

Role VHDL based Instructional Modules for Computer Organization and Architecture Course

Nikunja K. Swain, James A. Anderson, Darby Bradley, Daudi Muhamed
South Carolina State University
nkswain2001@yahoo.com, jaanderson@scsu.edu

Raghu B. Korrapati
Walden University
raghu.korrapati@waldenu.edu

Abstract

Studies show that students who used simulation prior to conducting actual experiments performed better than the students who conducted the laboratory experiments without conducting simulation first. One such simulation software package is VHDL. VHDL is an acronym of VHSIC Hardware Description Language. VHSIC means Very High Speed Integrated Circuits. VHDL is a formal language for specifying the behavior and structure of a digital circuit. It is one of the most used languages because it has shorter development times for electronic design and also has simpler maintenance. This paper will discuss design and development of interactive instructional modules for Computer Organization and Architecture course using VHDL. We expect this approach will result in improved learning outcomes for Electrical Engineering Technology and Computer Science majors in the hardware field.

Introduction

The engineering, science, and technology field, at present, is very dynamic due to recent advances in computer and other technologies. These advances have resulted in numerous computer programs to solve traditional and novel problems. These programs use the computer's increased computational capabilities and assist in the design, development, and control of complex systems in a matter of minutes. Automation is becoming part and parcel of every industry, and industries need a trained workforce to manage this new development. Engineering and technology graduates must have a comprehensive background covering a wider range of technical subjects. The graduates must be proficient in the use of engineering and scientific equipment, conducting experiments, collecting data, and effectively presenting the results [1-4]. This is especially true for the graduates of engineering, technology, and science. Furthermore, these graduates must be well-trained in courses and laboratories such as electric and electronic circuits; digital systems and microprocessors; computer programming; computer aided design; computer organization and architecture; electronic and data communications; networking; control and robotics; electric machines and power

systems; PLC and virtual instrumentation; and others. One cost-effective way of achieving this is through the use of simulation software programs, and a number of simulation software packages are available for these purposes. These software packages play an important role in education and are used to deliver training for all kinds of activities, from piloting sophisticated aircraft or ships to operating nuclear power plants or complex chemical processing facilities.

There are numerous uses of simulation, starting from simulation of simple electric circuits to complex tasks such as electromagnetic fields, heat transfer through materials, networking, computer circuits, game programming, electron flow in semiconductors, or beam loading with the ultimate objective of providing illustrations of concepts that are not easily visualized and difficult to understand. Simulators are also used as an adjunct to and, in some cases such as distance learning courses, as a substitute for actual laboratory experiments. In many instances, students are required to verify their theoretical design through simulation before building and testing the circuit in the laboratory. Studies show that students who used simulation prior to conducting actual experiments performed better than the students who conducted the laboratory experiments without conducting simulation first. Also, simulation is used to model large and complex systems. There is no doubt that simulation cannot replace the physical hands-on experience, but simulation can enhance the teaching and learning experience.

The objective of this paper is to discuss design and development of interactive instructional modules for Computer Organization and Architecture course using VHDL and its effect on student learning.

A Partial List of Simulation Software Used in EET and CS Programs

A partial listing of simulation software programs used in EET and CS programs are shown in Table 1 [5],[6]. These programs are either used as stand-alone teaching tools or in conjunction with other tools. For example, a student may use one package to get the experimental data and another a spreadsheet package, such as Excel, for plotting and data analysis. Table 1 lists a few of the most widely used EET and CS simulation software packages.

Table 1: List of Widely Used Software Packages in EET and CS

Name of Software	Primary Application Areas
PSPICE	Electric and Electronic Circuits (Analog and Digital)
Electronics Workbench (Multisim)	Electric and Electronic Circuits (Analog and Digital), Communication
VisSim	Electric and Electronic Circuits (Analog and Digital), Communication
Logic Works/Design Works/VHDL	Digital/Microprocessor Design
MatLab	Mathematics, Control Systems, Power Systems
Mathematica	Mathematics
MathCad	Mathematics
AutoCad	Computer Aided Drafting (CAD)
Simulink	Control and Power Systems
LabVIEW	Control, Signal Processing, Mathematical Simulation
Excel	Spreadsheet for Multipurpose Activities
UMPS	Microprocessors and Microcontrollers
UV151	Microprocessor and Microcontrollers
MASM	Microprocessors
DEBUG	Microprocessors
RSLOGIX	Programmable Logic Controller

Many of the software packages listed above are used in various electrical engineering technology courses at SCSU to assist the faculty and students in teaching and learning as shown in Table 2.

