IEEE TRANSACTIONS ON

EDUCATION

FEBRUARY 1981 VOLUME E-24 NUMBER 1 (ISSN 0018-9359)

A PUBLICATION OF THE IEEE EDUCATION SOCIETY

SPECIAL ISSUE ON MICROPROCESSORS IN ELECTRICAL ENGINEERING EDUCATION

(Note: Additional Papers to be Published in May 1981 Issue)

FOREWORD

PAPERS

rather minimal level and expand as budget and campus politics may permit.

Several responded that industry had been of minimal assistance in the task of initiating μP education into the engineering program. Because of the inherent attractiveness of this device, educators have largely succeeded despite inadequate support. The faculty has indeed learned "new tricks" and has been innovative! Just now they will have another opportunity to make some administrative decisions with the introduction of the new VLSI Intel 432. Some have just given up the minicomputer to accommodate the μP; it appears that the 32-bit Model 432 will force us to get back into that game which was just abandoned.

Essentially all μP education for electrical engineers is taught by the electrical engineering department. Computer science departments are at about 40 percent of that level. At the 10 percent level or less, as compared to electrical engineering departments, a number of other departments also teach μP's. The author suggests the impact of the μP on measurement, data handling, instrumentation, and control will become more evident to nonelectrical engineering departments in due time. This should present an opportunity for the electrical engineering department to extend a minimum prerequisite course to the nonelectrical engineer, including the basic sciences.

We must be impressed with the extent and very rapid acceptance of the μP into the electrical engineer's educational program. To use modern jargon, the school that does not offer education in μP's is "not with it." Too little is too little, and too much is much too much! Electrical engineering educators have the challenge of finding the golden mean. With the delightful intrigue of the μP and the pressure of going digital, let us not forget that digital logic is rather readily understood, but real digital problems require an analog background for solution, a systems approach, transmission lines for data transfer, and "juice" coming out of the wall outlets.

Jens J. Jonsson (M'51-SM'58), for a photograph and biography, see this issue, page 2.

A Microprocessor Laboratory for Electrical Engineering Seniors

DONALD F. HANSON, MEMBER, IEEE

Abstract—**An approach to the laboratory instruction of electrical engineering students in the principles of microprocessor and microcomputer interfacing is presented. Instructional apparatus including the Rockwell AIM-65 microcomputer and a homemade educational interfacing unit is described. Finally, the details of each laboratory experiment are given.**

I. Introduction

THE invention of the microprocessor and its increasing use in electronic design, control, and instrumentation has necessitated rapid changes in electrical engineering education. Most schools have already added microprocessor-related courses, and many have revised their curricula starting from the freshman year on up. Here in the Electrical Engineering (EE) Department at the University of Mississippi (UM), a definite need existed by 1977 for the addition of a computer architecture/microprocessor course and laboratory. At that time, while some students were asking questions about microprocessors, others had never heard of them. In 1977, without formal microprocessor training, two students built a microprocessor-controlled ping-pong game to earn their required senior design credits.

Unlike engineering programs at most institutions, engineering programs at UM are engineering science oriented. Our EE students are given a heavy dose of mechanical and civil engineering courses in addition to electrical engineering courses. Our goal is to provide the students with the basic engineering and scientific knowledge with which they can tackle a general engineering problem. Because of the small size of the EE Department (15 to 20 EE graduates/year), no elective courses can be taught. It is in this framework that the laboratory described here was developed. Only the fundamentals of microprocessor systems engineering are covered. The emphasis is on hardware and interfacing principles. More detailed software coverage is left to those schools with computer engineering curricula or to those which can afford to teach

Manuscript received March 13, 1980, revised June 28, 1980.
The author is with the Department of Electrical Engineering, University of Mississippi, University, MS 38677.

electives. At one time or another in their career it is very likely that most of our graduates will be required to design computer and/or microcomputer interfaces and software. It is, thus, necessary to give all our students the fundamental knowledge which will enable them to interface external devices to a microcomputer or microprocessor and to generate software for control of the external devices.

In 1977 a new faculty member (myself) was hired to initiate a microprocessor course (Advanced Digital Systems) at UM. I spent the first year getting up to speed on microprocessors by reading and attending short courses since I had had no previous experience with them. This course was first taught in 1978. That year, we had one KIM to work with in laboratory (for 12 students). Despite this limitation, we were able to do some worthwhile experiments which were described in [3]. In 1979 we were fortunate to receive a total of $9000 in equipment funds for our undergraduate program from the University, $5400 of which we used to equip the microprocessor laboratory.

