G. Bell

PDP-X Technical Memorandum # 40

Title: Memory Bus Description

Author(s): L. Seligman

Index Keys: Memory Bus
Bus
Parity
Memory
DMA Channel

Distribution Keys: A

Obsolete: 14

Revision: none

Date: January 31, 1968

Not used

[illegible]

INDEX

1.0 Introduction

The PDP-X memory bus is the interconnection facility between one processor module (or extremely fast IO device, standard or customer special) and one or more memory modules. Signals in the bus are interlocked to permit arbitrarily long lines without degrading performance when memory is close to the processor. The electrical properties of the memory bus are identical to the IO bus.

2.0 System Organization

The three modules which connect to the memory bus are a) the processor (an extremely fast IO device), b) the memory, and c) the arbiter. Each processor has one single-ended port for connection to the memory system; each memory has two double-ended ports as shown in accompanying Figure 1. Note that any configuration with only one local memory system and two or fewer processors does not require an arbiter.

When there are three or more processors in a system, an arbiter is required. The arbiter multiplexes the second port of each memory so that it is available to all the processors. It contains 4 single-ended input ports, one for each 4 possible processors, and a single output port. One arbiter may be plugged into the next to expand its capability in units of three.

The second memory port may be considered a direct memory access channel through which standard selector channel IO devices or custom-designed very high speed devices are connected. If two or more such devices are to be connected, as in Figure 2, the system has three or more processors and the arbiter, in the guise of a DMA multiplexor, is required.

In true multiprocessor systems where there are two or more local memory systems, the arbiter is also required. Figure 3 gives a diagram of a hypothetical 4 processor system. The memory local to a processor is defined to be the memory which may be accessed without using the arbiter. All processors may be simultaneously communicating with their local memory; only one processor may, at any instant, have access through the arbiter to the entire memory system. The arbiter must be the last connection to the memory bus.

2.1 Rationale

The system organization described above was chosen over the alternatives for the following reasons:

1. The majority of systems sold fit into the very basic system category with the minimal hardware of Figure 1 sufficing.

2. The second level of system complexity, corresponding to the largest PDP-9 systems, is included within the hardware of Figure 2.

3. The hardware of (1) and (2) is well understood and is consistent with current products.

4. The few multiprocessor systems now sold, including the 338 display, interconnected PDP-8's, and PDP-6/10 - 680 systems require only loose, low data rate interconnection although each processor is rather active. Since this trend is expected to continue, the loose interconnection scheme is sufficient.

5. If necessary, a more complex arbiter may be defined which would provide full interconnection capability. Note that the arbiter function here is particularly simple since all signal timing and address detection is done in the memories.

2.2 Naming

Each memory port examines the high order 5 bits of incoming addresses to determine if the request is directed to it. If it is, the memory immediately raises the ADDR EXIST line, and when ready, begins a cycle. Two sets of switches are provided in each memory, one on each port, to specify the high order address bits to which the memory will respond.

In systems without the optional protection feature, each processor sees its local memory named starting at address 0 and extending upward to the limit of the local addresses. Non-local memory is named consecutively upward from the top of local memory to the installed capacity. Note that each processor may address a total of 32K words; a particular memory might be addressed differently depending upon which processor was generating the address; and the two sets of switches in the memory would be set differently.

In systems with the protection feature installed, all memory is simply named starting at 0 and extending upward to the installed capacity; this address is applicable to both the local and common bus. A given address always, therefore, refers to the same physical location in the total memory system regardless of which processor has generated it.

2.3 Special IO Devices

A special customer IO device may be substituted for a processor in any system where the data rate of the device warrants it. The memory bus interface seen by the device is, of course, identical to the interface seen by a processor; the same conventions and timing relations hold. Such devices have an additional interface to the IO bus for control purposes. Maximum data transfer rate along the memory bus is determined by the full memory cycle plus arbiter delay plus any cable delay to the device. The practical rate approaches 300,000 words per second for simple devices, a µsec main memory, and the first processor implementation. Data rates in excess of 1,000,000 words per second are possible.

3.0 Operation

When a processor makes a memory request, the request is examined by each memory. If no memory responds, the arbiter passes the request to the entire memory system. This system operates efficiently so long as processors normally communicate only with their local memories. The arbiter determines that no local memory has responded by delaying the request signal and then using it to examine the ADDR EXIST line on the (local) bus. The arbiter forwards any requests for which no local address exists to the entire memory system along the common bus.

3.1 Read, Write, Pause

The processor initiates a memory cycle by placing the desired address on the data lines, control information on the control lines, and then raising REQ. The memory reads the address and acknowledges receipt by raising ADDR ACK. The processor may then lower REQ and disconnect the address.

