Design and Construction of the Very Simple Computer (VSC): A Laboratory Project for an Undergraduate Computer Architecture Course

R. A. Pilgrim
Computer Science and Information Systems Department
Murray State Univeristy, Murray Kentucky

Abstract

The Very Simple Computer (VSC) was designed in class and breadboarded in the laboratory by the students of an undergraduate-level computer architectures course. The computer uses small and medium scale TTL integrated circuits. It has an 8-bit word with a 3-bit op-code and a 5-bit address, resulting in 8 instructions and 32 words of memory. The VSC uses a rocker switch input and directly monitors (with LEDs) the bus for the output. This relatively inexpensive project (<$100 including power supply, ICs, breadboards and assorted hardware) is a natural for a small computer science department on a limited budget. In addition to providing the students with a sense of accomplishment, the VSC raises many practical architecture issues not encounted when using laboratory simulators. This paper outlines a method for incorporating the design of the VSC into regular classroom lectures, and offers a number of recommendations for instructors who would consider using the VSC in their own course.

Introduction

A number of educators have used simple computer designs to provide a basis for classroom instruction. Notable examples are Sajjan Shiva's ASC (A Simple Computer)[1] and A. K. Dewdney's SCRAM (Simple but Complete Random Access Machine).[2] While these computers have no practical purpose beyond their educational value, they are both candidates for classroom laboratory projects. While the design of the VSC was heavily influenced by these two machines, the decision to take the system to hardware produced a slightly different set of design goals. First of all, the VSC has an instruction set size of 8 rather than 16 as provided for in the ASC. This decision was based on the simpler hardware configuration resulting from the smaller instruction set. Next, the VSC provides 32 words addressable memory rather than 16 as with the SCRAM. This was done to provide an example of chip-select circuit design in the memory unit. A block diagram of the VSC is shown in Figure 1.

ACM-24thCSE-2/93-IN,USA

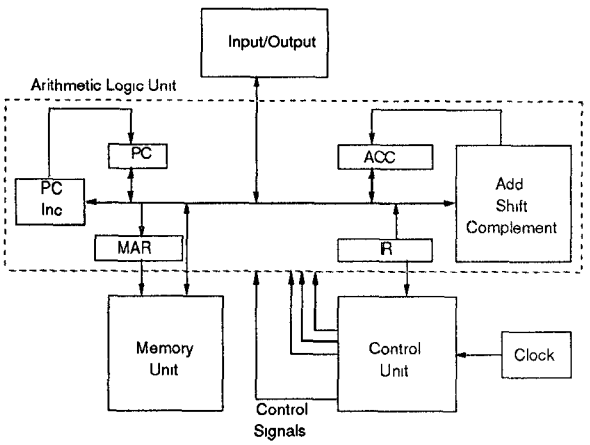

Figure 1: VSC Block Diagram

The breadboarding of each VSC functional module is assigned to one or two students as individual projects. These assignments are given with a design specification, indicating the format of inputs and outputs, expected control signals and required outputs of each module. The modules are tested independently and passed by the instructor. Toward the end of the course, the class begins the task of module integration. The integration effort is one of the most beneficial laboratory exercises of the course, usually requiring occasional conflict resolutions by the instructor.

1. The Instruction Set and Registers

One of the first issues addressed in the design of the VSC is the instruction set.

The 8-bit word is divided into a 3-bit operation code and a 5-bit address. This permits a maximum of 2^3 or 8 instructions. The instruction set selected for the VSC is shown in Figure 2.

Instruction	Register Transfers
LDA	ACC <- MEM[MAR]
STA	MEM[MAR] <- ACC
ADD	ACC <- LAT + MEM[MAR]
CMA	ACC <- ~(LAT)
BIP	**If** $ACC_7 = 0$ **then** PC<-$IR_{0\text{-}4}$
SHL	ACC <- LAT * 2
SHR	ACC <- LAT / 2
HLT	PC <- [11111]

Figure 2: The VSC Instruction Set, and Register Transfers

The accumulator (ACC) the instruction register (IR), the program counter (PC), the latch (LAT, shown in Figure 4, below) and the memory address register (MAR) are made with D-type flip-flops (74273). The actions shown in Figure 2 are the register transfers and arithmetic operations corresponding to each instruction. This instruction set theoretically provides for the implementation of any arithmetic or logical operation. However the utility of the VSC is restricted by a 32 word memory limitation.

The fetch-cycle, shown in Figure 3 below, is the same for all instructions. It begins with the loading of a word in memory (MEM) at address PC into the IR. The PC is then incremented with the help of the MAR as a temporary location for the value of PC. PC+1 is then placed in PC. The address portion of the IR is then transferred to MAR and the contents of the ACC is placed in the LAT.

