

B. Bertolucci and D. Horelick
Stanford Linear Accelerator Center
Stanford University, Stanford, California 94305

ABSTRACT

A programmable sequencer and a memory module have been designed and built to demonstrate high speed operation of the FASTBUS, and to study design implications of the FASTBUS specification. Both are implemented in ECL, and illustrate master and slave operation, arbitration circuit design, and logical and geographical addressing considerations.

INTRODUCTION

Early in the prototyping stage of FASTBUS development it was suggested to build a versatile high-speed sequencer, and a RAM based memory module to demonstrate high-speed operation of FASTBUS, and the possible varieties of data transfer protocols. These modules are the result of that effort, and are now running in a development system at SLAC. Discussion of running performance and software aspects of this project can be found in a companion paper entitled "A FASTBUS Demonstration System."

The design philosophy and nomenclature reported in this paper are compatible with the FASTBUS spec current during 1979-1980, although it should be recognized that changes have taken place since that time.

Both modules are built on FASTBUS kluge cards using conventional wire-wrap, except that each line is terminated at the "receiving" end in 100Ω to -2V, and routing is optimized for minimum wiring length. Due to wide availability, and ability to properly drive wire-wrap, MECL 10000 was the preferred logic class used in these designs.

GENERAL DESCRIPTION AND OPERATION

Memory

The memory is based upon ECL memory chips, 4 × 256, organized in a 32-bit wide by 256 word memory. Soft-loadable logical addressing is provided in a control space register accessible via geographic mode in startup, as provided in the FASTBUS standard. The entire memory located in "data space" is then addressable in the conventional two-cycle mode (address/data) or in the three cycle mode (module address/extended address/data). "Control space" is likewise addressable in two or three cycle mode. Figure 1 illustrates a simplified block diagram of the memory.

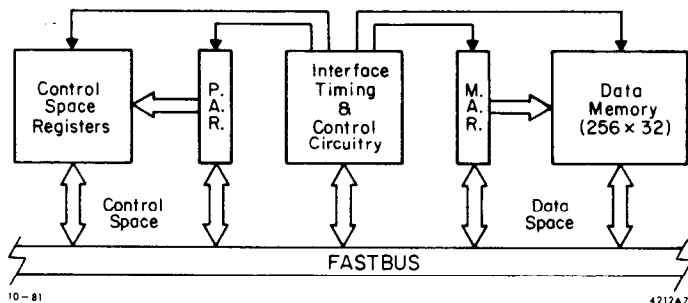


Fig. 1. Memory module simplified block diagram.

* Work supported by the Department of Energy, contract DE-AC03-76SF00515.

Block transfers of arbitrary length, in accordance with the FASTBUS standard, can be used to read or write sequentially from/to the RAM memory starting with the address initialized during the address portion of the cycle.

Parity generation and/or checking can be performed on all transfers, and a self-contained parity error counter can keep count of errors.

The RAM memory operates only in slave mode, and therefore contains no arbitration logic associated with master operation.

General specifications for the memory module are summarized in Table I.

TABLE I. Memory Module Specifications

Data Memory:	256 words × 32 bits
Memory Elements:	Fujitsu MB7071H 4bits × 256 words 10 nsec access time
Memories and Registers:	Loadable and Readable in Slave Mode
Number of chips:	92
Power:	-5.2V @ 4.1A. -2V @ 1.3A.
Symmetrical address implementation of data and control spaces.	
Parity optional on all transfers. 8-bit parity error counter.	

Sequencer

The sequencer is a programmable device based on the same ECL RAMS, which can act as a FASTBUS master and perform arbitrary sequences of FASTBUS protocols, storing and transferring 32 bits of data on the bus at high speed. Virtually all variations of FASTBUS protocols can be exercised, although at the present time block transfers are not yet implemented in the interest of expediency. The basic organization is shown in Fig. 2, which shows the essential components--the 256 word control memory containing the desired sequence of FASTBUS control signals, AS, DS, CB, NH, and the 512 word data/address memory containing the 32 bits of address or data. The specific location of the data/address memory to be accessed in a particular operation is contained in a 9-bit pointer which is part of the control word.

