAGE NUMBER
96890042	01	B	D	PWA-LRG MEM SUBSYS DATA 81NMT	DS	CBR 18	10/09/81		10/09/81	2/ 2 MF

FIND NUMBER	DW ST	PART NUMBER	QUANTITY	UNIT MEAS.	PART DESCRIPTION	IN/OUT STATUS	CHANGE ORD. NUMBER	DATE EFFECTIVE	MAKE/BUY PART TYPE	FN HC	S OR N
34	A	88924706	2.00	PC	RES NETWORK-SIP,8 PIN,1.K OHM	IN			PPP4		N
21	A	96720225	0.00	OZ	ADHESIVE-PERMA BOND 610	IN			RFE4		N
35	D	96740700	1.00	PC	PWB-MOS MEM INTFC (DATA)	IN			PPP4		N
36	C	96890043	REF	PC	LOGIC-LRG MEM SUDSYS DAT 81NMT	IN					N

NUMBER OF LINE ITEMS = 40
HIGHEST FIND NUMBER = 40

AA3706-1 REV 3-80 PRINTED IN U.S.A.

SCMD MFG INFO

Figure 7-4. LMS Data Printed Wiring Assembly (Sheet 2 of 6)

DWN			DATA SYSTEMS LA JOLLA DIVISION LA JOLLA CA. 92037	TITLE PWA - LARGE MEMORY SUBSYSTEM (DATA) BINMT	PREFIX DN	DOCUMENT NO. 96890042	REV. 01
CHKD							
ENG							
MFG				FIRST USED ON BA222-A		SHEET 1 OF 7	
APPR			CODE IDENT 09132				

SHEET REVISION STATUS

REVISION RECORD

REV	ECO	DESCRIPTION	DRFT	DATE	CHKD	APP
01	DS19333	CL. B PRE-RELEASED		12/2/81		AN

NOTES:
1. WIRE ADDITIONS USE F/N 38.

DETACHED LISTS

AA8166 REV. 8/71 PRINTED IN U.S.A.

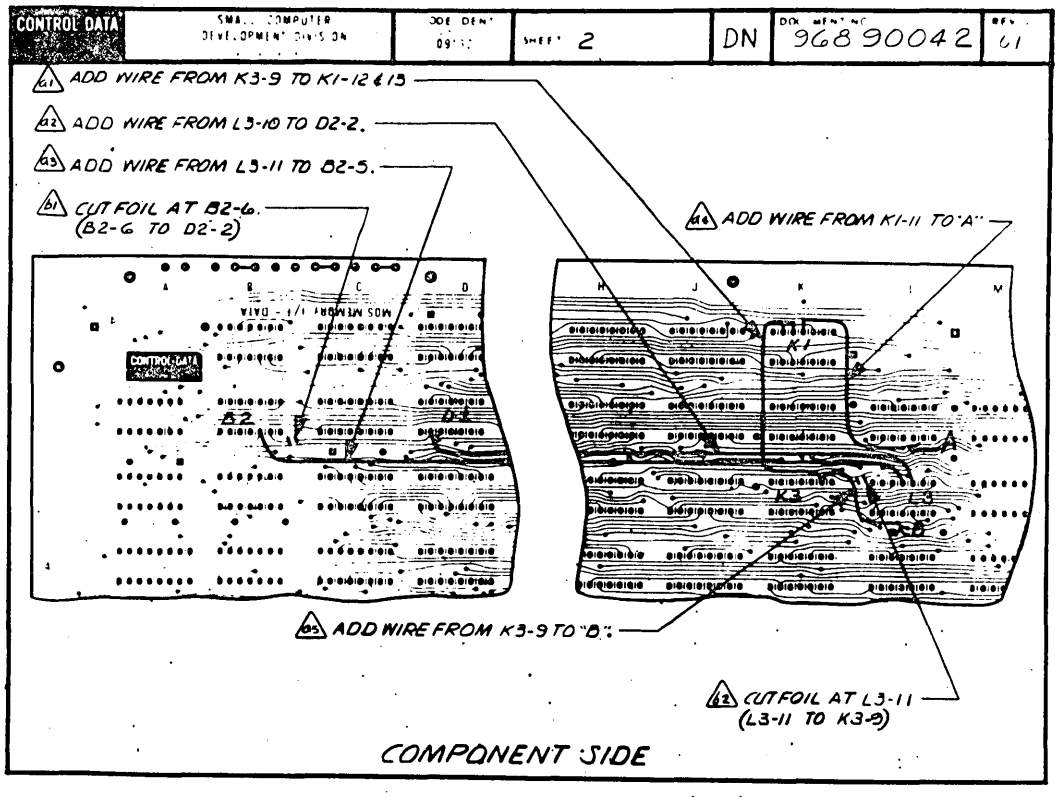
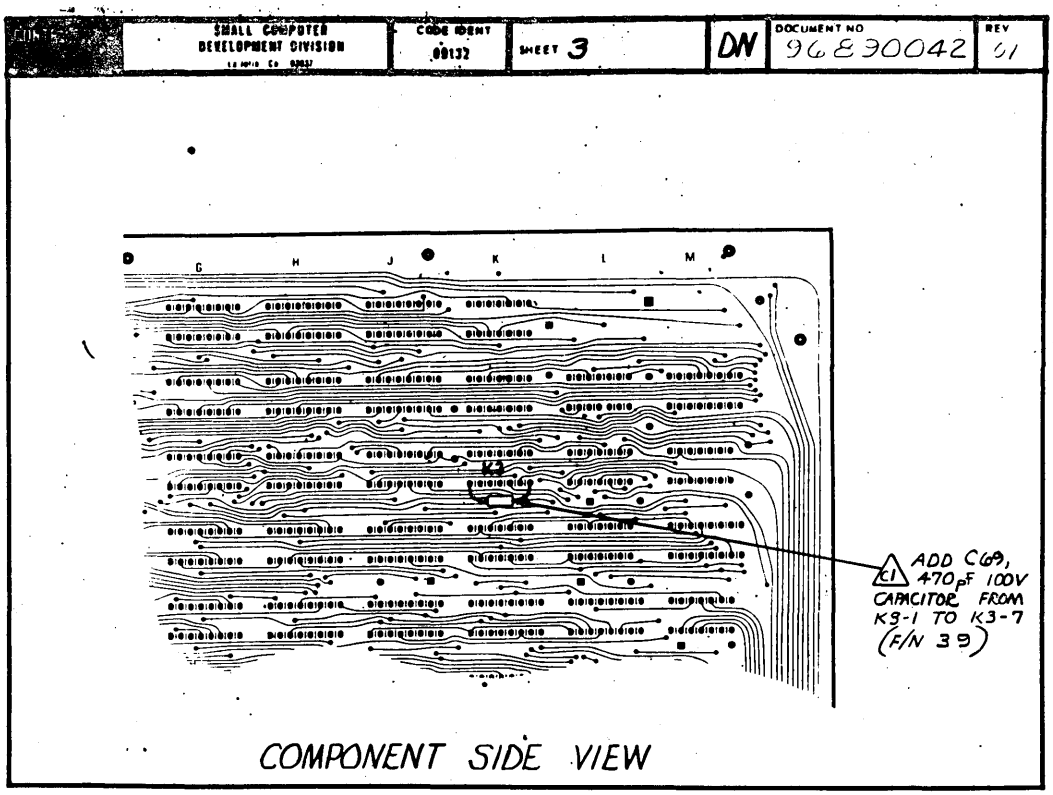


Figure 7-4. LMS Data Printed Wiring Assembly (Sheet 3 of 6)



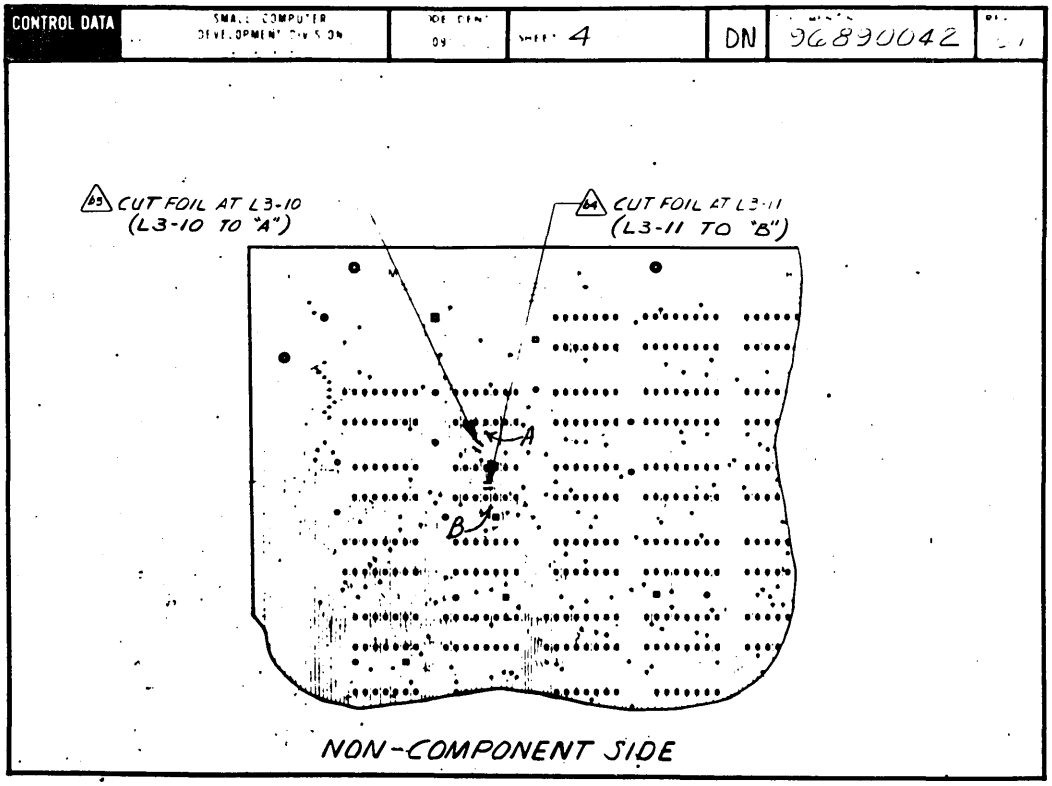
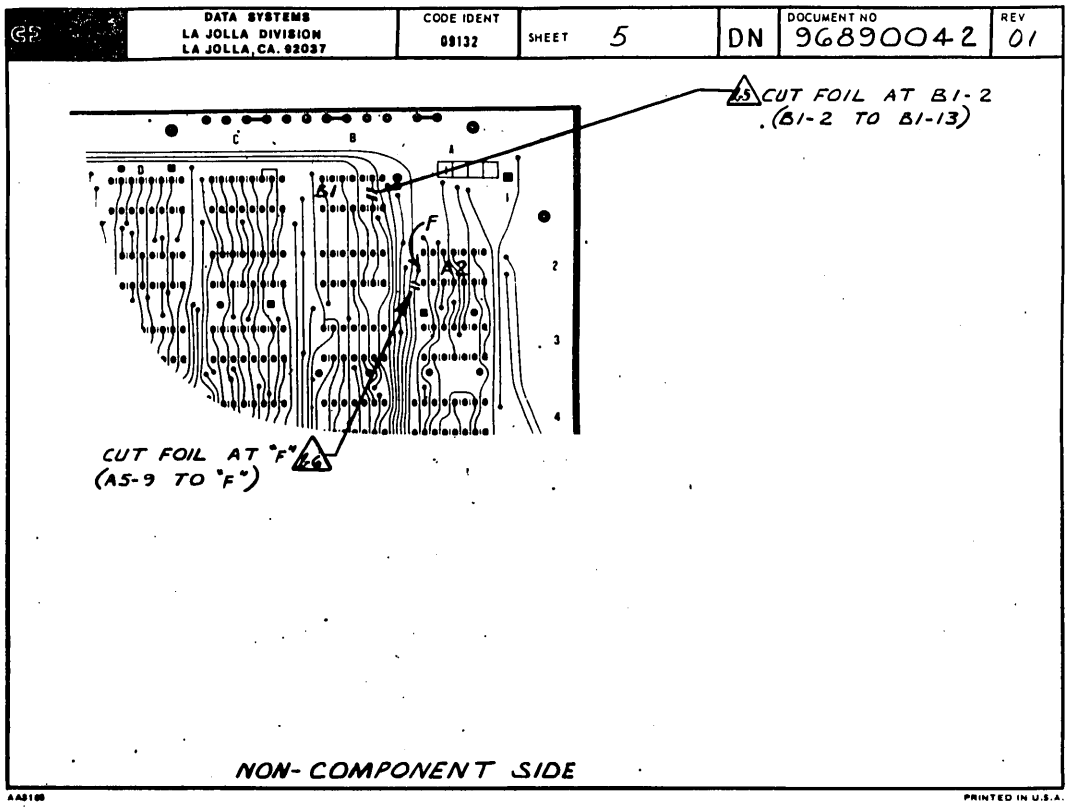


