A COMPUTER HARDWARE LABORATORY COURSE

Krishna K. Agarwal
Gary R. Boucher
Alfred L. McKinney
Reza Sanati

Department of Computer Science
Louisiana State University in Shreveport
Shreveport, LA 71115

ABSTRACT

This paper describes a computer hardware laboratory course for computer science students. It discusses the importance of exposing students to real hardware components in conjunction with the first course in theoretical computer architecture and organization. This course helps in improving their understanding of the operating principles of computer hardware. The various lab exercises that we have used over the last few years are included.

Many times students find the subject of digital electronics to be an abstract study far from familiar subject matter found in other programming or theory classes encountered in a computer science curriculum. This lab course introduces the student to hands on application of digital electronic theory. All information from students to date has indicated that the laboratory approach to teaching computer hardware is a positive reinforcement of theory learned in the lecture.

INTRODUCTION

Computer science students ordinarily take numerous software-oriented courses but they take only one or two computer architecture courses. However, with better understanding of the precise operation of computer hardware, they can work more easily on sophisticated software such as that required for operating systems and compilers. Another reason for a computer hardware lab course is that accreditation by CSAB (Computer Sciences Accreditation Board) requires that computer science students receive substantial computer hardware-oriented laboratory training.

Students who will work in the field of microprocessors and microcontrollers will need more hardware experience than even those who work with more conventional hardware. Many firms engaged in microprocessor applications have a somewhat blurred boundary between the function of the electrical engineering staff and the computer programming staff. For this reason it is essential to have a more well rounded educational background in the theory and hands on practice of digital electronics.

Students who deal with machine dependent software must understand the internal details of the computer, such as registers, data buses, memory organization, instruction sets, and input-output devices. An appropriate computer hardware lab enhances their understanding of these topics while building actual skills in construction and trouble shooting of digital circuits.

In a small college environment we cannot afford to build VLSI chips because of the expensive equipment involved in doing so. Instead, we can demonstrate the basic concepts involved by setting up an inexpensive lab. Other efforts in the same direction may be found in [Barbour 1993, Daulard 1994, Houghton 1994].
Many laboratory manuals used to teach digital electronics fail to introduce the student to a basic knowledge of factors such as current, voltage and resistance. Also many manuals start the student at the chip level without a knowledge of how signals are made to become logic high and low. This matter is taken up at the beginning and developed into resistor-transistor methods for implementation of logic gates. This approach makes what occurs inside the chip somewhat less mysterious and instills a better intuitive grasp of digital switching theory.

Following the introduction to digital switching, the essential concepts that need to be demonstrated are: gates and flip-flops, transfer of data between registers, counters, adders, clocked circuits. In addition, it is immensely helpful to look at how memory chips work. Other topics covered are maintenance of computer hardware such as locating bad memory chips, checking the calibration of floppy drives and power supplies. Many students proceed through a four year curriculum in computer science without having ever having done tasks as simple as changing circuit cards, checking power supply voltage, or installing a hard disk drive. This lab allows the student to become involved in computer maintenance from a board level perspective.

LAB DESCRIPTION

This is a required course with the following prerequisites: Structured Programming in Pascal (CS I), Advanced Programming (CS II, also in Pascal) and Discrete Finite Mathematics (Boolean algebra and logic). The corequisite is a course in Computer Architecture and Organization. The lab meets once a week for three hours at a time but provides only one credit hour for the course. The hardware lab and architecture courses prepare the students to take courses in data structures and assembly language programming.

Many of our students majoring in computer science have a weak science background with little or no knowledge of electrical concepts. Therefore, our first lab meeting includes a basic introduction to voltage, current, resistance and Ohm’s law. At this time we introduce them to lab supplies including resistors, connectors, power supplies, a breadboard and how to make a simple circuit using these items. A digital multimeter to measure volts, milliamps and ohms is provided with explanation.

In later labs, we introduce them to more and more sophisticated ICs (integrated circuits) to design and build several combinational and sequential circuits. A memory chip is also introduced in later experiments. Finally, experiments for diagnosing and maintaining personal computers are included.

LAB EXPERIMENTS

In this section we give more details about each lab experiment. In most labs, the students are first encouraged to build one or more circuits and then test them. They are then given one or two design exercises in which they first design the circuit and then build and test it to see if it meets the specifications.

Experiment #1:

Objective: To become familiar with resistors, transistors, switches, voltage and current. To use the multimeter to measure volts, milliamps, and ohms.

Method: Measure the values of various resistors with the digital multimeter. Build circuits with resistors in series and in parallel and measure the effective resistance, the current and various voltages. Build a NOT gate with a transistor and several resistors and test it. The concept of voltage division between two resistors in series is introduced. This concept is used to form a foundation for transistor logic switching.
Experiment #2:

Objective: To become familiar with bipolar transistors, LEDs (Light Emitting Diodes), truth-tables, gates and the relationship between them.

Method: Build a NOR gate with two transistors (and several resistors), combine the NOR gate with a NOT gate to construct an OR gate and build a NAND gate from two transistors. Construct a truth-table for each of the above circuits.

