Session ENG 206-116

Design and Development of Virtual Instrument (VI) Modules for an Introductory Digital Logic Course

Nikunja Swain, Ph.D., PE
South Carolina State University
swain@scsu.edu

Raghu Korrapati, Ph.D.
Walden University
rkorrapa@waldenu.edu

Abstract

Today’s test environment is more challenging than ever as pressure on increasing quality and meeting time-to-market continue to increase. Because of these recent advances in computer and other technologies, it is becoming evident that a sound foundation in computers and computer networking is critical for success in many kinds of computer-based work To address this, majority of the undergraduate programs have courses in Digital Logic, Computer Architecture, and Computer Networking with Digital Logic being the first course and prerequisite for other computer hardware courses. The objective of these courses are to provide the graduates with solid foundation in computers and their applications to solve real world problems. A problem for these students is the solution of problems in textbooks through the use of routine problem solving techniques such as equations and formulae. But the students’ over reliance upon formulae and routine use of technique in problem solving too often lead to poor performance in real world scenarios. Also, the students’ lack of comprehension of logical and mathematical concepts results in time wastage during laboratory experiments, misinterpretations of lab data and underachievement in standardized science and engineering tests that stress the fundamentals. This problem can be effectively addressed by improving the student’s conceptual understanding and comprehension of the topics through interactive learning and teaching with virtual instruments (VI) software package like LabVIEW.

This paper will discuss design and development of interactive instructional modules (VIs) for studying (a) Number Systems, (b) Logic Gates Truth Table and Logic Diagram, and (d) Half Adder and Full Adder.

I. Introduction

The engineering, science, and technology field at present is very dynamic. This is due to the recent advances in computer and other technologies. These advances are resulted in number of 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 matter of minutes. Automation is becoming a part and parcel of every industry and industries need trained workforce to manage this new development. As a result, the engineering, technology, and science programs are under pressure to incorporate use of computers into their curriculum so that their graduates can be well trained in the use and application of these changing technologies and serve the needs of the industrial community. To address this need, most of the engineering programs and some of the engineering technology programs have introduced courses, programs, and laboratories in Digital Design and Computer Architecture to provide the graduates with both the theoretical and practical knowledge and experience. The study of computers and networking requires a good background on Digital Design, and textbooks and software programs are currently used in academia to provide the students with such background. This allows the student to spend less time in writing the code to solve the problem and to spend more time to understand the concepts. One such program is LabVIEW from National Instruments Corporation. The acronym stands for Laboratory Virtual Instrumentation Engineering Work Bench. Originally designed for test and measurement applications, the program has been modified over the years to design and analyze various complex systems. LabVIEW is a graphical programming environment and is based on the concept of data flow programming. Data flow programming concept is different from the sequential nature of traditional programming languages, and it cuts down the design and development time of an application. It is widely accepted by industry, academia, and research laboratories around the world as a standard for data acquisition and instrument control software [1]. Since LabVIEW is based on graphical programming, users can build instrumentation called "virtual instruments (VIs)" using software objects. With proper hardware these virtual instruments can be used for remote data acquisition, analysis, design and distributed control. The built-in library of LabVIEW has number of VIs that can be used to design and develop any system. LabVIEW can be used to address the needs of various courses in a technology and science curriculum [2, 3, 4]. The objective of this paper is to discuss the application of built-in VIs in LabVIEW to develop VI modules to be used in an introductory Digital Design Course.

This paper is arranged as follows: Section II discusses various LabVIEW application areas. Section III discusses the VI module for Number System Conversion. Section IV discusses the VI module for Logic Gates Truth Table and Logic Diagram. Section V discusses the VI modules for Half Adder and Full Adder. Section VI presents the conclusion and discussion and Section VII presents the references.