Figure 7-4. LMS Data Printed Wiring Assembly (Sheet 4 of 6)



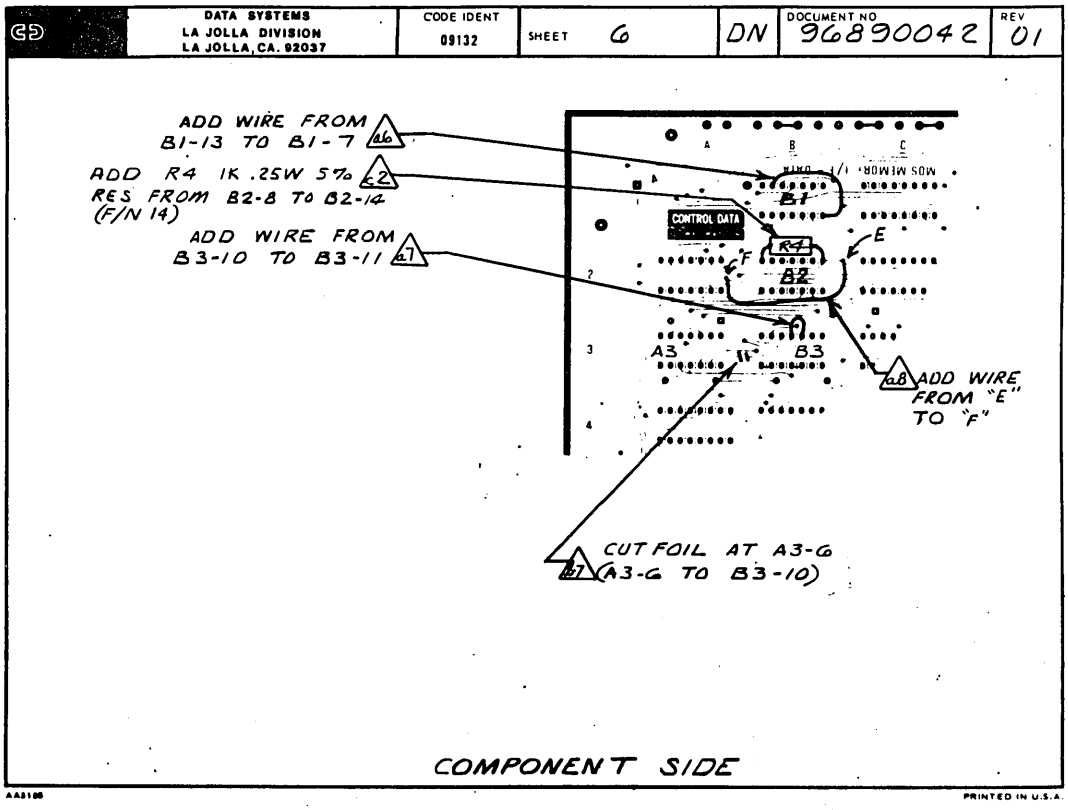


Figure 7-4. LMS Data Printed Wiring Assembly (Sheet 5 of 6)

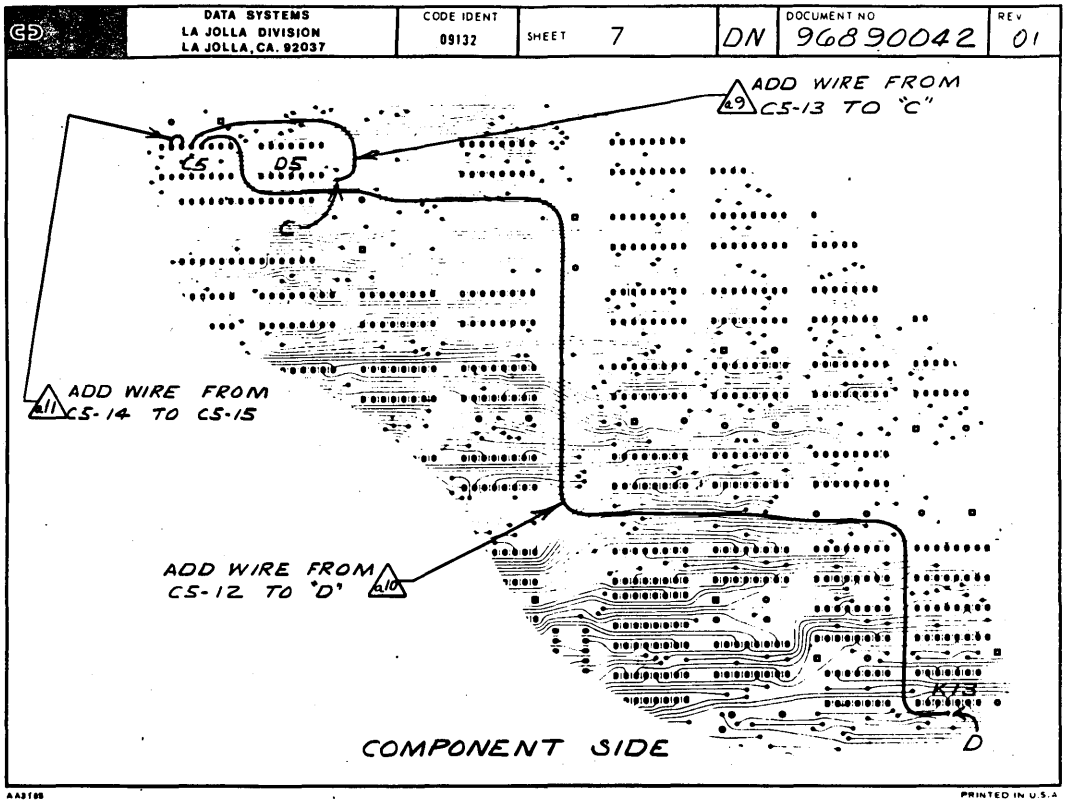
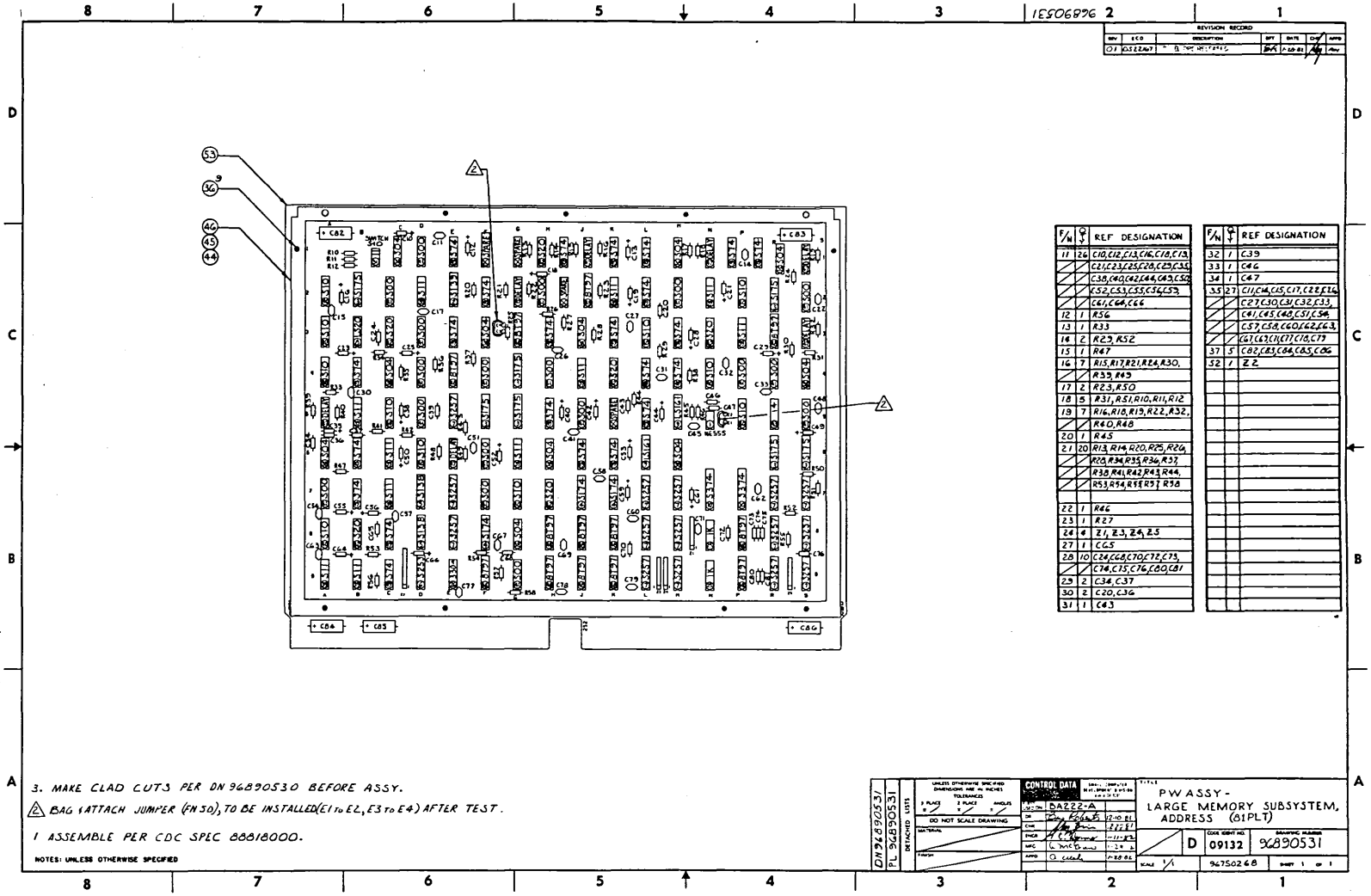


Figure 7-4. LMS Data Printed Wiring Assembly (Sheet 6 of 6)