II. UM's Microprocessor Laboratory

There are many different approaches to take in developing a microprocessor laboratory. Possibilities range from buying a $30 000 development system to a $200 microcomputer. It was decided that for our purposes sufficient equipment was needed to allow each student to connect his own interface and test his own program, individually or in pairs. For a laboratory class size of 12, this meant at least six complete laboratory stations were required. It was my goal to have the student complete each experiment, including the connection of wires if necessary, and the writing and testing of a program in less than a three-hour period. A versatile system was also desired for those students utilizing a microprocessor in a senior design project.

It was decided that a system was needed at each laboratory station which incorporated most of the following features:

1) program in assembly language or higher level language, not at the machine level;
2) not to be obselete soon;
3) adequate I/O available including keyboard input, printer output, peripheral interface adapters, tape I/O, and D/A and A/D converters;
4) solderless connection to interfacing pins to allow simple hookup of experiments;
5) properly designed for years of use in laboratory and in senior design projects with a minimal risk of damage by student ineptness;
6) low cost.

Based on these needs, it was decided to purchase seven Rockwell AIM-65 microcomputers. One is shown in Fig. 1 along with allied instructional apparatus. The AIM-65 had many of the desired features including:

1) low cost
2) alpha-numeric thermal printer
3) alpha-numeric LED display
4) full-sized ASCII keyboard

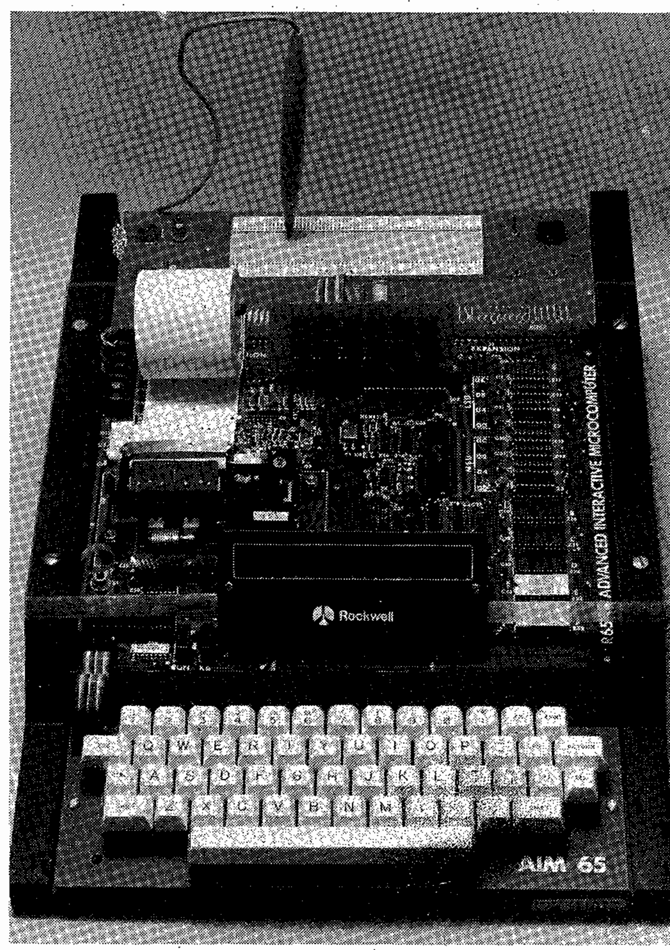

Fig. 1. AIM-65 microcomputer by Rockwell International.

5) advanced monitor program including register and instruction trace modes (*V* and *Z*)
6) mnemonic instruction entry mode (*I*)
7) TTY and audio cassette interfaces
8) 4K onboard RAM and 20K ROM.

In fact, it has most of the desired features except solderless interfacing, D/A output, and A/D input. In addition, the availability of Basic and assembler ROM chipsets and a wide range of peripheral devices should mean that it will not become obsolete quickly. One of the AIM-65's we purchased is used for a spare and for hooking various peripheral equipment to as the need arises. The other six are used in the laboratory itself. We have used the AIM-65's for over a year now and have experienced a minimum of reliability problems. We are well pleased with them.

III. An AIM-65 Educational Interfacing Unit

The laboratory instruction of electrical engineering students in the skills of interfacing external devices to a microcomputer often requires that many wires be routed from the microcomputer to the external devices. In an educational laboratory setting where time is at a premium, these wires must be set up and torn down within a single three-hour laboratory period. The basic AIM-65 system has a total of 88 signal points available for interfacing arranged in two groups of 44 pins each.

