EMULATION OF THE HP 21MX COMPUTER

Dan H. Barnes
and
Larry L. Wear
Department of Computer Science
California State University, Chico

Introduction

Recent advances in the field of semiconductor memory technology have made microprogrammable computers economically feasible. A number of mini computer manufacturers are currently selling user programmable computers. One of the earliest of these computers was the Hewlett-Packard 2100A. This paper describes how the microprogrammable feature of the 2100A was used to enable it to emulate a newer computer, the HP 21MX.

In the past, there have been numerous examples of newer computers simulating or emulating older machines [1][2]. The reason for this invariably was that a great deal of software had been developed for the older computer and users did not want to have to redesign application programs. This problem does not exist on the HP 21MX because the 21MX was designed so that it would use an expanded instruction set of the 2100. This meant that all programs developed for the 2100 would run without modification on the 21MX. However, since the 21MX was introduced, software has been developed which utilizes the expanded capability of the 21MX, and this software will not run on the standard 2100A.

The Computer Science Department of California State University, Chico, currently has both a 2100A and a 21MX in the department laboratory; and because we wanted to take advantage of the features of the 21MX but wanted to have interchangeability of programs between the two systems, we chose to emulate the 21MX on the 2100. A microprogram emulation was chosen rather than a software simulation because of the speed improvement gained by emulation.

Comparison of the HP2100 and 21MX Computers

The HP 2100 and 21MX computers are quite similar in their external appearance to the programmer with one major exception: The HP 21MX has two index registers, X and Y, and the 2100A has none. Table A below shows a comparison of registers and timing for the two machines.

Characteristic	HP 2100A	HP 21MX
Accumulators	A,B	A,B
Index Registers	---	X,Y
Scratchpad and General Purpose Registers	S1-S4 F,Q	S1-S12
Memory Cycle Time	980ns	650ns
Microprocessor Cycle Time	196ns	325ns
Control Store Memory (maximum)	1024	4096

TABLE A
Comparison of Hardware Characteristics
Between the HP 2100A and HP 21MX

In addition to the items described in Table A, there are several other differences between the two machines that should be explained. The 2100A [3] is basically a three bus machine. The R and S busses provide data paths to the arithmetic and logic unit (ALU), and the T bus provides a data path from the ALU to various registers. The 21MX [4], on the other hand, has only two primary busses, the S and T busses. The second data path, the ALU, was eliminated. This, however, required adding a register to temporarily store an operand, the L register, so that the ALU would have two operands available at execution time.

Another major difference between the two machines is that the 21MX is asychronous and the 2100A is synchronous. Although this feature has a considerable effect on system operation, it does not have a significant effect on the use of the machine from the microprogram level. One other difference between the machines which falls into this same category is the memory map option which is available on the 21MX. This allows the 21MX to address more than 32K by switching from one block of memory to another without having to resort to a time-consuming core swap from a peripheral device.

Goals and Methodology

The primary goal of our research was to develop a set of microprograms which would allow the HP 2100A to execute all programs written for the 2100MX that did not require memory mapping. It was decided that this must be accomplished without requiring that changes be made to the operating system programs or to the hardware itself. Two secondary goals were that 1) the control store memory required for the emulation should be no greater than 512 words and 2) that the emulator should run at the same speed as the 21MX itself. The maximum memory size was chosen so that there would be room for future expansion within the 768 words at available control store. The speed requirement was given the lowest priority and speed was sacrificed whenever conflict with the other two objectives occured.

Our approach to the problem was to examine the Extended Instruction Group (EIG) of the 21MX and to see how the microcode would have to be changed to allow the instructions to execute on the less sophisticated hardware. Because some instruction executions (I/O, Shift/Rotate and Alter/Skip) are controlled almost entirely by hardware and not microprograms, these instructions could be executed on either machine without modification. The instructions from the 2100A that required modification so that they would be compatible with the EIG instructions were the EAU instructions, 32 bit shifts, MPY and DIV, floating point.

