APPENDIX C

MEMORY MANAGEMENT UNIT

SERVICE MANUAL

Effective Pages February 1977

ii
C.1 through C.19

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

APPENDIX C

MEMORY MANAGEMENT UNIT

C.1 GENERAL

The Memory Management Unit (MMU) is a hardware option which provides memory addressing and protection facilities for the P857 system. The MMU is a system slave which operates under CPU Bus control. The three main functions of the MMU are:

- Address translation which extends memory addressing up to 128k words.
- Page Usage Timing for dynamic program relocation on a page basis.
- Full memory protection.

1 Page = 2 K words

C.2 Address Translation

The primary function of the MMU is to extend memory addressing up to 128k physical words while retaining the 32k-word logical address selection from the CPU instruction (Figures C-1 and C-2).

- A sixteen-segment table is pre-loaded with page addresses by a single Table Load instruction.
- Each CPU/memory transfer via the MMU (MMU translation) then uses the four most-significant address lines (S0-3) to select a table segment, and thus a page address, while the 12 least-significant address bits select one of the 2k words within the page.
- A Table Store instruction can be used by software to access the 16-word segment table for testing or dynamic relocation.

C.3 Page Usage Timer

The function of the Page Usage Timer is to keep track of which pages have not been accessed within a certain Page Residency Parameter (Window Interval). This parameter is based on the number of MMU translations and is derived from a software constant and an MMU hardware constant of 1024 (2^{10}).

- An eight-bit (0-255) software constant is loaded into the MMU by a WER instruction. This interval time count is used as a software constant common to all 16 page timers.
- Each MMU translation resets the timer for the addressed page, and is counted as an "interval" time unit.
- Each time the interval count is complete, a single page timer (located in the MMU scratchpad) is incremented, and the next sequential page timer is selected in preparation for the succeeding page-timer count.
- A non-addressed page thus counts to overflow after a number of MMU translations equal to the software constant times the hardware constant 1024 (2^{10}), as follows:
  -- Window interval (count 0-255).
  -- Sequential count through all 16 page timers (2^4).
  -- Full six-bit page timer (2^6).

C.4 Memory Protection

Each of the 16 words loaded into the segment table by the TL instruction includes, in addition to the page address, two memory-protection bits established by software.

- Bit 6 (E) -- The Page Error bit is set to restrict the page to System Mode. If an address translation in User Mode (FU set) attempts to access the page, the translation is blocked and MFAULTN is sent to the CPU.
- Bit 7 (R) -- The Read-Only Page bit is set to restrict the page to read operations only. If an address translation in User Mode attempts a write operation on this page, the translation is blocked and MFAULTN is sent to the CPU.

C.5 SOFTWARE CONSIDERATIONS

C.6 System/User Operations

The main memory is divided into a System part and a User part, with the System part always located at the beginning of the memory (lowest addresses). The System part contains the supervision programs (monitors) and their associated working areas (tables, buffers, etc.). The User part contains the user's running programs or real-time tasks, etc. The CPU operates in either System Mode or User Mode, and certain instructions are restricted to System Mode only.

C.7 The system part of memory is located in the beginning of memory and is limited to the first 32k word positions. Addressing the System part of memory is done with the 16-bit logical address directly from the CPU, and the MMU is not used. However, the System is able to extend its addressing by using the extended instructions (EL, ES, MVSU, MVUS) which use MMU translation.

C.8 The maximum length of a single user program is 32k words, divided into pages of 2k words. The CPU addresses a user location with the 16-bit logical address. The MMU translates the first four bits into a six-bit physical page address (which was previously loaded into the MMU). The remaining 12 bits from the CPU then select one of 2k words within the selected page.

C.9 System mode instructions used by the MMU:

- Table Load (TL) loads 16 physical page addresses and protection bits from memory into the MMU segment table. This must be done prior to address translations through the MMU.
- Table Store (TS) stores the 16-word segment table (including page addresses, protection bits, and page timers) from the MMU into memory.
- Write External Register (WER) loads a Window-Interval software constant into the MMU for use with the page timers. The WER instruction is not exclusive to MMU operations.

Address translation is usually performed by the MMU while the CPU operates in User Mode, and the MMU does memory-protection checking. There are, however, four System-Mode instructions which are used for address translation operations:

- Extended Load (EL) and Extended Store (ES) are used by the System to access the User area of memory to read or write a single word.
- Move Table User-to-System (MVUS) and System-to-User (MVSU) are used by the System to transfer blocks of data between the User area of memory and the System area.

The only difference the MMU notices for these System-Mode address translation operations is that the user-mode bit (FU) is not active, and the MMU does not do the memory-protection checking.

C.10 The Operating System must load the segment table every time a new User program is given control. The Operating System is also responsible for sending the 18-bit physical addresses of the User's I/O buffers to the I/O channels (I/O Processors or DMA controllers).

C.11 Page Replacement Facilities

C.12 Window Interval. The required window interval count (N) to obtain a specific page residency time (T) is: the time (T), divided by the average memory access time (θ), divided by the hardware constant (1024).

$T = 1024.\theta.N$ or $N = T \div \theta \div 1024$

For example, if a page residency time of 100ms is desired and the average memory access time (θ) is 2μsec, the window interval count (N) is:

$100\text{ms} \div 2\mu\text{sec} \div 1024 = 50$

Set into MMU by WER instruction.

Hardware constant (count through 16 timers of six bits each)

Average memory access time: produces one count each time the MMU is accessed for address translation.

C.13 Modified Page. Bit M (08) is set by the MMU whenever a write operation (Store instruction) is performed on a specific page. This feature indicates to the Operating System when a page needs to be swapped out before a new program segment is loaded at the same place. If not necessary, a direct over-writing is possible, resulting in a large saving of time.

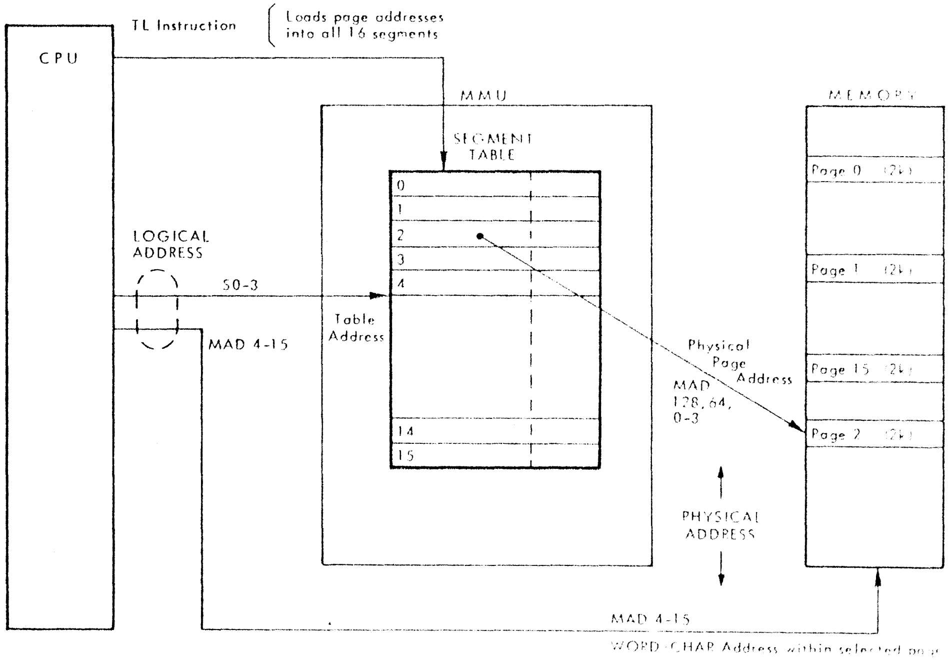
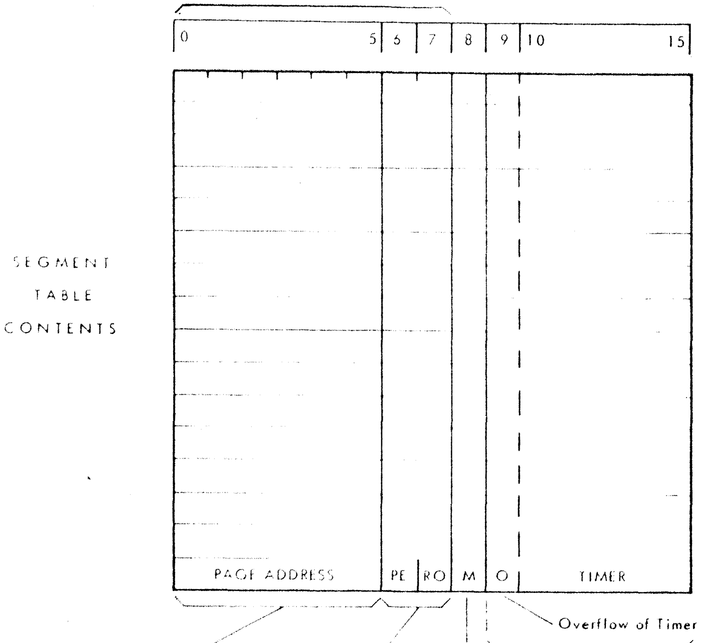


Figure C-1 MMU System Diagram

Loaded by TL instruction for a specific User program, and used during address translations.



