

## VISICOMP: The Visible Computer

George York and Ruth D. Fogg

United States Air Force Academy

### Abstract

*At the US Air Force Academy, we found our students have difficulty visualizing the inner workings of a basic computer without some type of training aid. This visualization problem has been noted by others [1][2].*

*Often students are taught computer architecture using a software programmer's model, along with a software tool such as an assembler/simulator. The simulator usually can single step through a program, displaying the results of the registers after each instruction. However, many students don't seem to follow this abstraction. They can't see inside the microprocessor chip.*

*Our solution was to let the students see what is inside a microprocessor by developing our own visual computer. Twenty years ago we developed EDUCOMP for this purpose and found the students enjoyed getting their hands on the hardware. Today we are replacing EDUCOMP with a newer version called VISICOMP, which is more visual and has many improvements, such as a larger memory.*

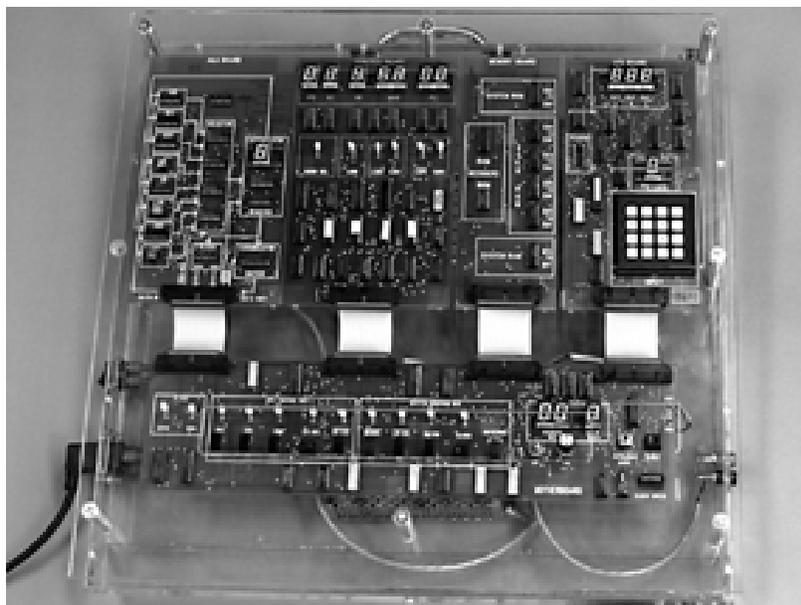


Figure 1. VISICOMP

*VISICOMP is physically partitioned into the main subsystems of a computer (ALU, Controller, Memory, I/O) and has a motherboard to allow the student manual control of the computer. VISICOMP's operation is visual to the student, since: (1) each subsystem uses SSI and MSI chips which the students have already*

learned in our course; (2) the status of the registers, signals, and busses are displayed directly on the subsystems; and (3) the students can manually control each subsystem to observe how each one works. After understanding the basics of the computer subsystems, the students then write their own assembly programs and hand translate them into machine code. They can run their programs in real time, or manually clock the controller to observe what happens during each state of an instruction's execution.

This paper will discuss the features of VISICOMP and how it is used in the classroom.

## Introduction

Many years ago, we at the US Air Force Academy like several other authors [1][2], recognized the best way to teach our students the inner workings of a basic computer is with a training aid. Instead of using a software model such as described in [1] [2] [3] and [4] which still have a high level of abstraction, we developed an actual computer called EDUCOMP.

EDUCOMP was typically introduced to the students in the last part of our sophomore Introductory Digital Systems course after the students have already been exposed to the design of combinational and sequential digital logic circuits. By combining the SSI and MSI chips which they are familiar into a simple computer, the students found it easy to understand how a computer works. This was the underlying premise of EDUCOMP which has been in use for the last 20 years or so. However, since EDUCOMP's memory was very small (sixteen 4 bit locations), it lacked of a common data bus, and not all the circuits were visible, we felt a need for an upgrade. The new computer, called VISICOMP, still uses a 4 bit data word but now has a common data bus and 256 memory locations (8-bit address). Moreover, it shows every chip used in each subsystem along with 7 segment displays and LEDs to display the contents all of the buses and registers. We introduced VISICOMP to our students in the Fall of '95 with very favorable results.

