CHAPTER 2

FUNCTIONAL DESCRIPTION

2.1 GENERAL

The MARK 12 is a 16-bit Data General Nova*-compatible CPU with 128K bytes of onboard static ram. In addition, the MARK 12 can access up to 16 megabytes of external memory located on MARK 12 memory expansion boards, each containing from 2 to 4 megabytes. The CPU uses a pipelined architecture, with a four instruction prefetch queue, and an autonomous Data Channel processor.

The MARK 12 CPU is an extension of that part of POINT 4 Data Corporation's product line that consists of the MARK 5, MARK 8, and MARK 9 computers. It uses the same size printed circuit board, and is compatible with the same peripheral controllers.

2.2 MARK 12 CPU SIGNALS

Signals common to the Data General Nova are defined in the Nova User's Manual (see list of reference documents). Signals unique to the CPU are described below. All are low-active signals, indicated by the "-" suffix.

2.2.1 → 32-BIT HIGH SPEED BUS TO EXPANSION MEMORY

The MARK 12 CPU features a 32-bit high speed synchronous extension memory bus used to access expansion memory boards in the MARK 12 system. Data is transferred on this "V-Bus" along 32 parallel, bidirectional, data lines. Control and status signals are carried along dedicated, unidirectional, control and status lines. This structure uses 41 pins on the 200-pin Nova backplane. The pins used are all taken from the set of pins reserved by the standard Data General Nova pinout for memory lines, that is, pins which are not permitted to be used by custom interface boards.

The 41 pins are defined as follows:

(from MARK 12 CPU Product Specification)

END

ned
de,
with

that
quent

comary
h the
ively,
gh the
onic.

STATUS

a DIA
hes on
cified
on is
t will
. The
n I/O

exactly
has a

which
imulato

n
state

The Hardware/Firmware Identification word is defined as follows:

Left byte = Hardware ID = 12 (octal), defines
MARK 12 CPU
Right byte = Firmware ID = 1 for initial release

The CPU Status Word is defined in Section 2.3.4.

2.3.5.2 LOAD PROGRAM FROM APL PROM

The NIO CPU instruction is used to read in any program contained in the APL PROM on the CPU board and to begin executing it.

A0 = Address in the APL PROMs from which the program will be loaded

A1 = Number of words to be loaded

A2 = Onboard memory address where the program will be loaded

A3 = Starting address in onboard memory to begin program execution

All accumulators are preserved.

2.3.5.3 EXPANSION MEMORY ACCESS

2.3.5.3.1 CPU ACCESS TO EXPANSION MEMORY

The MARK 12 CPU uses four pairs of I/O instructions to access expansion memory; one for multiple word transfers to or from onboard memory, one for single "block" transfers to or from onboard memory, and two for single word transfers to or from accumulator 2. One of the two pairs of I/O instructions used to transfer single words also allows Data Channel Map translation. However, all other expansion memory access instructions have no map translation. In all of the following descriptions, the mnemonic "MEM" represents the device code, which is 2 for expansion memory access.

				Single word xfers			
multiple words		block xfer		no DCH		DCH	
DIA	DOA	DIB	DOB	DIC	DOC	DIC	DOC

END

2.3.5.3.1.1 MULTIPLE WORD TRANSFERS

DON'T USE

DIA and DOA are used to transfer multiple words between expansion memory and onboard memory. The specifics are detailed as follows:

DIA ac,MEM Read from expansion memory to onboard memory.

DOA ac,MEM Write from onboard memory to expansion memory.

In both of the instructions above, all four accumulators have prescribed meanings, regardless of which accumulator is coded in the instruction. They are,

A0 = Number of words to transfer (maximum 400 octal). This value must be a multiple of 4; i.e., the two least significant bits are ignored. If (A0) = 0, no transfer is done.

A1 = Block number in expansion memory, where a block is defined as 256 words. The block number may range up to 77777 (octal) for 16 MB addressing.

A2 = Onboard address; i.e., address in onboard memory from which data is to be transferred. Must be a multiple of 4; i.e., the two least significant bits are ignored and taken as 00.