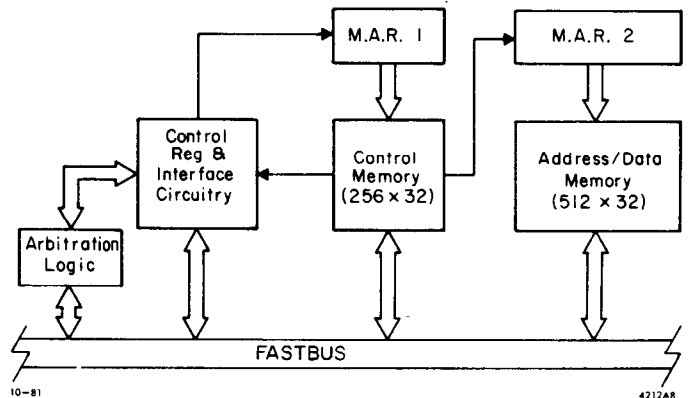


Fig. 2. Sequencer module simplified block diagram.

General specifications of the sequencer are shown in Table II.

TABLE II. Sequencer Module Specifications

Data/Address Memory: 512 words \times 32 bits
 Control Memory: 256 words \times 32 bits
 Memory Elements: Fujitsu MB7071H
 4 bits \times 256 words
 10 nsec access time
 Number of chips: 155
 Power: -5.2V @ 9A.
 -2V @ 2.1A.

All Memories Loadable and Readable in Slave Mode,
 Geographically Addressed

Parity optional on all transfers.
 16-bit parity error counter

protocol can be evaluated, directed to one or more memory modules.

MEMORY SUBSYSTEM

To simplify the construction work a common memory subsystem was developed based upon Fujitsu ECL memories MB7071H. These chips, each organized s 256 words by 4 bits wide, are specified to have a maximum address access time of 10 nsec.

Eight chips are organized into a 32 bit by 256 word memory on a small P.C. board. The memory module utilizes one of these boards, and the sequencer uses one for the control memory, and two for the data/address memory.

A CAMAC tester was designed and built for the memory subsystem and the boards were pretested using a computer controlled CAMAC test system.

MEMORY MODULE DESIGN NOTES

The reader will see from Fig. 3 that the memory module is organized internally around two busses (IN and OUT) buffered from the FASTBUS, each with its own parity circuitry. The memory itself has a eight bit pointer (MAR) and the 4 control registers have a two bit pointer (PAR). To achieve the proper timing in block transfer mode a 32-bit data out register buffers the memory data.

Word Formats for software design are illustrated in Fig. 4. These are essentially developed from the appropriate FASTBUS specification. Note the special feature in geographical addressing--bit 7 controls auto reset of the IA's (pointers) during geographical addressing so that the prior state of the pointers can be read if desired. Bit 3 of Control Register 2 also