F/N	Q	REF DESIGNATION
11	26	C10,C12,C13,C16,C18,C19,
		C21,C23,C25,C28,C29,C35,
		C38,C40,C42,C44,C45,C50,
		C52,C53,C55,C56,C57,
		C61,C64,C66
12	1	R56
13	1	R33
14	2	R29,R52
15	1	R47
16	7	R15,R17,R21,R24,R30,
		R35,R49
17	2	R23,R50
18	5	R31,R51,R10,R11,R12
19	7	R16,R18,R19,R22,R32,
		R40,R48
20	1	R45
21	20	R13,R14,R20,R25,R26,
		R28,R34,R35,R36,R37,
		R38,R41,R42,R43,R44,
		R53,R54,R55,R57,R58
22	1	R46
23	1	R27
24	4	Z1,Z3,Z4,Z5
27	1	CG5
28	10	C24,C62,C70,C72,C73,
		C74,C75,C76,C80,C81
29	2	C34,C37
30	2	C20,C36
31	1	C43

F/N	Q	REF DESIGNATION
32	1	C39
33	1	C46
34	1	C47
35	27	C11,C14,C15,C17,C22,C24,
		C27,C30,C31,C32,C33,
		C41,C45,C48,C51,C54,
		C57,C58,C60,C62,C63,
		C67,C69,C71,C77,C78,C79
37	5	C02,C03,C04,C05,C06
52	1	Z2

Figure 7-5. LMS Address Printed Wiring Assembly (Sheet 1 of 5)

CONTROL DATA CORPORATION

ASSEMBLY PARTS LIST

ASSEMBLY NUMBER	REV	CL	DW SZ	ASSEMBLY DESCRIPTION	DESIGN SOURCE	FIRST USAGE	RELEASE DATE	CHANGE ORD. NUMBER	PROCESSING DATE	PAGE NUMBER
96890531	01	B	D	PWA-LRG MEM SUBSYS ADDR A1PLT	DS	CBR 18	01/20/82		01/20/82	1/ 2 MF

FIND NUMBER	DW SZ	PART NUMBER	QUANTITY	UNIT MEAS.	PART DESCRIPTION	IN/OUT STATUS	CHANGE ORD. NUMBER	DATE EFFECTIVE	MANUF/BUY PART TYPE	PN NO	S OR N
1	A	15112100	1.00	PC	IC NE555	IN			PPP4		N
2	A	15117200	1.00	PC	1-C TTL DCRD2	IN			PPP4		N
3	A	15132000	7.00	PC	I.C. 74S175	IN			PPP4		N
4	A	15138800	3.00	PC	IC 74S174J	IN			PPP4		N
8	A	15145900	1.00	PC	IC TTL DUAL 4INPUT NAND 74LS20	IN			PPP4		N
9	A	15146800	2.00	PC	I C TYPE 74LS161J	IN			PPP4		N
10	A	15163310	2.00	PC	IC 74S374 OCTAL LATCH TTL	IN			PPP4		N
11	A	15164102	26.00	PC	CAP-TANT 12UF 6VDC 10PCT	IN			PPP4		N
12	C	24500027	1.00	PC	RES FXD .25W 33 OHMS	IN			PPP4		N
13	C	24500033	1.00	PC	RES FXD .25W 56 OHMS	IN			PPP4		N
14	C	24500039	2.00	PC	RES FXD .25W 100 OHMS	IN			PPP4		N
15	C	24500040	1.00	PC	RES FXD .25W 110 OHMS	IN			PPP4		N
16	C	24500043	7.00	PC	RES FXD .25W 150 OHMS	IN			PPP4		N
17	C	24500047	2.00	PC	RES FXD .25W 220 OHMS	IN			PPP4		N
18	C	24500051	5.00	PC	RES FXD .25W 330 OHMS	IN			PPP4		N
19	C	24500052	7.00	PC	RES FXD .25W 360 OHMS	IN			PPP4		N
20	C	24500061	1.00	PC	RES FXD .25W 820 OHMS	IN			PPP4		N
21	C	24500063	20.00	PC	RES FXD .25W 1000 OHMS	IN			PPP4		N
22	C	24500066	1.00	PC	RES FXD .25W 1300 OHMS	IN			PPP4		N
23	C	24500087	1.00	PC	RES FXD .25W 10000 OHMS	IN			PPP4		N
24	A	39452200	4.00	PC	RES NETWORK-VOLTAGE DIVIDR-SIP	IN			PPP4		N
55	C	52629949	15.00	IN	WIRE ELEC 30 GA WHITE	IN			PPP1		N
25	A	75009901	2.00	PC	RESISTOR MODULE 1K OHM 2 0/0	IN			PPP4		N
26	A	75009915	1.00	PC	RESISTOR PAC, 330 OHMS	IN			PPP4		N
27	A	75808515	1.00	PC	CAP-CERAMIC,100V,150PF	IN			PPP4		N
28	A	75808517	10.00	PC	CAP-CERAMIC,100V,220PF	IN			PPP4		N
29	A	75808519	2.00	PC	CAP-CERAMIC,100V,330PF	IN			PPP4		N
30	A	75808521	2.00	PC	CAP-CERAMIC,100V,470PF	IN			PPP4		N
31	A	75808523	1.00	PC	CAP-FXD,CER,100V,10PCT,680PF	IN			PPP4		N
32	A	75808524	1.00	PC	CAP-CERAMIC,100V,820PF	IN			PPP4		N
33	A	75808534	1.00	PC	CAP-CERAMIC,100V,5600PF	IN			PPP4		N
34	A	75808537	1.00	PC	CAP-CERAMIC,100V,.01MF	IN			PPP4		N
35	A	77830574	27.00	PC	CAP-CERAMIC,0.1UF,50V	IN			PPP4		N
36	A	88812400	9.00	PC	RIVET-SEMI-TUBULAR,BRS .312 LG	IN			PPP4		N
37	A	88880500	5.00	PC	CAP ELECT-ALUM 16VDC 100UF	IN			PPP4		N
7	A	88883300	15.00	PC	IC 74S257 QUAD 2-L D SEL/MUX	IN			PPP4		N

SCMD MFG INFO

PRINTED IN U.S.A.

CONTROL DATA CORPORATION

ASSEMBLY PARTS LIST

ASSEMBLY NUMBER	REV	CL	DW SZ	ASSEMBLY DESCRIPTION	DESIGN SOURCE	FIRST USAGE	RELEASE DATE	CHANGE ORD. NUMBER	PROCESSING DATE	PAGE NUMBER
96890531	01	B	D	PWA-LRG MEM SUBSYS ADDP A1PLT	DS	CBR 18	01/20/82		01/20/82	2/ 2 MF

FIND NUMBER	DW SZ	PART NUMBER	QUANTITY	UNIT MEAS.	PART DESCRIPTION	IN/OUT STATUS	CHANGE ORD. NUMBER	DATE EFFECTIVE	MANUF/BUY PART TYPE	PN NO	S OR N
38	A	88883700	11.00	PC	IC 74S04 SCHOTTKY TTL HEX INV	IN			PPP4		N
39	A	88884100	11.00	PC	IC 74S11 SCHOTTKY TTL TPL 3-I	IN			PPP4		N
40	A	88884200	11.00	PC	IC 74S10 SCHOTTKY TTL 3-I NAND	IN			PPP4		N
41	A	88884500	18.00	PC	IC 74S00 SCHOTTKY TTL QUAD 2-I	IN			PPP4		N
5	A	88885100	2.00	PC	IC 74S138 TTL DECODER/MUX	IN			PPP4		N
42	A	88885300	6.00	PC	IC 74S20 SCHOTTKY TTL DUAL 4-I	IN			PPP4		N
43	A	88896100	1.00	PC	IC 7414 TTL HEX SCHMITT TRIGER	IN			PPP4		N
44	C	88896600	1.00	PC	INSULATOR-CARD FRAME,UPPER	IN			PPP4		N
45	C	88896700	1.00	PC	INSULATOR-CARD FRAME,LOWER	IN			PPP4		N
46	C	88896800	1.00	PC	FRAME-CARD (CASTING)	IN			PPP4		N
47	A	88897300	7.00	PC	DELAY LINE-100 NSEC	IN			PPP4		N
48	A	88918800	13.00	PC	IC 8T97 TTL TRI-STATE HEX BFR	IN			PPP4		N
49	A	88923000	23.00	PC	IC 74S74 TTL DUAL D EDGE F/F	IN			PPP4		N
6	A	88924200	1.00	PC	IC 74S253 TTL TRI-ST D 4/1 MUX	IN			PPP4		N
50	A	88945400	2.00	PC	JUMPER 24AWG TEF INSULATED .50	IN			AYM4		N
51	A	94398002	1.00	PC	SWITCH,SLIDE-DIP-PCB,4 POS	IN			PPP4		N
56	A	96720225	0.25	OZ	ADHESIVE-PERMA BOND 610	IN			RFE4		N
52	A	96752401	1.00	PC	RES NETWORK,10 PIN SIP 330 OHM	IN			PPP4		N
53	D	96890230	1.00	PC	PWA-LARGE MEMORY SUBSYS ADDER	IN			PPP4		N
54	C	96890831	REF	PC	LOGIC-LG MEM SUBSYS ADDP A1PLT	IN					N

NUMBER OF LINE ITEMS = 56
HIGHEST FIND NUMBER = 56

SCMD MFG INFO

PRINTED IN U.S.A.

Figure 7-5. LMS Address Printed Wiring Assembly (Sheet 2 of 5)

DRN				TITLE PW ASSY- LARGE MEMORY SUBSYSTEM ADDRESS (81PLT)	PREFIX	DOCUMENT NO.	REV.
CHKD			DATA SYSTEMS LA JOLLA DIVISION LA JOLLA CA 92037		DN	96890531	01
ENG				FIRST USED ON		SHEET 1 OF 5	
MFG			CODE IDENT 09132				
APPR							

SHEET REVISION STATUS

REVISION RECORD

REV	ECO	DESCRIPTION	DRFT	DATE	CHKD	APP
01	DS22167	CL. B. PRE-RELEASED	BR	1/28/82		AN

NOTES:

DETACHED LISTS

AA8100 REV. 8/71 PRINTED IN U.S.A.