A3 = Bits 8-15: Word address offset into the block in expansion memory addressed by A1, where the transfer is to begin. Must be a multiple of 4; i.e., the two least significant bits are ignored and taken as 00.

Bit 0: If the msb of A3 is 1, the Error Detection and Correction circuitry is disabled. On a read, the data is read regardless of any error indication and no attempt at error correction is made. On a write, only the new data is written to the addressed quadruple-word in expansion memory, preserving the previously existing Error Correction bits.

Note: Transfer may not overlap from one block into the next. In other words, the sum (A0) + (A3) must not exceed 400 octal.

END

2.3.5.3.1.2 BLOCK TRANSFERS

DIB and DOB are used to transfer entire blocks (256 words) between expansion memory and onboard memory. The specifics are detailed as follows:

USE THIS

DIB ac,MEM Read a block from expansion memory to onboard memory.

DOB ac,MEM Write a block from onboard memory to expansion memory.

In both the above, A1 and A2 have prescribed meanings, regardless of which accumulator is coded in the instruction.

A1 = Block number in expansion memory

A2 = Onboard memory address QUAD WORD ALIGNED!

Note: Error Detection and Correction cannot be disabled as with the DOA/DIA instructions.

In all four of the above instructions, after execution the accumulators will be as follows:

A0 = Expansion memory status word (see Table 2-2)

A1 = Unchanged

A2 = Next onboard memory address, i.e., A2 is incremented by the number of words transferred

A3 = Unchanged

END

2.3.5.3.1.3 SINGLE WORD TRANSFERS

USE THIS

DIC and DOC are used to transfer single words between expansion memory and accumulator 2. The specifics are detailed as follows:

DIC ac,MEM Read one word from expansion memory into accumulator 2.

DOC ac,MEM Write one word from accumulator 2 into expansion memory.

In both of the instructions above, A1 and A3 have prescribed meanings, regardless of which accumulator is coded in the instruction. They are,

A1 = Block number containing the desired word in expansion memory

A3 = Bits 8-15: Word address offset of the desired word into the block in expansion memory addressed by A1.

Note: Error Detection and Correction cannot be disabled as with the DOA/DIA instructions.

In both of the above instructions, after execution the accumulators will be as follows:

A0 = Expansion memory status word (see Table 2-2)

Note: The status word returned is 0 if there are no error conditions to report; i.e., if the lsb = 0 then the entire status word is equal to zero.

A1 = Unchanged

A2 = The word read from expansion memory when using DIC; unchanged when using DOC.

A3 = Unchanged

2.3.5.3.1.4 SINGLE WORD TRANSFERS WITH DATA CHANNEL MAP TRANSLATION

Don't use

DIC/DOC ac,MAP are used to transfer single words between expansion memory and accumulator 2 using the Data Channel Map address translation features (refer to 2.3.5.3.2). The specifics are detailed as follows:

DIC ac,MAP Read one word from expansion memory into accumulator 2 with address translation. The mnemonic "MAP" represents the device code, which is 3 for Data Channel Map access.

DOC ac,MAP Write one word from accumulator 2 into expansion memory with address translation.

In both of the instructions above, A3 has a prescribed meaning, regardless of which accumulator is coded in the instruction. It is,

A3 = Logical address of the desired word in onboard or expansion memory, depending on the contents of the Data Channel Map.

Note: Error Detection and Correction cannot be disabled as with the DOA/DIA instructions.

After execution, the accumulators will be as follows:

A0 = Expansion memory status word (see Table 2-2), or 0 if word referenced is located in onboard memory, or if there are no error conditions to report.

A1 = Block address as read from Data Channel Map (refer to Section 2.3.5.3.2).

A2 = The word read from expansion memory when using DIC; unchanged when using DOC.