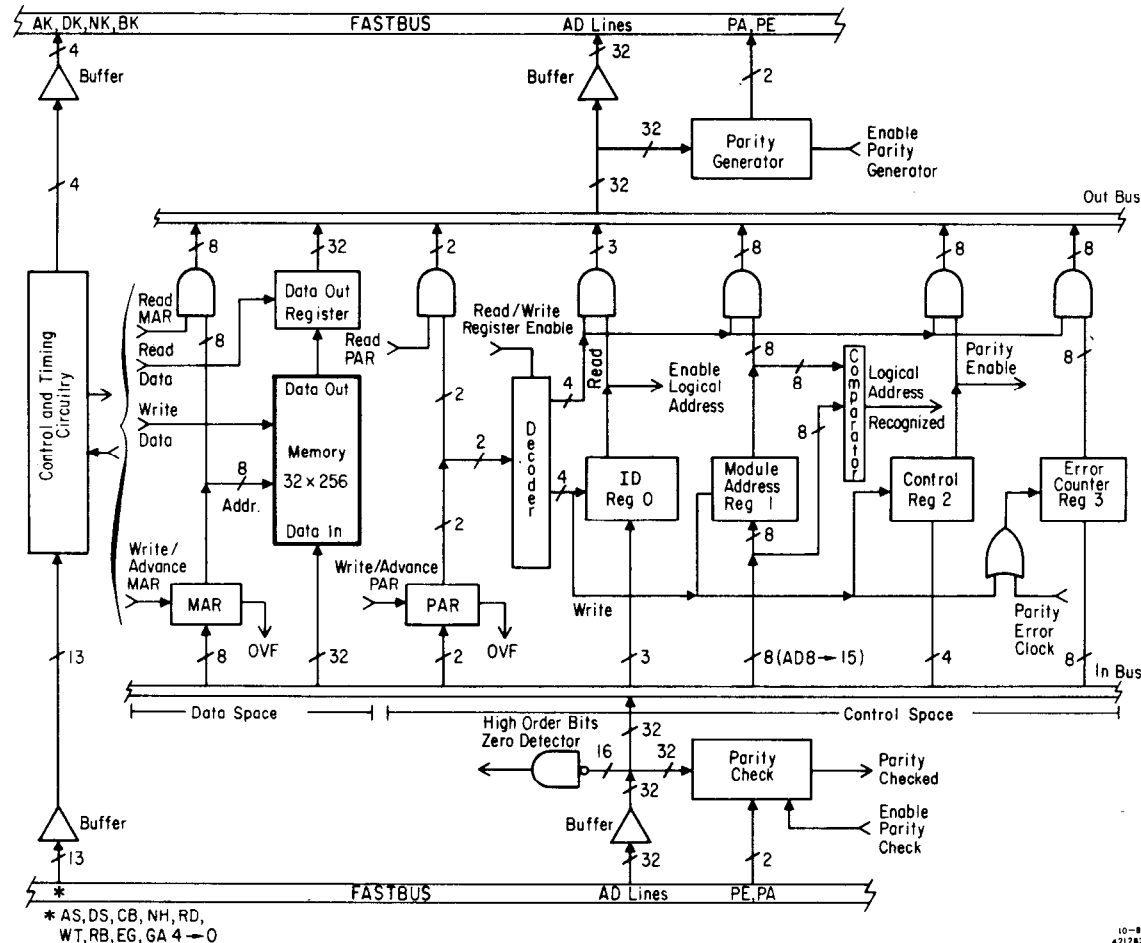


Fig. 3. Memory block diagram.