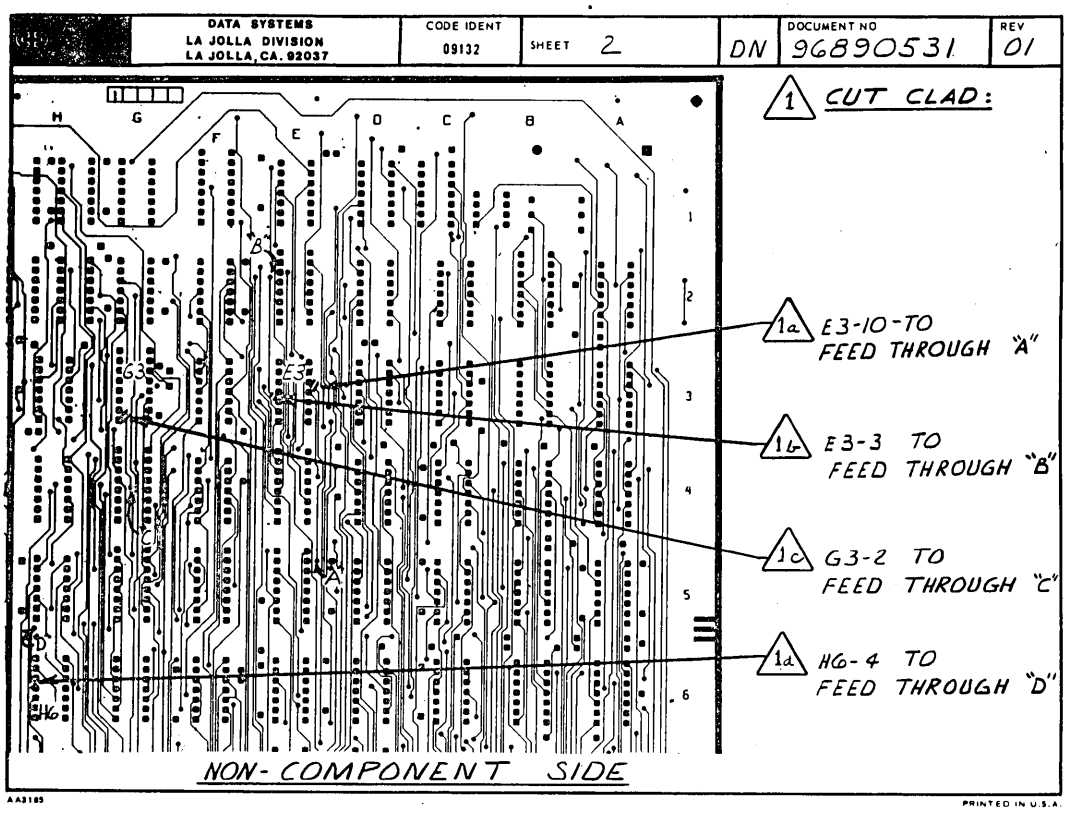
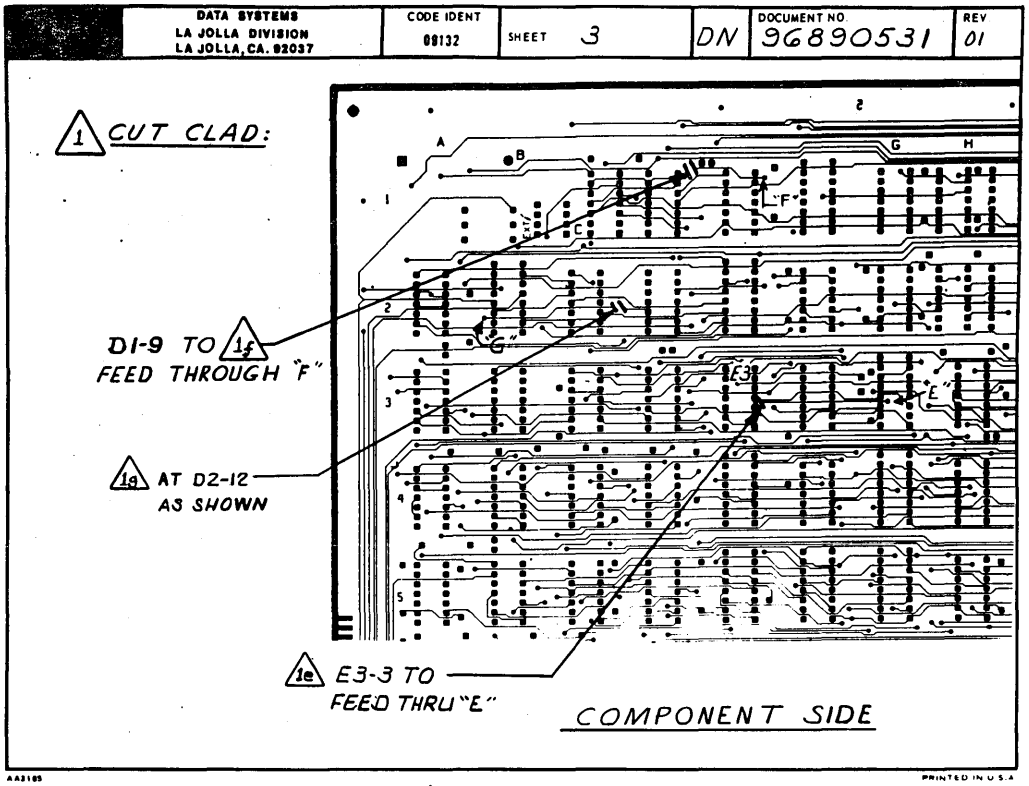


Figure 7-5. LMS Address Printed Wiring Assembly (Sheet 3 of 5)



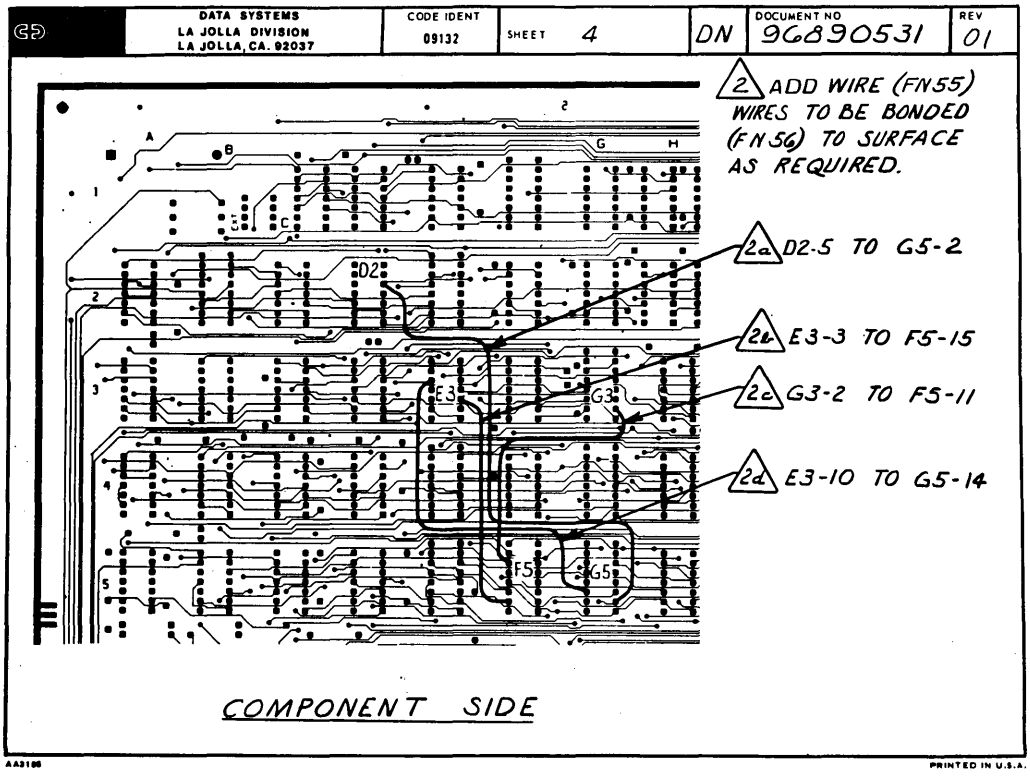


Figure 7-5. LMS Address Printed Wiring Assembly (Sheet 4 of 5)

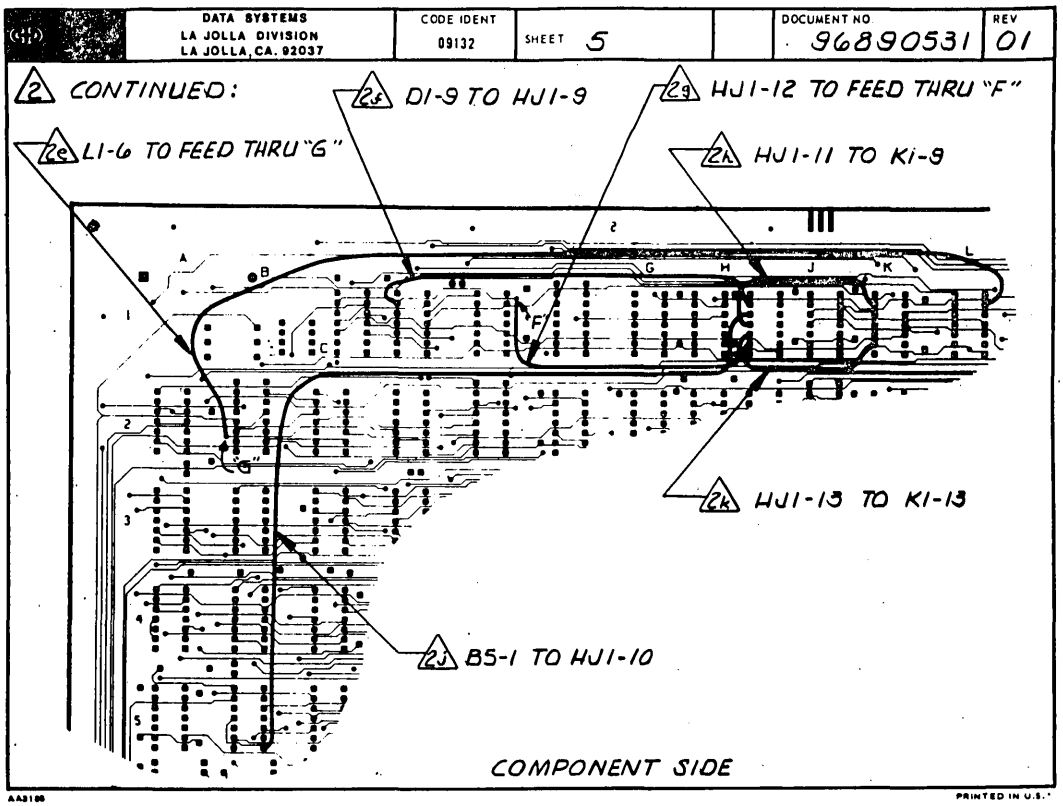


Figure 7-5. LMS Address Printed Wiring Assembly (Sheet 5 of 5)

LMS DIAGNOSTIC REFERENCE DATA A

This appendix describes the diagnostics used on the large memory subsystem (LMS). The diagnostics consist of four new Level I tests, three of which are derived from existing ODS diagnostic tests.

CAUTION

Refer to appendix B for any specific requirements and restrictions associated with the use of ODS tests when they are run with the LMS.

INTRODUCTION

This series of tests detects failures and provides the necessary action codes so the diagnostic decision logic tables (DDLTs) in section 6 of this manual may isolate failures associated with the following printed wiring assemblies:

Printed Wiring Assembly	Part/Equipment Number
Large memory subsystem, address PWA	96890530
Large memory subsystem, data PWA	96890042
Large memory array PWA	96890036

TEST NMEMONICS

The test mnemonics are as follows:

Instruction test	LMSIN
Large memory subsystem memory test	LMSM1
Large memory subsystem page file test	LMSPG
Large memory subsystem dual-CPU test	LMSDU

TEST PARAMETERS

The parameters for these tests follow the standard ODS parameter list structure. Device-specific parameters are in words 7_{16} through 15_{16}. The run parameters are displayed and modified according to the procedures described for Level I test in the ODS reference manual.

MEMORY REQUIREMENTS

All of the LMS tests are designed to execute within 64K words.

EXECUTION TIMES

The execution times of the LMS tests are as follows:

LMSIN	20 seconds
LMSM1	9 minutes (testing 128K, excluding section 4)
LMSPG	20 seconds
LMSDU	20 seconds

LMSIN - INSTRUCTION TEST

TEST IDENTIFICATION

40

PURPOSE

The instruction test validates the macroinstruction set, both basic and enhanced instructions.