Since we chose not to modify the hardware of the 2100A, two major obstacles had to be circumvented; 1) the fact that 2100A has fewer registers than the 21MX and 2) the limitations of the hardware mapper, which selects a particular microprogram for execution based on the contents of the instruction register. The lack of index registers was overcome by rewriting part of the standard instruction set for the 2100A so that it did not make reference to scratch pad registers S3 and S4. These registers were then used as pseudo index registers. Bypassing the second problem was less trivial. Because the 21MX has more instructions than the 2100A, the hardware mapper for it was designed to generate more entry address into control store than the 2100A. This, in turn, meant that some 21MX instructions mapped directly into the middle of a 2100A instruction. In particular, the microprogram that controls the divide instruction on the 2100A had to be split into parts to accomodate the extra entry points required for the 21MX instruction set.

Timing Considerations

A number of interesting observations can be made between two machines regarding timing. First, the older machine, the 2100A, has a faster microprocessor cycle time, 196ns vs. 325ns. Second, the main memory cycle time of the 21MX is faster, 650ns vs. 980ns. These combined with the difference in buss structure between the two machines make estimation of execution speed between the two machines difficult. One might reasonably expect the 2100A to be noticibly slower than the 21MX based on differences in memory speed. This is, however, offset by the 2100A microprocessor speed and its three buss architecture. Table B shows execution times for some of the instructions on both machines.

Instruction	HP 21MX Emulator	HP 21MX
LDA/B	1.96μs	1.94
MPY	10.78	12.32-13.30
CAX	2.94	2.27
LAX	5.88	4.875
JPY	3.92	4.55
FIX	5.88-8.82	6.50-12.02
LIA	1.96	2.59-3.89
ALF	1.96	2.6-2.92
SSA	1.96	2.59-2.92

TABLE B
Typical Instruction Execution Time Comparison
Between the 21MX Emulator and the HP 21MX

As the table indicates, the HP 2100A is faster than the newer machine in some cases. A definitive answer as to which machine is faster would require an analysis of instruction set mix such as given in the research by Ma [5].

Test and Evaluation

Because the purpose of developing the emulator was to allow a 2100A system to be upgraded, we decided that the only relevent test was to run all operating system programs on the emulator. To that end, three operating systems, DOS-M, DOS-IIIB, and RTEII were generated and exercised on the emulator system. No problems or incompatabilities were observed. User Programs written in 21MX assembly language were also assembled and executed and again there were no problems encountered.

Tests to determine the relative speed of the 2100A, the emulator, and the 21MX were run. However, because the relative speed of the machines was highly dependent on instruction mix, no quantitative results are presented. Instruction execution times for some instructions have been included as Table B above and a user could evaluate his own programs to determine speed degradation/improvement for a particular application. Although no specific numbers are given, we have observed that there was no noticeable difference in performance (with student user environment) between the standard 2100A system and the emulator system.

Conclusions and Extensions

The goals of our investigation were met to our satisfaction. We have an emulator which allows us to take advantage of the enhanced instruction set and new software available on the 21MX. The speed of execution is more than adequate. The amount of control store memory required is 728 words. This requires three modules of WCS which is all that can be used on the 2100A. It therefore would not be possible to add more features to the emulator without making some changes to our current system. One method that would allow the user to have one module of WCS, 256 words, available would be to burn the emulator into PROM and replace the standard 2100A PROM. This is feasible and is currently under consideration.

Work on this project has shown the advantages and limitations of using the 2100A as a host machine for emulation. A project that will lead to the development of a Varian V72 emulator for the HP 2100A has been initiated. Results of this project should be published upon its completion.

References

[1] Husson, Samir S., Microprogramming Principles and Practices, Prentice-Hall, Inc., Englewood NJ, 1970, pages 15, 82.

[2] Kurzban, Heines and Sayers, Operating Systems Principles, Mason/Charter Publishers, Inc., London, England, 1975, pages 338-344.

[3] Hewlett-Packard, 2100 Computer Microprogramming Guide, Cupertino, California, 1972.

[4] Hewlett-Packard, Microprogramming 21MX Computers, Cupertino, California, 1974.

[5] Ma, S. and Wear, L., "Dynamic Instruction Set Evaluation", The Seventh Annual Workshop on Microprogramming, pages 9-12, September 1974.