Western Technical College

10662134 Embedded Systems

Course Outcome Summary

Course Information

Description	With an ever increasing need, embedded systems are being used in commercial, consumer and residential products. Smart homes, cell phones, and MRI machines all use some form of embedded system. This course will introduce the student to an FPGA based soft processor. Assembly and C language programming will be used to understand the concepts of memory systems and devices, bus systems, microprocessor architecture, A/D and D/A conversions, serial and parallel communications, interrupts and interfacing to various external devices.
Career Cluster	Science, Technology, Engineering and Mathematics
Instructional Level	Associate Degree Courses
Total Credits	4
Total Hours	108

Textbooks

Programming in C (custom). ZyBook. **ISBN-13:** 978-1-5418-4715-6. Required.

Program Outcomes

1. Apply electronic theory to practice.
2. Operate test equipment.
3. Build electronic circuits and systems.
4. Evaluate the operation of electronic circuits or systems.
5. Communicate technical information.
6. Demonstrate effective programming skills.

Course Competencies

1. Explore the fundamental architecture of every computer system.

Assessment Strategies

1.1. Written Product

Criteria

You will know you are successful when

1.1. you explain the major blocks of a computer system and how they are connected.
1.2. you describe the architecture of the 68HC11 microcontroller.
1.3. you explain the interior block diagram of the 68HC11 microcontroller.
1.4. you list and describe the processor registers.

Learning Objectives

1.a. Describe the use of busses to connect computer elements.
1.b. Explain the three major functional units of a microprocessor.
1.c. Illustrate the typical registers inside the processor.
1.d. Compare and contrast various memory types.
1.e. Specify input/output functions present on most computers.
1.f. Compare the elements of the HC11 block diagram to the fundamentals of every computer system.
1.g. Use basic commands to control a microcontroller system board.
1.h. Explain the importance and function of the processor, memory, and input/output.

2. Explain how computer software controls the operation of a microprocessor based system.

Assessment Strategies

2.1. Written Product

Criteria

You will know you are successful when

2.1. you explain the differences between source code and machine code.
2.2. you write an instruction machine code listing and describing the opcode, operand and address of the operand.
2.3. you explain the difference between immediate, absolute, implied, indexed and relative memory addressing modes.
2.4. you describe the fetch and execute machine cycles of a processor performing a program instruction.
2.5. you draw and explain proper basic flowcharting symbols.

Learning Objectives

2.a. Describe the difference between source code and machine code.
2.b. Define opcode, operand, and address of an operand.
2.c. Explain the purpose of memory addressing modes.
2.d. Show how memory addressing modes work on the HC!!.
2.e. Explain how an instruction os fetched from memory and executed by the processor.
2.f. Use flowcharts to show the flow/function of a program.

3. Develop assembly language programs used to manipulate and move data.

Assessment Strategies

3.1. Written Product

Criteria

You will know you are successful when

3.1. you explain the operation of each of the flags in the Condition Code Register.
3.2. you write simple programs to load, store and move data within a microcontroller system.
3.3. you develop an assembly language program to add, subtract, multiply, divide and perform logic functions.
3.4. you demonstrate serial shifting of data in memory and processor registers.

Learning Objectives

3.a. Define the five Condition Code Register (CCR) status flags.
3.b. Write simple programs that move data using the load, store, and transfer instructions.
3.c. Perform simple arithmetic and logic operations using a variety of add, subtract and logic instructions.
3.d. Illustrate how serial shifting of data is possible in software.
3.e. Perform multiplication and division operations.

4. Create assembly language programs using branching and loops.

Assessment Strategies

4.1. Written Product

Criteria

You will know you are successful when

4.1. you explain the difference between the branch instruction and jump instruction used in the 68HC11.
4.2. you develop and test programs using the branch and jump instructions.
4.3. you evaluate proper program operation containing branch and jump instructions.
4.4. you write program code implementing the IF-THEN-ELSE and WHILE-UNTIL programming structures.
4.5. you develop programs to perform simple time delays.

Learning Objectives

4.a. Use branch and jump instructions in short programs.
4.b. Calculate the destination address and relative address when using branch instructions.
4.c. Evaluate conditional branch instructions.
4.d. Update the status flags using the compare instruction .
4.e. Use branch instructions to implement the IF-THEN-ELSE programming structure.
4.f. Implement finite loops using the WHILE and UNTIL programming structures.
4.g. Make a simple time delay using a counter in a finite program loop.

5. Access blocks data in memory using the indexed addressing mode.

Assessment Strategies

5.1. Written Product

Criteria

You will know you are successful when

5.1. you demonstrate the calculation of an effective address for an indexed mode instruction.
5.2. you write and demonstrate proper operation of a program used to sum a list of numbers.
5.3. you write and demonstrate proper operation of a program used to find and count a list of numbers using the indexed mode.
5.4. you develop a program to perform a memory-to-memory copy operation.

Learning Objectives

5.a. Calculate the effective address for an indexed mode instruction.
5.b. Perform summation on a list of numbers.
5.c. Find items in a list using indexed mode to index through the list.
5.d. Count items in a list using indexed mode to index through the list.
5.e. Accomplish memory-to-memory copy of data.