PARAMETERS

The run parameters for LMSIN are listed in table A-1.

TABLE A-1. LMSIN RUN PARAMETERS

Word	Initial Value (Hex)	Definition
TESTID	0040	Test identification number
PASCNT	0000	Test pass count
ERRCNT	0000	Test error count
1	0080	Control word
2	0001	Repeat count
3	1234	Sections selected
4	5678	Sections selected
5	9ABO	Sections selected
6	0000	Sections selected
7	0000	Equipment address (not used)
8	0000	Interrupt lines (not used)
9	0000	Logical unit (not used)
A	FFFF	Multi-level indirect switch
B	0001	Emulator (1 = version D or later)

Parameter B defines the version of the emulator used. A one signifies that version D or a later version of the emulator is being used.

ORGANIZATION

LMSIN is identical to ODS test MPINS with the following exceptions:

- The test ID is 40.
- There are differences in section B.

The MPINS test requirements and section descriptions in the ODS reference manual listed in the preface apply to LMSIN, with the exception of the parameter list and section B.

INITILIZATION

The initialization section is not selectable. It is executed once before executing the test.

This section initializes the low memory area that is to be used by the test sections. A table of a sliding one bits and a pointer to that table are placed into memory, starting at location 7. The memory layout after execution of the initialization section is shown in the ODS reference manual.

Section B – Version D Miscellaneous Instructions

Section B executes only if parameter B is non-zero. The section tests the operation of the miscellaneous instructions added in the version D emulator of the CYBER 18. The page registers are tested with 9-bit data (FF_{16}). The list of instructions is given in section 5 of the ODS reference manual.

MESSAGES

Action Codes

The LMSIN action codes are the same as those listed and described for MPINS in the ODS reference manual, with the exception of the test ID.

Error Codes

The error codes for LMSIN are the same as those listed in the ODS reference manual for MPINS except that the test ID is 40 (for example 40XX).

Action Messages

Refer to MPINS in the ODS reference manual for the messages applicable to LMSIN.

RESTRICTIONS

LMSIN uses an area of low memory from location 7_{16} to location $1D_{16}$ regardless of where it is loaded in memory, thereby destroying any data or code that resides there.

LMSM1 – LMS MEMORY TEST (MEUM1)

TEST IDENTIFICATION

41

PURPOSE

LMSM1 verifies that every word of the memory and the addressing line are functioning properly. It loads, stores, writes, and compares special patterns that are determined by the sections selected to be executed. It verifies that the parity logic bit is storing data correctly and functioning properly.

PARAMETERS

The run parameters for LMSM1 are listed in table A-2.

Parameter A defines the number of large memory array (LMA) printed wiring assemblies in the system. A maximum of 8 LMAs are permitted (4 LMAs in the internal CPU and 4 LMAs in the external CPU).

Parameter B defines the most significant four bits of the starting address for memory testing.

Parameter C defines the least significant sixteen bits of the starting address for the memory testing.

Parameter D defines the most significant four bits of the stopping address for the memory test.

Parameter E defines the least significant sixteen bits of the stopping address for the memory test.

Parameter F is of the format 000X, where X determines if memory sizing is to be performed (X=0 for memory sizing and X=1 for no sizing).

Parameter 10 is the number of times to extend random number pattern without reinitializing in sections 4 and 5. Each number will raise the test time for these sections by the factor selected.

Parameter 11 defines the first special data pattern to be used in the test.

Parameter 12 defines the second special data pattern to be used in the test.

ORGANIZATION

The test LMSM1 consists of the following sections:

Initialization

Section 1 - Addressing

Section 2 - Basic patterns

Section 3 - Marching ones and zeros

Section 4 - Moving inversion

Section 5 - Random data at random address

TABLE A-2. LMSM1 RUN PARAMETERS

Word	Initial Value (Hex)	Description
TESTID	0041	Test identification number
PASCNT	0000	Test pass count
ERRCNT	0000	Test error count
1	0088	Control word
2	0001	Repeat count
3	1235	Sections selected
4	6840	Sections selected
5	0000	Sections selected
6	0000	Sections selected
7	0000	Equipment address (not used)
8	0000	Interrupt lines (not used)
9	0000	Logical unit (not used)
A	0001	Number of 128K memory arrays (Format = 00RL) L (bits 0-3) number in local bank (1-4) R (bits 4-7) number in remote bank (1-4)
B	0000	Start address MSBs (0-$F)
C	0000	Start address LSBs (0-$FFFF)
D	0001	Stop address MSBs (0-$F)
E	FFFF	Stop address LSBs (0-$FFFF)
F	0000	Size memory (0 = Size, 1 = Do not size)
10	0001	Random pattern extender
11	CDCD	Special data pattern 1
12	4545	Special data pattern 2

Section 6 - Parity bit

Section 7 - Protect bit

Section 8 - Selectable pattern termination

Initialization

This section will size memory, if this operation is not prohibited by the operator, and compare this value with the number of 128K word blocks of memory (parameter word A). If these values do not match, an error is reported. This error may be due to an incorrect parameter in word A or a hardware error. The test will not continue until these values agree or until the sizing of memory is prohibited. For the test to execute, the following conditions must be met:

1. The start address must be less than or equal to the stop address, which must be less than or equal to the last address based on parameter entry A.

2. The start and stop address cannot contain any non-contiguous memory.

When these conditions are met, the section moves the loader and parameter list into the lowest 16K of memory to avoid destruction by the test.

Section 1, Addressing

To verify that locations are addressed correctly, a unique number is written into the last address of every 2K increment within the test area specified by the operator, except where the program is loaded (0-16K). After the writing, the locations are read and verified. The next part of the test is to write sequentially in every address from the stop address to the start address. In the same sequence all the locations are read and verified. The test area is then written in reverse order, stop to start. The addresses are then read and verified in the same order. The pattern written in each location is the AB register used for addressing the cell.

Section 2, Basic Patterns

This section verifies that every memory cell can be written to and read from using basic data patterns. The patterns used are 0000/FFFF and 5555/AAAA.

Section 3, Marching Ones and Zeros

This section verifies that writing into one memory cell does not affect the contents of another cell. This detects decoding problems. This section tests one 16K chip at a time. If the start address specified is not on the chip boundary, the part of the 16K chip from the start address to the upper boundary is tested before going to the next chip.

Section 4, Moving Inversion

This section checks the memory with a moving inversion pattern. Each 16K chip is tested separately.

Memory is backgrounded with zeros. In addition to both true and complimented data, moving inversion uses a sliding data pattern and marches both forward and backward by ones, twos, and fours, expanding in this manner up to a 16K chip address space.

Section 5, Random Data at Random Address

Each 16K chip will be tested using a random data pattern. A location will first be loaded with a random number. That location will then be tested with a random number pattern. When 160K addresses have been selected, the next 16K chip will be tested. If parameter 10 is set to a value greater than one, additional groups of patterns will be used without the random number generator reinitializing itself.

Section 6, Parity Bit Test

This section verifies the parity generator logic and the parity bit storage capability.

There are two subsections in this section.

Subsection 1

This subsection checks the parity generator logic by following the steps below for one location within the program:

1. Write a pattern.
2. Read the 16-bit data and verify correct parity.
3. Read the ECC status and verify the parity and protect bits.

These three steps are completed for a sliding ones pattern as well as the following patterns:

Data

0001
0000
FFFF
5317
D7B7
D3B7
D337

Subsection 2

This subsection writes a marching ones and zeros pattern on the parity bit chip. The parity bit is then checked by reading the ECC status that contains the parity bit. If parity error occurred, the status will indicate the error.

Subsection 2 tests the 16K chips one at a time. If the start address specified is not on a 16K boundary, the part of the 16K chip from the start address to the next 16K boundary is tested before going on to the next chip.

To set the parity bit (=1), the 16-bit data used is 8 and the protect bit is set. To clear the parity bit (=0), the 16 bit data used is 9 and the protect bit is set.

The following steps are completed for each 16K area:

1. Write 0 to all parity bits.
2. Set ADDR to the start address.
3. Read the ECC status of ADDR and check the parity.
4. Write 1 to the parity bit.
5. Increment ADDR by 1.
6. If ADDR is less than or equal to the stop address, go to step 3.
7. Set ADDR to the stop address.
8. Read the ECC status and check the parity.
9. Write 0 to the parity bit of ADDR.
10. Decrement ADDR by 1.
11. If ADDR is greater than or equal to the start address, go to step 8.
12. Read the ECC status of all locations from the start to the stop address and check the parity.

Section 7, Protect Bit (Local Bank Only)

This section verifies that the protect bits can be set and cleared. The pattern used is the marching ones and zeros. Since the protect bit cannot be cleared in full write mode, this section runs with the protect system enabled.

Only the protect bits in the local bank are tested due to not being able to read status of the protect bit in the remote bank.

This section consists of two subsections.

Subsection 1

This subsection runs without the protect system and verifies that all the protect bits are set to one.

Subsection 2

This subsection first outputs a message to enable the protect system.

The following steps are completed for each 16K area:

1. Set ADDR to the stop address.
2. Write 0 to the protect bit.
3. Read the ECC status and verify the protect bit is clear.
4. Write 1 to the protect bit of ADDR.
5. Increment ADDR by 1.
6. If ADDR is less than or equal to the stop address, go to step 2.
7. Read the ECC status of all locations from the start to the stop address and verify the protect bit is set.

Section 8, Selectable Pattern

Two 16-bit data patterns are defined in the parameter list. This section writes the first pattern in the first location, second pattern in the second location, and so forth until every location in the test area has been filled. The data is then read and verified.

After each read/write operation, parity error is checked.

MESSAGES

Action Codes

The action codes listed in table A-3 are issued when the test encounters an error condition.

TABLE A-3. LMSM1 ACTION CODES

Action Code	English Text	Description
4100-4107	ADRING ERR	Addressing error in 128K block 00-07
4110-4117	PARITY ERR	Parity error in 128K block 00-07
4110-4117	BAD DATA	Data error in 128K block 00-07
4110-4117	PARITY BIT	Parity bit error in 128K block 00-07
4110-4117	PROTECT BIT	protect bit error in 128K block 00-07
4180	PARAM ERR-LCL	Number of 128 K blocks specified in parameter A is different from value computed by program in local bank.
4181	PARAM ERR-RMT	Number of 128K blocks specified in parameter A is different from value computed by program in remote bank.
4182	PARAM ERR-ADR	Start and/or stop address is not correct. Either start is greater than stop or an address outside existing memory.
4190	MPF ERR	Memory page file error.

Error Mnemonics

ACV A	Actual value
ADDR	Address that was in error
EXV A	Expected value
PREG	Page register number in error

Action or Informative Messages

The following message can occur in section 1 of LMSM1:

ADDRESSING ERROR IN 64K-BLOCK xx

In all sections an informational message is displayed to indicate test progression:

INFORMATIVE MSG NOW TESTING xxxx,xxxx TO yyyy,yyyy

RESTRICTIONS

The LMSM1 restrictions are as follows:

- The test cannot be restarted from an error stop.
- The test cannot be restarted while executing a section.
- There is no subsection to repeat execution on a subsection. Looping on a location can be achieved by setting the start address equal to the stop address and repeating a section.

LMSPG - LMS PAGE FILE TEST

TEST IDENTIFICATION

48

PURPOSE

LMSPG verifies the operation of the memory page file (MPF).

