Suggested project topic 6: Design a 16-bit SIMD microprocessor

This project is a 16-bit ALU. 2's compliment calculations are implemented in this ALU. The ALU operation will take two clocks. The input to the ALU is a decoded instruction that contain the following fields, the opcode, operation length and the addresses of the registers that stored the operand. For example,

Instruction Add $s1, $s2 $s3 – the inputs to the ALU will be an opcode that indicate it is an add instruction and then 3 addresses to specify where the source and destination of the computation.

Another example : sll $s1, 7 - the inputs to the ALU will be an opcode that indicate it is an logic shift left instruction and then an addresses to specify where the source of the data is and then an operation field that indicate the shift distance is 7.

You can decide the coding of the opcode yourself. You can assume you have a 32 word register-field.

The operation of the ALU takes two cycles. The first clock cycle will be used to read values for th register file into the data registers. The second will be for performing the operations and write back the result to the register files. In your Verilog code modeling, you can use behavioral model to model the register file (i.e., you do not need to synthesis the register file) and you can assume a multi-port register file.

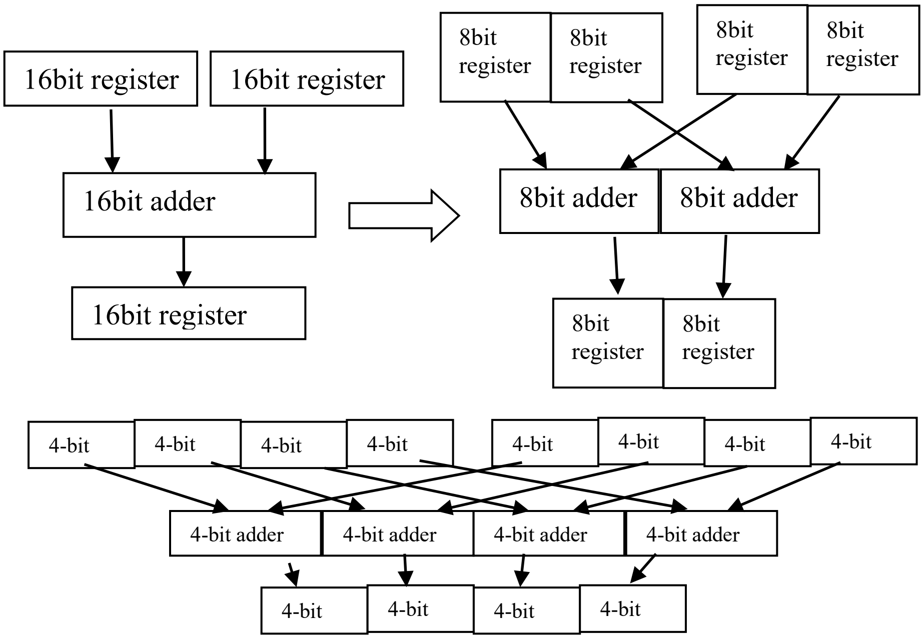
The following operations are carried out in the ALU:

Arithmetic Operations
- 2's compliment addition
- 2's compliment subtraction
- 2's compliment multiplication
- MAC – multiply-and-add operation

Shifting Operations
- Logical shift left, Arithmetic shift left
- Logical shit right

Logical Operations
- Bitwise and
- Bitwise or, xor
- Bitwise not

Now this processor is different from traditional processor. It is a SIMD processor, i.e. it supports single instruction multiple data. It means for every arithmetic computation, it has 3 modes, it can work as a single instruction working on 16bit data, or a single instruction working on 2 set of 8-bit data, or a single instruction working on 4 set of 4 bit data. E.g.



For every instruction, you have 3 versions, one for 16-bit operation, one for 8-bit operation and one for 4-bit operation. You have to design the datapath to support the SIMD operation and assign opcodes for the corresponding operation.