Physical Page Addresses. One segment accessed for each address translation

Memory Protection:

PE -- Page Error bit is set to restrict this page to System Mode only.

RO -- Read Only bit is set to restrict this page to read operations only.

• Controlled by MMU according to number of address translations, and a "Window Interval" loaded into MMU by a WER instruction.

Modified Page is set when a write occurs on the page, indicating that the backing store is now different from the main memory.

Figure C-2 MMU Segment Table Contents

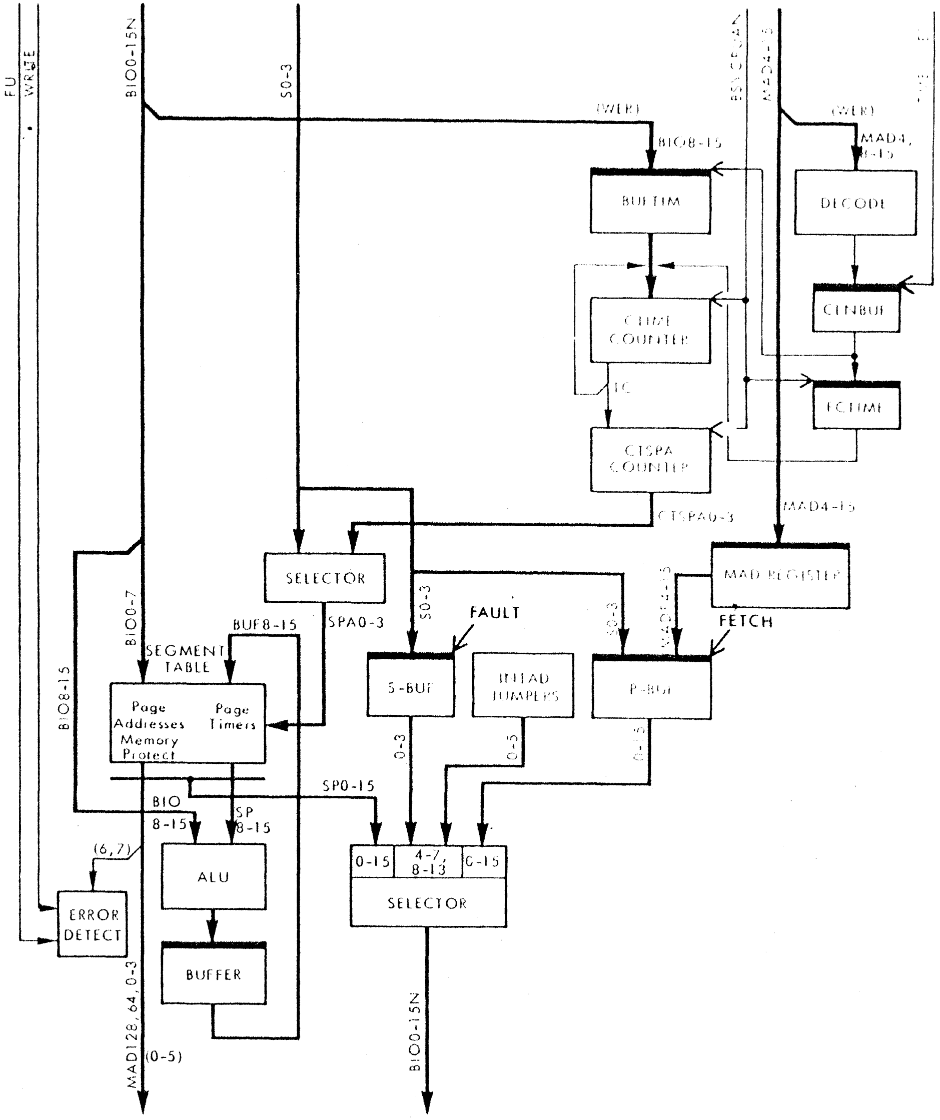


Figure C-3 MMU Block Diagram

C.14 LOGIC DESCRIPTION

The MMU logic description is organized into the different operating sequences of the MMU. A general block diagram of the MMU is provided in Figure C-3. The detailed logic is shown in Figure C-10 at the end the logic description. The different operating sequences used in the MMU are:

- Table Load
- Table Store
- WER -- Interval Time Loading
- Translation Operation
- Increment Timer
- Page Fault

C.15 Table Load

The CPU starts the Table Load operation (Figure C-4) in the MMU by sending TMMN. The CPU controls the operation with the signal BSYCPUAN which it sends for each of the 16 page addresses to be loaded into the MMU segment table. TRMN is received from the memory each time the data from the memory is valid. The TMMN signal resets counter CTSPA (via TLS and CTSPAZON) so that segment-table position 0 is selected for the first entry. The active signal TLSN enables CTSPA counting and selects the counter input to the SP-Address Selector with a high SPAS.

C.16 When the memory is ready with the data, TRMN goes active and removes the reset clamp STARTZON from the START flip-flop. The following high transition of OSC then sets START. The MMU clock T1, T2, T3, T4 runs when START is set, synchronized on OSC from the CPU. One word with the page address and memory-protection information is gated from BIO into the segment table at time T3. Also at T3, DONEMN is sent to the CPU to request the next word transfer.

C.17 Next Word Transfer. The CPU responds to DONEMN by dropping BSYCPUAN. The MMU increments the CTSPA count at the trailing edge of BSYCPUAN to select the next position in the segment table. START resets on the first rising OSC after BSYCPUAN drops, and is held reset when TRMN goes inactive. When memory is ready with data, TRMN goes active and the MMU

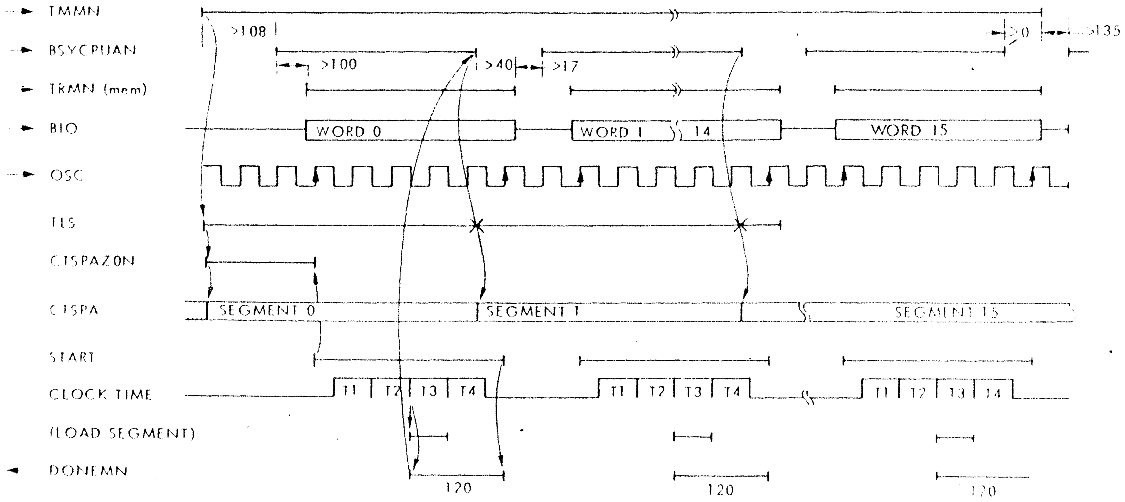


Figure C-4 Table Load Sequence

sequence (START, clock cycle, BIO → segment table, DONEMN) is repeated to load this word.

