Programmed Introduction to MIPS Assembly Language

Central Connecticut State University

QtSpim Edition, August 2015

This is a course in assembly language programming of the MIPS processor. It emphasizes the topics needed for study of computer architecture: bits, bit patterns, operations on bit patterns, and how bit patterns represent instructions and data. This course is equivalent to a semester-long junior college or university course (except, perhaps, for the emphasis on bit patterns).

The emphasis of the course is on understanding how computers work. This will provide a basis for further study of computer architecture and computer software. The MIPS processor, the subject of this course, has a well designed architecture and is particularly fruitful to study. However, the goal of the course is not to turn you into a MIPS programmer, but to give you an understanding of all computer systems.

The only equipment you need for this course is a PC. The only software you need is the SPIM simulator of the MIPS32 processor and a text editor. The simulator is available by free download (see appendix A). Example programs are presented using an MS Windows operating system. However, you can use any platform that runs SPIM. (There are many).

Assembly Language is normally taken the semester after a course in a high level programming language (such as Java or C). This course assumes that you have this background although no specific programming language is required.

Read the pages of this course *actively*. Think about and answer the question at the bottom of each page. (This style of tutorial is called *programmed learning*. It is very effective for technical material). Most questions call for just a little thought. Some call for pencil and paper. Keep a pencil and a scrap of paper next to your keyboard. Each chapter is about 15 pages long. Spend several minutes per page. You can read each chapter in about 30 minutes. However, for maximum benefit, you should run some of the example programs, write some programs of your own, and then think about your results. This may take several hours.

Part 1: Prelude to Assembly Language

Assembly language: what it is, why it is studied, and where it is used.

Part 2: Data Representation

Data: characters and integers. The *binary addition algorithm*.

- Chapter 5 — Characters. Quiz
- Chapter 6 — Number Representation. Quiz
- Chapter 7 — Binary and Hex Representation. Quiz Flash Cards
- Chapter 8 — Binary Addition and Two's Complement Representation. Quiz

Part 3: Running SPIM; Bitwise Logic

Running SPIM. MIPS programming. Bitwise logic operations.

- Chapter 9 — A Simple SPIM Program. Quiz
- Chapter 10 — MIPS Programming Model. Quiz
- Chapter 11 — Bitwise Logic with Immediate Operands. Quiz
- Chapter 12 — Shift Instructions and Logic Instructions. Quiz Programs

Part 4: Integer Arithmetic and Memory Access

Integer arithmetic. Moving data to and from memory.

- Chapter 13 — Integer Addition and Subtraction Instructions. Quiz Programs
- Chapter 14 — Integer Multiplication, Division, and Arithmetic Shift. Quiz Programs
- Chapter 15 — Memory Access**:** Loading and Storing Registers. Quiz Programs
- Chapter 16 — More Memory Access: Bytes and Halfwords. Quiz Programs

Part 5: Branches, Decisions, and Loops

Program flow**:** branch, jump, and set instructions; loops, and decisions.

- Chapter 17 — Jump and Branch Instructions. Quiz Programs
- Chapter 18 — Set Instructions and more Branch Instructions. Quiz Programs
- Chapter 19 — Structured Programming. Quiz Programs
- Chapter 20 — Programming Examples. Quiz Programs

Part 6: Extended Assembly Language

The assembler extends bare machine language. Registers have mnemonic names. *Pseudoinstructions* extend the bare hardware.

- Chapter 21 — The Extended Assembler. Quiz Programs
- Chapter 22 — The SPIM Exception Handler. Quiz Programs
- Chapter 23 — Instructions for Bitwise Logic and Math. Quiz Programs
- Chapter 24 — Branch Instructions, Set Instructions, and Indexed Addressing. Quiz Programs

Part 7: The Stack and Subroutine Linkage

Programs are divided into sections called subroutines. At run time, the stack is used to save and to restore a subroutine's values.

- Chapter 25 — The Run-time Stack. Quiz Programs
- Chapter 26 — Simple Subroutine Linkage. Quiz Programs
- Chapter 27 — Stack-based Linkage Convention. Quiz Programs

- Chapter 28 — Frame-based Linkage Convention, Variables, and Recursion. Quiz Programs

Part 8: Floating Point Data

Bit patterns are used to represent floating point numbers. More machine instructions are used to do floating point arithmetic.

- Chapter 29 — Binary Fractions. QuizA QuizB
- Chapter 30 — IEEE 754 Floating Point. Quiz
- Chapter 31 — Floating Point Arithmetic on MIPS. Quiz Programs
- Chapter 32 — Floating Point Comparison Instructions. Quiz Programs

Part 9: Data Structures in Assembly Language

- Chapter 33 — Dynamic Memory Allocation. Quiz Programs
- Chapter 34 — Data Structures. Quiz Programs
- Chapter 35 — Linked Lists. Quiz Programs
- Chapter 36 — Objects. Quiz Programs

Appendices

- Appendix A — Downloading and Installing SPIM.
- Appendix B — Register Use Chart
- Appendix C — MIPS Assembly Instructions
- Appendix E — Binary Addition Calculator (Applet)
- Appendix F — ASCII Chart
- Appendix G — SPIM Exception Handler Services
- Appendix H — Decimal Representation to Binary Conversion (Discussion)
- Appendix I — Decimal to Binary Converter (Applet)

Index

- Index

Main Tutorial Menu