Table 2: List of Simulation Software Package Used in EET and CS Programs at SCSU

Name of Software	Primary Application Areas
PSPICE, Electronic Workbench (MultiSim)	Electric and Electronic Circuits (Analog and Digital), Electronic Communication
Design Works/VHDL	Digital/Microprocessor Design
MatLab & Simulink	Mathematics, Control Systems, Power Systems
LabVIEW	Control, Signal Processing, Mathematical Simulation, Power Systems, Electric Circuits, and Electronics
Excel	Spreadsheet for Multipurpose Activities
MASM	Microprocessors
DEBUG	Microprocessors
EDSIM 51	8051 Microcontroller
RSLOGIX	Programmable Logic Controller

Examples of Application of Simulation Software to EET and CS Programs at SCSU

As presented in Table 2, various simulation software packages are currently being used by the EET and CS programs at SCSU to enhance teaching and learning. The faculty at SCSU have developed a number of modules for course and laboratory use. Packages like PSPICE, Multisim, MatLab, Simulink, LogicWorks, RSLogix, Debug, MASM, and LabVIEW are widely used by engineering and technology programs at other institutions, and there is sufficient information on these in textbooks and on the Web. Packages such as Design Works and EDSIM 51 are not that well-known and may not be widely used, but both of these packages have tremendous potential for enhancing student learning in computer organization and microcontroller courses. We will discuss these software packages and the instructional modules developed using these packages below.

Examples of VHDL Instructional Modules

VHDL [7] is a logic schematic creation and simulation program. It comes with many circuit symbols and models that can be used to design and simulate various types of digital circuits. VHDL comes with libraries of various types of components required to construct and simulate various types of circuits and systems. The objectives of these modules are to assist the student in validating the theoretical design process to have a better understanding of the concept. The student completes the design theoretically and derives the logic circuit. The student then builds the circuit in the VHDL simulator, simulates the circuit using appropriate inputs, and verifies the output to validate the theoretical design. Following this, the circuit can be built in the laboratory using physical components and tested using actual physical signals.

Number of instructional modules on Computer Architecture and Organization are possible with VHDL. In this paper, we will focus on the design of (a) Full Adder, (b) 4x1 Multiplexer, (c) Adder-Subtractor module, (d) a 4-bit arithmetic circuit module, (e) one stage of a logic circuit module, and (f) One stage of Arithmetic Logic Shift Unit. The basic building blocks in the above modules are 4:1 Multiplexer and Full Adder. Therefore the design of Full Adder and 4:1 Multiplexer first are presented first and then these are used as basic building blocks to complete the design and simulation of modules mentioned above. Detailed discussion on these modules can be found in [8].

Design of Full Adder

A full-adder is a combinational circuit that forms the arithmetic sum of three input bits. It consists of three inputs and two outputs. The truth table of a Full-Adder is shown in Table 1 and the logic diagram is shown in Figure 1. The corresponding VHDL design and simulations are shown in Figure 2.

Table 1: Truth Table of Full-Adder

A	B	CIN	Sum(S)	COU(C)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

The logic equations of S and C are the following:

$$C = AB + (A \oplus B)CIN$$

$$S = A \oplus B \oplus CIN = (A \oplus B) \oplus CIN$$

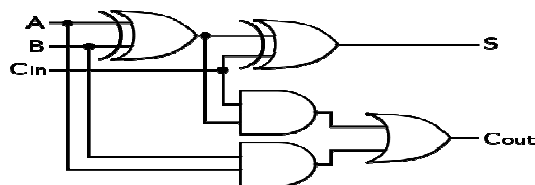


Figure 1: Diagram of Full-Adder

VHDL Design and Simulation

The figure below demonstrates VHDL Simulation Results for A = 1, B = 1, and CIN = 1. The simulated output values of S and COU(C) agrees with the values in Table 1 for A=1, B=1, and CIN =1.