IR <- MEM[PC]
MAR <- PC
PC <- MAR + 1
MAR <- $IR_{0\text{-}4}$
LAT <- ACC

Figure 3: VSC Fetch-Cycle

This last register transfer in the fetch cycle is a wasted operation for exactly half the instructions. (See instruction set.) The inclusion of the LAT register itself and the loading of LAT in the fetch cycle are to limit the execution of all instructions to a single clock cycle.

To demonstrate the capabilities of such a small instruction set, a number of homework assigments are given to write VSC assembly language programs to perform various arithmetic tasks. The fact that the students are involved in the selection of the instruction set, gives them a better perspective on software/hardware interface issues. The VSC also provides additional motivation for the study of ones-complement and twos-complement arithmetic.

2. The Arithmetic/Logic Unit

The design of the arithmetic/logic unit (ALU) is performed as part of the study of combinatorial circuit design and analysis. As shown in Figure 4, the ALU consists of a shifter, a complementer and an adder. The ALU uses two registers, the Latch (LAT) and the accumulator (ACC) which both require read-enables (RE). The ACC also requires a write-enable (WE) to for output to the bus.

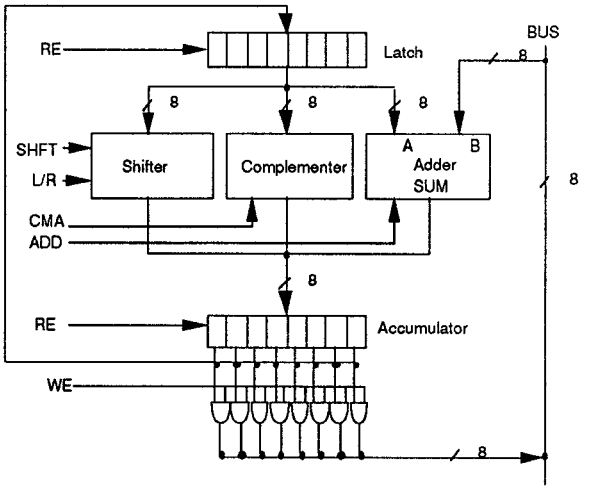

Figure 4: The VSC Arithmetic/Logic Unit

Each of the three functions of the ALU require write-enables so that the result of the desired operation can be presented to the ACC. It is noted that the branching instruction (BIP) which checks the most-significant bit of the ACC is not shown in Figure 4.

The ALU is built up from small and medium scale TTL logic in two or three laboratory sessions. The selection of chips for the adder, complementer and shifter are left to the student, who only has to meet bus interface requirements. The protocol for the

control lines are dictated to some degree by the ALU design. Some recommendations are given such as the use of separate control lines for shift direction and enable.

3. The Memory Unit

The memory unit is built using four 4 x 16 bit static random-access-memory (RAM) chips (7489). Since 32 eight-bit words are required for the VSC, the design of the memory unit includes the use of the MSB of the address as a chip select. The use of static rather than dynamic RAM greatly reduces the complexity of the VSC design. As noted in Figure 1, the VSC does not have a memory buffer register (MBR). This function is provided by the memory itself and the ACC.

4. The Control Unit and Timing Circuit

In order to accomplish the indicated register transfers and arithmetic operations, the control unit (CU) must provide the correct sequence of control signals (or enables) to the registers and other modules of the VSC. The design of the control unit is incorporated into the study of sequential circuit design and analysis.[3,4] In the design of the CU it is shown that the fetch-instruction operation requires a minimum of six primary clock cycles and the execute-instruction operation requires one additional clock cycle, see Figure 5.

