CSC3050 Assignment 3 Report (120090211)

Overview:

This program aims to simulate the ALU. There are 3 inputs with bandwidth 32 in the main module, including instruction, regA (address 00000), regB (address 00001). The output will be the result after calculation inside the ALU and flags (including zero, negative, and overflow) to be used in the later parts of the CPU. The bandwidth of the result will be 32 and that of flags will be 3. The program will first identify the opcode and function part of the given instruction code. Any combination of opcode and function corresponds to some control units. These control units will define several things such as the exact two data to be calculated later or whether the immediate number will be sign-extended or zero-extended, etc. Then, the ALU will calculate and change the flags if needed. All instructions required will appear on the test bench and the waveform will be stored in the test\_alu.vcd. The test bench will show the type name of each instruction, the opcode part, the function part, flags, and the result.

Processing Logic:

To solve the problems, I divide the whole program into 3 parts.

1. Identify the opcode and function and generate the control units. (inside the "control\_units" module)

There are 6 control units generated in this module, including ALU\_SRCA, ALU\_SRCB, EXTSEL\_IM, EXTSEL\_SA, EXTSEL\_SI, and ALU\_OP.

ALU\_SRCA: This control unit will define the first data to be calculated later. 1 denotes the SA part in the instruction, and 0 denotes data stored in Rs.

ALU\_SRCB: This control unit will define the second data to be calculated later. 1 denotes the Immediate part in the instruction, and 0 denotes data stored in Rt.

EXTSEL\_SA: This control unit will define whether the right shift operation will be sign-extended or zero-extended. 1 denotes sign-extended, 0 denotes zero-extended.

(Related instructions are shown in the data flow chart)

EXTSEL\_IM: This control unit will define whether the Immediate part of the instruction will be sign-extended or zero-extended. 1 denotes zero-extended, and 0 denotes sign-extended.

(Related instructions are shown in the data flow chart)

EXTSEL\_SI: This control unit will define whether the ALU will execute with signed numbers or unsigned numbers.

(Related instructions are shown in the data flow chart)

ALUOP: This control unit's bandwidth is 4.

0010 denotes addition

0000 denotes AND operation

0001 denotes OR operation

1001 denotes XOR operation

1100 denotes NOR operation

0110 denotes subtraction

0111 denotes comparison

1010 denotes shift left

1011 denotes shift right

1. Assign two data to be calculated later with the control units and do sign-extension if needed. (Inside the main "alu" module)

For the first data to be calculated later, if ALU\_SRCA is 1, it will be assigned by the zero-extended SA part of the given instruction. (i.e. instruction [10:6]) Otherwise, it will be assigned by the data stored in Rs.

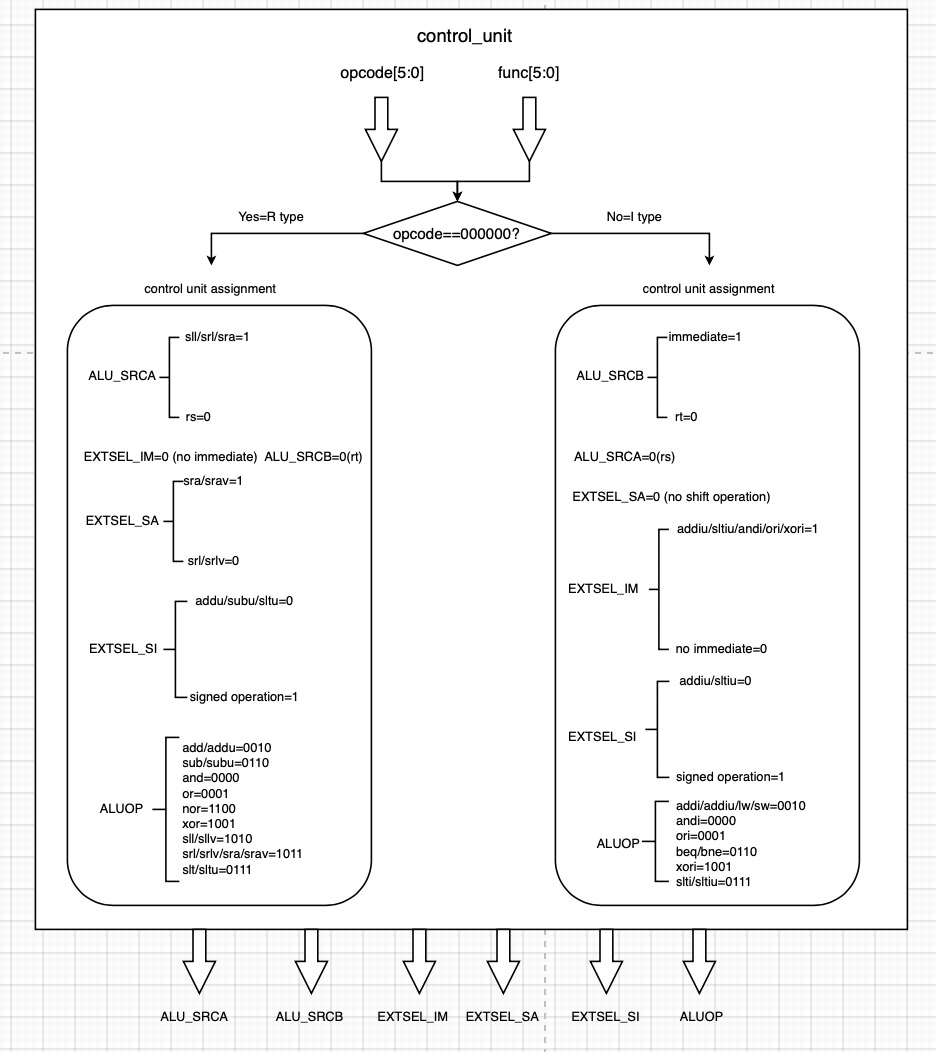
For the second data to be calculated later, if ALU\_SRCB is 1 and EXTSEL\_IM is 1, it will be assigned by the zero-extended Immediate part of the given instruction. (i.e. instruction [15:0]) If ALU\_SRCB is 1 and EXTSEL\_IM is 0, it will be assigned by the sign-extended Immediate part of the given instruction. Otherwise, if ALU\_SRCB is 0, it will be assigned by the data stored in Rt.

1. Define how two data will be calculated and return the result. (Inside the "alu\_calculate" module)

Based on the specific ALUOP, different kinds of operations will be conducted. EXTSEL\_SI will be used to identify operations involving signed numbers. EXTSEL\_SA will be only used in shift right operation to determine arithmetic shift or logical shift. Details can be seen in the data flow charts below. (See on the next page)

Data flow chart:

图示

描述已自动生成

图示

描述已自动生成

Implementation details:

1. Control\_Unit module:

This module is triggered by the opcode, function, input regA and input regB.

If any of them is changed the module will change the output control units immediately, indicating the next instruction now is executed.

1. ALU\_Calculate module:

This module is activated once the two data to be used for calculation are determined by the control units generated by the Control\_Unit module above. It will be triggered by these two data and all control units.

1. Result of beq/bne/slt/slti/sltu/sltiu

The result after calculation will be the difference of two address or value.