Conor McAvoy

ECE 250

Description of Processor

Instructions

All instructions have been tested and appear to function properly.

1-bit adder

The 1-bit adder subcircuit has three 1-bit inputs and two 1-bit outputs. The inputs are “in1”, “in2”, and “cin”. The outputs are “out” and “cout”. The function of the 1-bit adder is to add together in1, in2, and cin, which would typically produce a 2-bit output. This 2-bit output is divided into the least-significant bit which is assigned to out, and the most-significant bit, the carry bit, which is assigned to cout. In the ALU, the 1-bit adder is labeled as “+”.

ALU

The ALU subcircuit has two 16-bit inputs, one 2-bit input, one 16-bit output, and two 1-bit outputs. The 16-bit inputs are “in1” and “in2”, and the 2-bit input is “op”. The 16-bit output is “out”, and the 1-bit outputs are “not zero” and “neg”. The function of the ALU is to perform one of four distinct operations, as specified by op, on in1 and in2 and assign the result to out. The four operations are: add in1 and in2; subtract in2 from in1; invert in1; and xor in1 and in2 bitwise. These operations correspond to op values of 00, 01, 10, and 11, respectively. Additionally, the ALU outputs two flags: not zero, which is true when the output is not zero, and neg, which is true when the output is negative. These are used for branching instructions. In the processor, the ALU is labeled as “ALU”.

16-bit register

The 16-bit register subcircuit has one 16-bit input, three 1-bit inputs, and one 16-bit output. The 16-bit input is “D”, and the 1-bit inputs are “clk”, “en”, and “reset”. The output is “Q”. The function of the 16-bit register file is to store D when en is true (“enabled”) and the clk changes from true to false (“falling edge”). When a value is stored, the register constantly outputs it as Q, until a different value is stored, or reset is set to true, at which point the stored value reverts to all zeros. In the register file and the processor, the 16-bit register is labeled as “reg”.

Register file

The register file subcircuit has one 16-bit input, three 3-bit inputs, three 1-bit inputs, and two 16-bit outputs. The 16-bit input is “Rd val”, the 3-bit inputs are “Rd”, “Rs”, and “Rt”, and the 1-bit inputs are “clk”, “en”, and “reset”. The outputs are “Rs val” and “Rt val”. The function of the register file is to store eight 16-bit registers. A new value can be stored by specifying the destination register with Rd and the value with Rd val. The value will only be stored if en is true and the clk hits a falling edge. No value can be written to the first register which is permanently zero. Values can be read by specifying which register to read with either Rs or Rt. The corresponding value will appear in Rs val or Rt val, respectively. Setting reset to true will set all registers to zero. The register file operates with tri-state buffers rather than multiplexers for speed. In the processor, the register file is labeled as “reg file”.

Sign extend

The sign extend subcircuit has one 6-bit input and one 16-bit output. The input is “in” and the output is “out”. The function of the sign extend is to convert a 6-bit two’s complement number into the equivalent 16-bit two’s complement number for use in the ALU. This is as easy

as repeating the most-significant bit ten more times. The 6-bit immediate fields of immediate instructions are the only data that will pass through this subcircuit. In the processor, the sign extend is labeled as “sign ext”.

Shift left/right

The shift left/right subcircuits have one 16-bit input, one 3-bit input, and one 16-bit output. The 16-bit input is “in”, the 3-bit input is “shamt”, and the output is “out”. The function of these subcircuits is to shift in by shamt bits either left or right and assign that value to out. These both shift logically, that is, shift right does not preserve sign. In the shifter, these subcircuits are labeled as “sll” and “srl”, respectively.

Shifter

The shifter subcircuit has one 16-bit input, one 3-bit input, one 1-bit input, and one 16-bit output. The 16-bit input is “in”, the 3-bit input is “shamt”, and the 1-bit input is “l or r”. The output is “out”. The function of the shifter is to shift in by shamt bits either left or right, as specified by l or r, which is false for a left shift and true for a right shift. The shifted value is assigned to out. In the processor, the shifter is labeled as “shift”.

Control

The control subcircuit has one 16-bit input called “instruction” and the following outputs: “addr”, 12-bit; “immed”, 6-bit; “shamt”, 3-bit; “Rd”, 3-bit; “Rt”, 3-bit; “Rs”, 3-bit; “Rd is rt”, “write en”, “ALU immed”, “shift”, “shift right”, “load”, “store”, “bne”, “blt”, “input”, and “output”, all 1-bit; and “ALU op” and “jump”, both 2-bit. First, the circuit takes instruction and splits it up into its various fields. These are output directly, except for the opcode, which is used to determine the control flags (all the 1- and 2-bit outputs). Their meanings are detailed below.

Control flag	Meaning
Rd is rt	When true, indicates that the destination register is specified by rt, not rd. This only happens for I-type instructions, though it does not happen for all of them.
write en	When true, indicates that values are to be stored in the register file. This flag is false only for stores, branches, jumps (except jal), and output.
ALU immed	When true, indicates that the immediate field is to be used instead of rt as an input to the ALU. This is true for many I-type instructions.
shift	Indicates a shift is to be performed, whether left or right, by shamt bits.
shift right	When true, this causes the shift to be rightwards. Value does not matter if not shifting.
load	Indicates a lw instruction.
store	Indicates a sw instruction.
bne	Indicates a bne instruction.
blt	Indicates a blt instruction.
input	Indicates an input instruction.
output	Indicates an output instruction.
ALU op	Specifies the function of the ALU. 00 for addition, 01 for subtraction, 10 for not, and 11 for xor.
jump	Specifies types of jumps. 00 for no jump, 01 for j, 10 for jr, and 11 for jal.