A3 = Unchanged

END

TABLE 2-2 EXPANSION MEMORY STATUS WORD

Don't use

Bit	Symbol	Name and Significance
0	--	Reserved.
1	DON	Done Flag. Set by hardware to a 1 to indicate that the previous transfer was accomplished. A 0 indicates that no memory board with the specified block number has responded. In this case, ERR is also set to a 1.
2	EDC	Error Detection and Correction (EDAC) Option Flag. Set by hardware to a 0 if the EDAC option is installed.
3-10	C0/C7	These are the 8 EDAC check bits written or read along with the 64 data bits. Used for diagnostic purposes only.
11	VIR	Virgin Flag. This bit is set to a 1 by a DOA, DOB or DOC instruction (i.e., a write to expansion memory), and cleared by hardware at expansion memory power-up time. At Power-Fail Auto-Restart time, if this bit is 0, it indicates that Battery Back-up was not successful.
12	BAK	Battery Back-up Option Flag. Set by hardware to a 1 if the expansion memory BBU option is installed.
13	SER	System Error Flag. Set by hardware to a 1 to indicate that an error has occurred on a mapped data channel input (to write 1 word, the expansion memory must first read 4 words). Cleared by any instruction that reads the expansion memory status.
14	CER	Correctable Error Flag. Set by hardware to indicates that a correctable error has occurred on the last transfer; the error was corrected both in onboard memory and in expansion memory. Cleared by an instruction that reads the expansion memory status.

(from MARK 12 CPU Product Specification)

TABLE 2-2 EXPANSION MEMORY STATUS WORD (CONT.)

Bit	Symbol	Name and Significance
15	ERR	Error Flag. Set by hardware to a 1 to indicate that there has been an error on the last transfer. If Bit 14 is 0, it indicates that the error was uncorrectable. Cleared by any instruction that reads the expansion memory status read.

END

If it is desired to read the expansion memory status word without doing any memory transfer, this can be accomplished by a DIA ac,MEM instruction with zero in A0.

Any other I/O instructions with device code MEM will have no effect on the expansion memory.

2.3.5.3.2 DATA CHANNEL ACCESS TO EXPANSION MEMORY - DATA CHANNEL MAP

Don't Use

The MARK 12 CPU features a Data Channel Map which allows arbitrary Data Channel access to onboard memory, and block-mapped Data Channel access to expansion memory.

The Data Channel Map is actually a 256 word x 16 bit memory, where each word corresponds to one 256-word block in logical addressing space, and 15 of the 16 bits define the physical "block address" corresponding to that logical block. The 16th bit indicates whether a Data Channel transfer is to go into or out of onboard memory or expansion memory.

On power-up and after an IORST instruction, Data Channel mapping is disabled, i.e., all Data Channel transfers will go into or out of onboard memory without address translation. This is done so that the standard disk bootstrap routine will bring the initial block into onboard memory, where it can be executed.

When a Data Channel request occurs and mapping has been enabled by setting the Data Channel Mapping Enable bit in the CPU Status Word (see Section 2.3.4), the CPU reads the 16-bit address furnished by the requesting device, and applies the 8 most significant bits thereof as an address into the Data Channel Map. If the content of that word in the Data Channel Map has the most significant bit set, the Data Channel transfer is done into or out of onboard memory, otherwise it goes to expansion memory for the transfer. The 15 bits from the Data Channel Map are then combined with the 8 least significant bits of the given 16-bit address to form a 23-bit address, and the Data Channel transfer is done to or from the translated address.

If the software wishes to have Data Channel Mapping enabled, it must first set up the Data Channel Map appropriately, and then set the Data Channel Map Enable mode bit in the CPU Status Word.

END

For each block (256 words) in logical addressing space that is to be mapped into expansion memory, the following instruction must be executed:

DOB ac,MAP

where (regardless of "ac"),

MAP = 3 (device code)

A1 = M*100000 (octal) + B

where M = 0 for transfers to or from expansion memory

M = 1 for transfers to or from onboard memory

B = block address (up to 77777 octal for expansion memory or 377 for onboard memory)

A2 = Bits 0-7 = Logical block address given by the Data Channel device, i.e., bits 0-7 of the Data Channel address. Bits 8-15 = Ignored.

For any block that is not to be mapped, or to disable mapping after it has been enabled, use the same instruction to vector the block to itself; i.e., right half of A1 = left half of A2.

No warning is given here if expansion memory with the specified block number does not exist on the system. If Data Channel access to non-existent memory is later attempted, input will be discarded, and output will be zero-filled.

To read the content of the Data Channel Map, use a DIB ac,MAP with (A2) as above, and it will return the Map content in A1.

LAST PAGE

END