In the case of a Read or Pause operation as specified by the control lines, the memory will eventually read out the data, place it on the data lines, and raise RD RST to signal this to the processor. If Read, the memory completes its cycle automatically rewriting the data on the data lines until it receives MRLS from the processor. If Pause, the memory waits to receive MRLS. Upon receipt of MRLS, it disconnects its data word from the data lines and reads in the processor-sent data which it writes into memory. In either event, the memory signals that it has obtained all necessary data by sending MRLS ACK.

In the case of a Write operation, the memory never raises RD RST. Instead, it waits for the processor to respond with MRLS and a data word. It stores this word, responds with MRLS ACK and automatically completes its cycle.

The relevant flow diagrams are given in Figures 5, 6, and 7.

3.2 Parity

Parity is implemented for any local memory system by using memories with the parity feature and installing a parity control unit. The former consists primarily of 17th. bit electronics and a special memory core stack; the latter is an IO device.

The control unit calculates parity for all memory data leaving the processor and checks parity of all memory data entering the processor. To do so, it connects to the local memory bus between the processor and the closest memory to be checked. Bus control signals are delayed in the parity control unit to permit sufficient time for the parity calculation and, hence, the total memory cycle time is increased.

The memory bus signal CHECK PARITY is transmitted by memories equipped with the parity feature. Upon receipt of this signal, the parity control unit checks the parity of incoming data; it always calculates the parity for outgoing data. Use of this signal permits memory systems to be mixed, i.e., not every memory need have the parity feature.

Parity is checked on data coming to a processor through an arbiter, but not to a processor connected to the remote bus unless an additional parity control unit is installed. Figure 4 shows a possible configuration. Data between memory M_1 and the main processor P_1 is not checked and data may be read from that memory at the full cycle rate. All other data references are checked and those cycles are slower.

Can still
Solve this
by a [illegible]
[illegible]

bus arbiter

3.3 Hold and Priority

When a memory is receiving requests on both ports, it normally alternates between the two sources rather than servicing them in a wired priority order. That is, the port currently being serviced will not receive the next cycle if both are requesting. For IO devices transferring single words at up to about 300,000 words per second (twice the rate of the PDP-9 disk) this priority algorithm is equivalent to the algorithm used in current products. Two processors may both be connected to a single memory and each will run at approximately half speed.

In order to provide for service at data rates greater than 300,000 words per second, one additional line is provided in the bus, HOLD. The HOLD line is examined during a service cycle on a port. If it is high, no service is provided to the other port while this line remains high. HOLD may be used in the following ways:

1. Dedicate a memory system to one port so that devices transferring at, say, 700,000 words per second need not be heavily buffered. Synchronization problems are avoided.

2. Force transfer of blocks of words, i.e., a 32-bit buffer.

3. Lock another processor out of a memory area when a group of locations must be changed without interruption.

4. Provide a form of priority change when a device double buffer becomes loaded.

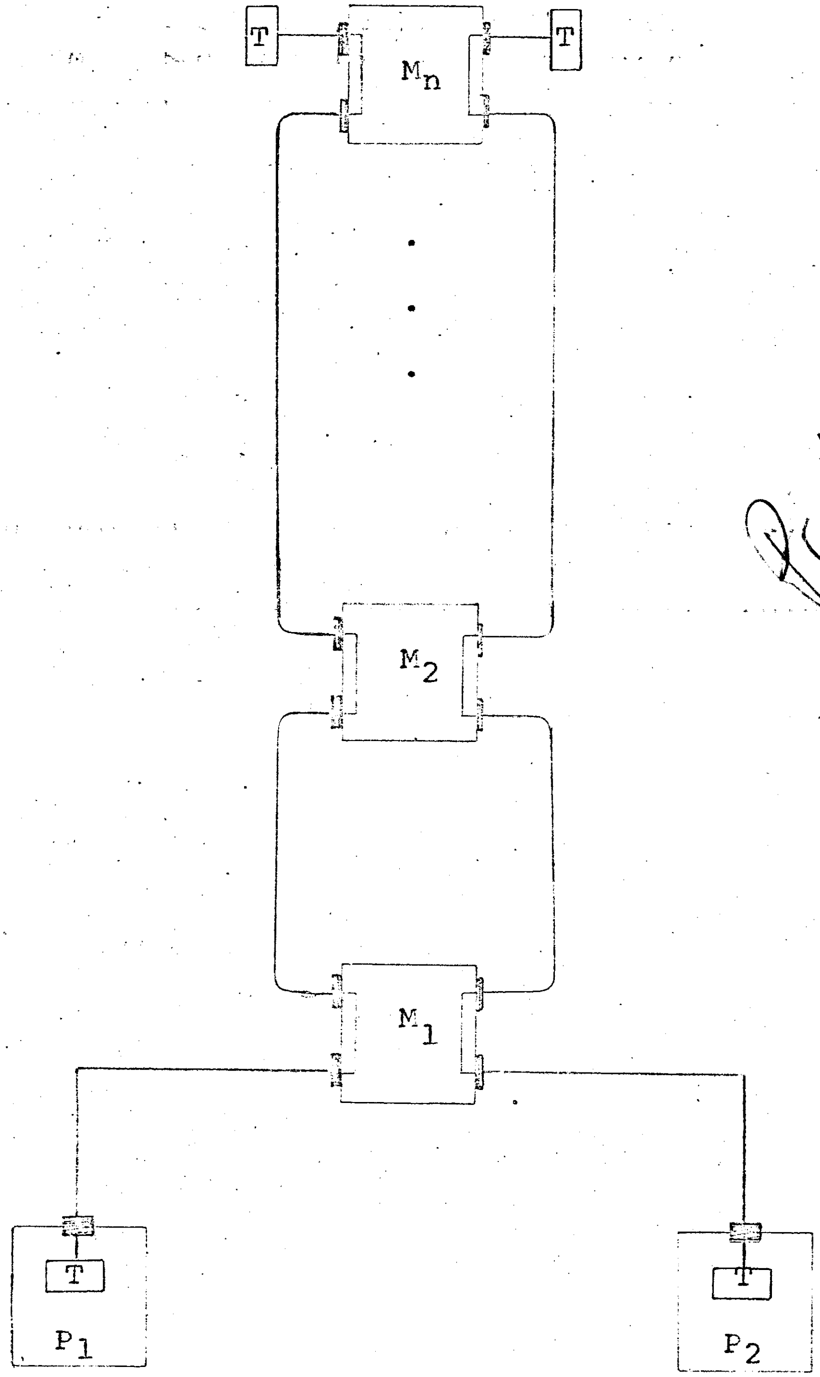
   Thus, a device would normally transfer its outer buffer only, but if both inner and outer buffers were full, it could, by raising HOLD, transfer both buffers in quick succession.

