Build a 4-input Multiplexor

Complete the code for a 4-input multiplexor below. You can assume you are given a 2 input mux.

```verilog
// output = A (when control == 00) or B (when control == 01) or
//        C (when control == 10) or D (when control == 11)
module mux4(out, A, B, C, D, control); // a 4-input multiplexor
  output out;
  input A, B, C, D;
  input [1:0] control; // <---- multiple bit signal (2-bits)

  // use control[0] and control[1] as inputs to multiplexors
endmodule // mux4
```

Draw the logic circuit for mux implemented above
Building blocks

Complete the code for a 1 bit logic unit below.

```verilog
// operation = AND (when control == 00) or OR (when control == 01) or
//     NOR (when control == 10) or XOR (when control == 11)
module logic_unit(out, control, A, B);
   // a 1-bit half adder
   output out;
   input [1:0] control;
   input A, B;
endmodule
```

Write the boolean expression for the zero, negative and overflow outputs in a 32 bit ALU assuming the inputs are 32 bit wide A and B and the output is 32 bit wide out.
Glue the blocks

Complete the code for a 1 bit ALU. Draw it in the empty space to make it easier to write the code.

```verbatim
// operation = + (when control == 010) or - (when control == 011) or
//        AND (when control == 100) or OR (when control == 101)
//        NOR (when control == 110) or XOR (when control == 111)
module alu1(out, carryout, A, B, carryin, control);
output out, carryout;
input A, B, carryin;
input [2:0] control;
endmodule // alu1
```