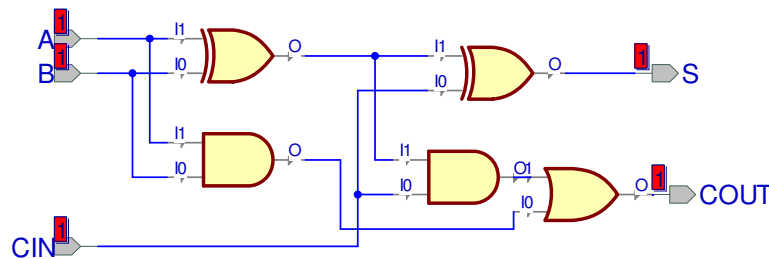


Figure 2: VHDL Design and Simulation

Design of 4:1 Multiplexer (4:1 MUX)

A multiplexer is a combinational circuit that receives binary information from one of 2n input data lines and directs it to a single output line. The selection of a particular output is determined by a set of selection inputs. A 2n-to-1 multiplexer has 2n input data lines and n input selection lines whose bit combinations determine which input data are selected for the

output. The function table for a 4-to-1-line multiplexer is shown in Table 2 and the corresponding diagram is shown in Figure 3. The corresponding VHDL design and simulations are shown in Figure 4.

Table 2: Function table for 4x1 MUX

Select Inputs		Output
S1	S0	Y
0	0	x0
0	1	x1
1	0	x2
1	1	x3

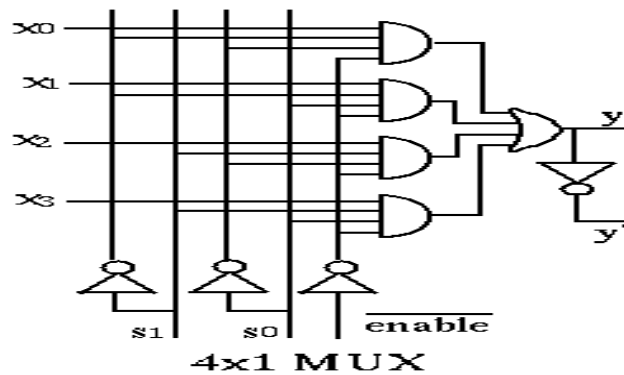
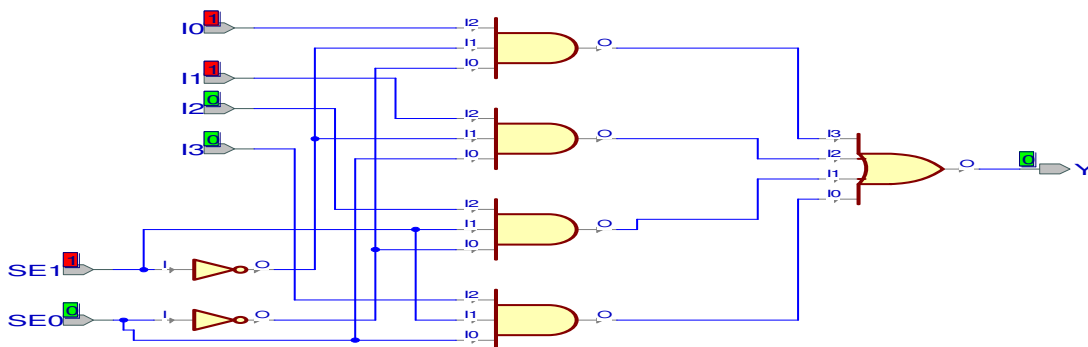


Figure 3: Diagram of a 4x1 MUX

VHDL Design, Code and Simulation of 4x1 MUX

VHDL Design of 4x1 MUX

VHDL Simulation Results for SE1 (S1) = 0 and SE0 (S0) = 2 [Since this is equal to 2, data in line 2 (I2) is selected at output (Y)].