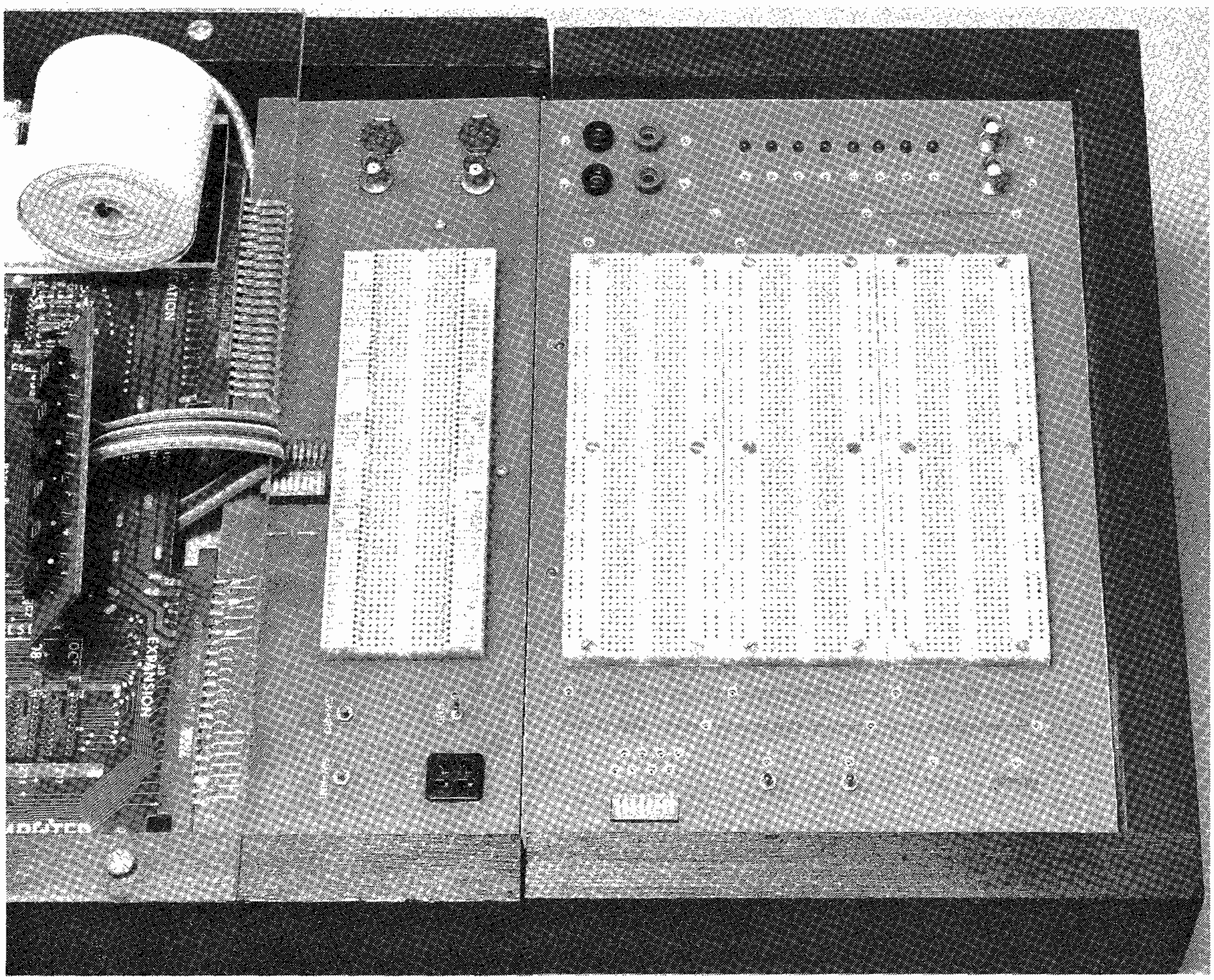

Fig. 2. The educational interfacing unit and breadboarding unit.

The usual way of interfacing to these pins is to solder wires to a 44-pin connector and slip this on the AIM-65. This would present a psychological barrier to many students under the time pressures present in a laboratory and would result in a great deal of wear on the AIM-65 and/or on the connectors.