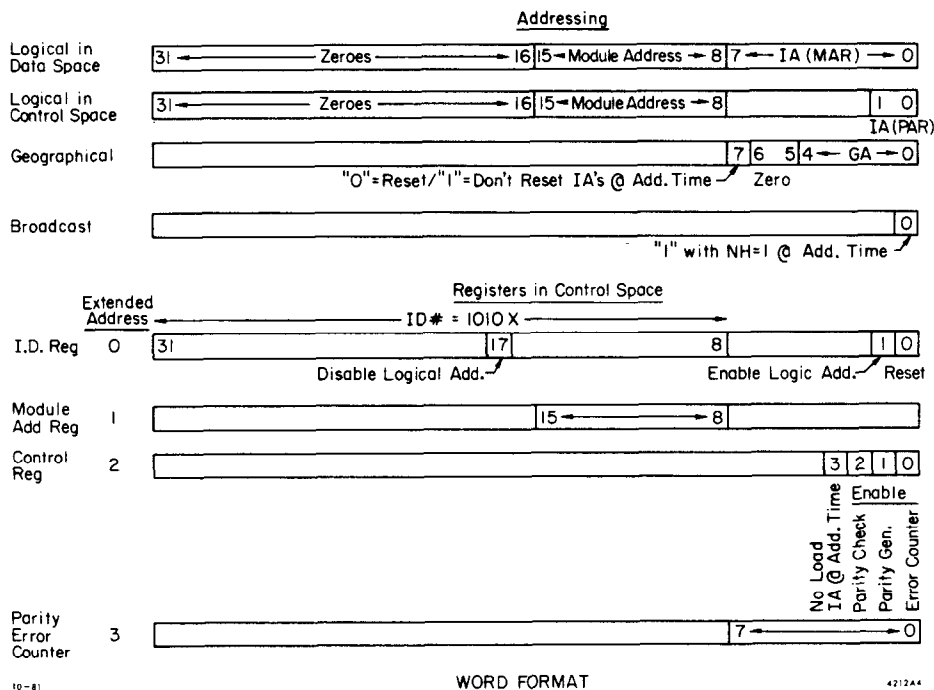


Fig. 4. Memory word formats.

permits direct control of loading of MAR and PAR during the logical address cycle for the same reasons as described above.

SEQUENCER MODULE DESIGN NOTES

The sequencer block diagram in Fig. 5 indicates a design philosophy evolving from the dual master-slave nature of the device. Since speed in slave operation was not considered critical, all registers associated with slave operation are on two additional busses, buffered "inbus" and buffered "outbus." This architecture shortened wiring and reduced bus loading in master operation for higher operating speed and easier debugging. The result is four internal busses in the sequencer without significantly increasing complexity.

For simplicity all accesses in slave mode are geographical, in control space. The case of multiple memories and registers in a single FASTBUS module was carefully considered and it was found that the use of TSR or Extended Address as a "pointer" made hardware design quite straightforward, although it must be recognized that this approach makes loading/reading the control and data/address memories in slave mode slightly awkward in terms of numbers of cycles.

Timing system design was quite challenging, and it was found that the best approach was separate design of master and slave timing, even if this did not optimize hardware. Even with this simplification, timing system design tends to be a somewhat iterative process.

The word formats shown in Fig. 6 are virtually complete for development of software support. In terms of sequencer operation the control word format is most instructive in showing how direct control of each FASTBUS timing/control line is effected, and how the control word contains an indirect address for the 32 bits of data. "Termination" and "arbitration" are bits added for respective control of unique sequencer features. The first bit completes the sequence; the second allows re-arbitration. Each control word thus corresponds to a FASTBUS "microcycle" in which one (or more) FASTBUS control line changes state.

TIMING EXAMPLE

Figure 7 illustrates the FASTBUS timing sequence for a particular operation--Address cycle/Data Write/Data Read--directed to the memory module. This drawing is primarily tutorial in nature, but it also points out the basic philosophy of the timing system design in each unit. Note that in the sequencer each timing cycle must allow for a delay of two memory accesses in each random operation, since the control memory word contains the address of the current data/address memory location. In the example shown a FASTBUS read corresponds to a sequencer memory write and vice versa. CB and NH are illustrated arbitrarily zero, corresponding to a random operation in data space, but of course similar control space operations are possible.

FUTURE PLANS

The present units are prototype wire-wrap units compatible with a prior FASTBUS specification. New versions of the memory and sequencer have been designed which include features compatible with the current version of the FASTBUS specification. This new sequencer also includes block transfer capability in master mode which was temporarily omitted from the original unit for expediency.

It is planned to make printed circuit versions of these designs, although it is expected that the memory would be increased in size to broaden its application.

Future studies include a pipelined approach to the sequencer to speed operation by overlapping memory accesses.

ACKNOWLEDGEMENTS

The authors acknowledge the contributions of E. Barsotti from Fermilab, who supplied the klugeboard for the memory, and L. Paffrath from SLAC who supplied the klugeboard for the sequencer.