4.0 Eus Signals

Name	Bits	Direction	Comment
DATA	17	both	bi-directional data and address (includes parity)
CON	2	to mem	controls read, write, or pause as in the MC bits of the control memory
REQ	1	to mem	memory request, raised 50 ns after address and control information above is set up
ADDR ACK	1	to proc	address acknowledge, raised by the memory signifying that it has read in an address and commenced
RD RST	1	to proc	read restart, indicates that the memory has placed the data word onto the DATA lines
MRLS	1	to mem	memory release, indicates that the processor has read the memory data and has (in case of Write or Pause) placed the information to be written onto the data lines
MRLS ACK	1	to proc	memory release acknowledge, signals receipt of MRLS and loading of a data word
ADDR EXIST	1	to proc (to arbiter)	address exists, rises within 100 ns of REQ when the memory recognizes the address. This occurs whether or not the memory is ready to perform a cycle.
CHECK PAR	1	to proc (to parity option)	check parity of this data, indicates that the cycling memory has the parity feature
HOLD	1	to mem	insures that this port receives subsequent cycle also
PWR	1		remote turn on
GND	1		heavy ground wire interconnection

5.0 Figures

Figure 1



P = processor module
M = memory module
T = bus terminator

[illegible] Kludge
Slower [illegible]?
Errors.