II. LabVIEW Application Areas

LabVIEW is extremely flexible and some of the application areas of LabVIEW [5] are Simulation, Data Acquisition & Data Processing. The Data Processing library includes signal generation, digital signal processing (DSP), measurement, filters, windows, curve fitting, probability and statistics, linear algebra, numerical methods, instrument control, program development, control systems, and fuzzy logic. These features of LabVIEW, will help us in providing an Interdisciplinary Integrated Teaching and Learning experiences that integrates team-oriented, hands-on learning experiences throughout the engineering technology and sciences curriculum and engages students in the design and analysis process beginning with their first year. LabVIEW can command DAQ boards to read analog input signals (A/D conversion), generate analog output signals (D/A conversion), read and write digital signals, and manipulate the on-board counters for frequency measurement, pulse generation, etc. The voltage data goes into the plug-in DAQ board in the computer, which sends data into computer memory for storage, processing, or other manipulation.

III. Number System Conversion

The students of science, engineering, and technology deal with different computer application areas and frequently encounter binary, octal, and hexadecimal numbers. At times they have to convert a number in one system into another which requires a) understanding the principles behind number system conversions and b) interactive modules to practice different conversions. Presented below a brief discussion of the principles behind number system conversion and the interactive module using LabVIEW (Number System Conversion VI)

A. Any Number System to Decimal Number System – *Rule 1*

Let us consider the source number as A = $(a_n\ a_{n-1} \ldots\ldots a_1 . a_{-1}\ a_{-2} \ldots\ldots a_{-n})_b$
where b = base and a = coefficients and the "a" values are between 0 and b-1.

Let the decimal equivalent of A be Deq
Deq = $a_n*b^{n-1} + a_{n-1}*b^{n-2} + \ldots\ldots + a_2*b^1 + a_1*b^0 \ .\ a_{-1}*b^{-1} + a_{-2}*b^{-2} + \ldots + a_{-n}*b^{-n}$

Example

Determine the decimal equivalent of binary number 111.11 or $(111.11)_2 = (\qquad)_{10}$?
Decimal equivalent Deq = $1*2^2 + 1*2^1 + 1*20\ .\ 1*2^{-1} + 1*2^{-2} = (4+2+1)\ .\ (.5+.25) = (7.75)_{10}$

B. Decimal Number System to Any Number System - *Rule 2*

Let D be the integer part of the decimal number and D1 be the fraction part of the decimal number. The decimal number therefore is equal to D.D1. Let "b" be base or radix of the destination number.

Converting Integer part to base "b"
Use process of repeated division. Process Terminates when quotient = 0

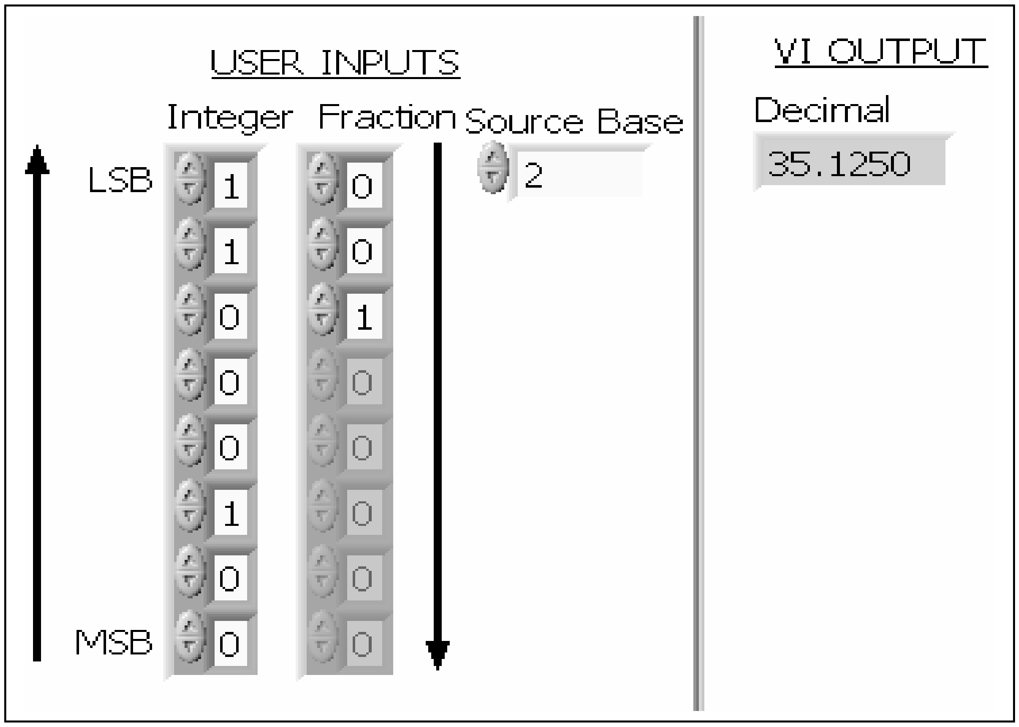
Converting Fraction part to base "b"
Use process of repeated multiplication. Process terminates when fraction part = 0