Simulation Waveform

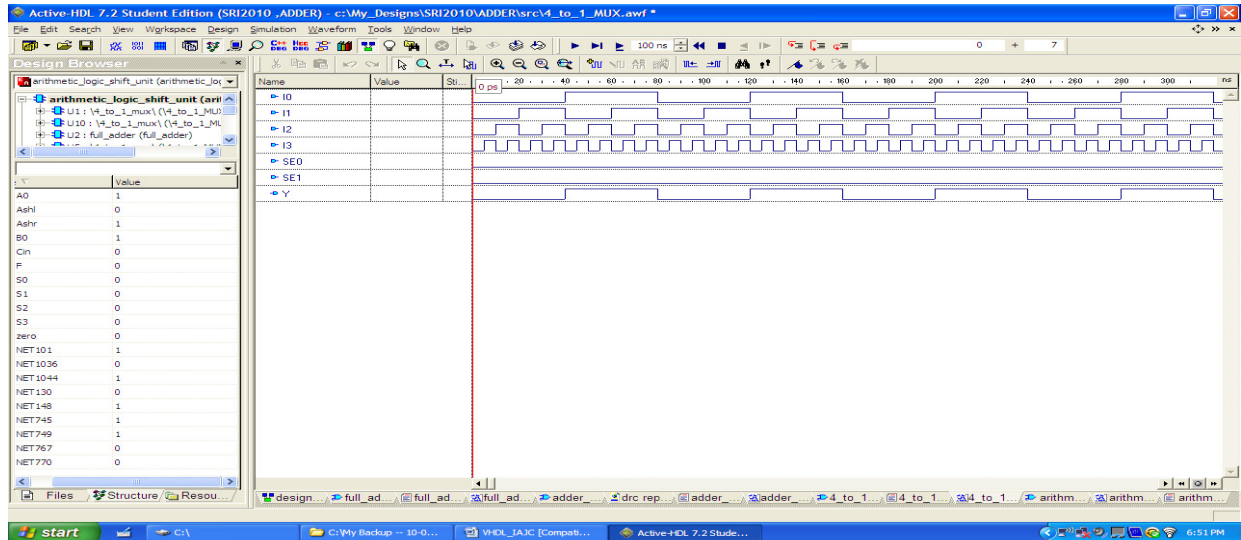


Figure 4: VHDL Design, Code and Simulation of 4x1 MUX

Design of Adder-Subtractor Module

The addition and subtraction operations can be combined into one common circuit by including an exclusive-OR gate with each full adder. A 4-bit adder-subtractor circuit is shown in Figure 5. The mode input M controls the operation. When $M = 0$ the circuit is an adder and when $M = 1$ the circuit becomes a subtractor. The corresponding VHDL design and simulation is shown in Figure 6.

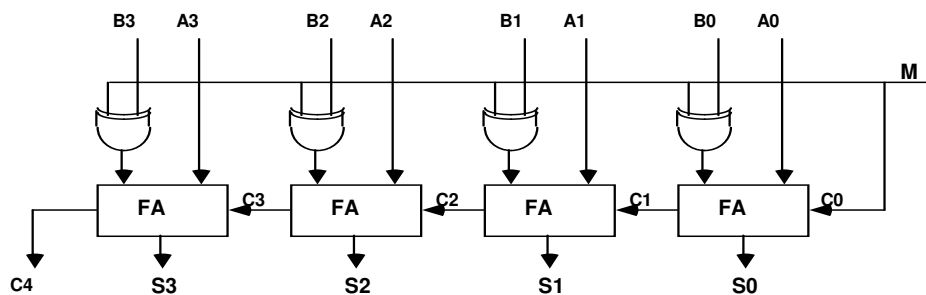
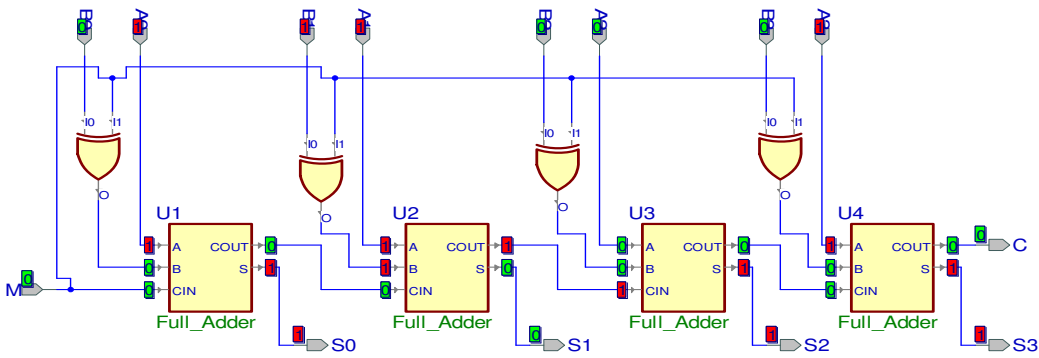


