COEN 122L: Project, Version with SUM

Design a Structural Model of a Pipelined CPU

The project is to design a structural model of a pipelined CPU with 13 instructions using Verilog HDL.

Description:

In this project, you are asked to design a 32-bit pipelined CPU for the given SCU Instruction Set Architecture (SCU ISA). The SCU ISA is described below.

- Register file size: 64 registers, each register has 32 bits.
- PC: 32 bits
- Word addressable
- Instruction format: Each instruction is 32-bit wide, and consists of five fields: `opcode, rd, rs, rt`, and `unused`. The format is as follows.

`Opcode (4 bits)`	`rd (6 bits)`	`rs (6 bits)`	`rt (6 bits)`	`unused(10 bits)`

The 13 instructions are defined in following table.

Instruction	Symbol	Opcode	rd	rs	rt	Function
No operation	**NOP**	**0000**	**x**	**x**	**x**	**No operation**
Load PC	**LDPC rd, X**	**1111**	**rd**	**x**	**x**	**$rd ← PC+X**
Load	**LD rd, rs**	**1110**	**rd**	**rs**	**x**	**$rd ← M[$rs]**
Store	**ST rt, rs**	**0011**	**x**	**rs**	**rt**	**M[$rs] ← $rt**
Add	**ADD rd, rs, rt**	**0100**	**rd**	**rs**	**rt**	**$rd ← $rs + $rt**
Increment	**INC rd, rs**	**0101**	**rd**	**rs**	**x**	**$rd ← $rs + 1**
Negate	**NEG rd, rs**	**0110**	**rd**	**rs**	**x**	**$rd ← -$rs**
Subtract	**SUB rd, rs, rt**	**0111**	**rd**	**rs**	**rt**	**$rd ← $rs – $rt**
Jump	**J rs**	**1000**	**x**	**rs**	**x**	**PC ← $rs**
Branch if zero	**BRZ rs**	**1001**	**x**	**rs**	**x**	**PC ← $rs, if Z = 1**
Jump memory	**JM rs**	**1010**	**x**	**rs**	**x**	**PC ← M[$rs]**
Branch if negative	**BRN rs**	**1011**	**x**	**rs**	**x**	**PC ← $rs, if N = 1**
SUM	**SUM, rd, rs, rt**	**0001**	**rd**	**rs**	**rt**	**See ***

* $\$rd = \sum_{i=\$rs}^{\$rs+\$rt-1} memory[i]$

Also, use the instructions in the SCU ISA to write two versions of assembly program of the computation of finding the sum of *n* numbers below.

(1) $A = \sum_{i=1}^{n} a_i$

The first version does not use the SUM instruction and the second one uses the SUM instruction.

You can create your new instructions to make the above coding easier. This computation will be used to test your CPU. You may not need five stages in your pipeline.

When you analyze the cycle time, you can use the following data: delay of memory (I and D memory): 2 ns., delay of register file: 1.5 ns., delay of ALU (adders): 2 ns. Ignore the delays of all other components.

Use the ALU attached in the last page.

The last instruction SUM is optional. However, four extra points will be added to both coen 122 and 122L if the SUM instruction is implemented correctly in your CPU.

Submission:

1. Report (30%): Please submit a written report to the TA On the last Friday, including the following

    - abstract (short description or outline of the project),
    - detailed description of the CPU design including the datapath and the truth table of your control.
    - test benchmarks/waveforms verifying the functions,
    - assembly code for calculating the sum in (1).
    - estimate the time need to execute your code for (1) based on your CPU design, and verify your estimate with simulation/waveform.

2. Demo (30%): You will demo your working programs to our TA. During the demo, the TA will provide you *n* random numbers and you will show the TA the result after running your program on your pipeline. You will also be asked to perform random but related tasks (for instance, change the address of data and demo the modified program), and be prepared to answer related project questions. The purpose of these questions is to make sure that you understand the project thoroughly.