In order to eliminate this problem, an educational interfacing unit was designed and built in order to be able to insert wires directly into a breadboard's fingers. This has the advantage that no soldering need be done and no connectors must be slipped on and off of the AIM-65 system. The educational interfacing unit is attached directly and permanently to the AIM-65. The unit consists of a solderless interfacing breadboard mounted on a printed circuit card as shown in Fig. 2. Also shown is a separate breadboarding unit. Each of the AIM-65's 88 interface lines is connected (after buffering) to a different set of contacts in the solderless breadboard. The location of each of the interface lines is clearly labeled as shown. This labeling format is similar to that used on the MMD-1 by E & L Instruments, Inc. In fact, it would be possible for some of the experiments to use the "OUTBOARD" concept discussed by Rony, Larsen, and Titus in [4]. The interface board also has eight rocker switches (SW1–SW8), two push-button switches (SW-NMI, SW-IRQ), and tape and teletype interfaces. These can also be seen in Fig. 2. Below the interface breadboard are two other boards, the buffer and electronics boards.

The buffer board, shown in Fig. 3, contains circuits which buffer each address and data line and most control lines present in the solderless breadboard. This protects the microprocessor from student wiring errors, and eliminates the need for the student to ever buffer any lines. The electronics board, shown in Fig. 4, contains two independent 6520 PIA circuits. The first PIA, in locations \$9C00–\$9C03, controls six 7-segment LED displays connected for multiplexing as shown in Fig. 5. This was chosen because in the author's view,

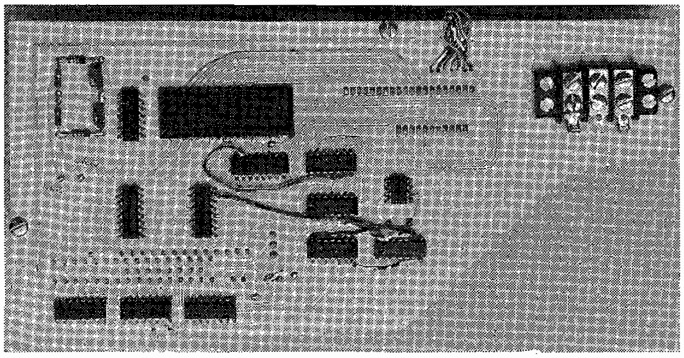

Fig. 3. The buffer board.

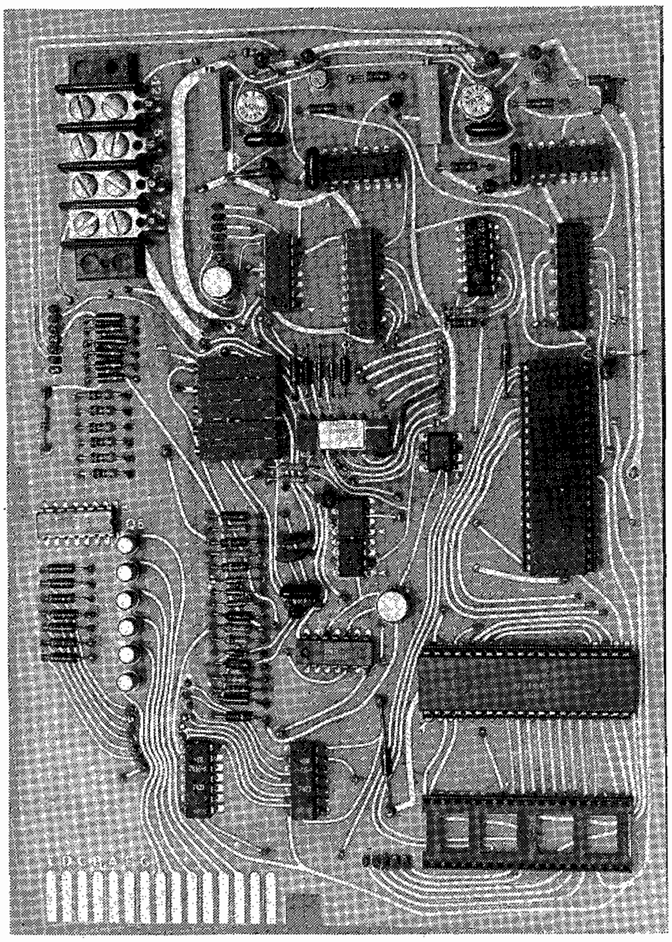

Fig. 4. The electronics board.

control of multiplexed LED displays provides students with a good learning experience. The second PIA, in locations \$9800-\$9803, controls two D/A converters and one A/D converter. A block diagram is shown in Fig. 6. Fig. 7 gives a block diagram of a sample/hold (S/H) circuit also present on the electronics board. These circuits were chosen to give the student exposure and experience in controlling analog I/O. The input, output, and control lines for these circuits are all present in the interface breadboard as shown in Fig. 8.