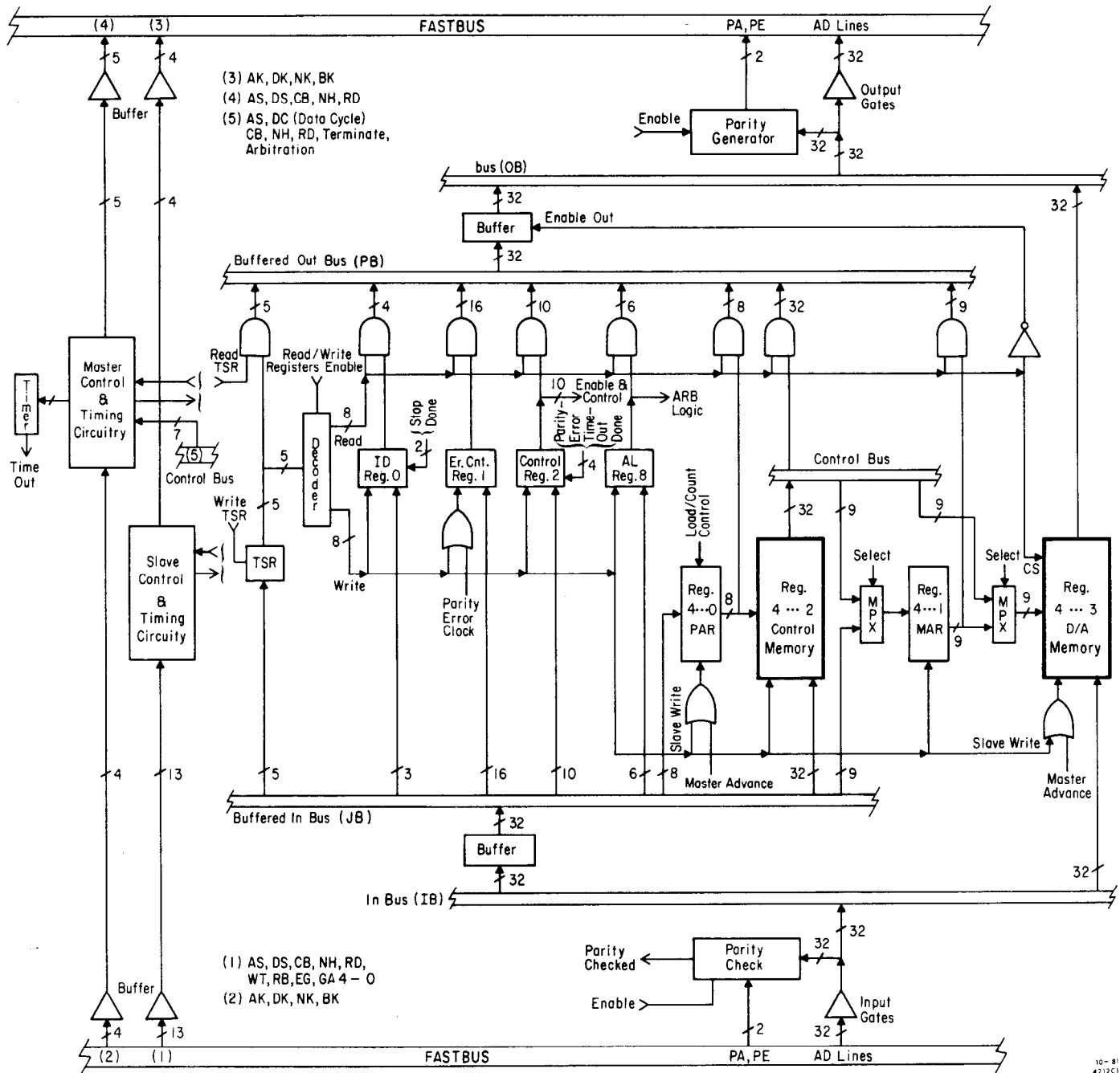
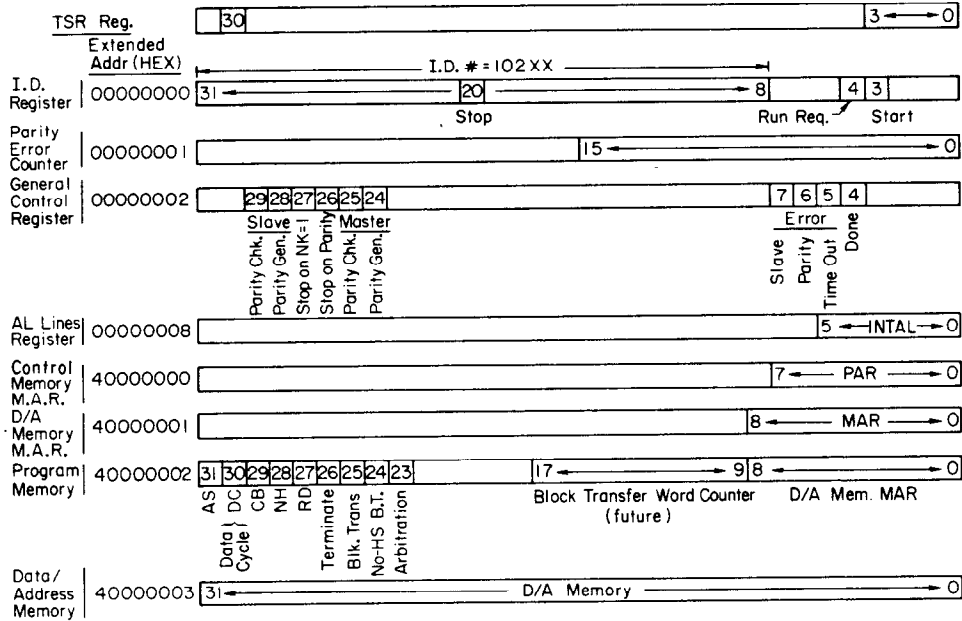