Figure 5: Diagram of 4-bit Adder-Subtractor

VHDL Design and Simulation results

$M = 0$ [Adder]: $A = 1011$, $B = 0010$; $A+B = 1101$ ($S_3S_2S_1S_0$)



M = 1 [Subtractor]: A = 1011, B = 0010; A-B = 1001 (S3S2S1S0)

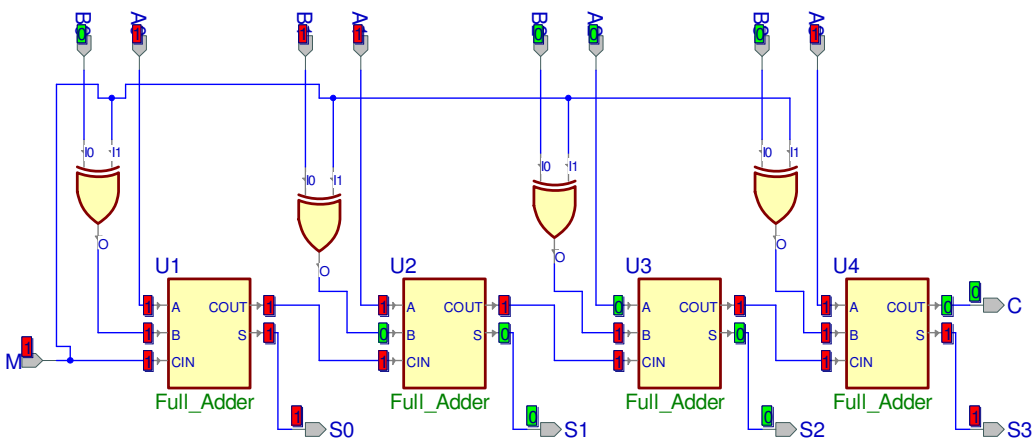


Figure 6: VHDL Diagram, Code and Simulation results for adder-subtractor

Four bit Arithmetic Circuit

The basic component of an arithmetic circuit is the parallel adder. By controlling the data inputs to the adder, it is possible to obtain different types of arithmetic operations. The 4-bit arithmetic circuit is shown in Figure 7 and the arithmetic operations are shown in Table 3. The VHDL diagram and simulation is shown in Figure 8.