PARAMETERS

The run parameters for LMSPG are listed in table A-4.

Parameter A defines the number of contiguous blocks of 128K words starting from location 0. A maximum of 8 blocks are allowed.

Parameter B defines the address of the first memory page file location to be tested in sections 2 and 5.

Parameter C defines the last memory page file location to be tested in sections 2 and 5. The maximum value of this parameter is $003F.

Parameter D defines the first data pattern (9 bits) to be used in section 2.

Parameter E defines the second data pattern (9 bits) to be used in section 2.

TABLE A-4. LMSPG RUN PARAMETERS

Word	Initial Value (Hex)	Description
TESTID	0048	Test identification number
PASCNT	0000	Test pass count
ERRCNT	0000	Test error count
1	0088	Control word
2	0001	Repeat count
3	1234	Sections selected
4	0000	Sections selected
5	0000	Sections selected
6	0000	Sections selected
7	0000	Equipment address (not used)
8	0000	Interrupt lines (not used)
9	0000	Logical unit (not used)
A	0001	Number of 128K memory arrays (Format = 00RL) L (bits 0-3) = number in local bank (1-4) R (bits 4-7) = number in remote bank (1-4)
B	0000	First MPF address tested
C	003F	Last MPF address tested
D	0155	Special data pattern 1 (9 bits)
E	00AA	Special data pattern 2 (9 bits)

ORGANIZATION

LMSPG consists of an initialization and a termination section and five test sections.

Initialization

This section verifies the parameters in the parameter list to insure that they are in the proper range. If the parameters are not correct, the test will not proceed. Just before completing, this section moves the loader and parameter list down next to the test so they will not be destroyed. The operator is required to reset the multi-level switch (MLI).

Section 1, Page File Addressing

This section verifies the addressing capability of the memory page file (MPF). It is composed of two subsections that load each location (register) in the page file with its respective address. Each location is read and verified to check the addressing capability and reread to check that the first read did not cause the original data to be lost.

Subsection 1

Subsection 1 writes the page file in ascending order (0 to $3F_{16}$).

Subsection 2

Subsection 2 writes the page file in descending order ($3F_{16}$ to 0).

Section 2, Page File Data

This section verifies the data retention properties of the memory page file and the control register. Two of the subsections write the data pattern specified in parameter words D and E into an area of consecutive locations in the page file whose boundaries are described by parameters B and C. Each location is read and verified to contain the proper data pattern.

Subsection 1

Subsection 1 writes all zeros in all MPF locations specified in the parameter list. It reads them and verifies the data. It then repeats the operation using all ones as data.

Subsection 2

Subsection 2 stores the data pattern specified by parameter D, then E, in consecutive locations in the page file. This alternating pattern is spread throughout the specified area.

Subsection 3

Subsection 3 is identical to subsection 2, except that the alternating pattern is parameter E followed by parameter D.

Subsection 4

Subsection 4 writes, reads, and verifies the upper page address bits in the control register. The data of zeros, ones, and the patterns specified in the parameter list are used.

Section 3, Memory Addressing

This section accesses all memory pages in page mode, to verify that all pages are addressed correctly. The section consists of two subsections.

Subsection 1

A single location in the page file is used to access all pages of available memory above the 8K words reserved for the test. The last word of each page is written with its corresponding page number. Each data location is then read and verified to contain its corresponding page number. Absolute location $3FFF_{16}$ is also read in absolute mode to verify that the lower 11 bits of the logical address are working in page mode (the expected value is 7). The test is then relocated to the upper 8K of the first 16K memory area to check the addressing of pages 0 through 3.

Subsection 2

All locations in the page file are used to address a single memory address ($3FFF_{16}$). For each page file location to be tested, the memory location is set to zero, and the page file location is set to page 7. (Page file locations 0 through 3 and 20_{16} through 23_{16} are set to 0 through 3 respectively, since this is the test program area.) The processor is placed in page mode and page 7 is written into the memory location. The processor is then placed in absolute mode, and the memory location is read and verified to contain 7.

Section 4, Multi-Level Indirect Test

This section verifies the operation of the multilevel indirect logic in page mode.

Subsection 1

A single-level indirect operation is attempted in page mode, and the retrieved data is verified to be as expected.

Subsection 2

A multi-level indirect addressing operation is attempted in page mode, with the multi-level indirect (MLI) addressing bit off. The retrieved data is verified to be as expected. (A message instructs the operator to manipulate the bit in the functional control register.)

Subsection 3

A multi-level indirect addressing operation is attempted in page mode, with the multi-level indirect addressing bit on. The retrieved data is verified to be as expected. (A message instructs the operator to manipulate the bit in the function control register.)

Termination

This section returns the loader and the parameter list to their original locations.

MESSAGES

Action Codes

The action codes listed in table A-5 are issued when the test encounters an error condition.

TABLE A-5. LMSPG ACTION CODES

Action Code	English Text	Description
4801	LOCAL MEM SIZE	Local memory size parameter error.
4802	REMOTE MEM SIZE	Remote memory size parameter error.
4803	FIRST MPF	First MPF parameter error.
4804	LAST MPF	Last MPF parameter error.
4805	MLI SW NOT OFF	Requested MLI off, but still on.
4841	MPF#MPF ADRS-AS	MPF location # MPF address (ascending).
4842	MPF#MPF ADRS-DS	MPF location # MPF address (descending).
4843	MPF PATTERNS	Alternating patterns to MPF data error.
4844	MPF-LOCAL PAGE	Write page number to last word memory page.
4845	MEM LOC $3FFF	Last word page 7 # 7.
4846	MPF-LOCAL PAGE	Write page number to last word memory page 0-3.
4847	ALL MPFS-PAGE 7	Write page mode, read absolute all pages.
4848	IA-SW OFF-MSBOFF	Data fetched 1-level indirect in page not correct.
4849	IA-SW OFF-MSBON	Data fetched multi-level indirect in page not correct.
484A	IA-SW ON-MSBON	Data fetched multi-level indirect MLI off not correct.
484B	MPF-EXTNL PAGE	Same as 4844 except to external page.

Error Mnemonics

ADAT	Actual data read
EDAT	Expected data value
FMPF	First memory page file register number
LMPF	Last memory page file register number
LMEM	Local memory size parameter
MEM	Memory size parameter
MPF	Memory page file register number
RMEM	Remote memory size parameter

Action Messages

There are no messages for LMSPG.

RESTRICTIONS

In a dual-CPU configuration, the CPU under test should be halted to prevent inteference which may cause errors.

A halt indicating a parity error may occur if memory parity errors are not cleared first by running the memory test LMSM.

LMSDU - LMS DUAL-CPU TEST

TEST IDENTIFICATION

49

PURPOSE

LMSDU tests the control between the 255x Network Processing Unit dual processor (dual-central processor unit) and the dual-MP† configurations. LMSDU also checks interrupts and memory interactions and the external memory protect fault mechanism.

NOTE

Throughout the LMSDU text, the multiplexer processor is referred to as CPU II and the base processor as CPU I. On a dual-MP system panel, these are marked as I (CPU B) and II (CPU A), respectively.

PARAMETERS

Table A-6 lists the run parameters. Parameters 8, A, and B indicate the macro (000x) or micro (00x0) interrupt lines in the respective processors. For the dual-MP system, parameter 8 refers to the SM115 local/SM208 external interrupt line in CPU I; parameter A refers to the SM206 local/external interrupt line in CPU II; and parameter B refers to the SM205 local interrupt line in CPU II. The default parameters, as shown, are for the dual-MP system. The parameters 8 and A only, indicate the SM205 local/external interrupt line for CPU I (0003), and the SM206 local/external micro interrupt line for CPU II (0030), respectively.

Parameter word C indicates to the test whether the configuration is the dual-MP system, for which the parameter is zero, or the 255x dual-processor system, for which the parameter must be set nonzero.

Parameter words D and E specify the micromemory page into which the microcontrol program is to be stored for CPU I and CPU II, respectively (normal range, 4 through F). The range of each parameter is verified in the initialization section, and section 1 uses all of the stated parameters.

TABLE A-6. LMSDU RUN PARAMETERS

Word	Initial Value (Hex)	Description
TESTID	0049	ID
PASCNT	0000	Pass count
ERRCNT	0000	Error count
1	0088	Control word
2	0001	Repeat count
3	1234	Sections selected
4	5678	Sections selected
5	0000	Sections selected
6	0000	Sections selected
7	0000	Equipment address (not used)
8	0005	Interrupt lines for CPU I
9	0000	Logical unit (not used)
A	0005	Interrupt line for CPU II
B	0003	SM205 interrupt line for CPU II (dual MPs only)
C	0000	Configuration flag (0 = dual MPs)
D	0007	CPU I micromemory page
E	0008	CPU II micromemory page

NOTE

The parameter values are checked only in the initialization section. If any parameter is changed during execution, the test should be restarted to ensure accurate results.

ORGANIZATION

The dual-processor test is organized into eight sections, plus an initialization and a termination section.

The test program is always loaded into CPU I, which becomes the master processor. It controls CPU II action by moving relevant test routines into and setting flags in CPU II memory. CPU I also detects error conditions in CPU II interrogating flags in CPU II memory. CPU I reports all errors from either processor. If a switchable console display is the system comment device, it must be

†The CYBER 18-25 and 18-30 processors are referred to throughout the LMSDU text as dual MPs or a dual-MP system.

switched to CPU I. All operator interaction is through the ODS executive (in CPU I).

With the exception of section 1, all subsections of all sections are executed regardless of previous error conditions.

Sections 1 and 8 of the test concern the control of CPU II by CPU I and the inter-processor interrupts (see tables A-7 and A-8).

TABLE A-7. 255x SYSTEM STATUS/MODE BIT ASSIGNMENTS

Status/ Mode Bit	CPU I	CPU II
SM115	Not used	MILIA status
SM205	Generates local macro interrupt	Generates macro interrupt in CPU I
SM206	Generates micro interrupt in CPU II	Generates local micro interrupt
SM208	Used to clear and start CPU II	Not used

TABLE A-8. DUAL-MP SYSTEM STATUS/ MODE BIT ASSIGNMENTS

Status/ Mode Bit	CPU I	CPU II
SM115	Generates local macro interrupt	MLIA status
SM205	Not used	Generates local macro interrupt
SM206	Generates macro interrupt in CPU II	Generates local macro interrupt
SM208	Used to clear and start CPU II	Generates macro interrupt in CPU I

Through the use of control programs in read/write micromemory in both processors and the escape to micro sequence (EMS) instructions, the status/mode bits are manipulated, and the expected micro and macro interrupts are handled and verified in each processor.

Sections 2 through 7 concern the memory interaction (that is, the simultaneous access of local and remote (external) memory by both processors and the protect system). Refer to table A-9 for the summary of memory operations.

There are six possible memory activities that can occur:

- Read from local memory
- Read from external memory
- Write to local memory
- Write to external memory
- Read-modify-write (RMW) to local memory
- RMW to external memory

Of the 36 possible combinations, 18 are tested in LMSDU. Of the remaining 18, 15 are tested as the program is loaded into the other processor; and three occur as the test programs are executing in both processors during the test.

These test routines are written and executed in a manner that ensures a high statistical probability of simultaneous memory access from both processors.

Section 1, Inter-Processor Control

This section verifies that CPU I can control CPU II by manipulating status/mode bit SM208. It consists of three subsections.

