

Digital Systems Laboratory for Teaching and Research

Rafic Bachnak, Dulal Kar, and Hesham Shaalan
Department of Computing and Mathematical Sciences
Texas A&M University-Corpus Christi

Abstract

The Department of Computing and Mathematical Sciences at Texas A&M University-Corpus Christi has developed a Digital Systems Laboratory that affords Computer Science and Engineering Technology students state-of-the-art training tools. The laboratory also enhances the ability of the College of Science and Technology to demonstrate science and engineering concepts to local high school students who participate in tours of the facilities on several occasions throughout the year. This paper presents the major components of the laboratory and describes some of the experiments developed for teaching a senior-level course in microprocessors and microcontrollers.

Introduction

Texas A&M University-Corpus Christi (A&M-CC) recently established a Digital Systems Laboratory (DSL) that provides interdisciplinary educational and research capabilities in several science and engineering areas¹. These areas include systems analysis and design, optimization and prototyping, hardware-software co-design, re-configurable architectures, peripheral simulation, modular design and integration, timing and state analysis, multi-layer board applications, mixed-signal simulation and design, system modeling and algorithm development, digital design methods, interfacing, and the use of microcontrollers as basic building blocks in data acquisition and control applications. The laboratory supports several courses and gives Computer Science (CS) and Engineering Technology (ET) students unique training opportunities and benefits.

Microcontrollers have been used in a variety of applications, including cameras, automobiles, printers, video games, toys, and phones. As a result, the study of Microcontrollers has been included in many engineering and technology curricula [1-3]. The Control Systems Engineering Technology Curriculum at A&M-CC includes a course that covers microprocessors and microcontrollers. The rest of this paper presents the laboratory equipment and describes some of the experiments that were developed in fall 2003 to support teaching the microprocessors and microcontrollers course..

¹ This project is funded in part by the US Army Research Office, award # DAAD19-03-1-0099

Laboratory Equipment

Computers, logic analyzers, development boards, and software form a basic set of tools required to teach advanced digital design techniques and microprocessor-based systems. In addition, electronic instruments such as power supplies, function generators, digital multimeters, and oscilloscopes must be available to allow the integration of hardware and software in an environment that addresses real world applications. The DSL consists of twelve workstations and supports a class of 24 students. The logic analyzers and PLD programmers will be placed in such a way that each of these units is shared by two workstations. The computers are networked and connected to the Internet. Detailed information about the major equipment follows.

- Computers (OptiPlex GX270T): Pentium 4 processor, 3 GHz, 1 GB SDRAM, 19" Dell UltraSharp 1901FP, 120 GB hard drive, Ethernet 10/100, 2.4X8 DVD-RW/12X10X32 CD-RW.
- Color Laser printer (HP Color LaserJet 4600N): Network ready, 17 ppm, 600 x 600 dpi, 96 MB RAM.
- Microcontroller boards (microMODUL-8051, Part #: KMM-207-C04 with 128 KB SRAM and 512 KB Flash): 12 MHz, RS 232 interface, AC adapter, serial cable, circuit diagram SBC and Development Board.
- Development Software (Keil PK51): Macroassembler, C compiler, code banking linker, debugger, extended linker, object-hex converter, simulator, and real-time kernel.
- Programmable logic device (PLD) programmers (Chipmaster 6000XP): 48-pin DIP/ZIF socket for 8- to 48-pin devices. Programs over 1200 different devices, including PAL, GAL, CPAL, CPLD, and FPGA.
- Logic analyzers (Agilent 1682A): 68-Channel 200 MHz State analysis, 800 MHz timing analysis, 20 GB hard drive, 1.44 MB floppy drive, 24XCD-ROM drive, 2 IEEE 1394 ports.
- Power supplies (Agilent E3631A): 6 V, 5 A and ± 25 V, 1 A, up to 200 W output power, front and rear terminals, GPIB and RS232 standard.
- Function generators (Agilent 33120A): 15 MHz sine wave, 15 MHz square wave, 100 kHz triangle, 100 kHz ramp, 50 mV to 10 V.
- Digital multimeters (Agilent 34401A): dc voltage, ac voltage, resistance, dc current, ac current, frequency, period, continuity, and diode. 6 ½ digits resolution, 0.0015% dc accuracy, 0.06% ac accuracy, 3 kHz to 300 kHz ac bandwidth.
- Oscilloscopes (Agilent 54641A): 350 MHz, HP BenchLink Software, automatic measurements of rise time, frequency, and pulse width.
- Digital Design Software (Cadence University Bundles for NT Servers): OrCAD Capture 9.2.3, OrCAD Layout, 44,000 library parts, 30+ netlist formats, FPGA programming, Pspice 9.2.2, Verilog and VHDL, mixed signal simulation, PCB design and analysis, system level design, verification, and IC packaging.
- Field Programmable System Level Integration Circuits (FPSLIC) University Program Design Laboratory Package (Atmel ATSTK94U): The ATKSTK94 is a design tool that includes a synthesis/VHDL Simulator, FPGA place and route, development board, FPSLIC device, 2 serial ports, and cables.