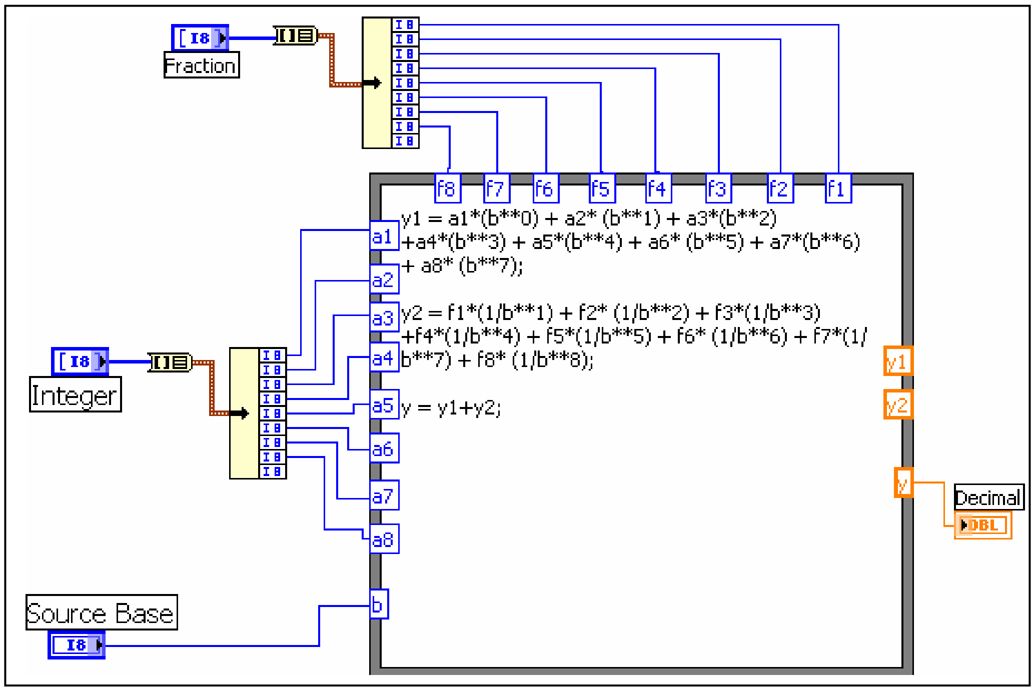
Example

Determine the binary equivalent of decimal number 7.75 or $(7.75)_{10} = (\qquad)_2$
In this example D = 7, D1 = 0.75, and b = 2.

Integer Part = 7	Converting integer part to binary		Fraction Part = 0.75	Converting fraction part to binary		Result
	Quotient	Remainder		Integer	Fraction	$(111.11)_2$
7/2	3	1	0.75*2	1	0.50	
3/2	1	1	0.50*2	1	0	
½	0	1				

Binary to Decimal Conversion VI (Front Panel)