INST.	BINARY	HEX	OPERATION
ONE WORD INSTRUCTIONS			
NOP	0000	0	NO OPERATION
NEG	0001	1	NEGATE THE ACCUMULATOR (2'S COMPLEMENT)
NOT	0010	2	COMPLEMENT THE ACCUMULATOR (1'S COMPLEMENT)
RAR	0011	3	ROTATE ACCUMULATOR RIGHT
TWO WORD INST. - SECOND WORD IS I/O PORT OR IMMEDIATE VALUE			
OUT	0100	4	OUTPUT ACCUMULATOR TO ADDRESSED PORT
IN	0101	5	INPUT ADDRESSED PORT'S VALUE TO ACCUMULATOR
ADDI	0110	6	ADD IMMEDIATE VALUE TO ACCUMULATOR
LDAI	0111	7	LOAD IMMEDIATE VALUE TO ACCUMULATOR
THREE WORD INST - 2ND AND 3RD WORDS ARE LO AND HI-ORDER ADDRESS			
AND	1000	8	AND ACCUMULATOR WITH VALUE AT ADDRESS
JMP	1001	9	UNCONDITIONAL JUMP TO ADDRESS
JZ	1010	A	JUMP TO ADDRESS IF ACCUMULATOR = 0
JN	1011	B	JUMP TO ADDRESS IF MOST SIGNIFICANT BIT = 1 (NEG)
OR	1100	C	OR ACCUMULATOR WITH VALUE AT ADDRESS
STA	1101	D	STORE THE ACCUMULATOR TO ADDRESS
ADDD	1110	E	ADD VALUE AT OPERAND ADDRESS TO ACCUMULATOR
LDAD	1111	F	LOAD VALUE AT OPERAND ADDRESS TO ACCUMULATOR

FIGURE 2. VISICOMP INSTRUCTION SET

## VISICOMP's Design

The architectural design became fixed by our desire to keep VISICOMP as small and simple as possible. With only four data bits, the range of numbers VISICOMP can interpret is from 0 - 15 in binary and -8 to +7 in two's complement representation. This also limits the number of instructions which can be decoded to 16, for a true RISC (reduced instruction set computer) design! VISICOMP's instruction set is shown in Figure 2. The table lists the assembly language mnemonic for the operation followed by the machine language and hexadecimal representation of the operation code and finally by a description of the function. As can be seen from the ADDI and ADDD, as well as the LDAI and LDAD instructions, there are two addressing modes used to introduce the student to the concepts of immediate and direct addressing.

The student can visually see the operation of the computer through a combination of LEDs and 7 segment displays. The address, data, and control buses are displayed, along with the registers (Accumulator, Program Counter, Status Register, Memory Address Register, Instruction Register, Present and Next State).

As seen in Figure 3, VISICOMP consists of five circuit boards, one for each major subsystem including the arithmetic logic unit (ALU), the controller, memory, and the input/output (I/O). The fifth board, called the motherboard, contains all the switches necessary for the student to take manual control of the computer.

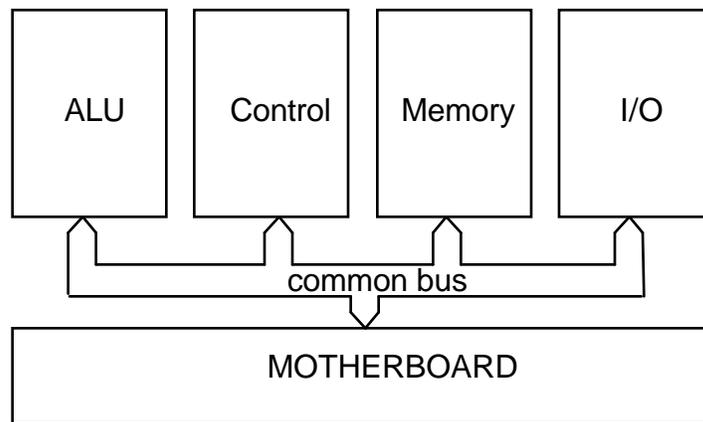


Figure 3. VISICOMP Block Diagram

The ALU board contains circuitry to perform eight ALU operations which are the logical AND, NEG, NOT, RAR, OR, IN, ADD and the Load Accumulator functions. Four (one for each data bit) 8-to-1 multiplexors select the desired function depending on the last 3 bits of the operation code. The selected data bits are then loaded into the accumulator. Combinational logic circuitry takes what is stored in the accumulator to provide the signals necessary for the conditional jump instructions (Zero and Negative Flags).

The major components of the control board are three registers, one for storing the current operation code (Instruction Register), another for storing the address of the operand (Memory Address Register), and another for storing the current address of the program (Program Counter). The contents of each register is displayed at all times with 7-segment displays. Combinational circuitry decodes the op-code and steps the state machine through the instruction cycle. The various control signals of the state machine are displayed with LED's, along with displays for the present state and next state of the controller's state machine. These let the student know the controller's status at any point during it's fetch, decode, and execute cycle.

The memory board simply consists of address decoding circuitry, a single PROM chip and five SRAM chips. VISICOMP's operating system (monitor routine) is stored in the ROM and uses one RAM chip for the system

RAM. The remaining 4 RAM chips are available for user programs, providing 64 memory locations which has been sufficient for most of the student programs.