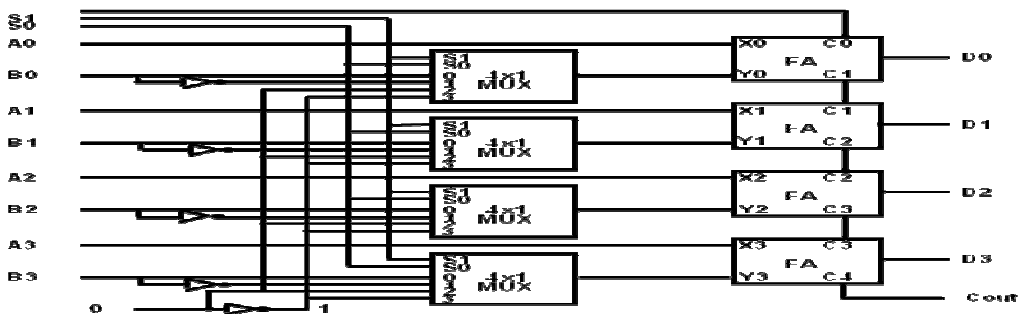


Figure 7: 4-bit arithmetic circuit

Table 3: Arithmetic Circuit function Table

S1	S0	Cin	Y	Output	Microoperation
0	0	0	B	$D = A + B$	Add
0	0	1	B	$D = A + B + 1$	Add with carry
0	1	0	B'	$D = A + B'$	Subtract with borrow
0	1	1	B'	$D = A + B' + 1$	Subtract
1	0	0	0	$D = A$	Transfer A
1	0	1	0	$D = A + 1$	Increment A
1	1	0	1	$D = A - 1$	Decrement A
1	1	1	1	$D = A$	Transfer A

VHDL Design

The figure shown below demonstrates the result of simulation for for $S1 = 0, S0=0$ and $Cin = 1$. From Table 3, $S1=0, S2=0$, and $DIN = 1$ corresponds to $A+B+1$. For $A = 1000$ and $B = 0001$, this corresponds to $1000+0001+1 (0001) = 1010$ and $Cout = 0$.

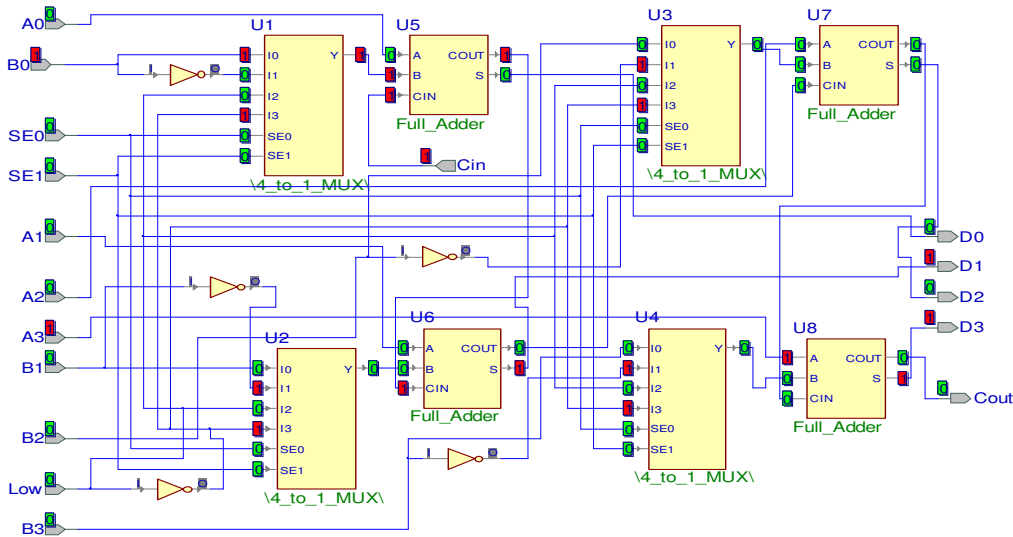
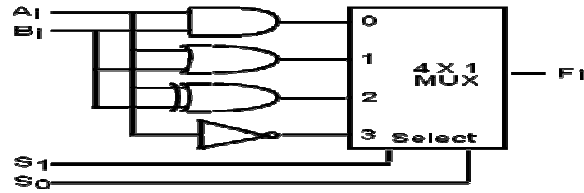


Figure 8: VHDL design and simulation of 4-bit arithmetic circuit.

Logic microoperations

Logic microoperations are very useful for manipulating individual bits or a portion of a word stored in a register. They can be used to change bit values, delete a group of bits, or insert new bit values into a register. Figure 9 shows the diagram and function table for one stage of a logic circuit. The circuit in Figure 9 was simulated using $S1=0, S0=0, A=1$, and $B = 1$. From Function Table this corresponds to AND logic operation with $Y = 1$ (since $A = 1$ and $B = 1$).