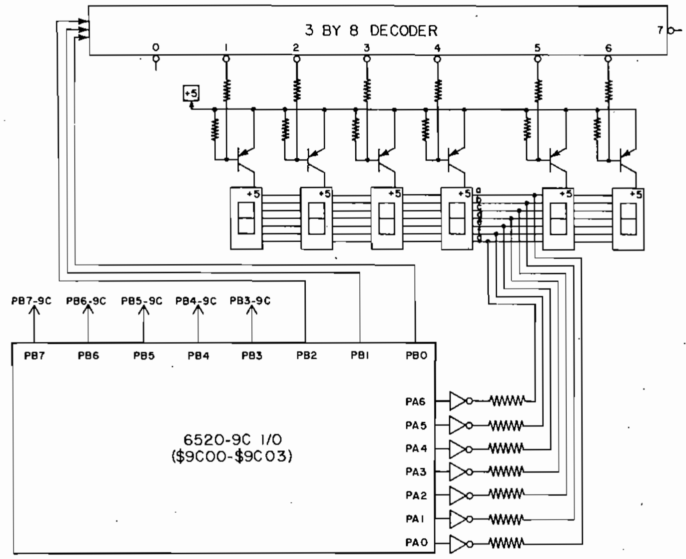

Fig. 5. The multiplexed LED displays.

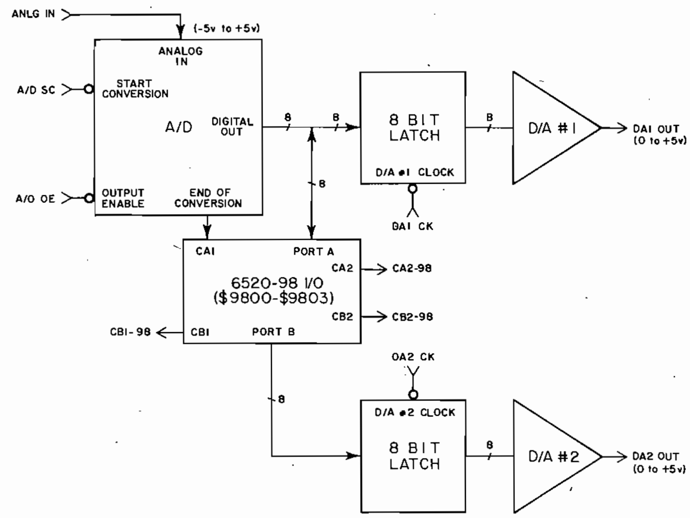

Fig. 6. D/A and A/D circuitry connections.

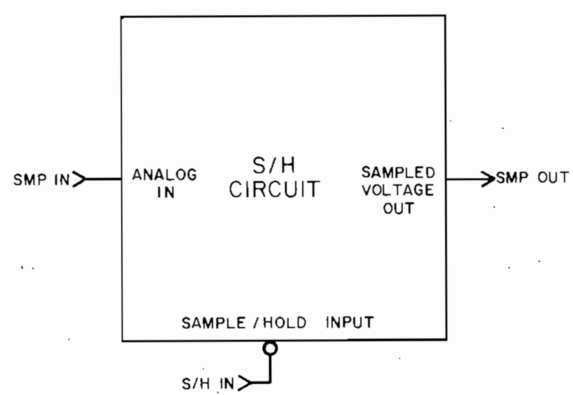

Fig. 7. Sample/hold circuit.