The microcontroller board and Keil software were used in fall 2003 to develop experiments for a microprocessor and microcontroller course. Test instruments were used for building and testing circuits.

Microcontroller Development Board

The kit consists of a microcontroller and development board from PHYTEC (microMODUL-8051, Part #: KMM-207-C04) that has the following features and parts: 12 MHz, AC adapter, user's manual, and circuit diagram. PHYTEC custom builds Single Board Computers (SBCs) in various sizes and configurations and provides development kits for them [4,5]. The SBC is plugged into a socket on the development board for programming and testing. Once the development stage is finished, the SBC can be removed from the development board and socketed or soldered to the user's hardware application. The serial cable allows connecting the board to a PC for programming, debugging, and testing.

The microMODUL-8051 offers more functionality than a standalone 8051 microcontroller (see Fig. 1).

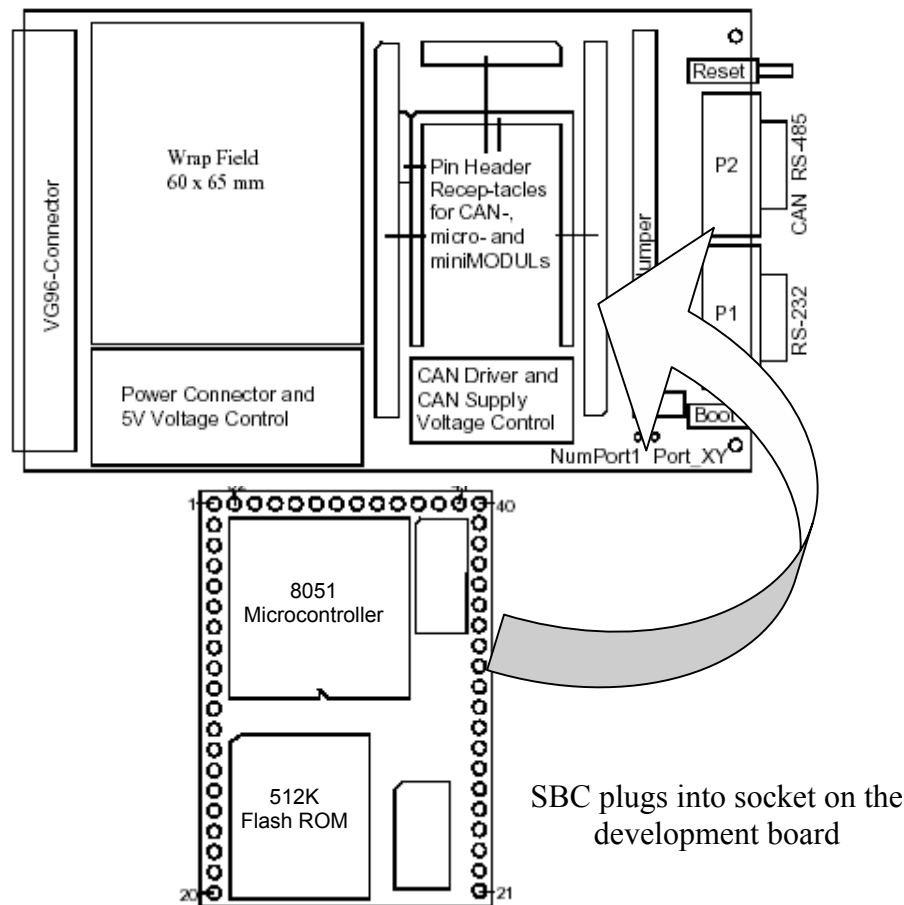


Fig. 1 The microMODUL-8051 SBC

The version available in the DSL includes the Infineon SAB C504-L microcontroller which is an extended version of the Siemens C501 8051-based microcontroller chip. Here is a summary of the main features of the C504-L:

- Fully compatible with the standard 8051 architecture
- Four 8-bit ports (2 useable)
- Three 16-bit timer/counters
- Capture/Compare Unit (CCU) for PWM signal generation and capturing
- 10-bit A/D converter with 8 multiplexed inputs
- 12 interrupt sources with two priority levels
- Programmable watchdog timer
- Power saving modes

In addition to the extended features of the 8051 chip, the microMODUL SBC board adds additional functionality to the microcontroller chip. Our particular board includes:

- 128K external SRAM memory
- 512K external Flash ROM
- RS-232 serial interface (can be configured as RS-485)
- All ports and applicable logic signals extend to pin headers at the edges of the board.

The Flash ROM device is for storage of the user program code and is easily programmed when the development board is connected to a PC.

The development board includes a 60 x 65 mm wire wrap field that can aid in system prototyping. On the end of the board is a standard sized VG-96 connector that can also be used when building custom applications. Features of the development board are listed below:

- Single power source via either 8V socket connection or VG-96 connector
- Reset and boot pushbutton switches
- Pin header sockets that accept either micro or miniMODULs
- DB9 socket for RS-232 communication
- DB9 plug that can be configured as a CAN or RS-485 interface
- VG-96 connector for developing custom applications

All of the microcontroller signals extend to standard width pin rows that line three sides of the SBC board. This way the board can be removed from the development board and plugged or soldered into a hardware application as if it were a “big chip”. The diagram in Fig. 2 shows the pin layout of the board. Alternate pin functions for ports 1 and 3 are shown next to their corresponding port pin numbers (consult the C504-L User’s Manual for further details).

