Fall 2021 CSE4117 Microprocessors
PROJECT PART of the FINAL EXAM

Due Date: 4.02.2022 Friday 23:59
Demo Date: Announced later.

Question 1 Verilog Implementation

You will implement a CPU board with two input devices (a timer and a keypad) and one output device (4-digit 7-segment led display) on FPGA.

-Write the Bird processor as described in the class as a Verilog module, but extend its address bus to 16-bits (the processor that you have implemented in 2nd project). It must be complete with push, pop, call and return instructions. Note that you have to modify jmp, jz and call instructions according to the new address bus width. (and also note that you have to modify your Assembler according to new design)
-Write a timer module in Verilog. This module will contain a 16-bit "current time" register. At (approximately) every second, it will increment this register. This timer module works via polling. Whenever the push button (you can choose any one of the two buttons on the FPGA board for this purpose) is pressed, the ready bit in the status register will become 1 and the CPU will read the current time register, which will make the ready bit 0. The CPU will display the contents of the timer register on the 7-segment display. The memory address in the address map corresponding to the registers of this timer and the 7-segment display will be decided by you.

-Write a keypad module which will receive input from the keypad. This keypad module also works via polling. When a key of the keypad is pressed, the key is recorded in its data register, and the ready bit of its status register becomes 1. When CPU reads the data from its data register, the ready bit will return back to 0. The memory addresses corresponding to these keypad registers will be decided by you. (Works as in 2nd project)

The keypad will be used as an adder.

- Since the decimal operations will be performed, only the digits 0-9 will be used for number entry.
- The "A" key will reset the whole system.
- The "*" will be used as the addition sign.
- When any digit is entered, the number will be displayed on the 7-segment display, then when another digit is pressed, the two-digit number will be displayed on the 7-segment display, and so on. If the entered number exceeds 4 digits, the last entered four digits will be stored and used in the calculation.
- After a number is entered and then "*" is pressed, the 7-segment display will be reset and the 2nd number will be read from the keypad. If "*" is pressed again, the addition will be done and the result will be displayed on 7-segment display, and so on.
- The system will keep the accumulated sum and add it to every newly entered number till the “A” key is pressed, which will reset the whole system.

Notice that, all **outputs** will be in **decimal** format.

Question 2 Logisim Implementation

In Logisim, add interrupts to the CPU you have designed for the 2nd project.
For this CPU, you will design an I/O system which will have two 16-bit switches with a push button and a timer as input and two four-digit 7-segment displays as output.

The system will

--input from the two16-bit DIP-switches via polling (ie, you also need to implement a push button) and output to a four-digit 7-segment display. The output will be unsynchronized. The program running on this system will read two 16-bit numbers as input from the switchboards and call a subroutine, passing these two numbers as arguments. The subroutine will add the two numbers and display the result on the 7-segment display and return (this part is very similar to 2nd project).

But in addition to these, you will also add a timer and one more four-digit 7-segment display to the system.

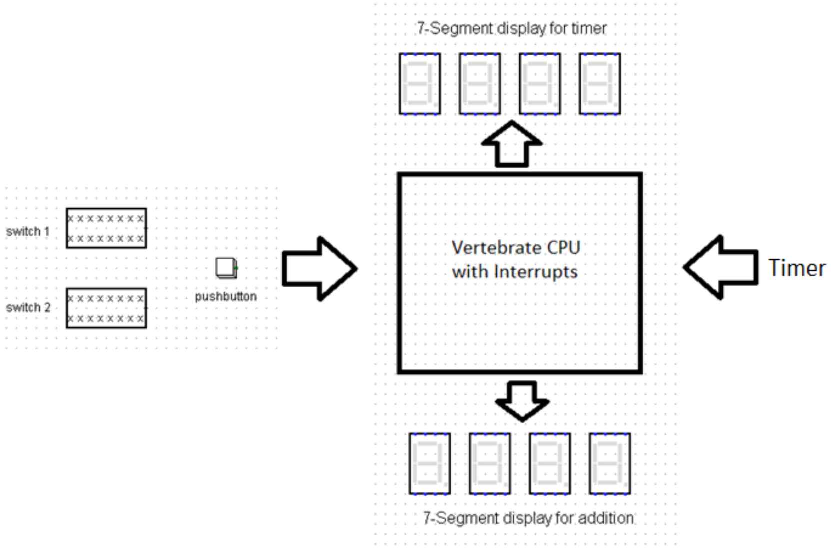
--The timer will have a 16-bit register. At (approximately) every second, it will increment this register and send an interrupt to the CPU. After receiving the interrupt, the CPU will go to an interrupt service routine (ISR), which will read this register and display its content on the second 7-segment display and return.

You must use the latest CPU (Vertebrate) shown in the class which is capable of handling interrupts. You must also use a PIC. You will choose which IRQ pin of the PIC you will use.

You will choose your address map (i.e, which addresses you will assign to the timer, switches, 7-segment displays etc.)

Your system must work at 4KHz clock of Logisim. When the system starts, I must be able to see the timer incrementing at (approximately) every second in one 7-segment display, while I must be able to perform additions by using the switchboard and the other 4-bit 7-segment display.

All the outputs will be displayed in decimal format.


7-Segment display for timer
switch 1
switch 2
pushbutton
Vertebrate CPU
with Interrupts
Timer
7-Segment display for addition