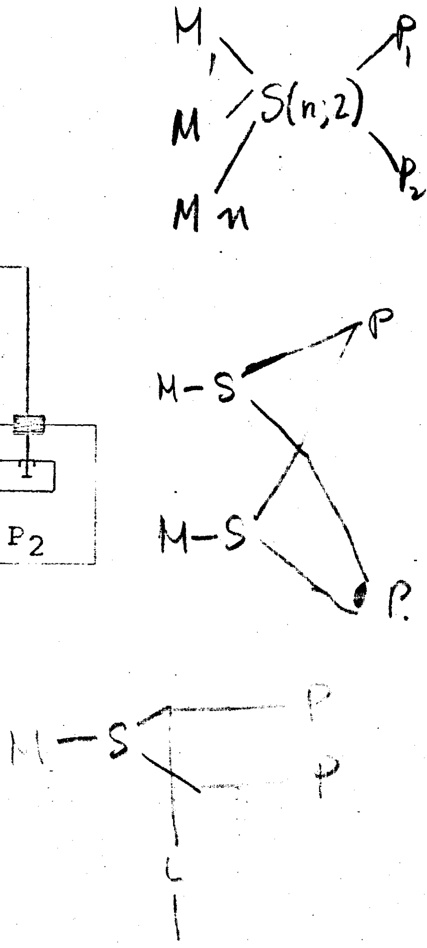


Figure 2

