1. Multiplexer

The following code contains errors. Find and fix all errors so that this circuit behaves as a 2-bit adder. Not all errors have the same grading weight!

```
// Full adder block
// Dataflow level
module full_adder (a1, a2, ci, s, co);  //lists full input/output
signal list
input  a1, a2, ci;       //input declaration
output sum, co;            //output declaration
assign sum = a1 ^ a2 ^ ci;
assign co <= (a1 & a2) | (a1 & ci) | (a2 & ci);

// 2-bit adder using hierarchical logic
// A 1-bit full adder is instantiated twice
module top (a1,a2,sum,c_in, c_out);
input [1:0]   a1;
input [1:0]   b1;
input         c_in;
output [1:1]  sum;
output        c_out , c_in;
//Intermediate wires for carry for block to block
wire c_out_0;  // 1-bit wire
fulladd2 fa0 (  a[0], b[0] , c_in, sum[0], c_out_0)
full_adder fa1 (.a1(a1[1]),.a2(b1[1]),c_in(c_out_0),
    .s(sum[1]),.c_out(c_out_0) );
endmodule
```

2. Accumulator

Write a behavioral Verilog module which implements the following function.  
A given module has three inputs:
input
reset
clk
Output:
sum

Register:
The accumulator module has an internal 16-bit register.

Behavior:

On a positive edge of clk it adds input, a 16-bit integer, to the register; on the negative edge of clk it places the new sum on its outputs (until the next negative edge).  When reset is high the register is set to zero and the output changes immediately.