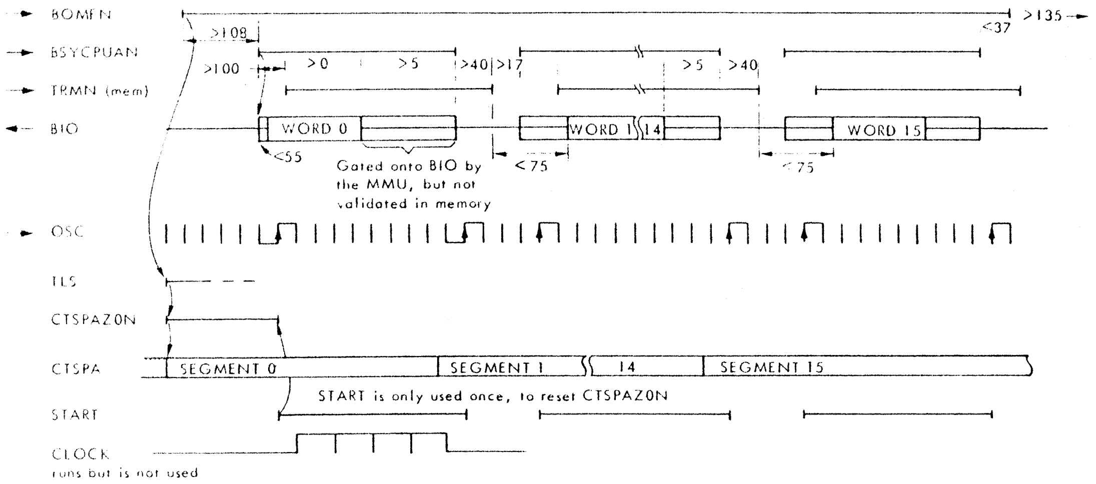
C.18 Last Word Transfer. The CPU drops TMMN after it receives the 16th DONEMN from the MMU. The inactive TMM signal in the MMU blocks further sending of DONEMN to the CPU. In all other respects, the MMU sequence for the last word transfer is identical to the previous transfers.

C.19 Table Store

The CPU starts the Table Store operation (Figure C-5) in the MMU by sending BOMFN. The CPU controls the operation with the signal BSYCPUAN which it sends for each of the 16 segment-table words to be transferred out to memory. TRMN is received from the memory each time the BIO data from the MMU has been accepted. The CPU uses TRMN to know when to initiate each transfer. The BOMFN signal resets counter CTSPA (via TLS and CTSPAZON) so that segment table position 0 is selected for the first entry. The active signal TLSN enables CTSPA counting and selects the counter input to the SP-Address Selector with a high SPAS.

C.20 The MMU uses BSYCPUAN and BOMFN to gate the first segment-table word (including page address, memory protection, and page timer information) onto the BIO to the memory. When the memory has accepted the data, TRMN goes active and removes the reset clamp STARTZON from the START flip-flop. The following high transition of OSC then sets START. The START signal resets CTSPAZON to allow the CTSPA counter to increment at each trailing edge of BSYCPUAN. START and the MMU clock continue to operate with each BSYCPUAN input, but are not used for the rest of the Table Store operation.

C.21 Each successive word is gated from the MMU segment table onto the BIO lines by the BSYCPUAN signal. The trailing edge of BSYCPUAN increments CTSPA to select the words sequentially from the segment table. The CPU terminates the operation after the 16th word transfer by dropping the BOMFN signal.



NOTES:

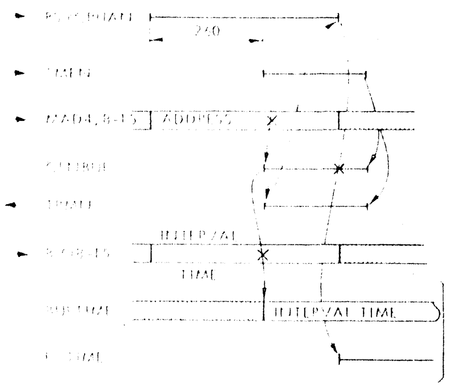
- Times in ns.
- OSC signal is not complete; shows only the relationship between edges of BSYCPUAN and OSC, for controlling START and clock times.

Figure C-5 Table Store Sequence

C.22 WER -- Interval Time Loading

The CPU loads the window interval software constant into the MMU with a WER instruction (Figure C-6). The CPU starts the operation by activating BSYCPUAN and, at the same time, placing the MMU address on the MAD08-15 lines and placing the window interval software constant on the BIO08-15 lines. When the MMU decoder logic detects its own address on the MAD lines and the WER instruction code (MAD04 = 0), signal MAREN is activated. With decoded-address signal MAREN active, the CPU timing signal TMEN sets flip-flop CLNBUF. The CLNBUF output gates the interval time from the BIO lines into the BUFTIME register and also, in conjunction with TMEN, generates the timing response signal TRMN back to the CPU.

C.23 The CPU responds to TRMN by terminating BSYCPUAN to the MMU. The trailing edge of BSYCPUAN, gated by CLNBUF, sets the FCTIME flip-flop. With FCTIME set and the interval time loaded in the BUFTIME register, the MMU is ready for page-time counting during the address-translation operations. When the CPU drops TMEN, the MMU resets CLNBUF (via CLBUFZ0, CLBUFZ0N) and drops the TRMN signal to the CPU.



Condition Ready for page-time counting during the Address Translation Operations.

Figure C-6 WER Sequence (Loading Window Interval Software Constant)

C.24 Translation Operation

The CPU activates signal TMMU to indicate a memory transfer requiring MMU address translation (Figure C-7). The MMU scratchpad address selector is switched to the S00-03 input by the inactive SPAS ($\overline{\text{FINCRN}}.\overline{\text{TLSN}}$). TMMU and BSYCPUAN together gate the physical page address (SP00-05) from the location selected by the logical page address (S00-03).

C.25 The START flip-flop is set on the first rising edge of OSC after BSYCPUAN is received. Each trailing edge of OSC then increments the MMU clock. The signals TMMU.START.$\overline{\text{FINCRN}}$ (=TRANSL) set the BUF08D selector to test the WRITE signal from the CPU. TRANSLN, together with $\overline{\text{TMMN}}.\overline{\text{FINCR}}$, sets the ALU to the Logical-Zero mode ($\overline{\text{FINCRN}}$ to the Mode input pin 8 selects Logical operations). The leading edge of T3 gates the addressed-word timer from SP08-15, through the ALU, to the Buffer: this resets the timer (bits 09-15) and sets bit 8 if the operation is a Write operation. The Buffer contents are reloaded into SP08-15 at the trailing edge of T3.

C.26 When the SP word is accessed by S00-03, the bits 6 and 7 are compared with the CPU command bits FU and WRITE. If bit 6 is set and FU is active (System Mode only bit and User Mode), ERRORN is generated. If bit 7 is set and WRITE is active (Read-Only bit and Write command) and FU is set (User Mode), ERRORN is generated and MFAULTN is sent to the CPU. This error condition prevents setting TMRF at T2, and blocks the sending of TMRF to memory, thus blocking the memory transfer.

C.27 If the memory check is correct, the TMRF flip-flop is set at T2 by the rising edge of OSC. TMRF and START together send timing signal TMRN to the memory. When the memory part of the transfer is complete, it sends TRMN to the CPU. The CPU responds by dropping BSYCPUAN to the MMU.