Fig. 5. Sequencer block diagram.



10-81 WORD FORMAT 4212A3

Fig. 6. Sequencer word formats.

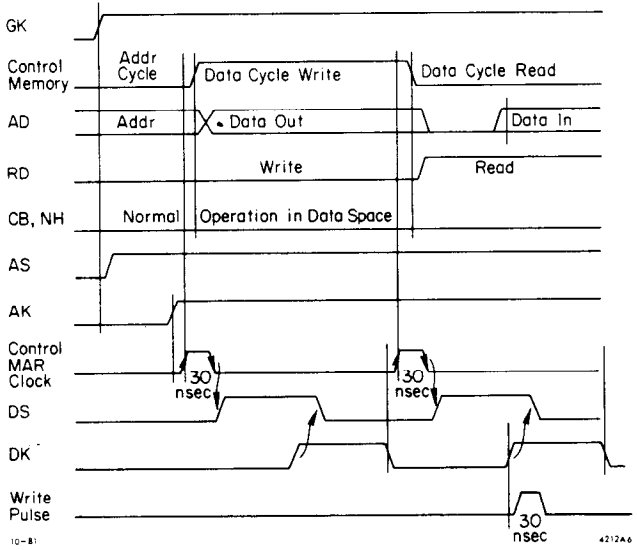


Fig. 7a. Timing diagram - sequencer.

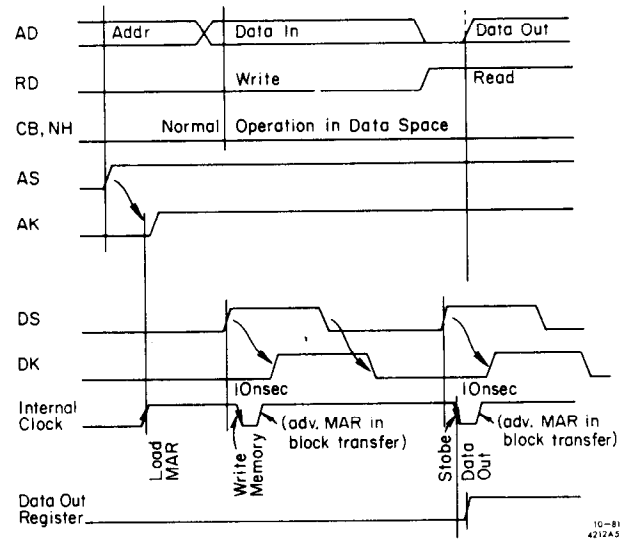


Fig. 7b. Timing diagram - memory.