Function table

S_1	S_0	Output	μ -operation
0	0	$F = A \wedge B$	AND
0	1	$F = A \vee B$	OR
1	0	$F = A \oplus B$	XOR
1	1	$F = A'$	Complement

Figure 9: Diagram and Function Table for one stage of a logic circuit

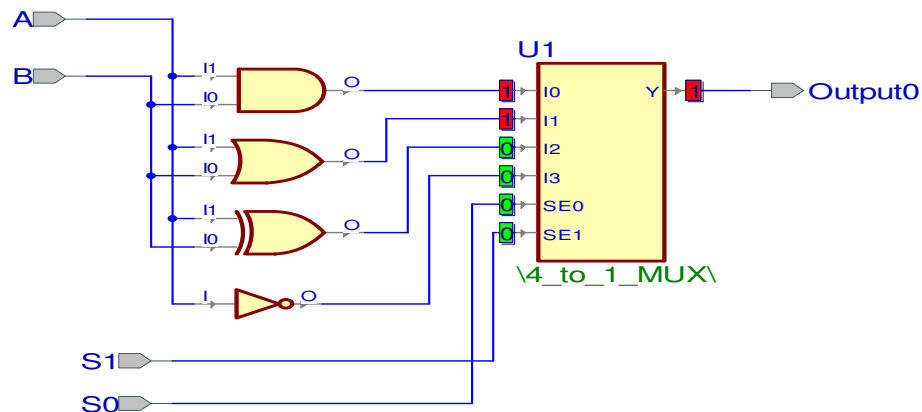


Figure 10: VHDL design and simulation of one stage logic circuit.

Arithmetic Logic Shift Unit

Instead of having individual registers performing the microoperations directly, computer systems employ a number of storage registers connected to a common operational unit called an arithmetic logic unit, abbreviated ALU. The arithmetic, logic and other circuits discussed before can be combined into one ALU with common selection variables. One stage of an arithmetic logic shift unit is shown in Figure 11 and Table 4 represents the function table.

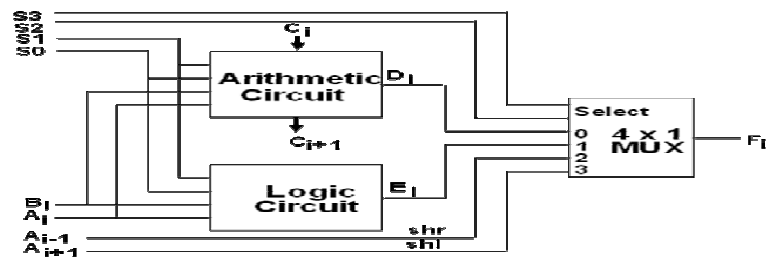


Figure 11: Diagram of one stage of arithmetic logic shift unit

Table 4: Function Table

S3	S2	S1	S0	Cin	Operation	Function
0	0	0	0	0	F = A	Transfer A
0	0	0	0	1	F = A + 1	Increment A
0	0	0	1	0	F = A + B	Addition
0	0	0	1	1	F = A + B + 1	Add with carry
0	0	1	0	0	F = A + B'	Subtract with borrow
0	0	1	0	1	F = A + B' + 1	Subtraction
0	0	1	1	0	F = A - 1	Decrement A
0	0	1	1	1	F = A	Transfer A
0	1	0	0	X	F = A ∧ B	AND
0	1	0	1	X	F = A ∨ B	OR
0	1	1	0	X	F = A ⊕ B	XOR
0	1	1	1	X	F = A'	Complement A
1	0	X	X	X	F = shr A	Shift right A into F
1	1	X	X	X	F = shl A	Shift left A into F

The block diagram shown in figure 11 was designed and simulated using S0=0, S1=0, S2=0, S3=0 and CIN =0. These input conditions correspond to F = A (Table 3). Figure 12 represents the corresponding VHDL simulation.