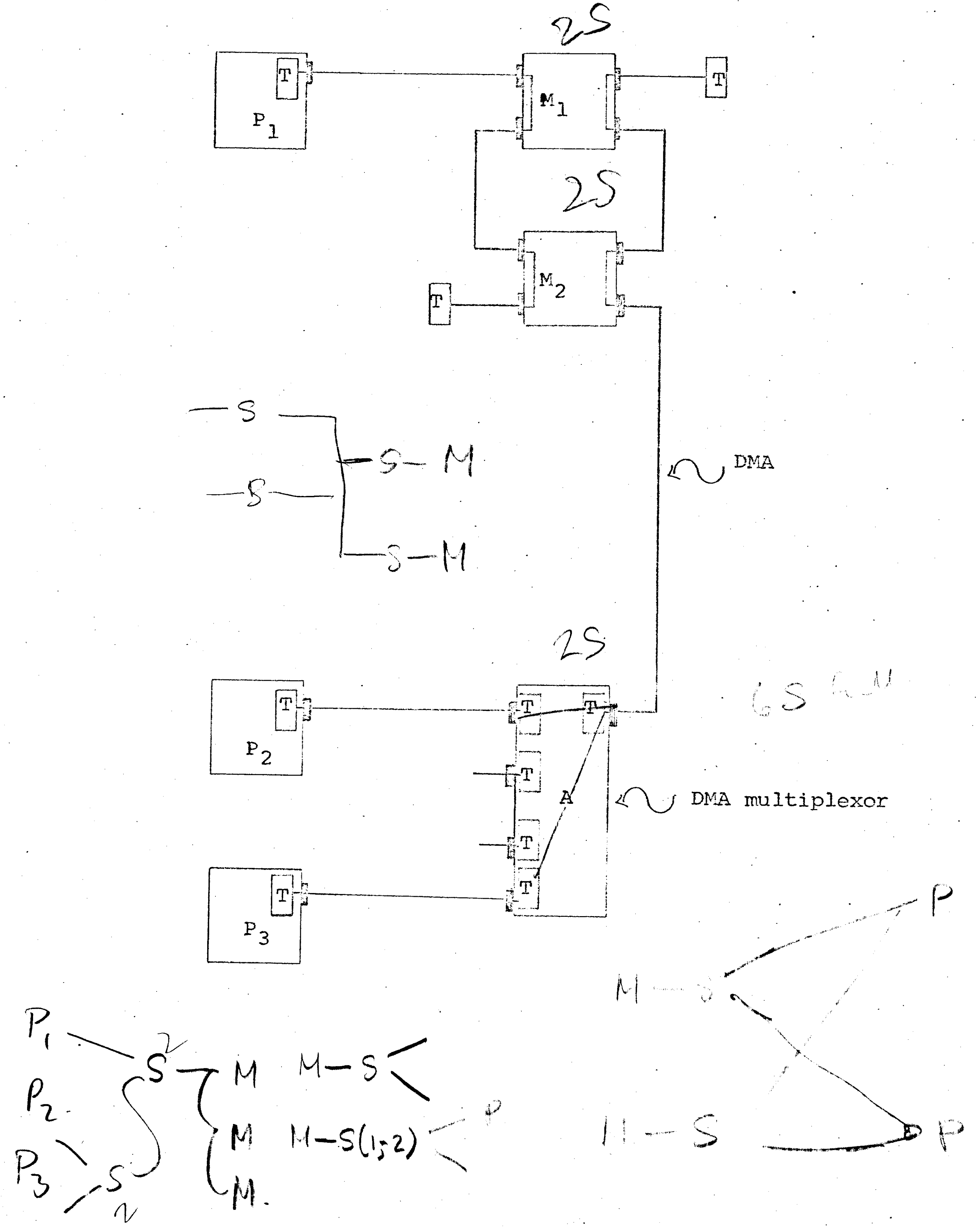
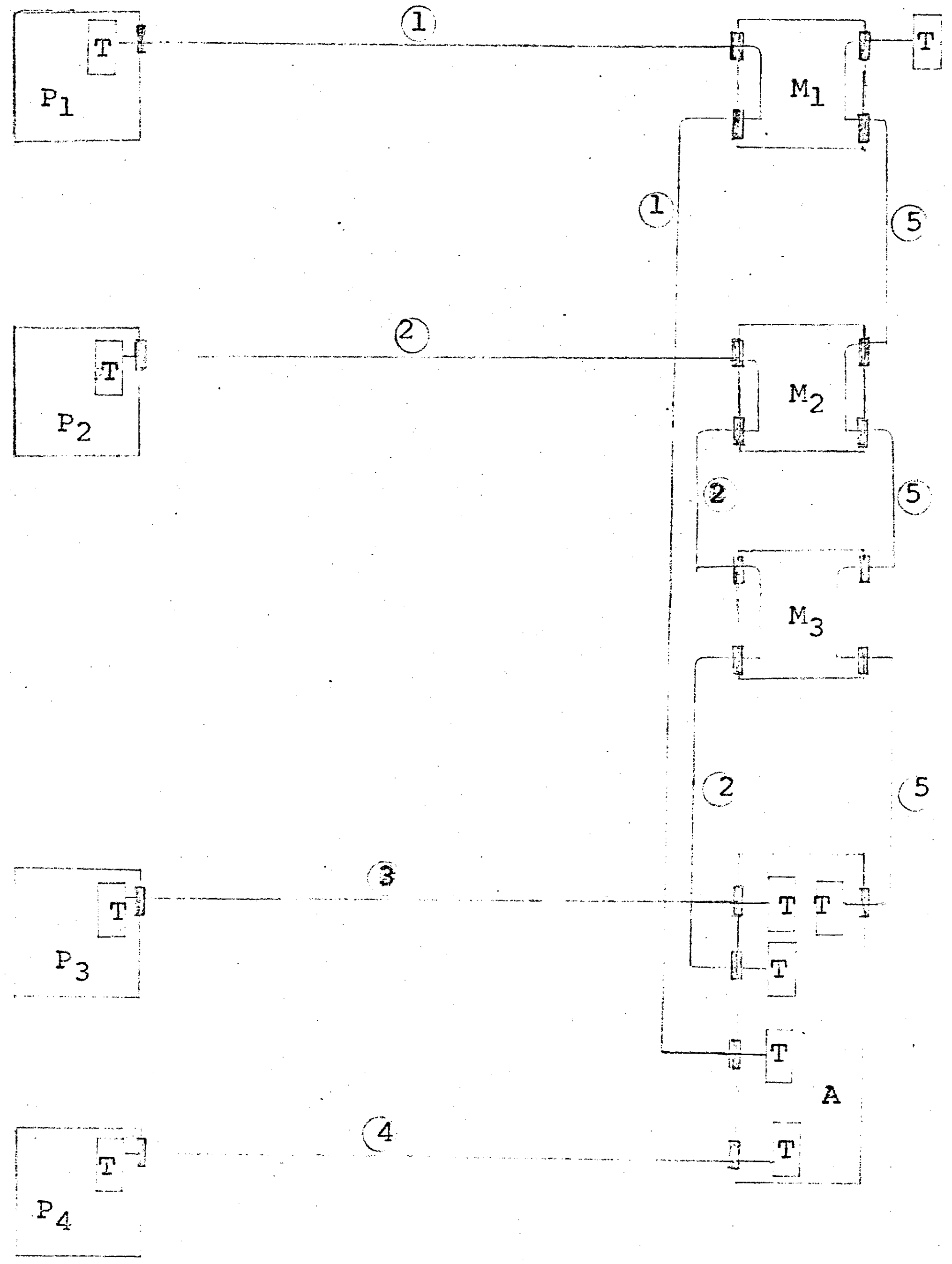
