2
7/6/77

UCID-17498

Lawrence Livermore Laboratory

SOME NOT-SO-COMMON WAYS OF USING MCS-8 I/O INSTRUCTIONS, LEN 22269

*looks like some sort
of report number —
see inside*

Eugene R. Fisher

MASTER

June 7, 1977

This is an informal report intended primarily for internal or limited external distribution. The opinions and conclusions stated are those of the author and may or may not be those of the laboratory.

Prepared for U.S. Energy Research & Development Administration under contract No. W-7405-Eng-48.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

REFER QUESTIONS TO: E. R. Fisher	LAWRENCE RADIATION LABORATORY	ELECTRONICS ENGINEERING ENGINEERING NOTE LIVERMORE, CALIFORNIA	UNIVERSITY OF CALIFORNIA	LEN 22269
ORIG E. R. Fisher	SOME NOT-SO-COMMON WAYS OF USING MCS-8 I/O INSTRUCTIONS			DATE 6/7/77
APPROVED				PAGE 1 OF 5

ABSTRACT

The I/O instructions for the Intel 8008 Microprocessor have versatilities that are not apparent to most users. The Standard lament is that there are only 8_{10} input instructions. The fact that there are only 24_{10} output instructions is sometimes belabored, but 24 is normally considered sufficient for most microprocessor applications. This Note discusses ways to expand the available number of both these instructions.

DISCUSSION

Could you use just a few more OUTPUT INSTRUCTIONS? How about 32? That is what you will have if you notice that the input instructions also put the accumulator data out on the Low Address in the same manner as an output instruction. The input instructions are not violated in any fashion--you will still input any data applied to the 8008 data bus. (See "APPLICATIONS" Section for an interesting application).

Then there is the old standard answer of decoding the output data which gives you $(24)(256) = 6,144$ output codes, but no data. (See Figure A). The following is an example of how to program the decoded version:

```

LAI
15          /Stepping motor #15

Out B      /Bank B

LAI
326        /Motor #326

OUT C      /Bank C

```

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

This solution should find application where a large number of pulses are required in an area remote from the processor.

The INPUT INSTRUCTION has suffered much abuse from users while, in fact, the instruction is very versatile. If 8_{10} input instructions are too few, how about having $2,048_{10}$? This number is available by again decoding the Accumulator (AC) data. You now have a large number of input instructions with little additional hardware. One SN74154 gives you 15 extra input instructions. To issue an input instruction you must set the AC to the input you desire, but (as shown below) this is easier than reading a remote memory location and you don't use the H and L registers (as shown in the second example below).

(Example 1)

```

LAI          /DVM #15

15

INP B        /DVMS SELECT/READ

```

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

(Example 2)

```

LHI          /DVMS SELECT
77
LLI          /DVM #15
15
LAM          /READ

```

APPLICATIONSBCD/BIN Conversion in One Instruction

Because the input instruction also OUTPUTS the AC, using a hardware BCD/BIN Converter Card (LEA 68-9068-52) you can now do this application as shown in Figure B, using the following program:

```

.
.
.          /ENTER WITH BCD IN THE AC
INP B      /CONVERT AND READ ANSWER

```

This requires no modification to the MCS-8.

ROM Code Conversion in One Instruction

How would you like to program your own PROM and use it for a data-addressed look-up table? An example could be a sine table, but instead let's look at an ASCII-to-BAUDOT/BAUDOT-to-ASCII Converter. The program is as follows:

```

.
.
.          /ENTER WITH 8-BIT ASCII OR 5-BIT BAUDOT
.          /IN THE AC
INP 2      /CONVERT - ANSWER IN THE AC

```

This application uses a standard PROM in a standard memory location with 128₁₀ memory locations remaining for subroutines. The application is possible because the current

implementation of the MCS-8 stores the A register in the low address buffers, and the device code for the input instruction in the high address buffers. Therefore, the memory now has the required high and low addresses.

Memory, or Page Numbers for the Input Instructions

	HIGH ADDRESS	LOW ADDRESSES
INP0	1	AC
.	3	
INP2	5	
.	7	
.	11	
.	13	
.	15	
INP7	17	

You can see that the page numbers must be the numbers shown and only the numbers shown. Only 8₁₀ ROMS, you say? Think of it this way: that's 2048 points! Any unused locations can still be used for program and, in fact, the memory can be READ/WRITE and can provide dynamic use of this application; all with only one small hardware modification! (The data input multiplexer for the 8008 must be told we are going to use this feature; therefore, the modification shown in Figure C must be made.) The modification inverts the S0 and S1 lines on an input instruction, and allows the MCS-8 to read the memory as a one-instruction I/O device without changing the H and L registers. Now aren't you glad these pins are available to the user?

STRAYS

Now, a point of interest on the RESTART instructions: There are two instructions on the MCS-8 that are never used. They are the HALTS. The reason: it is hard to get the processor going after halting without starting all over again, and then you lose your place (and registers). So, if you hate to waste anything and you want to use your HALTS (both of them), it is very easy to implement a true CONTINUE switch. Figure D shows that if you cause a RESTART instruction and present a NON OPERATION (NOP) instruction such as LBB or LAA, the 8008 will step off the HALT and continue with the next instruction with all flags and registers intact. (See the INTEL users manual on the RESTART instruction). On the MCS-8 we can get a NOP by holding bits D₇ and D₆ high and lowering D₀ and D₂ to give LAA or a NOP.

SUMMATION

There is a need for people to publish their applications of microprocessors so that other users can learn from the existing applications and not be continuously "reinventing the wheel". The point is: someone may have a better way to do your application, but how will the next person know the best way to do the job - unless we publish both applications?

An even greater problem is that the microprocessors are so new that it is difficult to evaluate their full capabilities. Applications, both successful and the other kind, will help the rest of us to do a better job.

LOW ADDRESS

FIG. A

FIG. B

SKETCH
NUMBER

FIG. C

FIG. D

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research & Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately-owned rights.

NOTICE

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Energy Research & Development Administration to the exclusion of others that may be suitable.