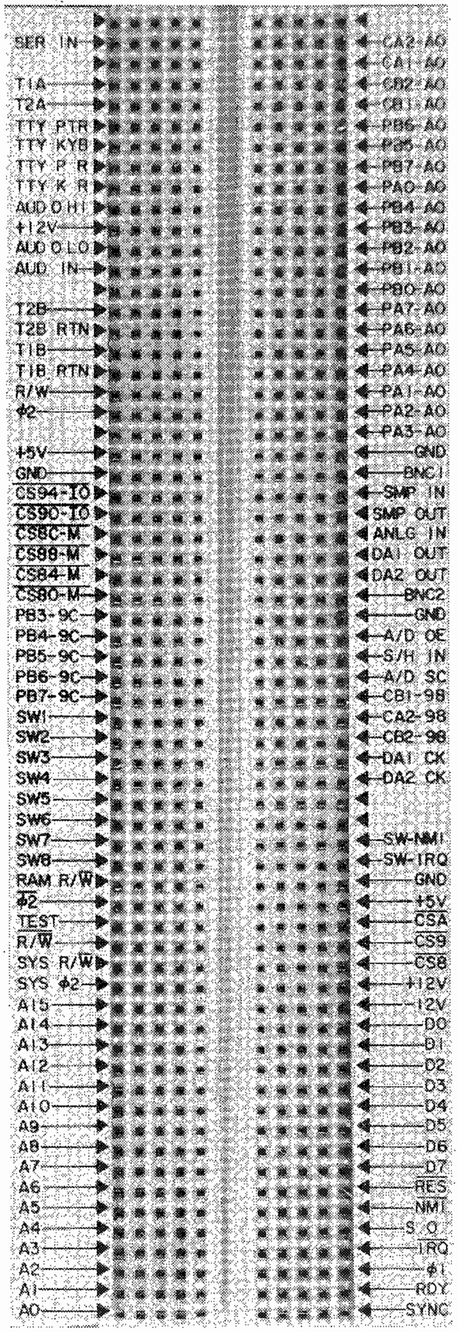

Fig. 8. Close-up of interface breadboard.

The top and bottom 22 lines on each side (88 total) are the (buffered) AIM-65 connections. The middle 20 lines on each side are input, output, and control lines for the various experiments described in the next section.

IV. Laboratory Course Content

In developing the specific laboratory topics, my goal was to provide interesting and achievement-oriented goals while inspiring the students' imaginations and giving them new insights into the digital world. Twelve three-hour experiments were written. Since our students are not required to take an assembly language programming course, it was important to reinforce the basic software principles learned in the first five experiments while introducing new concepts on the timing and control of interfaced devices in the last seven experiments. Laboratory reports were required for all of the experiments. In two of the experiments the students actually performed a slightly different experiment than that which is reported here. This was because of parts and equipment limitations that have now been alleviated. Table I gives a list of the titles of the experiments. A paragraph about the content of each experiment follows.

TABLE I
List of Experiments

1) Introduction to the AIM-65 System.
2) 6502 Addressing Modes.
3) 6502 Jumps, Subroutines, Branches, and Loops.
4) 6502 Arithmetic, Logic, Shift and Rotate Instructions.
5) Real-Time Data Conversion.
6) Interrupt Requests ($\overline{NMI}$, $\overline{IRQ}$).
7) 6502 Memory Expansion.
8) 6522 I/O Port and Polling.
9) 6522 Timer.
10) 6520 I/O Port and Multiplexed Displays.
11) Digital-to-Analog Converters.
12) A/D Conversion using Handshaking.

Experiment one provided an introduction to the AIM-65 system. The students followed through the examples given by Camp, Smay, and Triska in [1]. Once this was completed, they entered and ran a 24-hour clock program written by De Jong [2]. This exercise offered them a glimpse of what was to come and gave them practice in entering and running programs.

In experiment two, 6502 Addressing Modes, the students were given eight short programs to type in. Each program illustrated a different mode of addressing a data element. The students ran the programs using the register and instruction trace modes, V and Z, of the AIM-65 monitor program and then verified their printouts at home. Immediate, zero page, zero-page indexed X, absolute, absolute indexed X, absolute indexed Y, indexed indirect X, and indirect indexed Y addressing modes were all covered.

The third experiment, 6502 Jumps, Subroutines, Branches, and Loops, gave the students six more programs to enter. They ran these programs with the instruction trace Z and/or the register trace V ON. At home they verified their printouts by following through the programs and verifying each program operation.

6502 Arithmetic, Logic, Shift and Rotate Instructions was the title of experiment four. This experiment was composed of two parts. In the first part the students were given four programs to type in. The four programs were for:

1) 16-bit binary addition (ADC)
2) three digit BCD addition (ADC)
3) 16-bit binary subtraction (SBC)
4) three digit BCD subtraction (SBC).

For each of these programs, the student had to type in and run six different cases. Later they had to verify the results on the printout at home. In the second part they were required to type in and verify the operation of a program which utilized logical and shift instructions. At the completion of this experiment the student had used some form of almost all instructions and had used all addressing modes. In the rest of the experiments the students had to write programs on their own to control one or more external devices.

Experiment five, Real-Time Data Conversion, had the students write a subroutine for converting packed BCD (two digits to a byte) to ASCII. This type of conversion is common in those situations where BCD arithmetic results have to be printed. A main program (supplied to the students) read two 10-digit BCD numbers from the AIM-65 keyboard, added them, and stored the result in packed BCD. The students were required to convert this packed BCD result to ASCII and print and display the result on the AIM-65 printer and display. This experiment familiarized the students with AIM-65 monitor subroutines PHXY, PLXY, READ, OUTDIS, and OUTPRI.