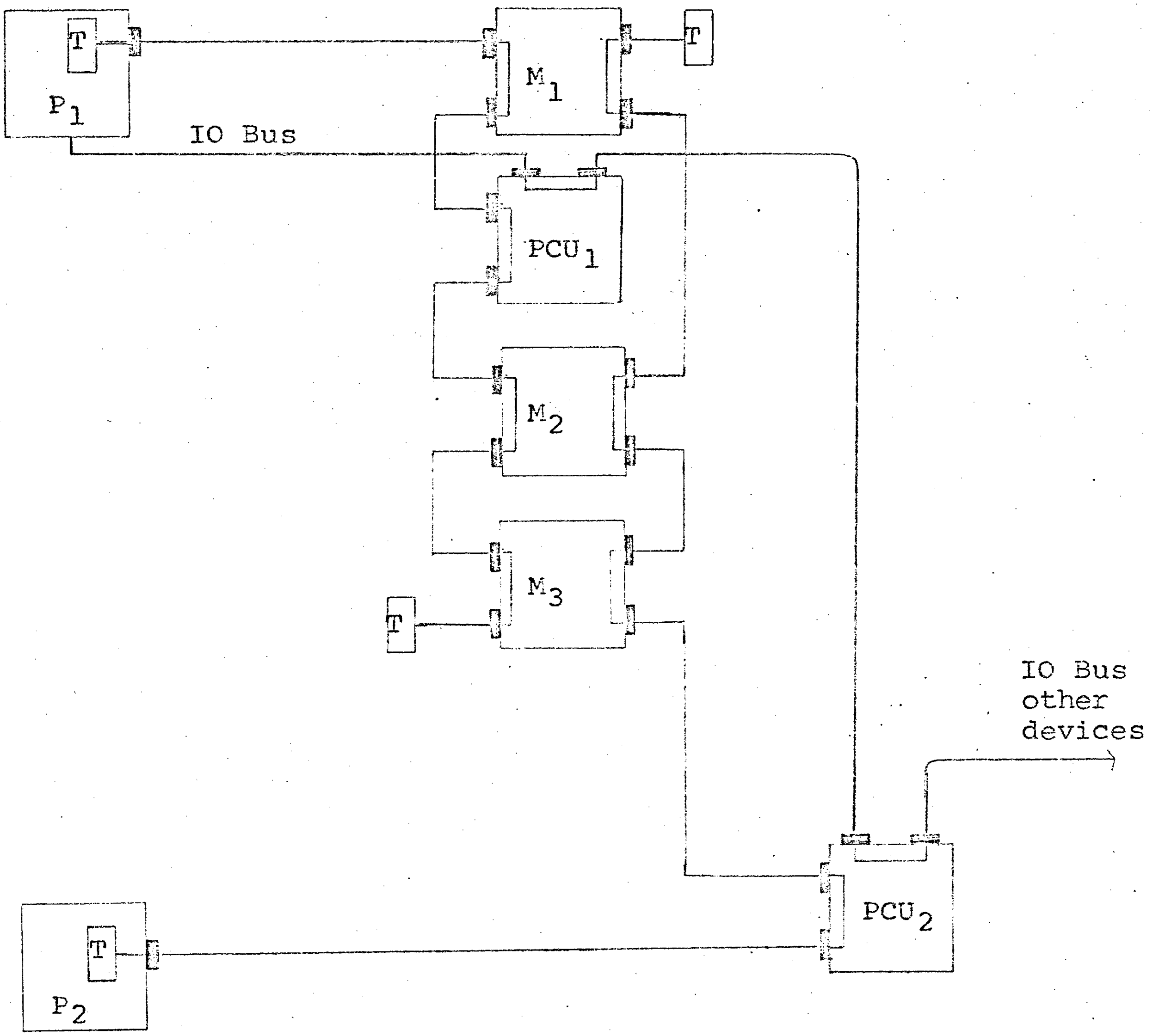