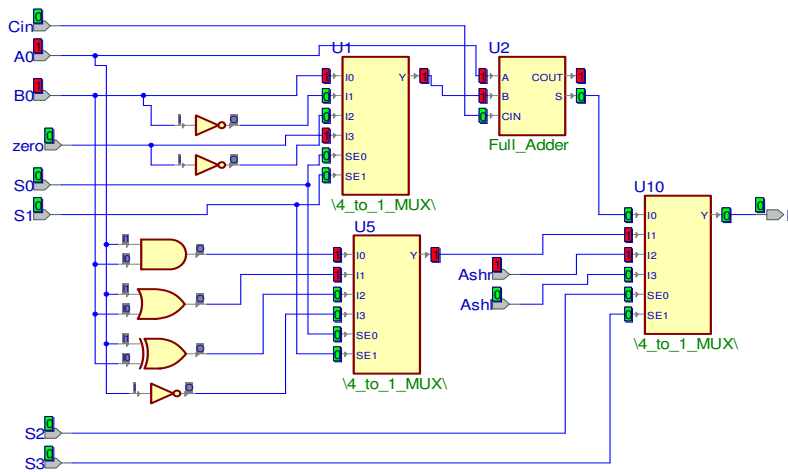


Figure 12: VHDL design and simulation

Conclusions

Students in computer architecture courses must understand both Hardware and Software requirements of a computer system and must be able to design, test, and evaluate various design alternatives in a simulation environment. VHDL is an essential CAD tool that offers designers an efficient way for implementing and synthesizing the design on a chip. It helps to build a higher level of abstraction to think about digital systems that is much more concise than digital circuits. This paper discussed design and simulation steps for few of the modules usually encountered in a computer architecture course. The modules are progressive in nature, meaning, previous modules are used as a component in the subsequent module. Because of the space limitations of this paper, we were not able to present other modules. These modules can be used in conjunction with other teaching aids to enhance student learning in various courses and will provide a truly modern environment in which students and faculty members can study engineering, technology, and sciences at a level of detail.

Acknowledgement

This work was funded in part by a grant from the NSF-HBCU-UP/RISC grant and NSF grant EPS-0903795-2010702 to South Carolina State University. We are thankful to the NSF and SCSU for providing us with this help.

References

- [1] Swain, N. K., Korrapati, R., Anderson, J. A., “Revitalizing Undergraduate Engineering, Technology, and Science Education Through Virtual Instrumentation,” NI Week Conference, Austin, TX, 1999.
- [2] Elaine L., Mack, Lynn G, “Developing and Implementing an Integrated Problem-based Engineering Technology Curriculum in an American Technical College System,” *Community College Journal of Research and Practice*, Vol. 25, No. 5–6, pp. 425–439, 2000.
- [3] Buniyamin, N, Mohamad, Z., “Engineering Curriculum Development: Balancing Employer Needs and National Interest—A Case Study,” Retrieved from ERIC database, 2000.
- [4] Kellie, Andrew C., et al., “Experience with Computer-Assisted Instruction in Engineering Technology,” *Engineering Education*, Vol. 74, No. 8, pp 712–715, 1984.
- [5] URL: <http://www.idsia.ch/~andrea/simtools.html#libraries>
- [6] URL: <http://thelearningpit.com>
- [7] URL: <http://www.aldec.com>
- [8] Mano, Morris M. *Computer System Architecture*, Prentice Hall, 1993.

Biography

NIKUNJA K. SWAIN is a professor at South Carolina State University. Dr. Swain has more than 25 years of experience as an engineer and educator. He has more than 50 publications in journals and conference proceedings; has procured research and development grants from the NSF, NASA, DOT, DOD, and DOE; and reviewed multiple books on computer-related subjects. He is also a reviewer for ACM Computing Reviews, IJAMT, CIT, ASEE, and other conferences and journals. He is a registered Professional Engineer in South Carolina.

JAMES A. ANDERSON’S areas of specialization are in electro-optics, solid-state devices/microelectronics, and microwave and optical communications. He has performed research and design at various industries, worked as a consultant and professional engineer, and has been a university professor and Dean of the School of Engineering Technology and

Sciences (SETS) at South Carolina State University. Currently, he serves as Professor and Manager of HBCU/UP grant at South Carolina State University.

RAGHU KORRAPATI is a faculty member in the Applied Management and Decision Sciences program in the School of Management at Walden University, Minneapolis, MN. Dr. Korrapati has published numerous papers and acted as reviewer for multiple journals and conferences. Currently, he is the editor in chief of the *International Journal on Applied Technology* (IJAMT).

MUHAMED DAUDI and DARBY BRADLEY are undergraduate students in the Mathematics and Computer Science Department at South Carolina State University.