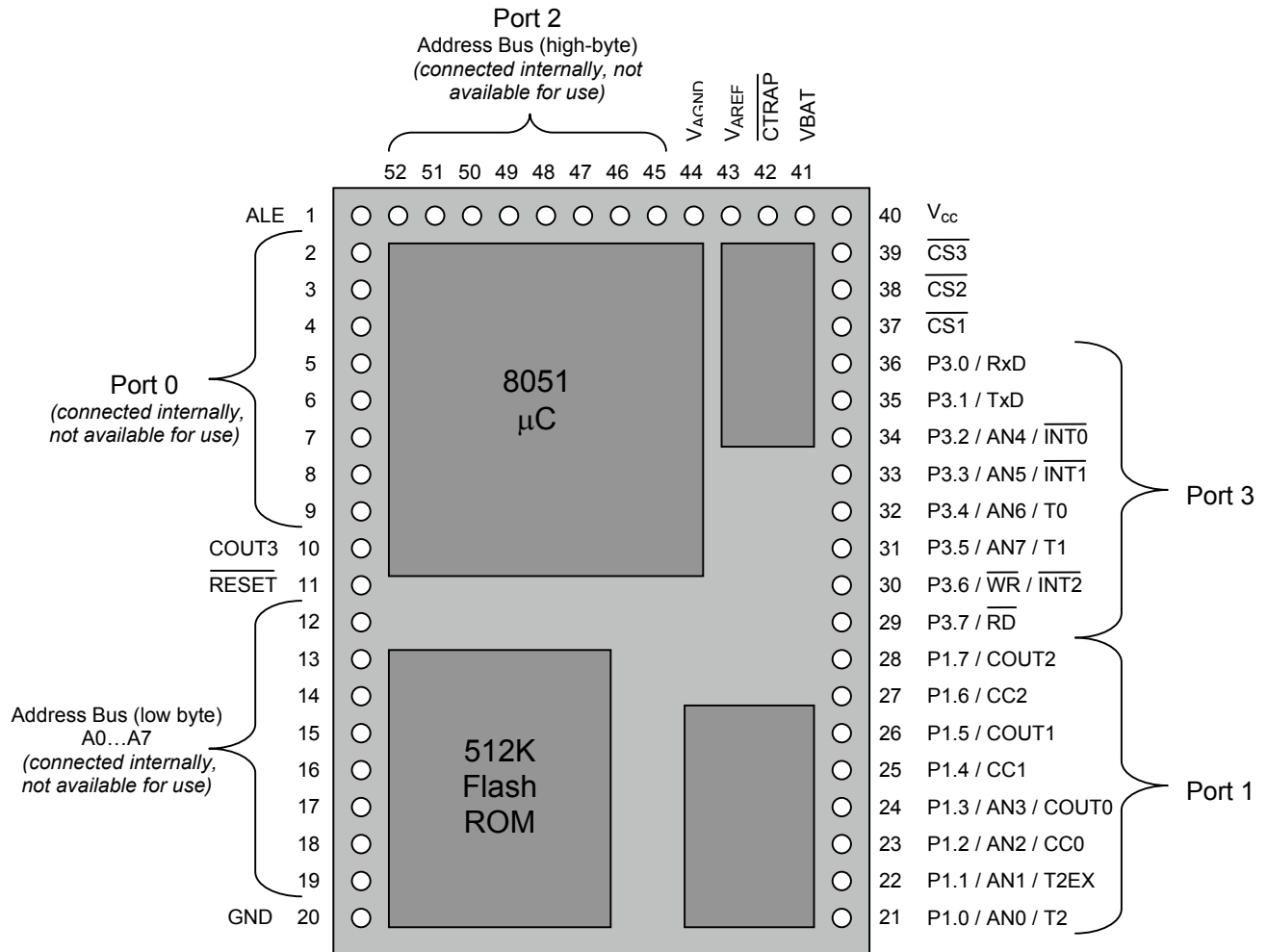


Fig. 2 Pin Layout of the 8051 Board

Keil PK51 Software Development Kit

The tools included with the PK51 Developer's Kit are: μ Vision2 for Windows™, the C51 ANSI Optimizing C Cross Compiler, the A51 Macro Assembler, the BL51 Linker/Locator, the LIB51 Library Manager, the OH51 Object-HEX Converter, and the RTX-51 Real-time Operating System [6]. The μ Vision2 IDE allows managing and developing projects in a cohesive windowed environment. With the μ Vision2 integrated environment, the compiling, assembling, and linking can be done in one step.

The μ Vision2 IDE includes a symbolic, source-level debugger that is ideally suited for fast, reliable program debugging. The μ Vision2 Debugger offers the following two operating modes:

- **Simulator** - this mode allows the user to configure the μ Vision2 Debugger as a *software-only product* that simulates most of the features of the 8051 microcontroller family without actually having the target hardware. You can

test and debug your embedded application before the hardware is ready. μ Vision2 simulates a wide variety of peripherals including the serial port, external I/O, and timers. The peripheral set is automatically selected when you select a CPU from the device database for your target.

- **Keil Monitor 51** interface - this mode allows the user to connect the 8051 hardware to the μ Vision2 debugger. This way the user can use the debugging interface to test application programs in the target hardware.

Using the debug monitor requires a few extra steps. The following steps illustrate the process with a project called Lab9.UV2.

1. Start the μ Vision2 IDE.
2. Create a new project named **Lab9.UV2** and specify the device as the **Infineon C504-L**.
3. Click on the menu **Project – Options for Target** to bring up the Options for Target dialog box. Under the **Debug** tab, choose the option **Use: Keil Monitor-51 Driver**. Also, put a check in the box: **Load Application at Startup**.
4. Click on the **Settings** button. Make sure the options are selected as shown in Fig. 3.
5. Create **Lab9.a51** (type the file and save it).
6. Add Lab9.a51 to your project.
7. Build the target then download the **Mon51.hex** file to the microcontroller. This is done using the **FlashTools98** utility. This file can be found in the directory:

C:\PHYBasic\um8051\Tools\mon\Keil\64kb

NOTE: Leave the serial cable connected to the board. It is used by the Monitor51 program to communicate with the board during testing and debugging.