C.28 During the first translation operation following a WER instruction, flip-flop FCTIME is set. When BSYCPUAN drops at the end of the first translation, the complement of the software constant is gated from the BUFTIM

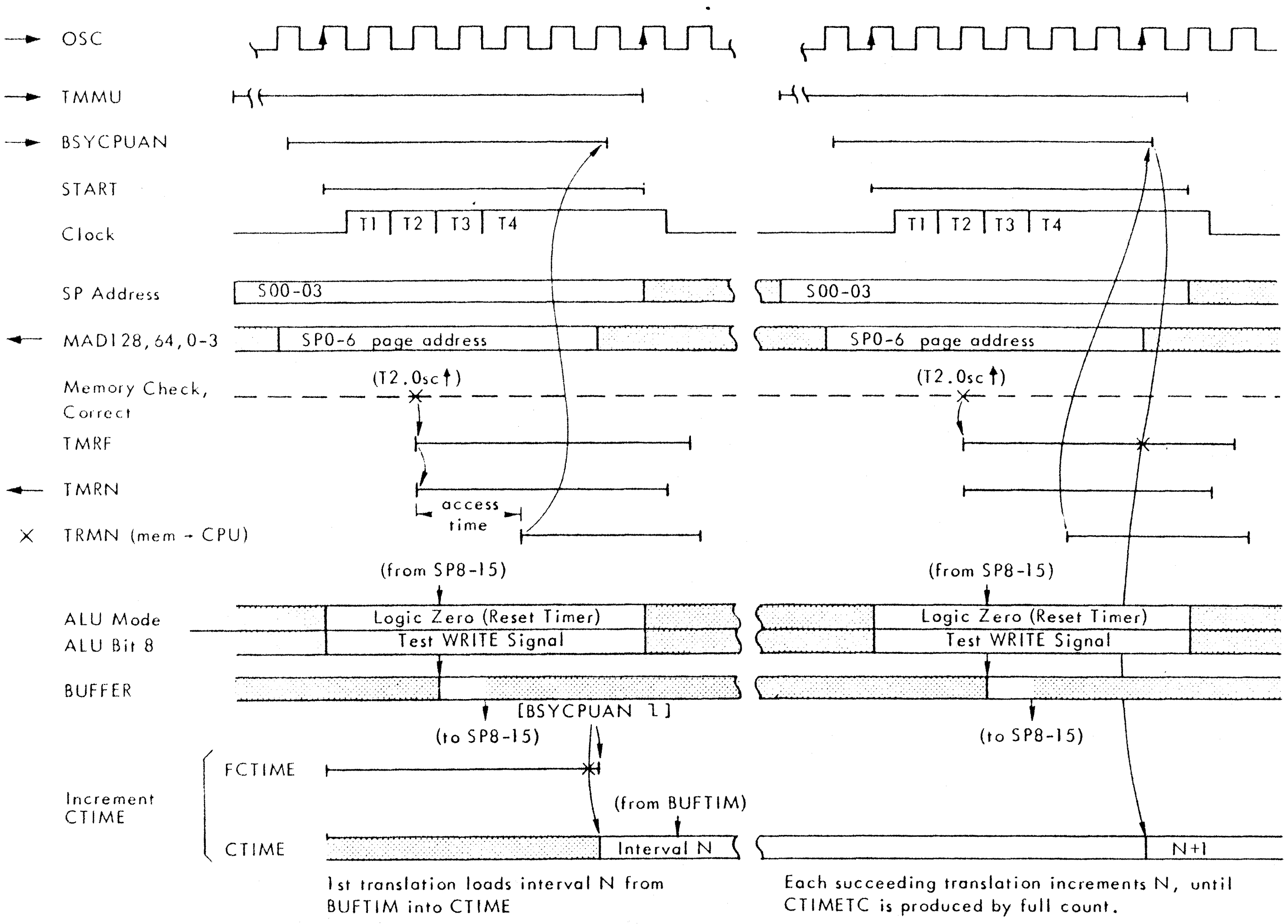


Figure C-7 Translation Sequences

register into the CTIME counter. When BSYCPUAN drops at the end of all succeeding translations, CTIME is incremented.

C.29 Increment Timer

The complement of the software constant is loaded into CTIME from the BUFTIM register (Figure C-8). Each translation increments CTIME so that it will be full when the number of translations equals the software constant. When the trailing edge of BSYCPUAN increments CTIME to full, CTIMETC is produced. The next rising edge of OSC then sets flip-flop FINCR. The same edge of OSC also resets TMRF (normal translation sequence). With TMRF removed from the CET input of CTIME, the CTIMETC output drops.

C.30 The FINCR flip-flop activates SPAS so that the scratchpad address selector selects the CTSPA source. The active FINCR signal also sets the ALU and the BUF08D selector to the arithmetic A+1 mode. At the leading edge of time T3, the scratchpad page timer (bits 08-15) selected by CTSPA is gated through the ALU into the BUFFER. At the trailing edge of T3, the buffer contents are loaded back into the scratchpad, now incremented by 1. FINCR is reset by the T4 clock, and the inactive FINCR signal then resets T4 to end the increment timer sequence.

C.31 TMRF is generated during the succeeding translation sequence and, since CTIME still contains its full count, the CTIME output is reactivated. At the end of the translation then, CTIMETC causes the trailing edge of BSYCPUAN to increment the CTSPA counter (for selecting the next page timer) and to reload the CTIME counter with the complement of the software constant from the BUFTIM register.

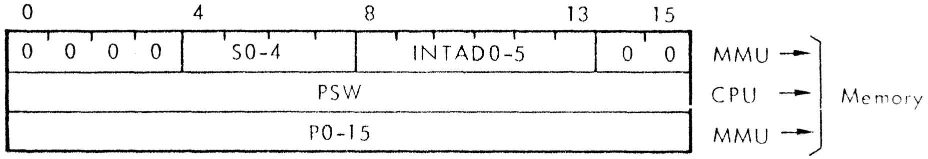
C.32 Page Fault

A page fault is detected during a normal Translation sequence if a User Mode operation (FU set) accesses a System-Mode-only page (bit 6 set) or if a Write operation is attempted (WRITE signal) in User mode on a read-only page (bit 7 set). The resultant ERROR signal inhibits the TMRF flip-flop and blocks the sending of the memory timing signal TMRN. The flip-flop FAULT is set at time T2 (Figure C-9). FAULT generates the MFAULTN signal to the CPU, gates the S00-03 bits into the S-Buf register, and sets the BIOS0 flip-flop.

C.33 BIOS0 and BIOS1 are now set (=1,0) to select the P-Buf source via the BIO Select circuit. The CPU responds to MFAULT by sending BOMFN and BSYCPUAN to the MMU. These two signals together gate the P-Buf contents (P00-15) onto the BIO lines to the CPU. At the trailing edge of BOMFN, BIOS0 is reset and BIOS1 is set (=0,1) so they select the S-Buf and INTAD sources via the BIO Select circuit. The next time BSYCPUAN and BOMFN are received together from the CPU, the MMU gates the segment-table address (S00-03) and the MMU interrupt code (INTAD0-5) onto the BIO lines.

C.34 The CPU activates BSYCPUAN once between the two BOMFN signals to transfer its PSW to memory. Since BOMFN is not active, however, there is no action taken in the MMU at that time.