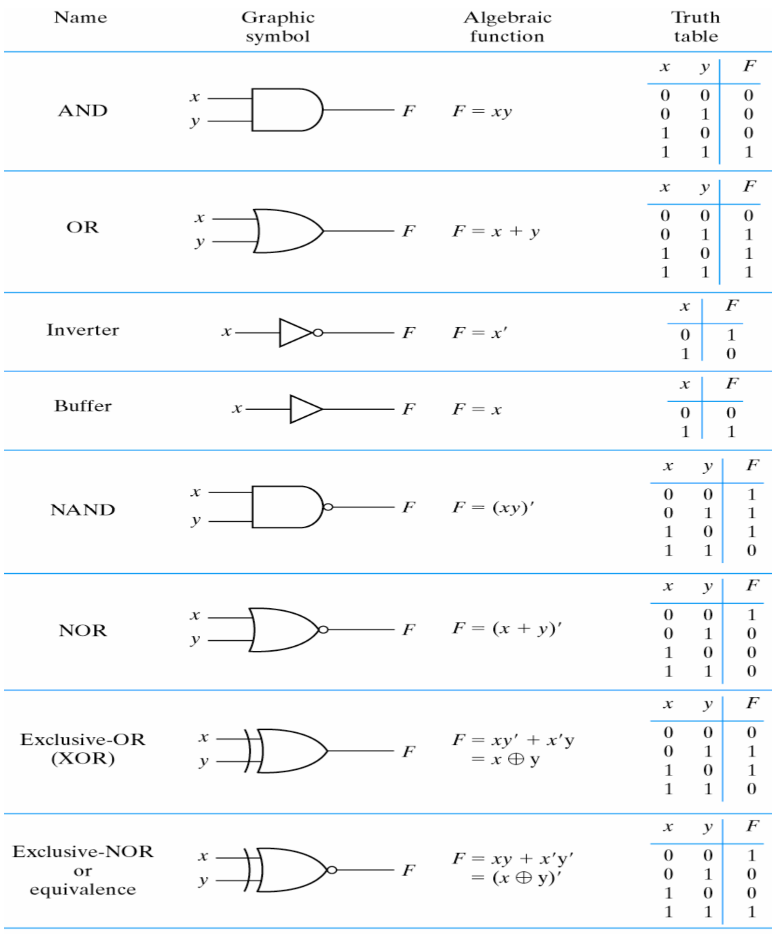


```
y1 = a1*(b**0) + a2* (b**1) + a3*(b**2)
+a4*(b**3) + a5*(b**4) + a6* (b**5) + a7*(b**6)
+ a8* (b**7);

y2 = f1*(1/b**1) + f2* (1/b**2) + f3*(1/b**3)
+f4*(1/b**4) + f5*(1/b**5) + f6* (1/b**6) + f7*(1/
b**7) + f8* (1/b**8);

y = y1+y2;
```

Figure 1 – Number Systems Conversion VI

IV. Logic Gates Truth Table and Logic Diagram

A logic gate is an elementary building block of a digital circuit. Most logic gates have two inputs and one output. There are seven basic logic gates: AND, OR, XOR, NOT, NAND, NOR, and XNOR. The following is the logic symbol, Logic Equation, and Tryth Table for these gates and is reproduced from [6].



