

MICROPROGRAMMERS GUIDE TO THE CRMS
PROCESSOR-MEMORY CONFIGURATION

Charles A. Grant

July 12, 1972

Revised July 30, 1973

Systems Group

Technical Document

Center for Research in Management Science

University of California

Berkeley

This work was done as part of the Systems Development effort under
National Science Foundation Grant GS-32138.

A META⁴ configured as either the SIMPLE or APL microprocessor will have access to 64K words of core memory. Access to memory and all other external devices is accomplished through four input/output register pairs (Registers 4-7):

D-BUS



B-BUS

A memory access transfers a 34-bit word between the microprocessor and memory. For writing into memory, these words are taken from the D-bus registers MDL(16), MDR(16), and the 2 high-order bits of IODEVICE. When reading from memory, the bits come into B-bus registers MDL(16), MDR(16) and the 2 high-order bits of MEMSTATUS.

There are four different kinds of accesses that can be made to memory:

(a) Read.

A microinstruction which accesses MADDR or MDL on the D-bus with the MR bit turned on (and MW off) will request the initiation of a read to be made from the memory word at the location specified by the contents of MADDR.

(b) Write.

A microinstruction which accesses MADDR or MDL on the D-bus with the MW bit turned on (and MR off) will request the initiation of a write to be made to the memory word at the location specified by the contents of MADDR.

(c) Split.

A microinstruction which accesses MADDR or MDL on the D-bus with both the MR and MW bits turned on will request the initiation of a read cycle to the word addressed by the contents of MADDR. The write part of the memory cycle will be inhibited. The only legal memory access to initiate after a split is a continue (any other request will cause the microprocessor to hang).

(d) Continue.

If the last memory access initiated was a split, then a microinstruction which accesses MADDR or MDL on the D-bus with the MW bit turned on (and MR off) will cause the completion of that cycle to occur, with the data to be written back into the memory being taken from the D-bus registers.

More specific rules about use of the memory access registers follow:

(a) D-bus accesses to MDL and MADDR.

If a previous memory reference has been requested by the microprocessor, and the read cycle part of that access has not yet completed, then any access to MDL or MADDR on the D-bus will cause the microprocessor to pause until that read cycle is completed, plus 90 extra nanoseconds.

MDL and MADDR on the D-bus are actual registers and their values are changed only when this is explicitly done by an instruction. However, it should be noted that regardless of whether or not MADDR is changed by an instruction between the initiation of a split access and the initiation of the corresponding continue, the write will actually be done to the word originally read from.

(b) B-bus accesses to MDL and MEMSTATUS.

Data read from memory is guaranteed to be valid in the B-bus registers MDL and MEMSTATUS only during the interval between the end of the memory read cycle of the access that fetches the data, until the beginning of the next read cycle initiated on the behalf of the processor.

The interval of time between the request for a memory access and its actual completion is a variable, and not a constant, due to possible interference because of accesses of other microprocessor connected to memory. However, if MDL is accessed on the B-bus and a requested memory cycle has not yet completed, then the microprocessor will hang until the cycle completes.

The data does not become valid on the B-bus, however, until slightly after the read cycle completes. Turning on the PZ bit in the instruction which accesses MDL will guarantee that valid data is being read with an exception during the first half of a split cycle. Due to data settle problems at the I/O interface of the memory data register, upon completion of the first half of a split cycle request, MDL should not be used as an operand of arithmetic operations. If it is necessary to use MDL in this manner, MDL should first be copied to a temporary register, or,

if none are available, a NOP should be inserted, such as $0 \leftarrow \text{MDL PZ}$.

It is an error to make a B-bus access to MEMSTATUS with the PZ or IO bits on.

At this point, it should be pointed out that the SIMPLE microprocessor and the APL microprocessor are slightly different. For the SIMPLE machine, bits 8-13 of IODEVICE act as a device to select to multiplex the use of MDR among several devices and memory. When data is stored into MDR on the D-bus, it is always available to the memory port; but when reading from MDR on the B-bus, the device code must be 0 to obtain data read from memory. Furthermore, after a device has been selected by changing the device select bits of IODEVICE, there is a period of settling time that must elapse before data can be read from MDR.

The discussion below applies, as is, to the APL microprocessor and, under these conditions, for the SIMPLE microprocessor.

(c) D-bus accesses to MDR and IODEVICE

When a continue cycle is requested, the data to be written from MDR and bits 0 and 1 of IODEVICE must already be present in these registers. However, when a write request is made, MDR and/or bits 0 and 1 of IODEVICE may be stored into the D-bus following the write request, and the updated value is what will actually be stored in memory. This is true for a period of time--here called MDRTIME.

(d) B-bus accesses to MDR.

Data is valid in MDR on the B-bus during the same period of time as in MDL and MEMSTATUS; however, the PZ and IO bits are ignored when

reading the contents of MDR on the B-bus. Thus, to be sure that data is valid in MDR on the B-bus, it will usually be necessary to access MDL on the B-bus first, using the PZ bit.

One note should be made by the microprogrammer: in a microinstruction, the PZ and IO bits are the same as the MW and MR bits respectively. This can cause problems for the microprogrammer not aware of this, e.g.,

MDL ← MDL PZ ;

This instruction will pause until the data from the last read becomes valid, copy MDL on the B-bus to MDL on the D-bus and then initiate a memory write. Two instructions would have to be used to avoid starting a memory write.

Timing Parameter Values

The values below represent current expectations with regard to the timing parameters given. They are subject to minor changes.

<i>ROM access time</i>	<i>30-45 ns</i>
Basic microinstruction time	90 ns
LOAD microinstruction time	210 ns
Stumble step time	30 ns

A stumble step occurs on any microinstruction in which one or more of the following occurs:

- (a) The PZ (MW) bit is on.
- (b) The J bit is on in RR format microinstruction.
- (c) The operation is a branch.
- (d) MADDR or MDL accessed on D-bus.

Full memory cycle	900 ns
Memory read cycle	450 ns
Max. interference by APL processor	900 + 360 ns
MDRTIME	210 ns
Interval after write initiated that MDR and IODEVICE can be safely modified.	
Device select settle time	150 ns