Stack after Page Fault Interrupt



S = page number
P = program counter or aborted instruction

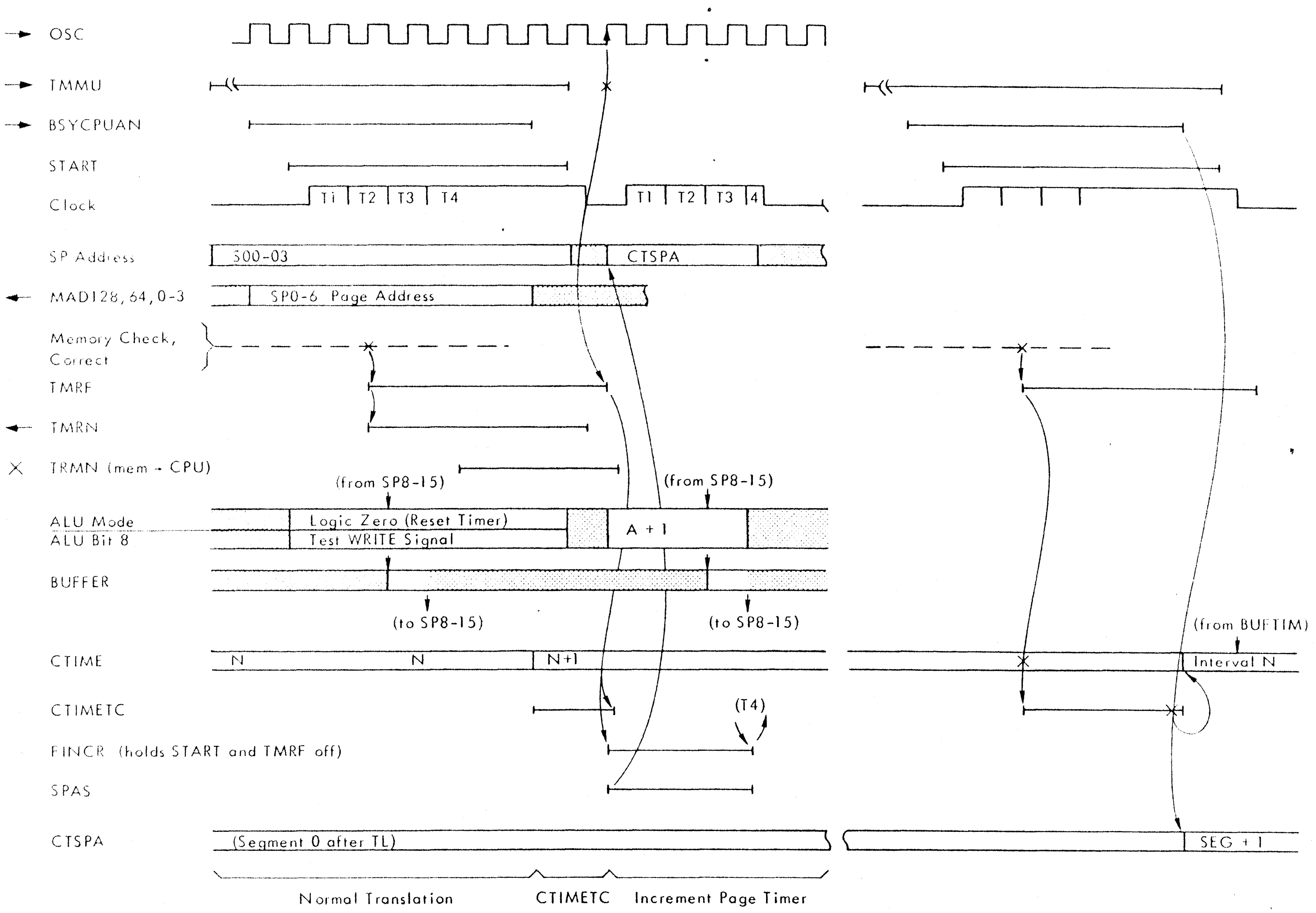


Figure C-8 Increment Timer Sequence

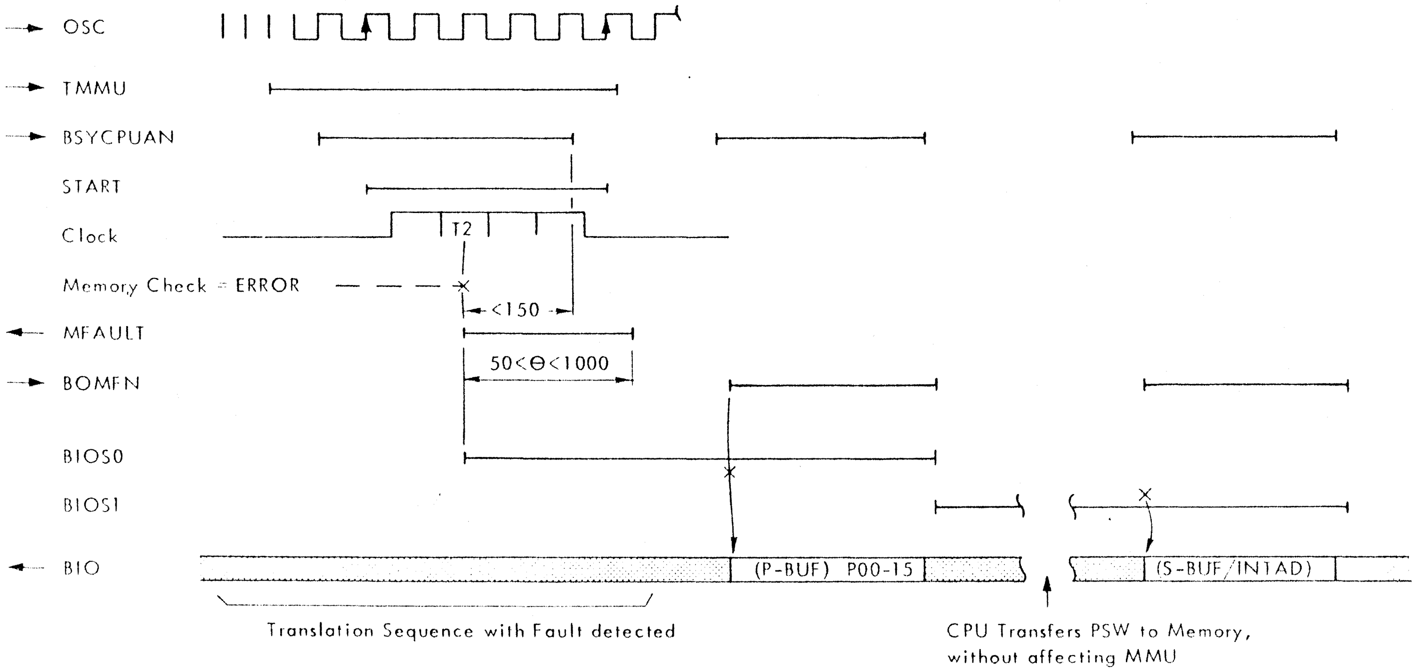


Figure C-9 Page Fault Sequence

C.35 INPUT/OUTPUT SIGNALS

CPU → MMU Signals	
BSYCPUAN	CPU has Bus control. Validates Bus signals and controls all MMU operations.
BOMFN	Table Store control signal. Page-fault stack-loading control signal: 1st -- to read aborted instruction-counter value; 2nd -- to read logic page address which set page fault and the program level of the page-fault job, coded on the MMU card.
FU = 1 = 0	CPU in User Mode. Any memory violation sets page fault. CPU in System Mode.
GFETCH	Fetch cycle is executed by CPU. Instruction counter value is loaded into MMU P-Buf register.
OSCFLO	CPU clock signal, used by MMU for internal timing.
S00-03	Logical page address from CPU P-register.
TMMN	Table Load control signal.
TMMU	Translation control signal; for page-address translation, memory protection check, and memory activation by the MMU.

MMU → CPU Signals	
DONEMN	MMU reply during Table Load as each segment is loaded.
MFAULTN	Page fault is detected during translation.
MMUABS	Held at 0v (inactive) when MMU card is in place.

BUS → MMU Signals	
BIOON-15N	Data used during Table Load and WER operations.
CLEARN	Clear signal for initializing system.
TMEN	WER instruction timing for loading the software constant into the MMU BUFTIM register.
TRMN	Memory reply: -- During Table Load and Table Store validates data. -- During translation releases memory activation.
MAD04-15	Address lines to: -- select the MMU during WER instruction, and -- store the CPU instruction-counter value during each Fetch instruction cycle.
WRITE	CPU command for store cycles. Used during translation to test for page fault (read-only page and User Mode), and sets segment table Modified-Page bit (8) of accessed page (if no page fault).

MMU → BUS Signals	
BIOON-15N	Data used during Table Store or Page-Fault stack loading operations.
MAD128,64, 0-3	Memory address lines for physical page addresses from the segment table.
TMRN	Activates memory during page address translation, if there is no page fault.
TRMN	External register reply to CPU when software constant is loaded during WER instruction.

C.36 CIRCUIT BOARD AND COMPONENTS

The Memory Management Unit is implemented on a double-sided printed circuit board. Component locations are shown on Figure C-11. The MMU uses connector 3 for GP Bus connections and power, and connector 5 for special CPU control signals. The MMU parts list is given in Table C-1. A list and guide of the integrated circuits used by the MMU is provided in Figure C-12.

