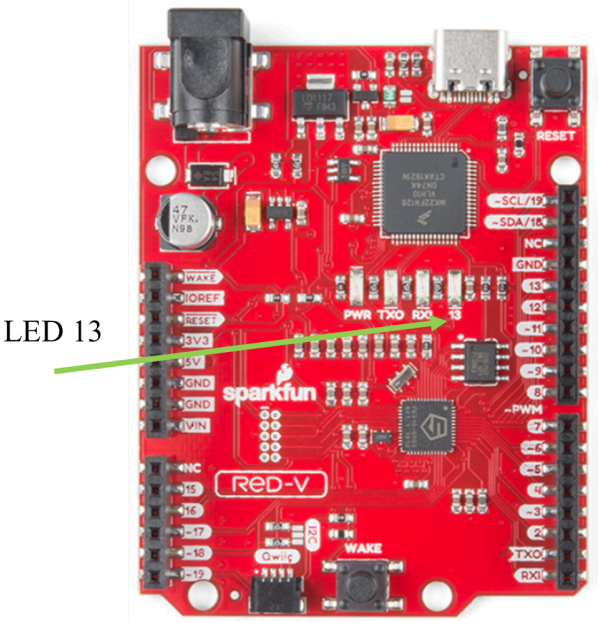
Computer Architecture - CS2323. Autumn 2024

Lab-5 (LED blinking on RED-V board)

Important: Handle the board carefully as these are delicate electronic devices and can be damaged due to mishandling.

Having used Ripes simulator until now, we will execute a simple program on a real RISC-V processor available on the Sparkfun RED-V Redboard. This assignment can be done in groups of upto **two**. Please note that the board has 32-bit registers (RV32 architecture) and hence ld/sd will not work. We must use lw/sw instead to read/write to memory addresses.



We would use FreedomStudio as a tool instead of Ripes for this lab exercise (as Ripes does not allow working with the board). The instructions for the same are captured as a separate document here. The software will already be installed in the lab systems. You can take the board and cable from the lab staff and connect to the PC using the USB port. Be careful about the file-names etc. to match what is given in the document, else you will see unwanted errors.

The Sparkfun RED-V board contains an LED on the board (labeled 13 next to the LED). Following are the steps for controlling the LED.

1. The 32-bit word at address 0x10012004 should be written with 0x00000000 (once at beginning) - this tells the system that no pin acts as input pin.
2. The 32-bit word at address 0x10012008 should be written with 0x00000020 (once at beginning) - this tells the system that the pin in position-5 from LSB acts as output.
3. To glow the LED: The 32-bit word at address 0x1001200C should be 0x00000020.
   To turn-off the LED: The 32-bit word at address 0x1001200C should be 0x00000000. - these write a value of 1 or 0 to the pin in position-5, making them ON or OFF.

Write an assembly program using RISC-V instructions in the following steps:

1. Glow the LED and turn-off the glowing LED to check that you are able to control the LED
2. Perform the above two operations in a loop to blink the LED available on the RED-V board. Please note there must be some appropriate delay between the ON and OFF steps of the LED to see a blinking effect by our eyes, else you may just observe that LED remains constantly ON. The blinking speed should be tunable based on a value that you put in the .data segment. Change the blink speed and see the effect.
3. Optional (for exploration): Modify the program to have different durations for ON and OFF and create various blinking patterns.

You may develop the code upfront and use the lab hours primarily to tune and run the program on the board.
The FreedomStudio follows a slightly different format for the assembler directives and you may modify these parts when using FreedomStudio:

```
.section .data            # (instead of .data in Ripes)
L1: .word 100000       #delay count to be loaded from memory

.section .text
.global main   # (add this to indicate main is a global function, need not be there in Ripes)
main:
        la x3, L1
        #YOUR CODE FOLLOWS HERE. The address of the data segment is available in x3
```

Submission instructions:
Submit the assembly code as a file named Lab5_ROLLNUM1_ROLLNUM2.s (e.g., Lab5_CSYYBTECHXXXXX_CSYYBTECHZZZZZ.s) in moodle. Also mention the names of the team members in comments in the assembly code file. Only one student of the group should submit the assignment. The working demo with LED blinking on the board should be shown to the TA by both the team members to receive marks for the assignment.

Important: Handle the board carefully as these are delicate electronic devices and can be damaged due to mishandling.