Microprocessor Systems **Week 5 Lab 3: Introduction to MIPS32 Assembly**

I. Learner Objectives:

At the conclusion of this lab, participants should be able to:

- Design and implement subroutines in MIPS32 assembly
- Interface with basic peripheral modules in MIPS32 assembly

II. Prerequisites:

Before reading this lab, participants should be able to:

- Outline the lab related to ports
- Summarize the unit related to registers

III. Keywords:

MIPS32 assembly

IV. Pre-lab:

None.

V. Background:

This lab will introduce you to the wonderful world of MIPS32 assembly. The idea behind the lab is to introduce you to basic low level design and programming. You will be using a switch peripheral module and on-board LEDs to construct a basic calculator.

As you should already know, the following table lists the instructions for using I/O Port registers. The *n* in the names for the registers may represent A - G. Each of the registers is 32-bits, where each of the bits may be set to read/input or write/output. 1 represents input and 0 represents output. The appropriate TRISn register should be set to indicate whether or not to read or write from/to a bit of the port. If all of the bits in the TRISn register are set for read, then contents from the port will be stored in the corresponding bit position in PORTn. If all of the bits in the TRISn register are set for write, then the contents to write to the port will be stored in the corresponding bit position in LATn. Refer to Table 1. Use the MIPS32 assembly instruction LW (Load Word) to read from a port and SW (Store Word) to write to a port.

Operation	TRISn	PORTn	LATn
Read (RD)	1	Content read from port	N/A
Write (WR)	0	N/A	Content written to port

Table 1: Read and Write Operations

Table 2 summarizes the pin connections between the Digilent 6-pin 4-slide switch peripheral module (plugged into the JH-07:10 connector) and the Microchip PIC32MX460F512L microcontroller. Note that you may find the port and function for the pin in the Cerebot MX4cK / chipKIT Pro MX4 Reference Manual document.

Cerebot Pin	MCU Port/bit	Switch
JH-07	RE08	SW1
JH-08	RD00	SW2
JH-09	RD08	SW3
JH-10	RD13	SW4

Table 2: Port to Switch Mapping

Table 3 summarizes the pin connections between the on-board LEDs and the Microchip PIC32MX460F512L microcontroller. Recall your goal is to map the state (OFF/ON) of a switch to the state (OFF/ON) of a LED.

Cerebot Pin	Port/bit	Peripheral LED
JK-01	RB10	LD1
JK-02	RB11	LD2
JK-03	RB12	LD3
JK-04	RB13	LD4

Table 3: Port to Peripheral LED Mapping

VI. Lab:

Before you begin writing MIPS32 code for the lab make sure you have one end of the USB cable plugged into the Cerebot MX4cK / chipKIT Pro MX4 micro-B "DEBUG" port and the other end into one USB-A port on your PC. Also, be sure to plug in the switch module that came with your Digilent board package into the JH-07:10 connector of the Cerebot MX4cK / chipKIT Pro MX4 board. The top pins correspond to 01:06 and the bottom to 07:12. Turn on power to the board by flipping the switch next to the "DEBUG" port.

In this lab you will create calculator of sorts using MIPS32 assembly. Thus, you will create a source file type than in previous labs. While you are creating a new source file ensure you select the .s assembly file type as shown in Figure 1.