C.37 ADDRESS/INTERRUPT U-LINKS

The U-link connections for both the external-register address of the MMU and the interrupt level address are shown on Figure C-11. For both groups of U-links, the least-significant bits are at the top and the most-significant are at the bottom. The examples shown on Figure C-11 are set up for the standard external-register address and standard interrupt level of the MMU (address /80 and level 47_{10}).

INT /2F (= 47_{10})

DA /80

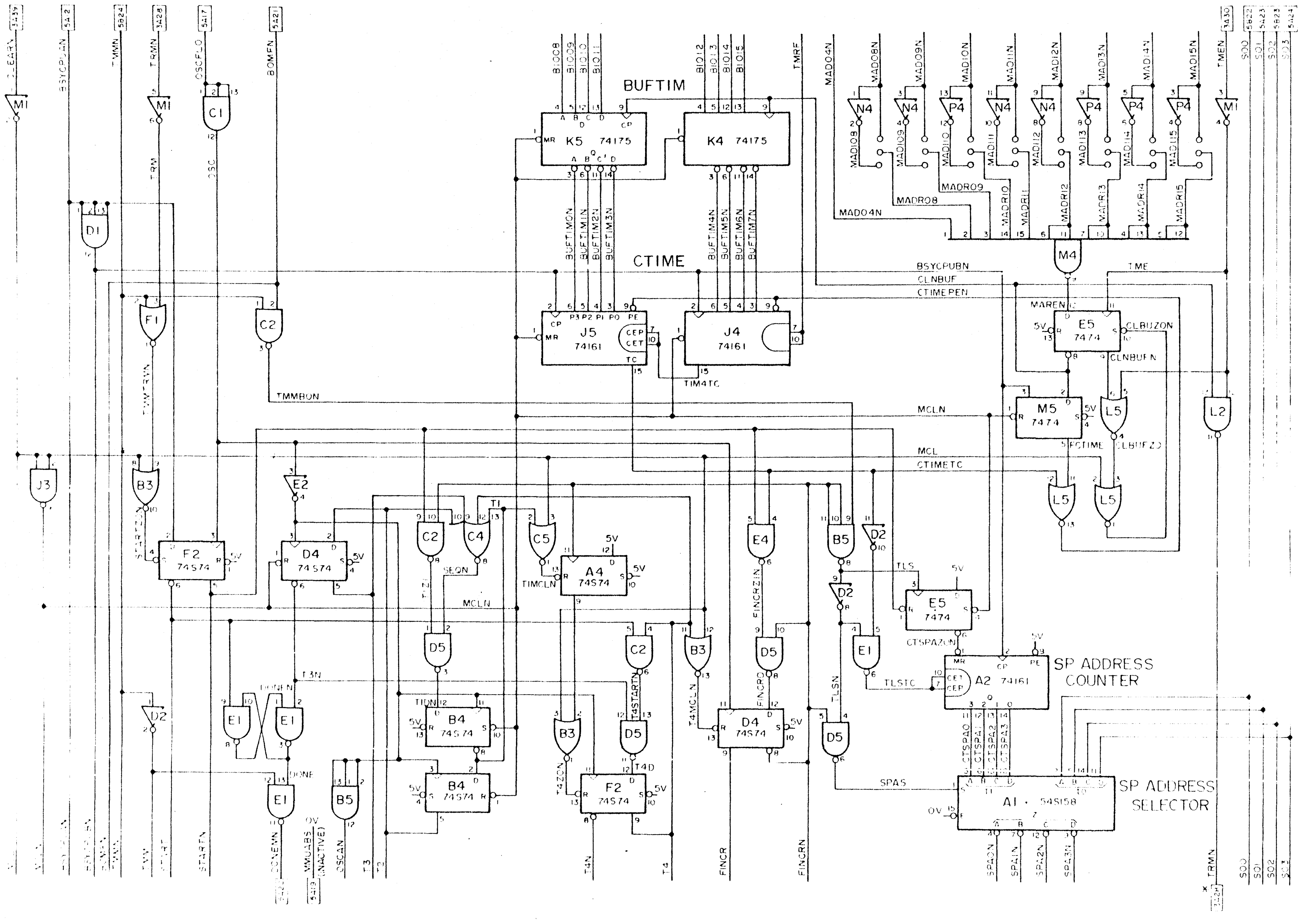


Figure C-10a MMU Logic Diagram

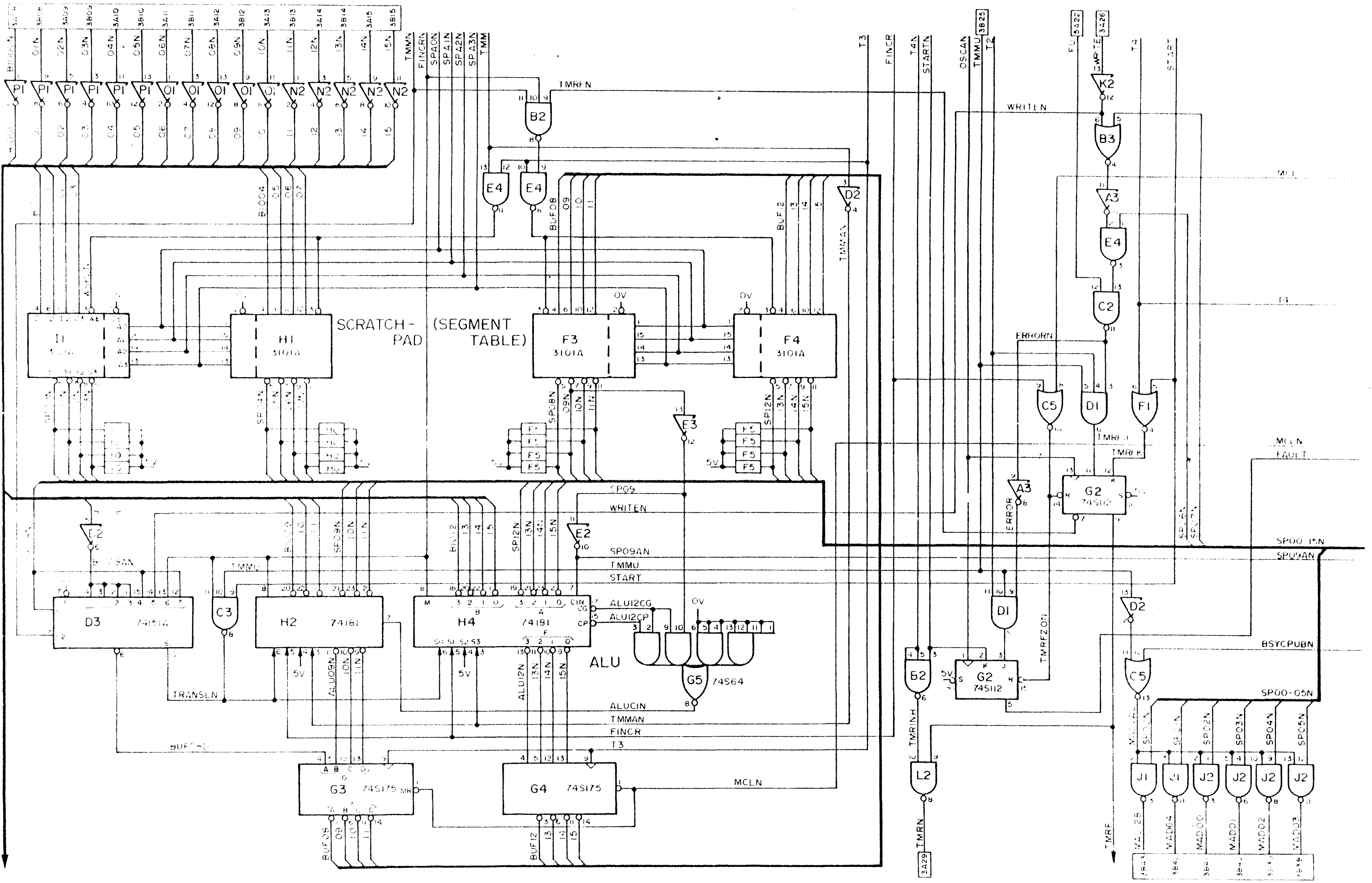


Figure C-10b   MMU Logic Diagram

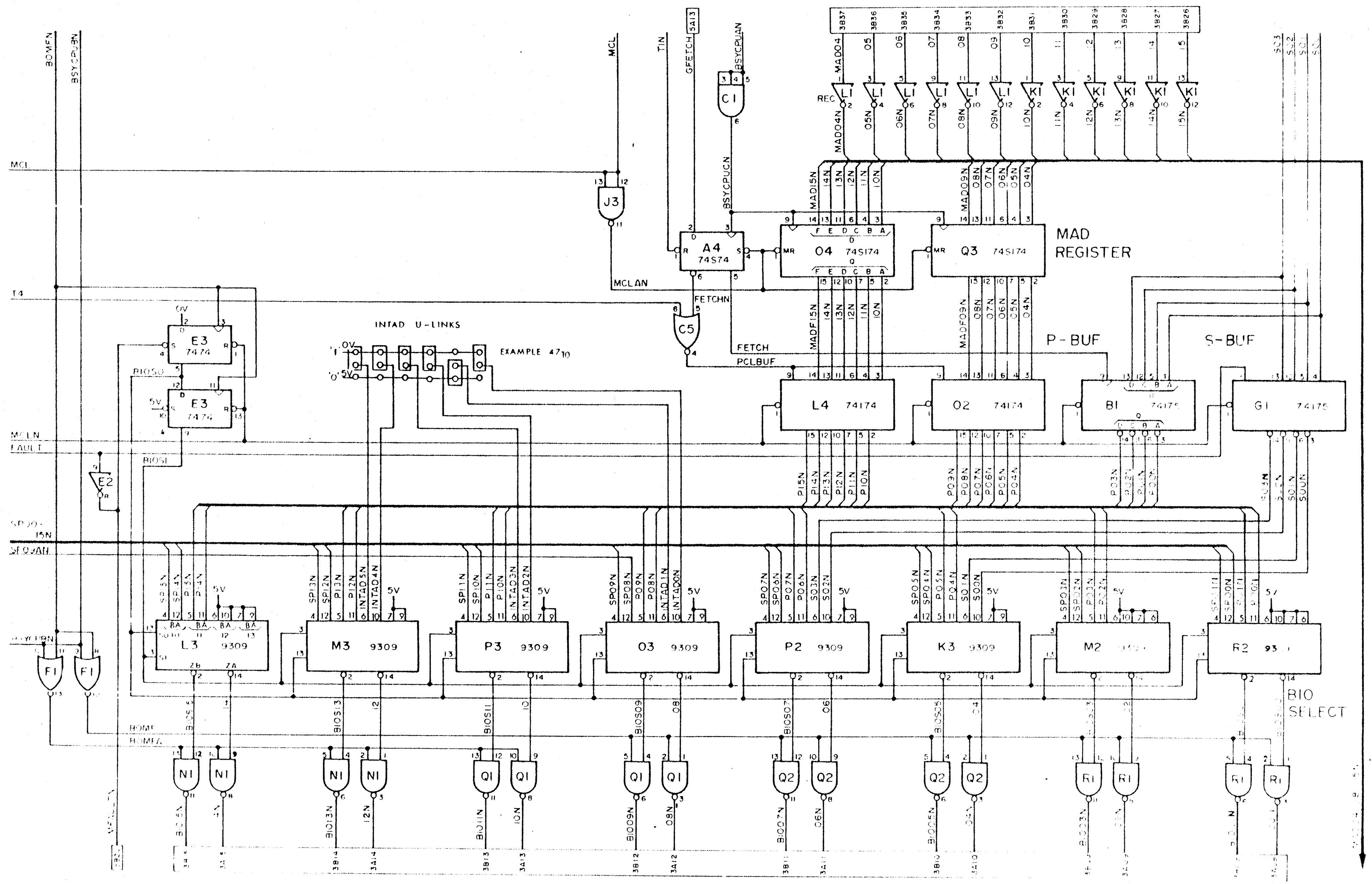


Figure C-10c MMU Logic Diagram

REV

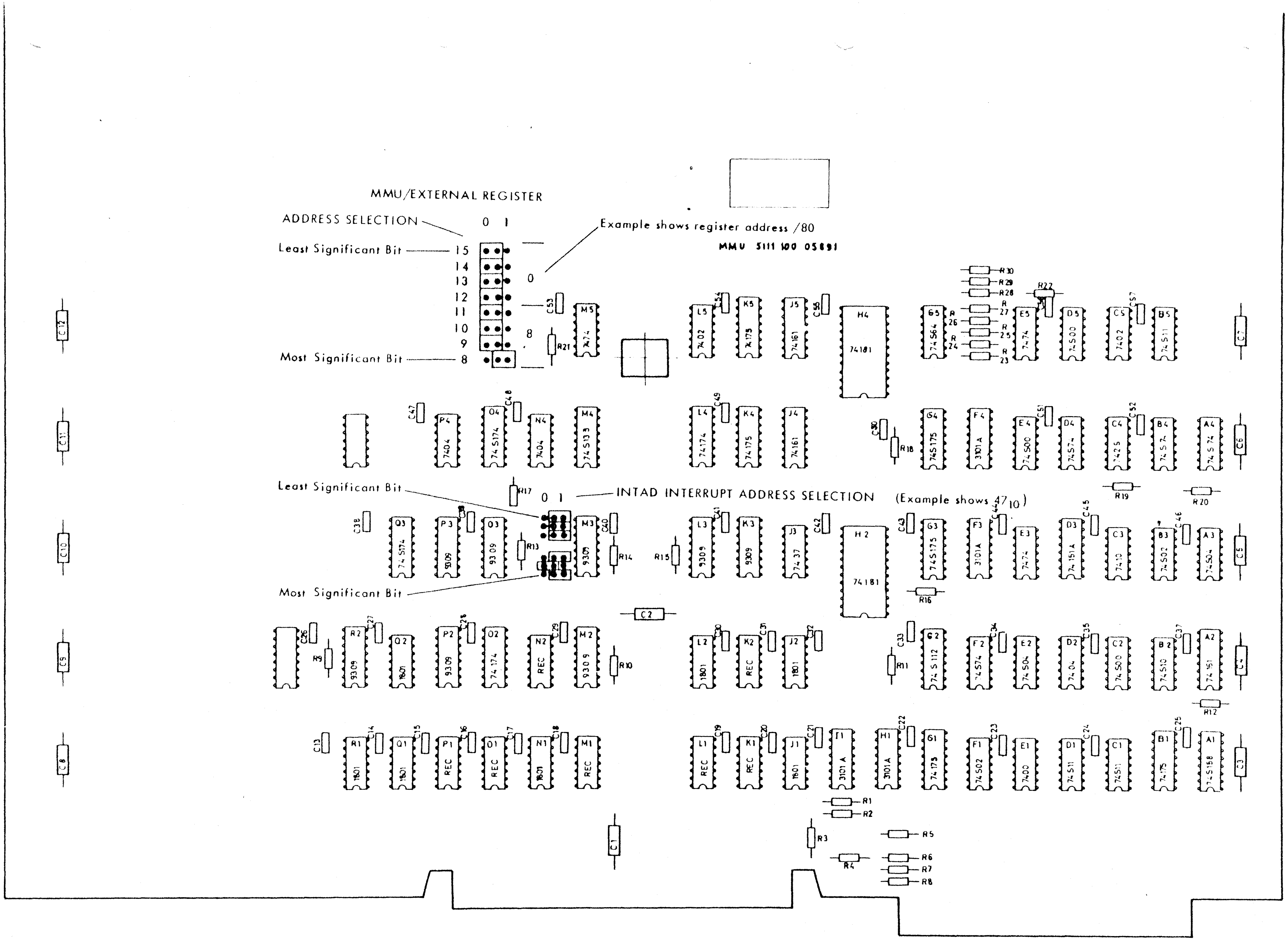


Figure C-11 MMU Card Layout

Table C-1 MMU Parts List

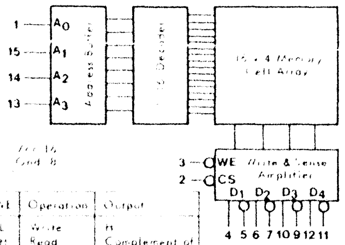
Reference	Description	12NC Code
	Printed circuit	5111 100 05891
A1	Integrated circuit 74S158	
B1 G1 K4 K5.	Integrated circuit 74175	
B5 C1 D1.	Integrated circuit 74S11	
E1.	Integrated circuit 7400	
B3 F1.	Integrated circuit 74S02	
E3 F4 H1 I1.	Integrated circuit 3101A	
J1 J2 L2 N1 Q1 Q2 R1.	Integrated circuit 1801	
K1 K2 L1 M1 N2 O1 P1.	Integrated circuit REC 0613	
A2 J4 J5.	Integrated circuit 74161	
B2.	Integrated circuit 74S10	
C2 D5 E4	Integrated circuit 74S00	
D2 N4 P4.	Integrated circuit 7404	
A3 E2	Integrated circuit 74S04	
A4 B4 D4 F2	Integrated circuit 74S74	
G2	Integrated circuit 74S112	
H2 H4	Integrated circuit 74181	
K3 L3 M2 M3 O3 P2 P3 R2.	Integrated circuit 9309	
I4 O2.	Integrated circuit 74174	
C3.	Integrated circuit 7410	
C4.	Integrated circuit 7425	
C5 E5.	Integrated circuit 7402	
D3.	Integrated circuit 74151A	