NOTE

This section must always be the first section selected and cannot be bypassed since it performs the initialization of CPU II. A parameter error message is displayed if section 1 is not the first section selected.

Subsection 1

CPU I moves a control program into CPU II macromemory and attempts to start CPU II by setting SM208 off and then on. The expected action in CPU II is verified by CPU I through interrogation of a flag word in CPU II.

Subsection 2

CPU I verifies that CPU II may be halted by resetting SM208 and restarted by setting SM208.

TABLE A-9. MEMORY INTERACTION SECTIONS AND SUBSECTIONS

		Operation in CPU I					
		Read I	Read II	Write I	Write II	RMW I	RMW II
Operation in CPU II	Section 2 Read II		Subsection 1			Subsection 2	Subsection 3
	Section 3 Read I		Subsection 1	Subsection 2	Subsection 3	Subsection 4	Subsection 5
	Section 4 Write II			Subsection 1	Subsection 2	Subsection 3	Subsection 4
	Section 5 Write I				Subsection 1	Subsection 2	Subsection 3
	Section 6 RMW II					Subsection 1	Subsection 2
	Section 7 RMW I						Subsection 1

Subsection 3

CPU II is instructed to enter an idle loop that continually monitors a flag word set by CPU I. Then CPU II is instructed to load its micromemory with a control program. (Micromemory write is always enabled in both processors. Refer to tables A-7 and A-8.)

Section 2, Read II from II

Section 2 consists of three subsections.

Subsection 1

This subsection simultaneously reads CPU II memory from CPU II and reads CPU II memory from CPU I.

Subsection 2

This subsection simultaneously reads CPU II from CPU II and performs a read-modify-write to CPU I from CPU I.

Subsection 3

This subsection simultaneously reads CPU II from CPU II and performs a read-modify-write to CPU II from CPU I.

Section 3, Read I from II

Section 3 consists of five subsections.

Subsection 1

This subsection simultaneously reads CPU I from CPU II and reads CPU II from CPU I.

Subsection 2

This subsection simultaneously reads CPU I from CPU II and writes CPU I from CPU I.

Subsection 3

This subsection simultaneously reads CPU I from CPU II and writes CPU II from CPU I.

Subsection 4

This subsection simultaneously reads CPU I from CPU II and performs a read-modify-write to CPU I from CPU I.

Subsection 5

This subsection simultaneously reads CPU I from CPU II and performs a read-modify-write to CPU II from CPU I.

Section 4, Write II from II

Section 4 consists of four subsections.

Subsection 1

This subsection simultaneously writes CPU II from CPU II and writes CPU I from CPU I.

Subsection 2

This subsection simultaneously writes CPU II from CPU II and writes CPU II from CPU I.

Subsection 3

This subsection simultaneously writes CPU II from CPU II and performs a read-modify-write to CPU I from CPU I.

Subsection 4

This subsection simultaneously writes CPU II from CPU II and performs a read-modify-write to CPU II from CPU I.

Section 5, Write I from II

This section consists of three subsections.

Subsection 1

This subsection simultaneously writes CPU I from CPU II and writes CPU II from CPU I.

Subsection 2

This subsection simultaneously writes CPU I from CPU II and performs a read-modify-write to CPU I from CPU I.

Subsection 3

This subsection simultaneously writes CPU I from CPU II and performs a read-modify-write to CPU II from CPU I.

Section 6, RMW II from II

This section consists of two subsections.

Subsection 1

This subsection simultaneously performs a read-modify-write to CPU II from CPU II and to CPU I from CPU I.

Subsection 2

This subsection simultaneously performs a read-modify-write to CPU II from CPU II and to CPU II from CPU I.

Section 7, RMW I from II

Section 7 simultaneously performs a read-modify-write to CPU II from CPU I and to CPU I from CPU II.

Section 8, Inter-Processor Interrupts

Section 8 verifies that the inter-processor interrupts provided by the hardware and the interconnecting cables function as specified (see tables A-7 and A-8).

The section consists of eight subsections, each of which determines whether it is executed based on the input to the parameter list of the configuration type. Subsections 1, 2, 5, and 6 are executed for a dual-MP system; and subsections 2, 4, 7, and 8 are executed for the 255x Network Processing Unit.

All the inter-processor interrupts are verified, first locally by having each processor interrupt itself (subsections 1 through 4 (see tables A-7 and A-8), and then processor to processor. All interrupts, both micro and macro in both processors, are verified to occur on the interrupt line specified in the parameter list.

Subsection 1

This subsection is executed only for a dual-MP system; it verifies that if SM115 is set in CPU I, a local macro interrupt is generated on the specified line.

Subsection 2

This subsection is executed for both systems. It verifies that if SM205 is set in CPU II for a dual-MP system and in CPU I for the 255x system, a local macro interrupt is generated on the specified line.

Subsection 3

This subsection is executed only for a dual-MP system, it verifies that if SM206 is set in CPU II, a local macro interrupt is generated on the specified line.

Subsection 4

This subsection is executed only for the 255x system, it verifies that if SM206 is set in CPU II, a local micro interrupt is generated on the specified line.

Subsection 5

This subsection is executed only for a dual-MP system, it verifies that if SM206 is set in CPU I, a macro interrupt is generated on the specified line in CPU II.

Subsection 6

This subsection is executed only for a dual-MP system, it verifies that if SM208 is set in CPU II, a macro interrupt is generated on the specified line in CPU I.

Subsection 7

This subsection is executed only for the 255x system, it verifies that if SM205 is set in CPU II, a macro interrupt is generated on the specified line in CPU I.

Subsection 8

This subsection is executed only for the 255x system, it verifies that if SM206 is set in CPU I, a micro interrupt is generated on the specified line in CPU II.

MESSAGES

Table A-10 lists the LMSDU action codes.

RESTRICTIONS

Section 1 must always be the first section selected since it performs all the preliminary testing and start-up of CPU II.

TABLE A-10. LMSDU ACTION CODES

Action Code	English Text	Description	Additional Information
4901	SEC 1 NOT FIRST	Section 1 must be the first section selected.	SEC1 = First section select word
	LINE B PARAMETER	The CPU I interrupt line parameter value is not in the range $1\text{-}F_{16}$ or $10_{16}\text{-}FO_{16}$.	LINE = CPU I line parameter value
	LINE A PARAMETER	The primary or secondary CPU II interrupt line parameter value is not within the range $1\text{-}F_{16}$ or $10_{16}\text{-}FO_{16}$.	LINE = CPU II interrupt line value
	PAGE PARAMETER	The parameter value for the CPU I or II micromemory page is not within the range 1-F.	PAGE = CPU I/II micromemory page parameter value
	B2O5-LCL-NO INT	No local interrupt occurred when SM2O5 was set in CPU I.	EDAT = Expected line number
	B2O5-LCL-OTHER	When SM2O5 was set in CPU I, a local interrupt occurred on a line other than the specified line.	EDAT = Expected line ADAT = Actual line
	B2O5-LCL-MULT	More than one local interrupt occurred when SM2O5 was set in CPU I.	EDAT = Expected line ADAT ADAT†
	A2O5->B-NO INT	No interrupt occurred in CPU I when SM2O5 was set in CPU II	EDAT = Expected line number
	A2O5->B-OTHER	When SM2O5 was set in CPU II, an interrupt occurred in CPU I on a line other than that specified.	EDAT = Expected line ADAT = Actual line
	A2O5->B-MULT	More than one interrupt occurred in CPU I when SM2O5 was set in CPU II.	EDAT = Expected line ADAT ADAT†
4902	A NOT RUNNING	CPU II did not start from zero when CPU I reset and set SM2O8.	
	A NOT STOPPED	CPU II did not halt execution when CPU I reset SM2O8.	
	A NOT IDLING	CPU II did not enter its idle loop when restarted from zero.	
	CPU-A TIMEOUT	CPU II did not return to its idle loop within three seconds after a request to initiate an operation (may be hung up).	
4903	B115-LCL-NO INT	No local interrupt occurred when SM115 was set in CPU I.	EDAT = Expected interrupt line
4904	B115-LCL-OTHER	When SM2O5 was set in CPU I, a local interrupt occurred on a line other than that specified.	EDAT = Expected line ADAT = Actual line

†The first four interrupts are indicated in the order 4-3-2-1 (that is, 0247 indicates three interrupts on lines 7, 4, and 2).

TABLE A-10. LMSDU ACTION CODES (Contd)

Action Code	English Text	Description	Additional Information
4904 (Contd)	B115-LCL-MULT	More than one local interrupt occurred when SM115 was set in CPU I.	EDAT = Expected line
	A208->B-OTHER	When SM208 was set in CPU II, an interrupt occurred in CPU I on a line other than that specified.	EDAT = Expected line ADAT = Actual line
	A208->B-MULT	More than one interrupt occurred in CPU I when SM208 was set in CPU II.	EDAT, ADAT† = Expected line
4905	A205-LCL-NO INT	No local interrupt occurred when SM205 was set in CPU II.	EDAT = Expected interrupt line
	A206-LCL-OTHER	SM206 was set in CPU II, local interrupt occurred on a line other than that specified.	EDAT = Expected line ADAT = Actual line
	A206-LCL-MULT	More than one local interrupt occurred when SM206 was set in CPU II.	EDAT, ADAT† = Expected line
	B206->A-OTHER	When SM206 was set in CPU I, the interrupt in CPU II occurred on a line other than that specified.	EDAT = Expected line ADAT = Actual line
	B206->A-MULT	More than one interrupt occurred in CPU II when SM206 was set in CPU I.	EDAT, ADAT† = Expected line
4906	A205-LCL-OTHER	When SM206 was set in CPU II, a local interrupt occurred on a line other than that specified.	EDAT = Expected line ADAT = Actual line
	A205-LCL-MULT	More than one local interrupt occurred when SM115 was set in CPU II.	EDAT, ADAT† = Expected line
4907	A206-LCL-NO INT	No local interrupt occurred when SM206 was set in CPU II.	EDAT = Expected interrupt line
4908	B206->A-NO INT	No interrupt occurred in CPU II when SM206 was set in CPU I.	EDAT = Expected line number
4909	A208->B-NO INT	No interrupt occurred in CPU I when SM208 was set in CPU II.	EDAT = Expected line number
490A	RDAFA-RMBFB-RMWB	Following the simultaneous execution of RDAFA and RMBFB, the data in the RMW location does not equal 64_{16}.	DATA = Actual data content of RMW location
	RDAFA-RDAFB-DATA	During the simultaneous execution of RDAFA and RDAFB, a data miscompare error was detected by CPU 1.	FLAG = Number of data errors

†The first four interrupts are indicated in the order 4-3-2-1 (that is, 0247 indicates three interrupts on lines 7, 4, and 2).