2S
T
P1
M1
T
2S
M2
T
S
S—M
S
S—M
DMA
2S
T
P2
T
T
T
A
DMA multiplexor
T
T
T
P3
6S
P
M—S
P
P1
S
M
M—S
P2
M
M—S(1;2)
P
P3
S
M.
M—S
P

Figure 3



A = arbiter
Note - there are 5 busses shown above

Figure 4



*Note: PCU is parity control unit.

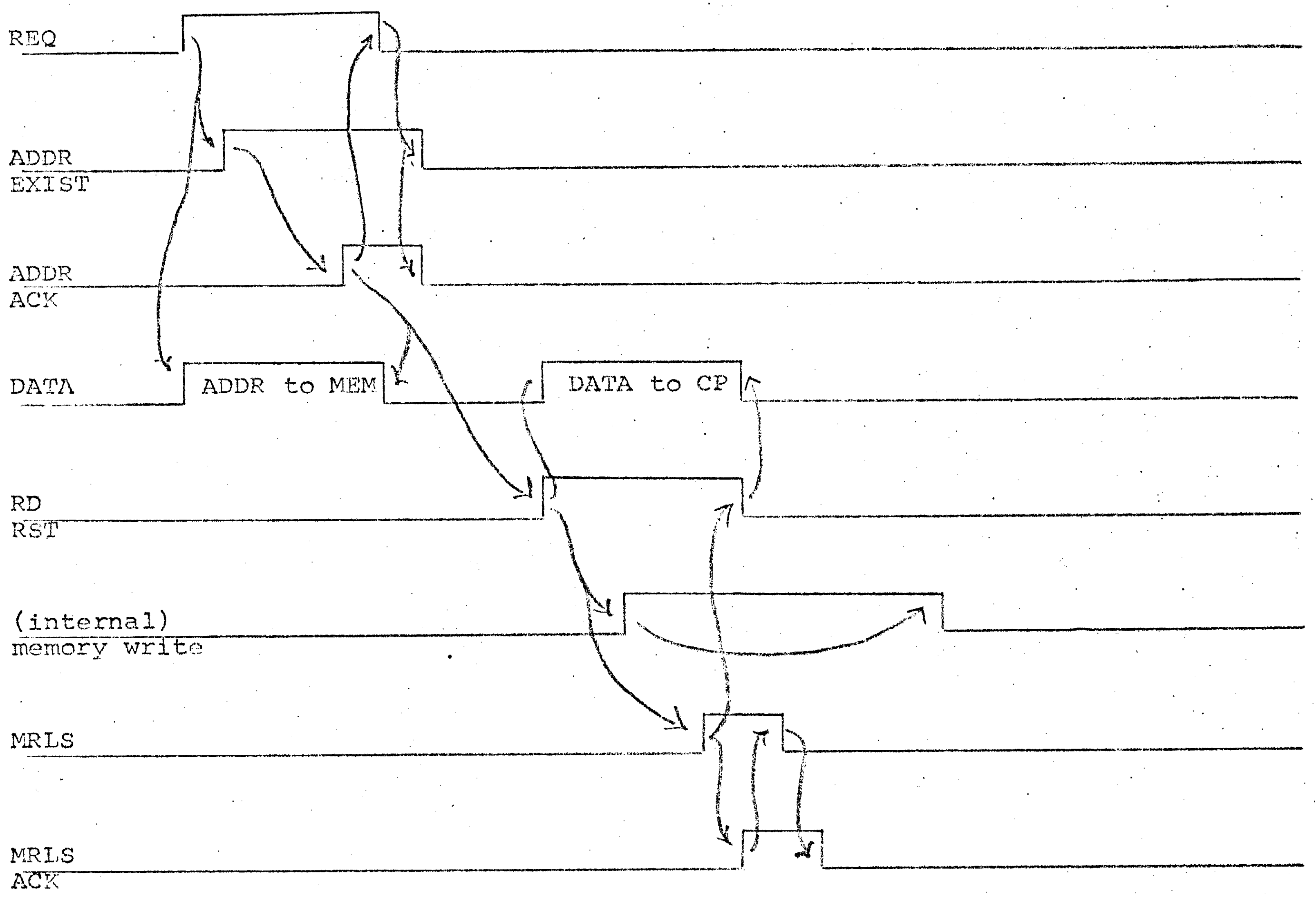


Figure 5 READ

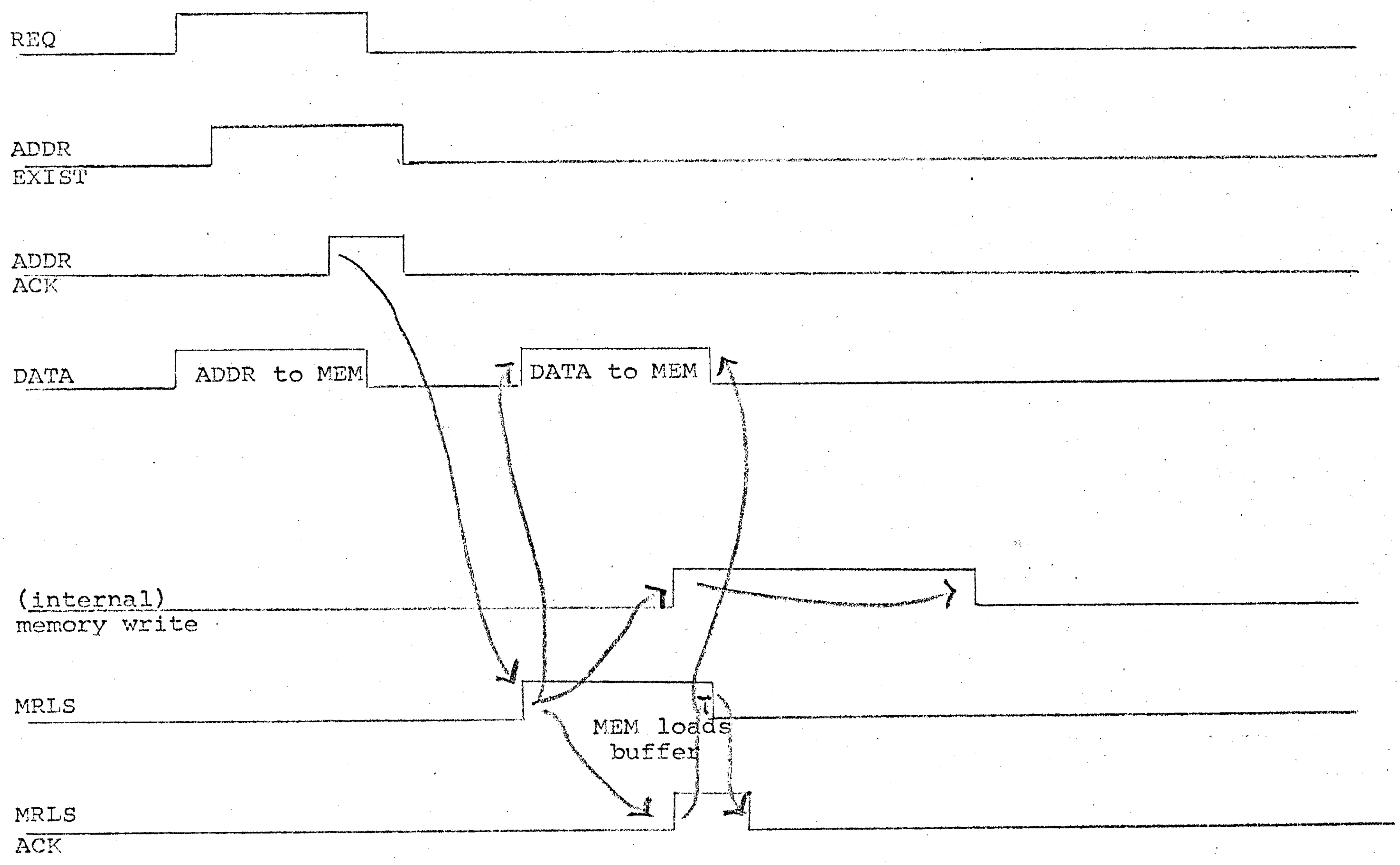


Figure 6 WRITE

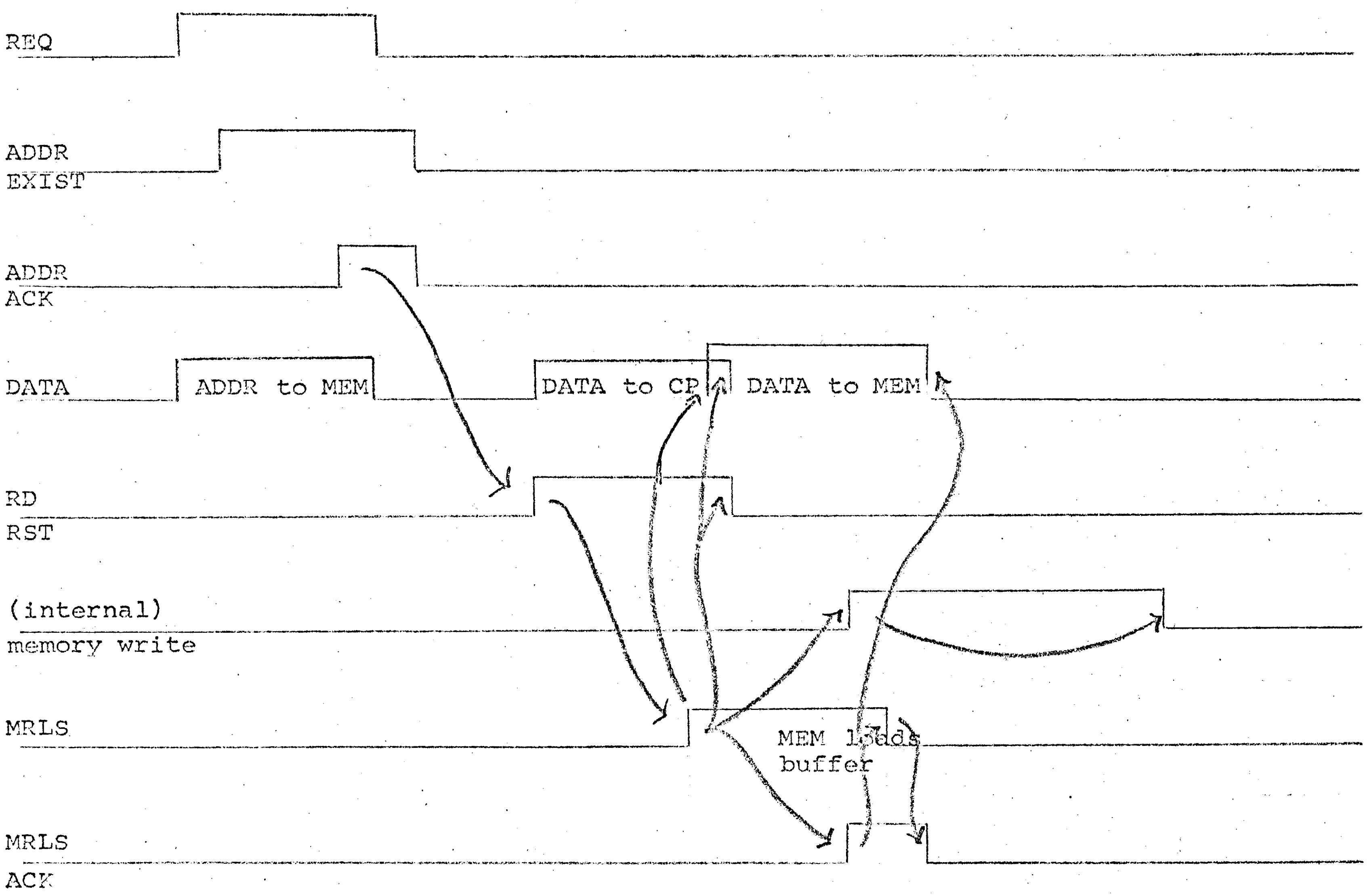


Figure 7 READ Modify WRITE