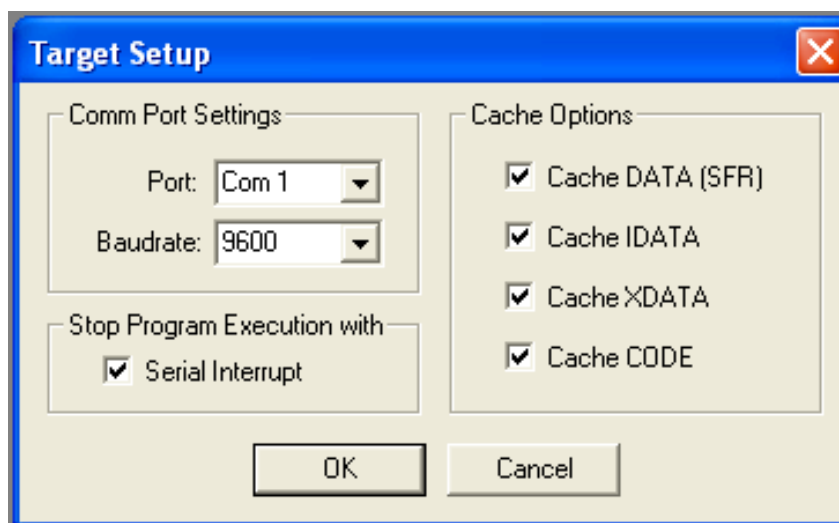


Fig. 3 Settings Options for Keil Monitor 51

Phytec Tools

FlashTools98 is a software program used to interact with the flash EEPROM located on the microcontroller module. This software allows the user to download an application onto the microcontroller flash memory. The available microcontroller module has 512K memory. This memory is divided into 8 banks of 64K each. The first bank (bank 0) is reserved for the preprogrammed microcontroller firmware. The remaining banks are available for the user application. This means 7 banks of 64K are available for the user application. The software has a safety mechanism that ensures that the system bank (bank 0) is not overwritten. Execution of the user application should always start in bank 1. The steps for downloading a user application to the firmware are outlined below.

1. Reset the target hardware and set it into flash programming mode by simultaneously pressing the Reset (S1) and Boot (S2) switches on the development board and then releasing first the Reset (S1) and then, several seconds later, the Boot (S2) switch.
2. Start the FlashTools98 program.
3. The **Communication Setup** tab is displayed first. Specify the proper serial port and baud rate (**COM2** and **9600 Baud**, for example). Click on the **Connect** button. A dialog box will appear to remind you that the microcontroller board should be in “bootstrap” mode.
4. Click **OK** when you are sure the board has been properly reset as described in step 1. This should start communication between the PC and the microcontroller board.
5. Click on the **Bank Utilities** tab. In the **Bank Erase** window, click on **Bank #1** to highlight it as shown in Fig. 4, then click on the **Erase Bank(s)** button.

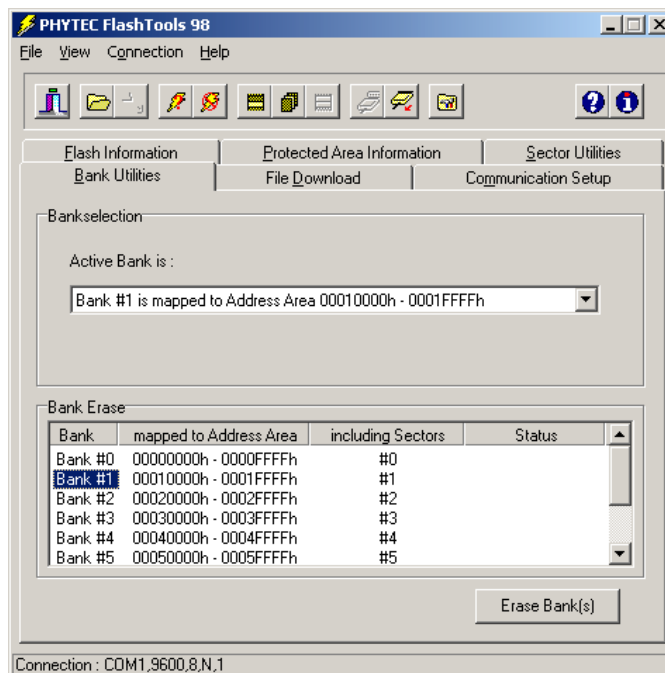


Fig. 4 Erasing Bank 1

6. Click on the **File Download** tab. Next, click on the **File Open** button. This should bring up a dialog box where you can choose the *.HEX file you created using μ Vision2. Choose the file and click **Open**.
7. Click the Download button to begin downloading your application to the board. You can watch the status of the download of the Lab9.hex program into external Flash memory in the Download window.
8. Disconnect from the target hardware after the download is finished by clicking on the **Communication Setup** tab, and then clicking the **Disconnect** button.
9. Exit FlashTools98.
10. Press the reset button (S1) to start the program running on the microcontroller.

Using the Lab to Teach Microcontrollers

As mentioned earlier, the DSL was used in fall 2003 to support teaching a course that covers microprocessors and microcontrollers [7, 8]. Nine exercises were developed using the Keil software tools to program the 8051 board. The following paragraphs briefly describe these exercises.