6. Develop software programs effectively using subroutines.

Assessment Strategies

6.1. Written Product

Criteria

You will know you are successful when

6.1. you explain the push and pull instructions used in conjunction with the stack.
6.2. you develop and test simple programs using the subroutine control instructions JSR, BSR and RTI.
6.3. you create programs using nested subroutines.
6.4. you test subroutines found in the microcontroller's ROM monitor program.

Learning Objectives

6.a. Use push and pull instructions to move data to and from the stack.
6.b. Explain the use of subroutine control instructions: JSR, BSR and RTI.
6.c. Describe how parameters are passed to subroutines.
6.d. Use nesting to call subroutines from other subroutines.
6.e. Access subroutines in the BUFFALO monitor program.

7. Write and implement source code using an assembler.

Assessment Strategies

7.1. Written Product

Criteria

You will know you are successful when

7.1. you demonstrate techniques of writing source code that is readable.
7.2. you define the contents of S records and interpret what the fields of each contain.
7.3. you utilize assembler directives, labels and directives to enhance source code.
7.4. you explain the significance of S19 files used in the 68HC11 microcontroller.

Learning Objectives

7.a. Demonstrate techniques of writing source code that is readable.
7.b. Utilize assembler directives, labels and directives to enhance source code.
7.c. Explain the fields in the listing file.
7.d. Define the contents of S records and interpret what the fields of each contain.

8. Explain the operation of memory used in a computer system including RAM, ROM, and EEPROM.

Assessment Strategies

8.1. Written Product

Criteria

You will know you are successful when

8.1. you explain the HC11's memory map including its RAM, ROM, and EEPROM on-chip memory devices.
8.2. you explain how address decoding is used in computer systems.
8.3. you describe the steps needed to erase and program an EEPROM.
8.4. you explain how memory can be expanded on the 68HC11.

Learning Objectives

8.a. Demonstrate the concepts of address decoding.
8.b. Show how the on-chip memory resources are allocated on the HC11.
8.c. Use control bits to activate the EEPROM and ROM.
8.d. Explain how to erase and write EEPROM locations.
8.e. Demonstrate how to connect external memory devices to the HC11.

9. Create software programs for general purpose input/output (I/O).

Assessment Strategies

9.1. Written Product

Criteria

You will know you are successful when

9.1. you describe the input/output ports on the 68HC11.
9.2. you create proper code to input and output data on the HC11 B, C, D and E ports.
9.3. you explain the concept of data masking.
9.4. you demonstrate I/O handshaking.

Learning Objectives

9.a. Write program code to output data onto the Port B pins.
9.b. Program the data direction of the Port C pins.
9.c. Use the C Port for input and output of data.
9.d. Output and input external system data using handshaking techniques.
9.e. Receive parallel data at the HC11 from an external device using Port E.

10. Develop software programs using interrupts and resets.

Assessment Strategies

10.1. Written Product

Criteria

You will know you are successful when

10.1. you explain how interrupts are used in computer systems.
10.2. you describe the difference between maskable and nonmaskable interrupts.
10.3. you write a program demonstrating the use of interrupts and resets.
10.4. you test a software program for proper interrupt operation.

Learning Objectives

10.a. Describe the process of servicing an interrupt.
10.b. Enable and mask interrupts using global and local control bits.
10.c. Demonstrate the use of a vector table and vector jump table.
10.d. Describe the impact of hardware resets on all hardware systems.

11. Interface I/O devices to the microcontroller using A/D and D/A control.

Assessment Strategies

11.1. Written Product

Criteria

You will know you are successful when

11.1. you calculate the range, step voltage and resolution of an analog-to-digital converter.
11.2. you describe the four major block of the HC11's A/D conversion system.
11.3. you create and test a program used to input analog data into the processor.
11.4. you create and demonstrate a program used to control the D/A conversion process to an output interface device.

Learning Objectives

11.a. Explain the process of analog-to-digital conversion.
11.b. Describe the four major functional blocks of the HC11 analog-to-digital system.
11.c. Write software code for HC11 single and multichannel conversions.
11.d. Interface a solid state temperature sensor to the HC11.
11.e. Construct and test an interface to control a DC motor using the HC11.

12. Develop asynchronous and synchronous serial communications programs.

Assessment Strategies

12.1. Written Product

Criteria

You will know you are successful when

12.1. you differentiate between asynchronous and synchronous serial communications.
12.2. you write a program to enable the HC11 to transmit and receive data serially.
12.3. you explain the operation of the HC11's serial peripheral interface when using the Port D pins.
12.4. you verify the proper operations of software programs using serial communications.

Learning Objectives

12.a. Explain the difference between asynchronous and synchronous serial communications.
12.b. Configure the HC11 for Serial Communications Interface (SCI) BAUD rate.
12.c. Describe the function of the Serial Peripheral Interface (SPI).

13. Create simple software programs for the microcontroller using C programming.

Assessment Strategies

13.1. Written Product

Criteria

You will know you are successful when

13.1. you explain the difference between assembly language and a higher-level language such as C.
13.2. you develop simple programs using statements written in C.
13.3. you develop a C program used for interfacing external I/O devices.
13.4. you test and verify the operation of C programs for the 68HC11 microcontroller.

Learning Objectives

13.a. Understand the basic concepts of programming in a higher-level language.
13.b. Write simple C statements and expressions.
13.c. Write short C programs to accomplish specific tasks on the HC11.