E3 F5 M5.	Integrated circuit 7474	
G3 G4.	Integrated circuit 74S175	
G5.	Integrated circuit 74S64	
I3.	Integrated circuit 7437	
M4.	Integrated circuit 74S133	
O4 O3.	Integrated circuit 74S174	
R9-22.	Resistor 1KΩ, 0.25W, 5%.	
R1-8, 23-30.	Resistor 560Ω, 0.25W, 5%.	
C1,2.	Capacitor 47μF, 10V, FITCO.	
C3-12.	Capacitor 10μF, 25V, FITCO.	
C13-35, 37-57.	Capacitor 10nF, 20%, ceramic.	
	U Link DCW 06	

1801

(see 7400)

3101

3101 HIGH SPEED FULLY DECODED 64 BIT MEMORY



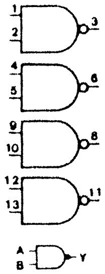
Vcc 16
Gnd 8

CS	WE	Operation	Output
L	L	Write	H
L	H	Read	Complement of Written Data
H	L		H
H	H		H

7400

7400
S00
H00
03[1]
37[2]
38
1801[1]

1 open collector
2 buffer

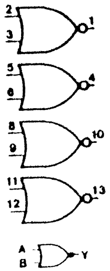


$Y=\overline{AB}$

Vcc 14
GND 7

7402

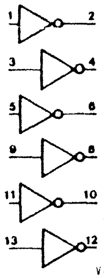
QUADRUPLE 2-INPUT
POSITIVE-NOR GATE



Vcc 14
GND 7

7404

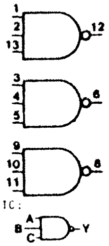
7404, S04, H04
7406 o.c. 30V
7414 sh. Trig
7416 o.c. 15V



Vcc 14
GND 7

7410

7410

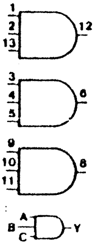


POSITIVE LOGIC:
$Y=\overline{ABC}$

Vcc 14
GND 7

7411

7411



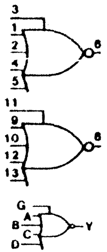
POSITIVE LOGIC:
Y=ABC

Vcc 14
GND 7

7425

7425

4-INPUT NOR
WITH STROBE



POSITIVE LOGIC:
$Y=\overline{G(A+B+C+D)}$

Vcc 14
GND 7

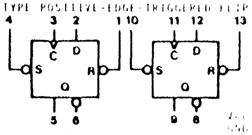
7437

(see 7400)

7474

7474, H74, S74

DUAL D-TYPE POSITIVE-EDGE-TRIGGERED FLIP-FLOPS



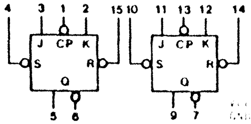
Vcc 14
GND 7

FUNCTION TABLE					
INPUTS				OUTPUTS	
S	R	C	D	Q	$\bar{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H	H
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q_0	$\bar{Q}_0$

74112