Exercise 1: Overview of the 8051 development board, Pin-Layout of the microMODUL-8051, Keil PK51 software development tools, creating a project, building the executable code, and debugging program.

Exercise 2: Create, assemble, and execute an assembly language program to understand the Microcontroller registers and stack operations.

Exercise 3: Create, assemble, and execute an assembly language program to understand the Jump, Loop, and Call Instructions.

Exercise 4: Create, assemble, and execute an assembly language program to understand microcontroller addressing modes.

Exercise 5: Create, assemble, and execute an assembly language program to understand microcontroller port Programming. Build a circuit, then download and test the program.

Exercise 6: Write a program that calculates the average of eight decimal numbers and stores the result in memory.

Exercise 7: Write a program that use the 8051 timers. Build a circuit, then download the program to test the proper operation.

Exercise 8: Write a program that uses the external 8051 interrupts. Build a circuit to test the program.

Exercise 9: Create, assemble, and execute an assembly language program to understand the Analog-to-Digital Interfacing capabilities of the C504 Microcontroller.

Conclusion

This paper described a new Digital Systems Laboratory that provides interdisciplinary educational and research opportunities and affords Computer Science and Engineering Technology students state-of-the-art training tools. The paper also presented experiments that were developed to teach a senior-level course in microprocessors and microcontrollers.

References

1. T. Caseres, Z. Combs, and J. Ochoa, "An advanced microcontroller systems course for upper-level undergraduate curriculum," Proc. Of the 2002 ASEE Annual Conference, Session 2432, 6 pages, Montreal, CA, June 2002.
2. H. Chaya, "An embedded systems course using the OOPic microcontroller," Proc. Of the 2002 ASEE Annual Conference, Session 1420, 7 pages, Montreal, CA, June 2002.
3. T. Hamrita, "Microcontrollers in the biological and agricultural engineering curriculum at the University of Georgia," Proc. Of the 2002 ASEE Annual Conference, Session 1526, 6 pages, Montreal, CA, June 2002.
4. Siemens. C504 Starter Kit Version 1.0 11.96. Publication Siemens AG, Bereich Halbleiter, Marketing Kommunikation, Balanstraße 73, 81541 München; 1996
5. Siemens. C504 8-bit CMOS Microcontroller User's Manual 06.96. Publication Siemens AG, Bereich Halbleiter, Marketing Kommunikation, Balanstraße 73, 81541 München; 1996
6. Keil Software, Inc. A51 Assembly and C51 Compiler in Topical Indexes, August 2001
<http://www.keil.com/support/topics.htm>.
7. Muhammad Ali Mazidi and Janice Gillispie Mazidi, The 8051 Microcontroller and Embedded System, Prentice Hall, 1999.
8. Walter Triebel and Avtar Singh, The 8088 and 8086 Microprocessors: Programming, Interfacing, Software, Hardware, and Applications, 4th Ed., Prentice Hall, 2003.

Biography

RAFIC BACHNAK

Rafic (Ray) Bachnak is a Professor in the Department of Computing and Mathematical Sciences (CAMS) at Texas A&M University-Corpus Christi (A&M-CC). He received his B.S., M.S., and Ph.D. degrees in Electrical and Computer Engineering from Ohio University in 1983, 1984, and 1989, respectively. Dr. Bachnak was previously on the faculty of Franklin University and Northwestern State University.

DULAL KAR

Dulal C. Kar received the B.Sc.Engg. and the M.Sc.Engg. degrees from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh and the MS and the Ph.D. degrees from North Dakota State University, Fargo, North Dakota. Currently he is a faculty in the Department of Computing and Mathematical Sciences at Texas A & M University – Corpus Christi, Texas. Previously, he was a faculty in the Department of Computer Science at Virginia Polytechnic Institute and State University, Virginia,

Mountain State University, West Virginia, and Bangladesh University of Engineering and Technology, Bangladesh.

HESHAM SHAALAN

Hesham Shaalan, Ph.D., is an Associate Professor of Engineering Technology at Texas A&M University – Corpus Christi. He received his BSEE and MEE from the University of Houston in 1986 and 1987, respectively, and his PhD from Virginia Tech in 1992. He previously served as a faculty member at Mississippi State University and Georgia Southern University. He is a member of Eta Kappa Nu Honor Society, American Society of Engineering Education (ASEE), and Senior member of IEEE and the IEEE Power Engineering Society. Research interests include Power Systems, Utility economics, and Software Engineering.