TABLE A-10. LMSDU ACTION CODES (Contd)

Action Code	English Text	Description	Additional Information
490A (Contd)	RDAFA-RMAFB-RMWA	Following simultaneous execution of RDAFA and RMAFB, the content of the RMW location in CPU II does not equal 64_{16}.	DATA = Content of RMW location
	RDBFA-RDAFB-DATA	During the simultaneous execution of RDBFA and RDAFB, an error was detected by CPU I.	FLAG = CPU I error flag†
	RDBFA-WRAFB-ADAT	Following the simultaneous execution of RDBFA and WRAFB, the CPU II memory data is not as expected.	DATA = Actual data
	RDBFA-RMAFB-RMWA	Following the simultaneous execution of RDBFA and RMAFB, the RMW location data in CPU II does not equal 64_{16}.	DATA = Actual data
	WRBFA-WRAFB-ADAT	Following the simultaneous execution of WRBFA and WRAFB, the data in CPU II memory is not as expected (all zeros).	DATA = Error count
	WRBFA-RMAFB-RMWB	Following the simultaneous execution of WRBFA and RMAFB, the RMW location data in CPU I does not equal 64_{16}.	DATA = Actual data
	RMBFA-RMAFB-RMWA	Following the simultaneous execution of RMBFA and RMAFB, the RMW location data in CPU II does not equal 64_{16}.	DATA = Actual data
490B	RDBFA-WRBFB-BDAT	Following the simultaneous execution of RDBFA and WRBFB, the data in CPU I memory is not as expected (all zeros).	DATA = Actual data
	RDBFA-RMBFB-RMWB	Following the simultaneous execution of RDBFA and RMBFB, the RMW location data in CPU I does not equal 64_{16}.	DATA = Actual data
	WRAFA-WRBFB-BDAT	Following the simultaneous execution of WRAFA and WRBFB, the data in CPU I memory is not as expected (all zeros).	DATA = Error count

†CPU I/II error flag definitions:

1xxx Parity error detected
2xxx Protect fault detected
3xxx Power interrupt detected
0xxx A read miscompare error was detected; xxx is number of miscompares (hexadecimal).

TABLE A-10. LMSDU ACTION CODES (Contd)

Action Code	English Text	Description	Additional Information
490B (Contd)	WRAFA-RMBFB-RMWB	Following the simultaneous execution of WRAFA and RMBFB, the RMW location data in CPU I does not equal 64_{16}.	DATA = Actual data
	WRAFA-RMAFB-RMWA	Following the simultaneous execution of WRAFA and RMAFB, the RMW location data in CPU II does not equal 64_{16}.	DATA = Actual data
	WRBFA-RMBFB-RMWB	Following the simultaneous execution of WRBFA and RMBFB, the RMW location data in CPU I does not equal 64_{16}.	DATA = Actual data
	RMAFA-RMBFB-RMWB	Following the simultaneous execution of RMAFA and RMBFB, the RMW location data in CPU I does not equal 64_{16}.	DATA = Actual data
490C	RDAFA-RMBFB-CPUA	During the simultaneous execution of RDAFA and RMBFB, an error was detected by CPU II.	FLAG = CPU II error flag†
	RDAFA-RMAFB CPUA	During the simultaneous execution of RDAFA and RMAFB, an error was detected by CPU II.	FLAG = CPU II error flag†
	RDAFA-RDAFB-CPUA	During the simultaneous execution of RDAFA and RDAFB, an error was detected by CPU II.	FLAG = CPU II error flag†
	WRAFA-WRBFB-ADAT	Following the simultaneous execution of WRAFA and WRBFB, the data in CPU II memory is not as expected (all zeros).	DATA = Error count
	WRAFA-RMBFB-CPUA	During the simultaneous execution of WRAFA and RMBFB, an error was detected by CPU II.	FLAG = CPU II error flag†
	WRAFA-RMBFB-ADAT	Following the simultaneous execution of WRAFA and RMBFB, the data in CPU II memory is not as expected (all zeros).	DATA = Error count
	WRAFA-RMAFB-CPUA	During the simultaneous execution of WRAFA and RMAFB, an error was detected by CPU II.	FLAG = CPU II error flag†

†CPU I/II error flag definitions:

1xxx Parity error detected
2xxx Protect fault detected
3xxx Power interrupt detected
0xxx A read miscompare error was detected; xxx is number of miscompares (hexadecimal).

TABLE A-10. LMSDU ACTION CODES (Contd)

Action Code	English Text	Description	Additional Information
	WRAFA-RMAFB-ADAT	Following the simultaneous execution of WRAFA and RMAFB, the data in CPU II memory is not as expected (all zeros).	DATA = Error count
	RMAFA-RMBFB-CPUA	During the simultaneous execution of RMAFA and RMBFB, an error was detected by CPU II.	FLAG = CPU II error flag†
	RMAFA-RMBFB-RMWA	Following the simultaneous execution of RMAFA and RMBFB, the RMW location data in CPU II does not equal 64_{16}.	DATA = Actual data
490D	RDBFA-WRBFB-CPUA	Following the simultaneous execution of RDBFA and WRBFB, an error was detected by CPU II.	FLAG = CPU II error flag†
	RDBFA-WRAFB-CPUA	During the simultaneous execution of RDBFA and WRAFB, an error was detected by CPU II.	FLAG = CPU II error flag†
	RDBFA-RMBFB-CPUA	During the simultaneous execution of RDBFA and RMBFB, CPU II detected an error.	FLAG = CPU II error flag†
	RDBFA-RMAFB-CPUA	During the simultaneous execution of RDBFA and RMAFB, an error was detected by CPU II.	FLAG = CPU II error flag†
	WRBFA-WRAFB-CPUA	During the simultaneous execution of WRBFA and WRAFB, an error was detected by CPU II.	FLAG = CPU II error flag†
	WRBFA-WRAFB-BDAT	Following the simultaneous execution of WRBFA and WRAFB, the data in CPU I memory is not as expected (all zeros).	DATA = Error count
	WRBFA-RMBFB-CPUA	During the simultaneous execution of WRBFA and RMBFB, an error was detected by CPU II.	FLAG = CPU II error flag†
	WRBFA-RMBFB-BDAT	Following the simultaneous execution of WRBFA and RMBFB, the data in CPU I memory is not as expected (all zeros).	DATA = Error count
	WRBFA-RMAFB-CPU	During the simultaneous execution of WRBFA and RMAFB, an error was detected by CPU II.	FLAG = CPU II error flag†
	WRBFA-RMAFB BDAT	Following the simultaneous execution of WRBFA and RMAFB, the data in CPU I memory is not as expected (all zeros).	DATA = Error count

†CPU I/II error flag definitions:

1xxx	Parity error detected
2xxx	Protect fault detected
3xxx	Power interrupt detected
0xxx	A read miscompare error was detected; xxx is number of miscompares (hexadecimal).

TABLE A-10. LMSDU ACTION CODES (Contd)

Action Code	English Text	Description	Additional Information
	RMBFA-RMAFB-CPUA	During the simultaneous execution of RMBFA and RMAFB, an error was detected by CPU II.	FLAG = CPU II error flag† 1
	RMBFA-RMAFB-RMWB	Following the simultaneous execution of RMBFA and RMAFB, the RMW location data in CPU I does not equal 64_{16}.	DATA = Actual data
490E	WRAFA-WRAFB-CPUA	During the simultaneous execution of WRAFA and WRAFB, an error was detected by CPU II.	DATA = Error count
	WRAFA-WRAFB-ADAT	Following the simultaneous execution of WRAFA and WRAFB, the data written into CPU II memory is not all zeros.	DATA = Error count
	RMAFA-RMAFB-CPUA	During the simultaneous execution of RMAFA and RMAFB, an error was detected by CPU II.	FLAG = CPU II error flag†
	RMAFA-RMAFB-RMWA	Following the simultaneous execution of RMAFA and RMAFB, the RMW location data in CPU II does not equal $C8_{16}$ (2 times 64_{16}).	DATA = Actual data
4910	FAULT BIT NOT ON	In CPU I memory, the ECC bit does not indicate that the protect bit was set by CPU II.	
4911	FAULT BIT ON	In CPU I memory, the ECC bit does not indicate that the protect bit was reset by CPU II.	
4912	FAULT BIT NOT ON	In CPU II memory, the ECC bit does not indicate that the protect bit was reset by CPU II.	
4913	FAULT BIT ON	In CPU II memory, the ECC bit does not indicate that the protect bit was reset by CPU I.	
4914	NO FAULT INT	No fault interrupt occurred in CPU II when an unprotected instruction in CPU II wrote to a protected area in CPU I memory.	FLAG†
4915	FAULT INT	A fault interrupt occurred in CPU II when a protected instruction in CPU II wrote to a protected area in CPU I memory.	

†CPU I/II error flag definitions:

1xxx	Parity error detected
2xxx	Protect fault detected
3xxx	Power interrupt detected
0xxx	A read miscompare error was detected; xxx is number of miscompares (hexadecimal).

TABLE A-10. LMSDU ACTION CODES (Contd)

Action Code	English Text	Description	Additional Information
4916	FAULT INT	A fault interrupt occurred in CPU II when a protected instruction in CPU II wrote to an unprotected area in CPU I memory.	FLAG†
4917	NO FAULT INT	No fault interrupt occurred in CPU I when an unprotected instruction in CPU I wrote to a protected area of CPU II memory.	FLAG†
4918	FAULT INT	A fault interrupt occurred in CPU I when a protected instruction in CPU I wrote to a protected area in CPU II memory.	FLAG†
4919	FAULT INT	A fault interrupt occurred in CPU I when a protected instruction in CPU I wrote to an unprotected area in CPU II memory.	FLAG†

†CPU I/II error flag definitions:

1xxx	Parity error detected
2xxx	Protect fault detected
3xxx	Power interrupt detected
0xxx	A read miscompare error was detected; xxx is number of miscompares (hexadecimal).

ODS TEST LIMITATIONS B

This appendix lists the ODS tests that should not be run on an LMS-based system and the ODS tests that will give a known failure due to changed system timing.

ILLEGAL ODS TESTS

he following tests should not be executed on an MEU-based system, since legitimate errors will occur:

MPINS
MPMOS
MOSMA
PAGE1
DUCPU

ALLOWABLE ODS TEST ERRORS

Some ODS tests will have errors when executed on an LMS-based system, either due to unsupported CYBER 18 features or system timing. These errors are allowable and are not hardware failures.

BCLA1

Due to the large memory the BCLA1 test fails after turning on the protect switch with the following error:

```
BCLA1 REJECT NOT RCVD
CODE= 2F05, SECT= 0002, PASS= 0001, ERRO= 0001,
    EQCD= 1500, ILNS= 0003
QREG= 1506, AREG= 0064
BCLA1 SUSPENDED ERR
```

Type GO after the error to continue the test.

REQUIRED ODS TEST CONDITIONS

Some ODS tests must have specific conditions set up prior to test execution or the test will fail in an unrecoverable manner. This is due either to unsupported CYBER 18 hardware features or different system timing.

SDLC1

Since the LMS does not support the CYBER 18 DMA memory address error (DMAMAE) feature, this status never occurs in a DMA controller status word. Section 7, which tests the DMAMAE feature, must therefore be removed from the SDLC1 run parameter list prior to test execution.

COMMENT SHEET

MANUAL TITLE: CDC AT429, BA222 Large Memory Subsystem
Hardware Reference/Maintenance Manual

PUBLICATION NO.: 60476080 REVISION: E

NAME: _____

COMPANY: _____

STREET ADDRESS: _____

CITY: _____ STATE: _____ ZIP CODE: _____

This form is not intended to be used as an order blank. Control Data Corporation welcomes your evaluation of this manual. Please indicate any errors, suggested additions or deletions, or general comments below (please include page number references).

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

STAPLE

STAPLE

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MINN.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION

Technical Publications Department
2401 North Fairview Avenue
St. Paul, Minnesota 55113

CUT ALONG LINE

FOLD

FOLD

03377

CORPORATE HEADQUARTERS, P.O. BOX 0, MINNEAPOLIS, MINN. 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

LITHO IN U.S.A.

CONTROL DATA CORPORATION