In experiment six, Interrupt Requests ($\overline{\text{NMI}}$, $\overline{\text{IRQ}}$), the students used the switches, SW-NMI and SW-IRQ (bounce-free) to send a $\overline{\text{NMI}}$ or $\overline{\text{IRQ}}$ signal to the microprocessor and then display and print the letters "NMI" or "IRQ," respectively, on the AIM-65. Nonmaskable interrupts are edged triggered on the 6502 so the switch signal SW-NMI can be wired directly to the $\overline{\text{NMI}}$ line. Since the $\overline{\text{IRQ}}$ line is level sensitive, the student had to wire up a flip-flop between SW-IRQ and $\overline{\text{IRQ}}$. In addition, the student was required to use software in the interrupt routine to clear the flip-flop by bringing $\overline{\text{CS8}}$ (for instance) low.

Experiment seven, 6502 Memory Expansion, was developed to have the students interface two 2114 memory chips to the AIM-65 by using one of the chip select signals shown on the left row in Fig. 8. These signals are $\overline{\text{CS80}}$-M, $\overline{\text{CS84}}$-M, $\overline{\text{CS88}}$-M, and $\overline{\text{CS8C}}$-M. The M means that they are suitable for hooking to the CS line on a 2114 memory. The data lines, ten address lines, and R/W are also used. The students were required to write a program to systematically check out all four pages of the new memory and print and display a message indicating whether the memory was good or bad.

Experiment eight, 6522 I/O Port and Polling, acquainted the students with the AIM-65 user 6522 PIA chip. The students interfaced the eight rocker-type switch signals (SW1–SW8) to port B of the user 6522 (PB0-A0 through PB7-A0). Either one of the momentary switches SW-NMI or SW-IRQ was connected to CB1-A0 and CB2-A0 was monitored with a logic probe. The students were required to write a program to generate a square wave on CB2-A0 whose period was proportional to the binary number present on the eight switches (SW1–SW8) times roughly 1 ms. Their program had to poll bit four of the interrupt flag register (interrupts disabled) once every cycle to determine if the user wanted the computer to read in a new number off of the switches. The user pushed the switch connected to CB1-A0 to indicate to the processor that new data was ready.

In experiment nine, 6522 Timer, the students were introduced to the 6522 timer one and were required to write a program to produce an accurate 5 s pulse on CA2-A0 every time a switch (SW-IRQ or SW-NMI) connected to interrupt pin CA1-A0 was pushed. In the main program, bit one of the interrupt flag register was to be polled for as long as it was clear (0). If it was high (indicating that the switch on CA1-A0 had been pushed), the timing interval was to be started. Since timer one can be programmed to generate continuous interrupts 50 ms apart, the completion of one hundred timer interrupts gave the required 5 s pulse interval.

The title of the tenth experiment was 6520 I/O Port and Multiplexed Displays. It used the hardware shown in Fig. 5. The students had to write a program to display a word of their choice on the LED displays by using multiplexing; "boPPEr" and "Abdul" were a couple of words chosen. The experiment also introduced the students to the 6520 I/O port.

In the eleventh experiment, Digital-to-Analog Converters, the students were given a general flowchart for a programmable ramp generator program. This experiment was developed to use one of the D/A converters shown in Fig. 6. CB2-98 can be programmed to give a short negative pulse every time data is stored on the 6520 port B. If CB2-98 is so configured and is connected to D/A#2 CLOCK (DA2 CK), port B is latched into the input of D/A#2 just after new data are stored on port B. The normal mode of operation that the students' programs were to follow was to output one cycle of a ramp, scan the keyboard once and, if neither *H* nor *D* was down, output the next cycle of the ramp. If either *H* or *D* was down, however, the user was to enter two hexadecimal characters which corresponded to the height *H* of the ramp or the delay *D* between ramp steps. With an input *H*10, the ramp was to have 17 levels (the D/A input was to count from 0 to 16) and then repeat. A *D*10 was to initialize a location in a delay subroutine to 10 hexadecimal. This would cause a delay of roughly 17 units of 4 μs each (ignoring JSR and RTS). The delay subroutine was to be called after incrementing the D/A input by one. This experiment acquainted the student with AIM-65 monitor subroutines RCHEK, READ, and PACK.

The final experiment, A/D Conversion Using Handshaking, used the hardware shown in Figs. 6 and 7. The students were required to sample a low-frequency sine wave input with the sample/hold circuit, convert the sampled voltage to digital, and output the same amplitude of sine wave to one of the D/A converters. The signals PB3-9C through PB7-9C (see Figs. 5 and 8) were to be used to control the S/H and A/D circuits. When A/D start conversion (A/D SC) was brought low, conversion was to start. The handshaking line, "end of conversion," immediately went low and returned high when the conversion was complete. This line is connected to CA1 to serve as a handshake line and if properly programmed will set bit seven of control register *A*. This bit was to be polled until conversion was complete.