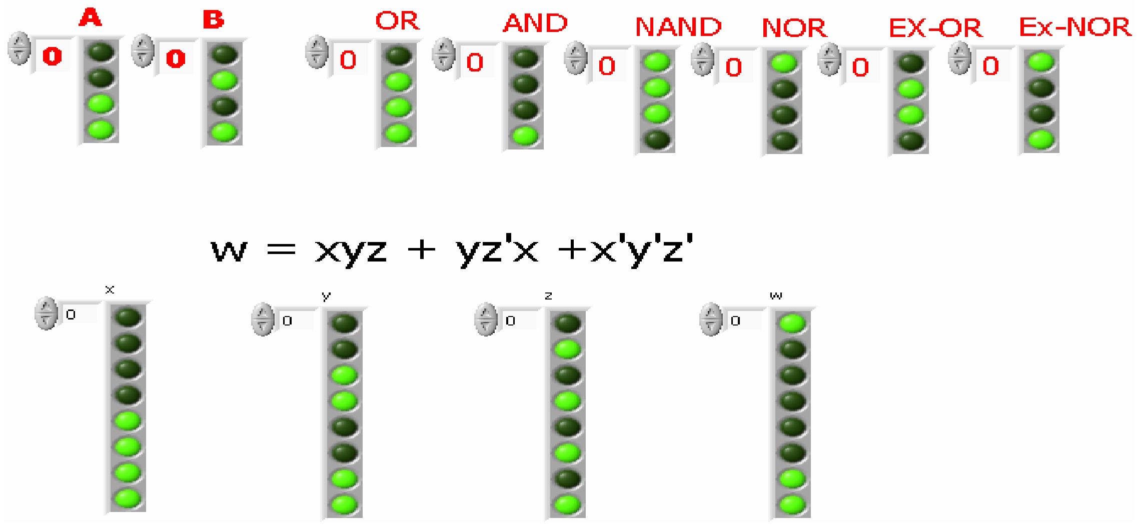
Name	Graphic symbol	Algebraic function	Truth table
AND	x, y → F	$F = xy$	x y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR	x, y → F	$F = x + y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Inverter	x → F	$F = x'$	x | F 0 | 1 1 | 0
Buffer	x → F	$F = x$	x | F 0 | 0 1 | 1
NAND	x, y → F	$F = (xy)'$	x y | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR	x, y → F	$F = (x + y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
Exclusive-OR (XOR)	x, y → F	$F = xy' + x'y$ $= x \oplus y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
Exclusive-NOR or equivalence	x, y → F	$F = xy + x'y'$ $= (x \oplus y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

Fig. 2-5 Digital logic gates

Figure 2 – Logic Gates: Symbol, Equation, and Truth Table

A digital system is usually represented by a logic equation and implemented using the gates. For example, the logic equation w = xyz + yz’x + x’y’z’ can be implemented using INVERTERS, AND gates, and OR gate. This can be also simplified by using Boolean algebra or Karnaugh’s Map (K-Map). Using Boolean Equation, w can be simplified to w = xy + x’y’z’. A Truth Table of this logic equation will produce 1s at xyz combination of (0, 0, 0), (1,1,0) and (1,1,1) and 0s at any other xyz combinations. Figure 3 presents the VI for the Truth Table of Logic Gates, Truth Table for the Logic Equation, and Logic Diagram for the logic equation.