The last two boards, the I/O board and the motherboard, is where the students get their fingers into the action. The I/O board contains the hex keypad for data entry and three 7-segment output displays as well as an I/O port for external connections.

The bottom board, the motherboard, continually displays the address, data, and control bus. It also has 12 switches and two address knobs which let the students take manual control of the ALU, controller, memory, and I/O boards.

Each board has access to any other board by way of the data, address, and control busses which pass through the motherboard. The data bus is a bi-directional bus to pass data between all the subsystems. Due to this common bus, the boards are interchangeable.

### **Operating VISICOMP**

To allow the student to have a hands-on experience to understand computer architecture and assembly language programming, VISICOMP has three operating modes: automatic, semi-automatic, and manual. In automatic mode, VISICOMP runs like a regular Von Neuman computer. It has a small operating system which recognizes four commands which allow the user to enter, view, edit and run assembly language programs. The entire operating system uses less than 200 memory locations.

In semi-automatic mode, the system clock is replaced by a manually controlled clock. With manual control of the clock, the student can step through the instructions of his or her assembly program or the operating system, one clock pulse at a time. At each clock pulse, the contents of all the registers and busses can be compared to what the student expected. Here, the student receives immediate feedback on how well he or she understands what is going on.

The last mode, manual mode, allows the student to take total control of the computer. By setting the appropriate switches the user can manually read and write to memory or the I/O ports and execute ALU instructions.

### **How VISICOMP is Used in the Classroom**

The goal of the last quarter of the Introductory Digital Systems course is to pull together all the combinational and sequential circuit design concepts into a single capstone example. At first glance, since all the circuits of VISICOMP are visible, the students can recognize larger versions of the circuits they had to design in their previous labs. In order to fully understand and appreciate VISICOMP, the students must be introduced to assembly language programming. Through a series of lessons which combine lecture with lab time, the students begin to understand how VISICOMP works. They start by writing simple assembly language programs. Once they can make VISICOMP perform some simple functions through their programs, the students begin to understand the interplay between the four subsystems. Running VISICOMP in semi-automatic mode, they then understand how assembly language instructions are fetched, decoded and finally executed. Once they have mastered this, they finally use VISICOMP in manual mode to perform several functions manually, such as reading and writing to memory and I/O and controlling the ALU.

The students are given a Reference Manual which explains how VISICOMP works and an Operators Manual which tells them how to run VISICOMP. Attached to the operator's manual is a complete set of logic

schematics showing each chip and wire used in VISICOMP. A series of three lab exercises gives the student practice running the computer in all three operating modes.

### **Results**

At the end of the course, we solicited for feedback and in general the results were very favorable. The students were intimidated by VISICOMP at first. Having to learn assembly language programming at the same time as trying to sort out the details of VISICOMP was tough for many. But being able to view the contents of each register and all the control signals simultaneously made understanding how a computer works a simple matter of interpreting the registers contents and comparing it to what was expected. Moreover, after learning to manually control all of the computer subsystems, the students were enthusiastic and felt they understood how a computer operates.

### **References**

- [1] Moser, A. T., "Animated Simulator for 68000 Microcomputer Architecture," ASEE Annual Conference Proceedings, June 1995, pg 179 -181.
- [2] Henderson, W. D., "Animated Models for Teaching Aspects of Computer Systems Organization," IEEE Trans. on Education, Vol. 37, No. 3, pp. 247-256, August 1994.
- [3] Miller, D. S., "Computer-1 -- A Modern Computer to Introduce Computer Organization and Assembler Language Programming," ACM SIGCSE Bulletin, Vol. 15, No. 1, pp. 271-277, 1983.
- [4] Cutler, M., "A Microprogrammed Computer Simulator," IEEE Trans. on Education, Vol. E-30, No. 3, August 1987, pp. 135-141.

### **Authors**

**GEORGE YORK**

George graduated from the US Air Force Academy with a BSEE in '86 and from the University of Washington with a MSEE in '88. He developed guidance computers for missiles at USAF Wright Laboratories from '88-'92. Then he served two years as an exchange engineer at the Korean Agency for Defense Development. Currently he is teaching Microcomputer System Design courses at the US Air Force Academy.

**RUTH D. FOGG**

Ruth D. Fogg currently teaches the Digital Logic Design and Electrical Circuits courses at the US Air Force Academy. She earned a BSEE ('83) and MSEE ('85) from the University of New Hampshire and a Ph.D from the University of Colorado ('95) where she developed a CAD tool for the layout of integrated-optic circuits. Prior to pursuing her doctorate, she worked 6 years in lasers and optics at the USAF Philips Laboratory in NM.