V. Conclusion

One approach to the laboratory instruction of electrical engineers in microprocessor systems engineering has been described here. This program was developed to provide each and every one of our graduates with the basics of microcomputer interfacing and programming. While this provides only the basics, it is felt to be sufficient for those individuals not planning to specialize in computers or microcomputers and to be a good start for those that are. The microprocessor equipment described here has been used for several senior design projects. The AIM-65 system, along with the solderless interface, allows for easy testing and debugging of microprocessor-based senior design projects.

ACKNOWLEDGMENT

The author would like to express his appreciation to R. Cronin and K. Pruett for their help in the construction of the laboratory apparatus.

REFERENCES

[1] R. C. Camp, T. A. Smay, and C. J. Triska, *Microprocessor Systems Engineering*. Portland, OR: Matrix, 1979, Appendix A.
[2] M. L. De Jong, "A simple 24 hour clock for the AIM-65," *Micro*, p. 10:5, Mar. 1979.
[3] D. F. Hanson, "Microcomputers in electrical engineering education at the University of Mississippi," in *Proc. 2nd Annu. Symp. Microprocessors and Education*, Huntsville, AL, Dec. 9, 1978, pp. 1-3-1-6.
[4] P. R. Rony, D. G. Larsen, and J. A. Titus, "Appendix 4: Outboards," in *The Bugbook V*. Derby, CT: E & L Instruments, Inc., 1977.

Donald F. Hanson (S'67–S'71–M'75) was born in Urbana, IL, on March 5, 1946. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Illinois at Urbana-Champaign, Urbana, in 1969, 1972, and 1976, respectively.

For the 1976–1977 school year, he was an Assistant Professor with the Department of Electrical Engineering, Iowa State University, Ames. He is currently an Assistant Professor with the Department of Electrical Engineering, University of Mississippi, University. His research interests include the development of mathematical and numerical techniques for solving boundary-value problems of electromagnetic theory, and digital and analog electronics applications including microprocessors and microcomputer interfacing.

Dr. Hanson is a member of Sigma Xi, Tau Beta Pi, and Eta Kappa Nu.

A Course Sequence in Microprocessor-Based Digital Systems Design

TREVOR MUDGE, MEMBER, IEEE

Abstract—**This paper gives an overview of a sequence of three courses designed to educate students in the use of microprocessor-based digital system design. The first course in the sequence is intended to satisfy the educational needs of undergraduates whose main focus of interest is not digital system design. The complete sequence is intended to provide a firm background in microprocessor-based digital system design for students who planned to continue in our graduate program and whose graduate work would involve them in experimental computer science. The three courses are part of a larger commitment to revitalize experimental computer science in the Department of Electrical and Computer Engineering at the University of Michigan. An overview of the lab equipment is given. The lab experiments are outlined and their pedagogical purpose briefly discussed.**

I. INTRODUCTION

THIS paper gives an overview of a sequence of three courses designed to educate students in the use of microprocessor-based digital systems design. These courses are presently being offered by the Electrical and Computer Engineering (ECE) Department and the graduate program in Computer, Information, and Control Engineering (CICE) at the University of Michigan. Design is taught in the context of microprocessors and other related very large scale integrated (VLSI) components, including ROM and RAM memory chips, PLA's, timers, sequencers, PIO's, SIO's, multipliers, and floating point attached processors. These VLSI components are taken principally from the Zilog Z80 family, the Advanced Micro Devices Am2900 family, and the Intel 8086 family. Fig. 1 shows a diagram of the relevant course sequence, including related courses. The three courses on which this discussion centers are: ECE 365—Digital Computer Engineering, ECE 366—Digital Computer Engineering Laboratory, and ECE 466—Digital Design Laboratory. ECE 365 is intended for juniors, ECE 366 for seniors, and ECE 466 for seniors or first year graduate students. Alongside the course sequence diagram is the catalog description of each course and the credit it is worth in hours. The direction of the arrows in the course sequence shows the order in which the courses must be taken—viewed in reverse the arrows define the prerequisite courses for each course.

Originally, in the period 1968 through 1978, 365 and 366 were taught using five PDP8 minicomputers—four for ECE 365 and one for 366—and a number of attache case "logic lab" kits that contain a 5 V power supply, push buttons, switches,

Manuscript received July 31, 1980.
The author is with the Computer, Information, and Control Engineering Program and the Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48109.

0018-9359/81/0200-0014$00.75 © 1981 IEEE