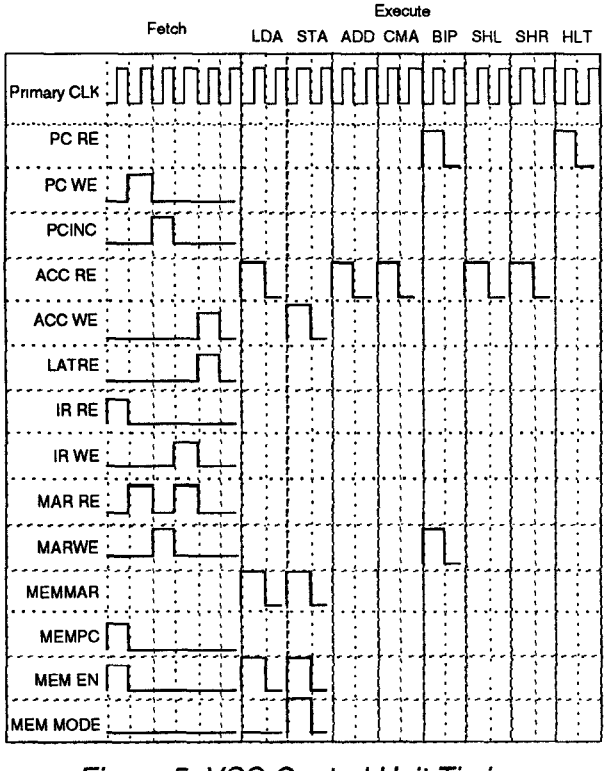

Figure 5: VSC Control Unit Timing

Each shaded group of signals in Figure 6 represents a required output function for the timing circuit. The groupings are made between two or more control signals with equivalent phase. The timing circuit individually sets lines 0 through 7 high in sequence. The CU consists of a timer circuit (NE555) , a divide-by-8 counter (74177) and a 3-to-8 decoder (74138).

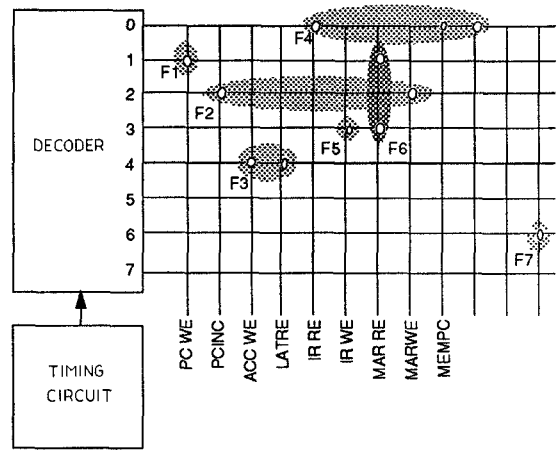

Figure 6: VSC Control Unit Functions

The lines shown are read-enables (RE) and write-enables (WE) for the various registers and memory. The line at node F7 provides the control signal for the execute instruction operation.

By the time the class reaches this point in the VSC design, the students are taking a very active role in design decisions. This is, at least in part, due to the fact that any significant changes necessitate additional laboratory work for one or more of them.

5. The Input/Output Interface and Bus

The input/output interface should be provided by the instructor or labratory assistants, since it requires mostly labor which falls outside the course subject matter. The instructions and data are entered in the VSC one word at a time. This is done by setting each data word on eight data switches and the desired destination address in memory on five address switches. A manual memory write-enable push button (debounced) performs the data entry. The output is obtained by directly viewing the data bus with a set of eight light-emitting diodes (LEDs). The address of the desired word in memory is set on the five address switches

and a manual read-enable push-button puts the selected word on the bus.

The bus itself is 8-line parallel and is used for both instruction and data transfers. The registers are connected to the bus through open-collector NAND gates. This forces the student to become familiar with negative-logic design consideration.

Preliminary Results and Recommendations

The VSC has been used in our undergraduate computer architecture class for the last three years. In terms of creating a machine that works, moderate success was achieved the first year. In the second year, individual modules were verified but a late design change resulted in some incompatibilities during module integration. In both classes, however, the students gave the project good marks in student evaluations and comments. Two students who had taken computer architecture without the laboratory have voluntarily built a version of the VSC on their own.

While any claims of the educational value of this project are anecdotal at best, it is apparent that the opportunity to effect the design of a working computer provides a motivation not realized in lectures alone. In such a rapidly changing area as computer technology, it is easy to forget that even small-scall integrated digital logic technology is new to the student. Sometimes, more ambitious laboratory projects using current large-scale integrated circuit technology will expose the student to more practical hardware at the expense of a deeper understanding of the underlying principles.

A project like the VSC gives students a clearer understanding of the basics of computer hardware by offering them a point design from which to compare and contrast other approaches. The design of individual modules is motivated by the knowledge that their work will be part of a larger system and must perform according to design specifications. The VSC is offered as a project for a computer science laboratory rather than a computer engineering laboratory since many of the timing and noise issues encountered in engineering projects are not addressed. (For example the VSC primary clock circuit runs at 10-100 Hertz which virtually eliminates problems with crosstalk.)

For anyone considering such a project, it is strongly recommended that the ICs, power supplies, breadboards and assorted hardware be purchased from discount electronics stores. Unless your university already has a fully equipped electronics laboratory, it is not cost effective to purchase chip-level digital electronics from educational supply houses.

References

1. Shiva, Sajjan G., *Computer Design and Architecture*, Little Brown and Co., Boston, Mass., 1985.

2. Dewdney, A. K, *The Turing Omnibus: 61 Excursions in Computer Science*, Computer Science Press, 1989.

3. Clements, Alan, Principles of Computer Hardware, PWS-Kent Publishing Co., Boston, Mass., 1993.

4. Shiva, Sajjan G., Introduction to Logic Design, Scott, Foresman and Co., Boston, Mass., 1988.