CSE4117 Microprocessors HW3 [draft]

Due date:
Demo date: Will be announced later.

Question 1: SystemVerilog Part
You will implement a modified version Bird in SystemVerilog and write the specified software that will run on it.

http://www.marmaralectures.com/reference-for-second-project/

The above link contains the code for the Bird CPU, the keyboard, 7-segment display, the main module, and the assembler with some missing sections (marked with the remark "to be added").

- **Increase the address bus of the BIRD CPU given in the class to 16 bits.** All the other properties of the BIRD CPU, ie. data bus width, number of registers etc., must remain same. Do all the necessary changes to the instruction set, FSM and assembler. Do not add or remove instructions to the original Bird instruction set. In other words, the only change you will make is to upgrade the CPU's address space to 64k.
- Modify and complete all the codes given in the link,
- Load it into the FPGA kit,
- Connect the keyboard and the 7-segment monitor to FPGA and run.

Write an assembly program in **Modified Bird** which will turn your hardware into a pocket calculator which is capable of adding and multiplying numbers.

- 'A' key will act as sum (ie, +)
- 'M' key will act as multiplication (ie, *)
- 'R' key will act as reset
- It is assumed that all key presses are of short duration, ie, only two characters will be generated for each key press.

There will be no "previlege" in calculations, ie, if you enter 3A2M10 the result will be 50, not 23. Or, in other words, + and * will have the same privilege.

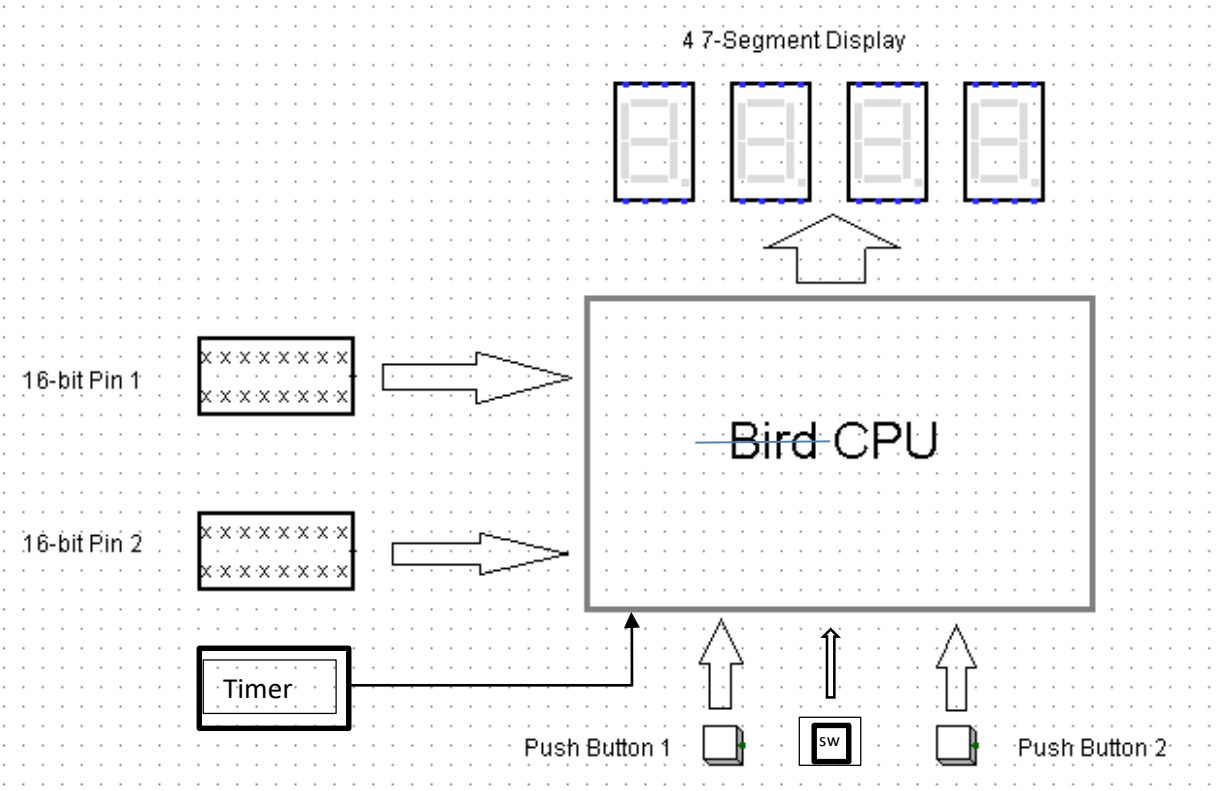
In your code, **you must have at least two function calls**, "call addit" and "call mult", to perform addition and multiplication after you finish entering a number and press M or A.

Your calculator must read the numbers from the keyboard and display them in **decimal** on seven segment display. To display your result in decimal, you must write an additional piece of code (**Assembly code, not in Verilog**) to convert your hexadecimal result into decimal (or binary coded decimal, BCD). You must do this in software and **NOT** in hardware. Some hints on how this can be done is described in the following link:
https://my.eng.utah.edu/~nmcdonal/Tutorials/BCDTutorial/BCDConversion.html
Disregard "BCD conversion in hardware" part.

There is an accompanying video which shows how the system you will build should work. (Note that in this video numbers are hexadecimal but you will implement it as decimal)
Note: You should write everything in SystemVerilog, ie, NO VERILOG !!!

Question 2: Logisim Part

You will implement Vertebrate defined in question 1 in Logisim and write the specified software that will run on it.



You have;
--Vertebrate CPU (Defined in question 1)
--Two 16-bit switchboards, switchboards 1 and 2
--Two Push Button, PB1 and PB2
--One 4*7-segment display
--One timer
--One switch

Your assembly program will have a variable, called “sum”, which is initialized to sum=0 at the beginning. Then your code will continuously poll the 16- bit switchboards (16-bit pins).

-- If PB1 is pressed, you will read the number from switchboard 1, calculate
sum=2*sum+(entered number),
-- If PB2 is pressed, you will read the number from switchboard 2, calculate
sum=sum*(entered number),

The timer keeps the time by regularly incrementing its internal counter register by one approximately every second and send to CPU an interrupt. CPU will read the time from the counter register and assign it to a variable called “sum”.

When switch is up, the CPU will display the variable “ time”. When switch is down, CPU will display the variable “sum”.

All numbers must be converted to decimal before being displayed. **The conversion process must be done in software in a function written for this purpose. Hardware solutions won’t be accepted.**

Have you implemented conversion to 16 bit address bus, do all instructions work properly?
Do you manage to take input from keyboard?
Do you manage to send the output to 7-Segment Display?
Do you manage to run your program correctly and get correct results?
Do you manage to print your results as decimal or hexadecimal?
Where did you put your devices? (ie. draw your address map below)
Draw your flowchart of your assembly program?