Logic Gates Truth Table and Truth Table of Logic Equation VI – Front Panel



Logic Gates Truth Table and Truth Table of Logic Equation VI – Diagram Panel

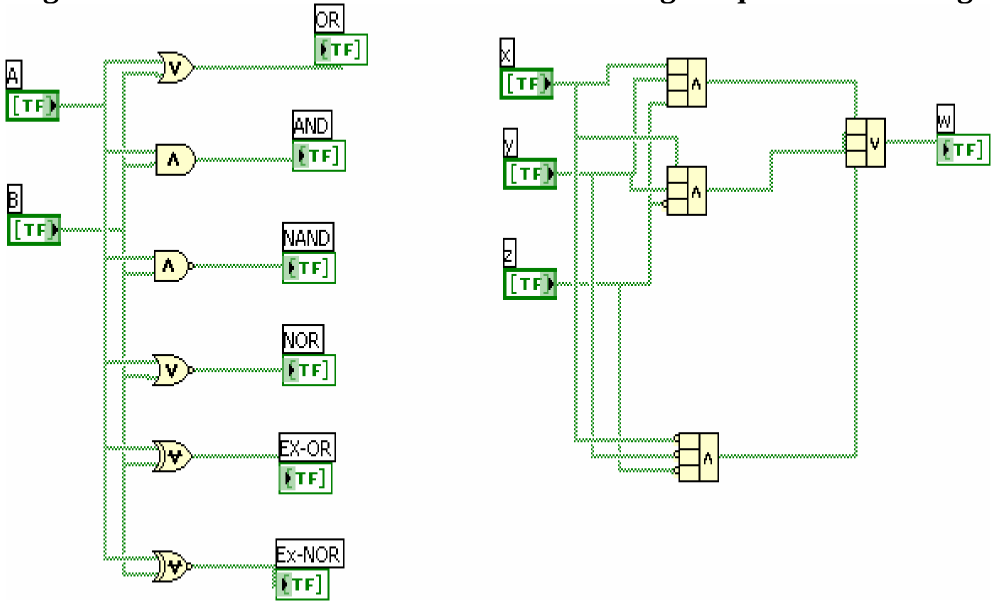
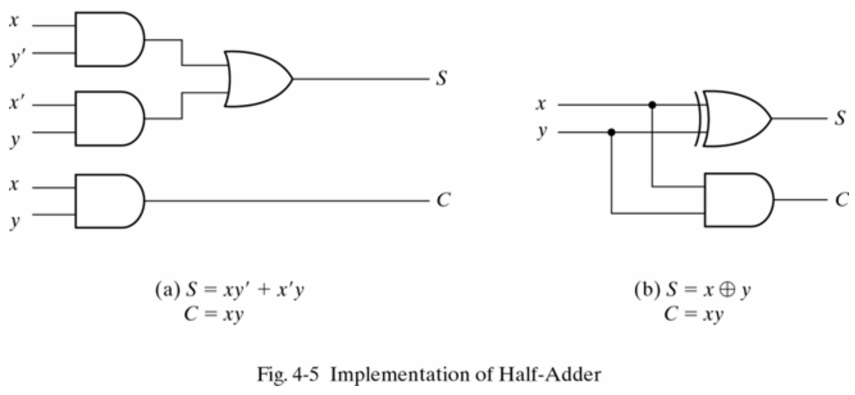


Figure 3 – Logic Gates and Logic Equation (Truth Table) VI

V. Half Adder and Full Adder VI

The logic diagram of a Half-Adder and a Full-Adder is shown in Figure 4. This figure is borrowed from [6]. A Half-Adder is used to add two single bits and a Full Adder is used to add two bits with carry. The VI modules for Half-Adder and Full Adder are shown in Figures 5 and 6, respectively [7].



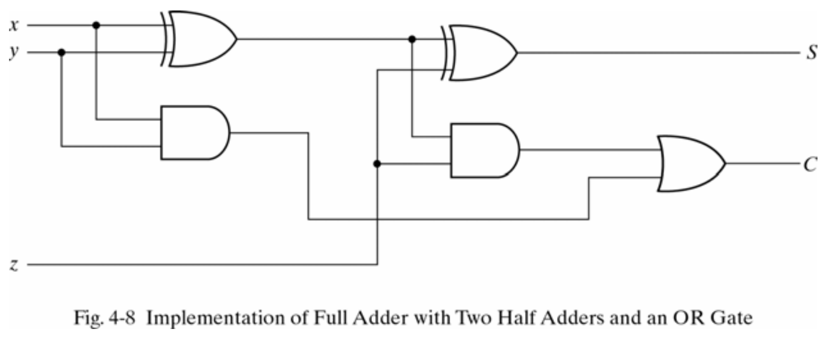
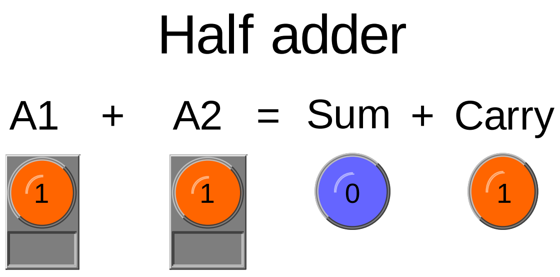

x
y
z
S
C
Fig. 4-8 Implementation of Full Adder with Two Half Adders and an OR Gate