Experiment #3:

Objective: To become familiar with logic gates on integrated circuit chips.

Method: Build a majority function with three inputs, three NOT gates, four AND gates and a 4-input OR gate. Build a half-adder with an XOR gate for the sum and an AND gate for the carry. Construct a full-adder with two half-adders. Construct a 8-bit full-adder using two 4-bit full-adder ICs.

Experiment #4:

Objective: To become familiar with decoders and multiplexers.

Method: Use two 2-bit to 4-bit decoders and a few other gates to construct a circuit to light an LED when a certain 4-bit input is given to the two decoders. Use an 8-bit multiplexer chip to construct a 3-input majority circuit.

Experiment #5:

Objective: To become familiar with displays and D flip-flops.

Method: Use a decoder, along with other gates and a seven segment display to construct a circuit to display the digits 4 and 9. Using several NAND and NOT gates construct a D flip-flop. Compare this to a flip-flop IC chip.

Experiment #6:

Objective: To become familiar with counters.

Method: Use a four flip-flop counter chip to display binary numbers from zero to fifteen. Modify the counter to count from zero to four and reset itself to zero at the next pulse. Design a counter to count from zero to fifty-nine.

Experiment #7:

Objective: To become familiar with shift registers.

Method: Construct a circuit to store eight bits of information in a shift register. Connect an LED to the output of the shift register and using a pulser to provide a clock pulse, show that your circuit works by retrieving the data that was stored previously. Attach another shift register to the output of the first one and transfer the data sequentially from the first one to the second one by using the pulser. Use an LED at the output of the second register to demonstrate that the data was successfully transferred.

Experiment #8:

Objective: To become familiar with Random Access Memory.

Method: Construct a circuit using static RAM chips to store a word of data. Connect LEDs to the input/output terminals of the RAM and retrieve the data stored in the chip. Repeat the experiment by storing and retrieving data from at least three different locations.

Experiment #9:

Objective: To become familiar with memory chips, floppy disks and power supplies.

Method: Given a personal computer with at least one faulty RAM chip, use either the built-in diagnostic routines or run diagnostic software to detect the bad chip. Replace the
faulty chip so that there are no errors at the time of booting-up the system. A computer with a faulty floppy disk is provided. Run diagnostic routines to determine what the problem may be. If data errors are reported, try cleaning the disk drive heads with a head-cleaning kit. The heads may need re-alignment if timing errors are reported. Adjust the speed of the drive motor if possible. If the drive still does not work, replace it with a good floppy drive. Test the output voltages of a power supply, check if they are within specifications and replace the power supply if necessary.

Experiment #10:
Objective: To become familiar with the maintenance of hard disk drives and graphics cards.

Method: Given a hard disk which is not functioning, open it and document the number of platters and read/write heads. Connect it to a working power supply and document the motions that the heads go through when power is turned on. Given a computer with a working hard disk (but no valuable data), do a low-level and then a higher level format of the hard disk. Document the sequence of commands, the number of cylinders, read/write heads and the disk capacity. Run diagnostics on the graphic card which is in the computer and document your observations. If the card does not function according to its specifications, replace it with another one and repeat the experiment.

Experiment #11:
Objective: To become familiar with assembling a personal computer.

Method: Given an empty case, a power supply, a motherboard with memory chips, a graphics card, a monitor, a keyboard and a floppy disk drive with its controller, assemble the computer and boot it up with the proper diskette. Run diagnostics to test all the components. Install a hard disk and its controller card in the computer, do a low-level format, a high-level format, run the diagnostics and install the operating system on the hard disk. Verify that the system boots up from the hard disk.

CONCLUSIONS

We consider it important that all students of computer science should be exposed to a hands-on laboratory environment. By the end of the semester it is expected that the students have learned the following:

- Fundamentals of electronics (voltage, current, resistance, Ohm's law).
- Knowledge about components (breadboard, power supply, resistors, diodes, transistors, LEDs, a variety of Integrated Circuit chips).
- Building gates (NOT, NOR, OR, NAND) with transistors and resistors.
- Using IC chips to make a variety of combinational circuits (the majority function, half-adder, full-adder, 8-bit adder, decoder).
- Using a 7-segment LED to display output data.
- Building flip-flops from gates (1-bit memory).
- Building counters.
- Shifting data from one register to another.
- Writing data to a RAM chip and reading it.
- Detecting bad RAM chips in a computer.
- Testing, cleaning and aligning floppy drives.
- Testing a power supply.
- Studying hard disks, lower and higher level formatting.
- Testing graphic display cards.
- Assembling a computer, installing disk drives, loading software on a hard disk.
A few students have difficulty with the lab but most find this lab very useful and rewarding, especially the part which deals with the maintenance of personal computers. After the students finish this lab course, they should feel comfortable with computer components, which can be very important for them in their interaction with computing equipment at home and at work. Hopefully, this paper has provided a good model for the goals and experiments of a computer hardware lab. The authors will provide more details on the lab experiments or further dialog on the lab course upon request.

REFERENCES