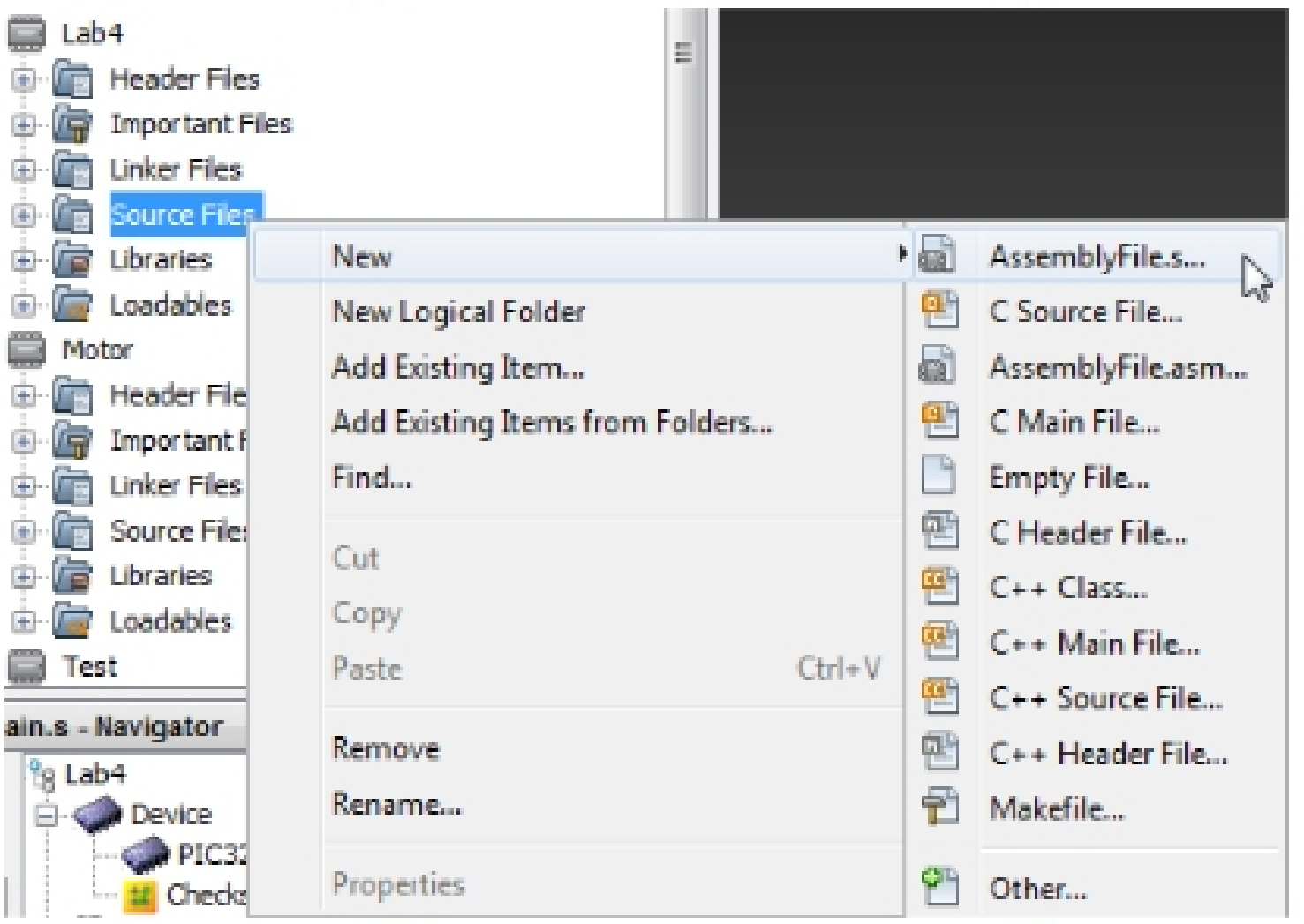

Figure 1: ASM32 Assembler

If the .s file type does not show up initially in the menu, click on "Other..." file types.

In this lab you will be performing calculations like the following, using the switch module:

Select OPERATOR (with SW1 and SW2), according to the table below; if after a couple of seconds no switches are flipped, then assume Addition
Select OPERAND1 (with SW3 and SW4)
Optional: Select OPERAND2 (with SW3 and SW4), only required for binary operations

You will use the switch module to select an operation to perform along with the operands to use in the calculation. The following table (Table 4) demonstrates the state of the switches corresponding to operations.