Figure 4 – Half Adder and Full Adder

Half Adder VI – Front Panel **Half Adder VI – Diagram Panel**



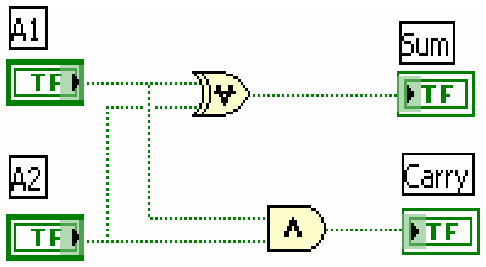
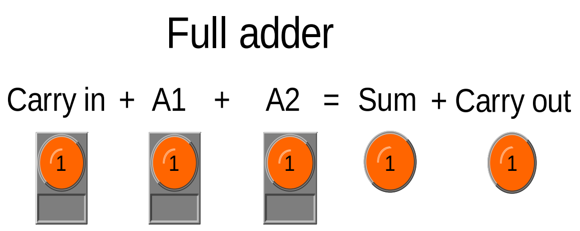


Figure 5 – Half Adder VI

Full Adder VI – Front Panel **Full Adder VI – Diagram Panel**



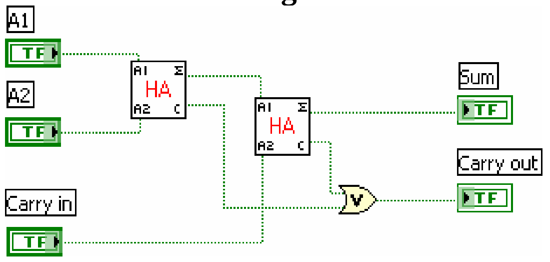


Figure 6 – Full Adder VI

VI. Conclusion/Discussion

The VI modules presented in this paper are tested with the input values from various examples in textbooks and results matched with the results of the examples. The authors have developed number of other VIs for other courses and used them as needed. The modules presented in this paper are used in Digital Logic and Computer Organization courses. A number of software packages are currently being used in engineering, technology, and sciences curriculum. While some of them require programming background, others are designed for specific course requirements. LabVIEW has features and built in virtual instrument modules identical to most of the features found in all these software packages. No or minimal programming knowledge is necessary to design and develop the VI modules. LabVIEW Therefore, one can use LabVIEW only to address the needs of various courses. This will be beneficial for students and faculty and introduce standardization across the curriculum.

VII. References

[1]. Chugani, M., Samant, A., and Cerna, N. LabVIEW Signal Processing, Prentice Hall, NJ 07458, 1998.

[2]. Anderson, J. A., Korrapati. R. B., & Swain. N. K., "Digital signal processing using virtual instrumentation". *Proceedings of SPIE Vol. 4052*.

[3]. Korrapati, R. B. & Swain. N. K., "Study of Modulation using Virtual Instruments". *Proceedings of National Conference on Allied Academies, Spring 2000.*

[4]. Swain, N. K., Anderson, J. A., & Korrapati. R. B. "Computer based virtual engineering Laboratory (CBVEL) and Engineering Technology Education". *2000 Annual ASEE Conference Proceedings.*

[5]. Lisa Wells and Jeferey Travis, LabVIEW for Everyone, Graphical Programming Even Made Easier, Prentice Hall, NJ 07458, 1997.

[6]. Mano, Morris M., Digital Design, Prentice Hall, NJ, 2002.

[7]. “Fundamentals of Digital Electronics”, retrieved on August 12, 2006 from http://www.ni.com.

Biography

Nikunja Swain is currently a Professor at the South Carolina State University. Dr. Swain has 25+ years of experience as as an engineer and educator. He has close to 50 publications in journals and conference proceedings and he has procured research and development grants from the NSF, NASA, DOT, DOD, and DOE. He has reviewed books on TCP/IP, Microprocessors, and Information Systems and a reviewer for ACM Computing Reviewes, IJAMT, CIT, ASEE, and other conferences and journals. Dr. Swain is also an adjunct faculty for Walden University, and Webster University. He is a registered professional engineer in the state of South Carolina.

Raghu Korrapati, Ph.D is a faculty in Applied Management and Decision Sciences Program in School of Management at Walden University, Minneapolis, MN, USA. His email address: rkorrapa@waldenu.edu.