74S112

J-K NEGATIVE-EDGE-TRIGGERED FLIP-FLOPS



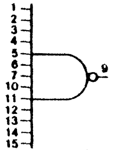
Vcc 16
GND 8

FUNCTION TABLE						
INPUTS					OUTPUTS	
S	R	C	J	K	Q	$\bar{Q}$
L	H	X	X	X	H	L
H	L	X	X	X	H	H
L	L	X	X	X	H	H
H	H	↓	L	L	Q_0	$\bar{Q}_0$
H	H	↓	H	L	H	L
H	H	↓	L	H	L	H
H	H	↓	H	H	TOGGLE	
H	H	H	X	X	Q_0	$\bar{Q}_0$

74133

74S133

13-INPUT NAND GATES

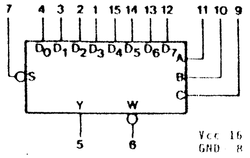


Vcc 16
GND 8

Figure C-12a MMU IC Guide

74151

DATA SELECTOR/MULTIPLEXER



FUNCTION TABLE

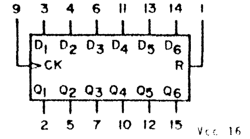
INPUTS				OUTPUTS	
SELECT			STROBE		
C	B	A	S	Y	W
X	X	X	H	L	H
L	L	L	L	D0	D0
L	L	H	L	D1	D1
L	H	L	L	D2	D2
L	H	H	L	D3	D3
H	L	L	L	D4	D4
H	L	H	L	D5	D5
H	H	L	L	D6	D6
H	H	H	L	D7	D7

D0,D1...D7 = the level of the D respective input

74174

74174

HEX D-TYPE FLIP-FLOPS WITH COMMON CLOCK AND RESET

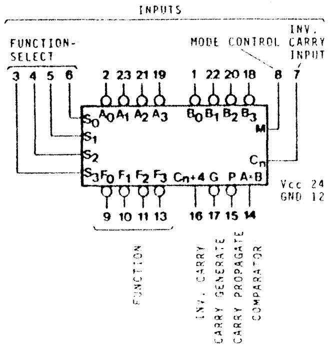


FUNCTION TABLE

INPUTS			OUTPUTS	
R	C	D	Q	$\bar{Q}$†
L	X	X	L	H
H	↑	H	H	L
H	↑	L	L	H
H	L	X	Qo	$\bar{Q}$o

74181

ARITHMETIC LOGIC UNIT



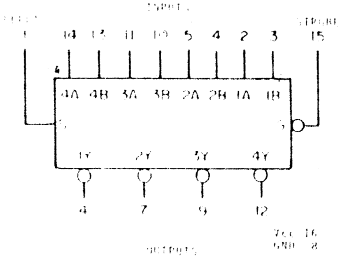
REC 0613

(see 7404)

74158

74158

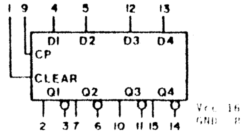
QUAD 2-TO-1-LINE DATA SELECTORS/MULTIPLEXERS



74175

74175

QUAD D-TYPE FLIP-FLOPS



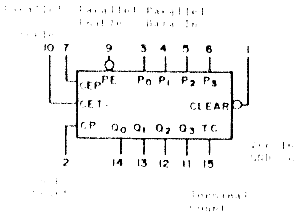
FUNCTION TABLE

INPUTS			OUTPUTS	
R	C	D	Q	$\bar{Q}$
L	X	X	L	H
H	↑	H	H	L
H	↑	L	L	H
H	L	X	Qo	Qo

74161

9310 · 74161

DECADE COUNTER
BINARY COUNTER



9309

9309

DUAL, 4-INPUT MULTIPLEXER

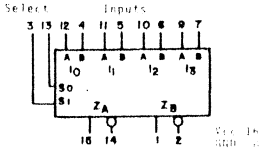


Figure C-12b MAMIL IC Guide

APPENDIX D

FLOATING POINT PROCESSOR

SERVICE MANUAL

Effective Pages March 1977

ii	through	iv
D.1	through	D-28

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES