Review Problem 9

- What does the number 100011_2 represent?

2's comp represents -29

Unsigned 35

S-M -3

MIPS sub x0, x0, x0

machine code

ASCII "#"

Debugging Complex Circuits

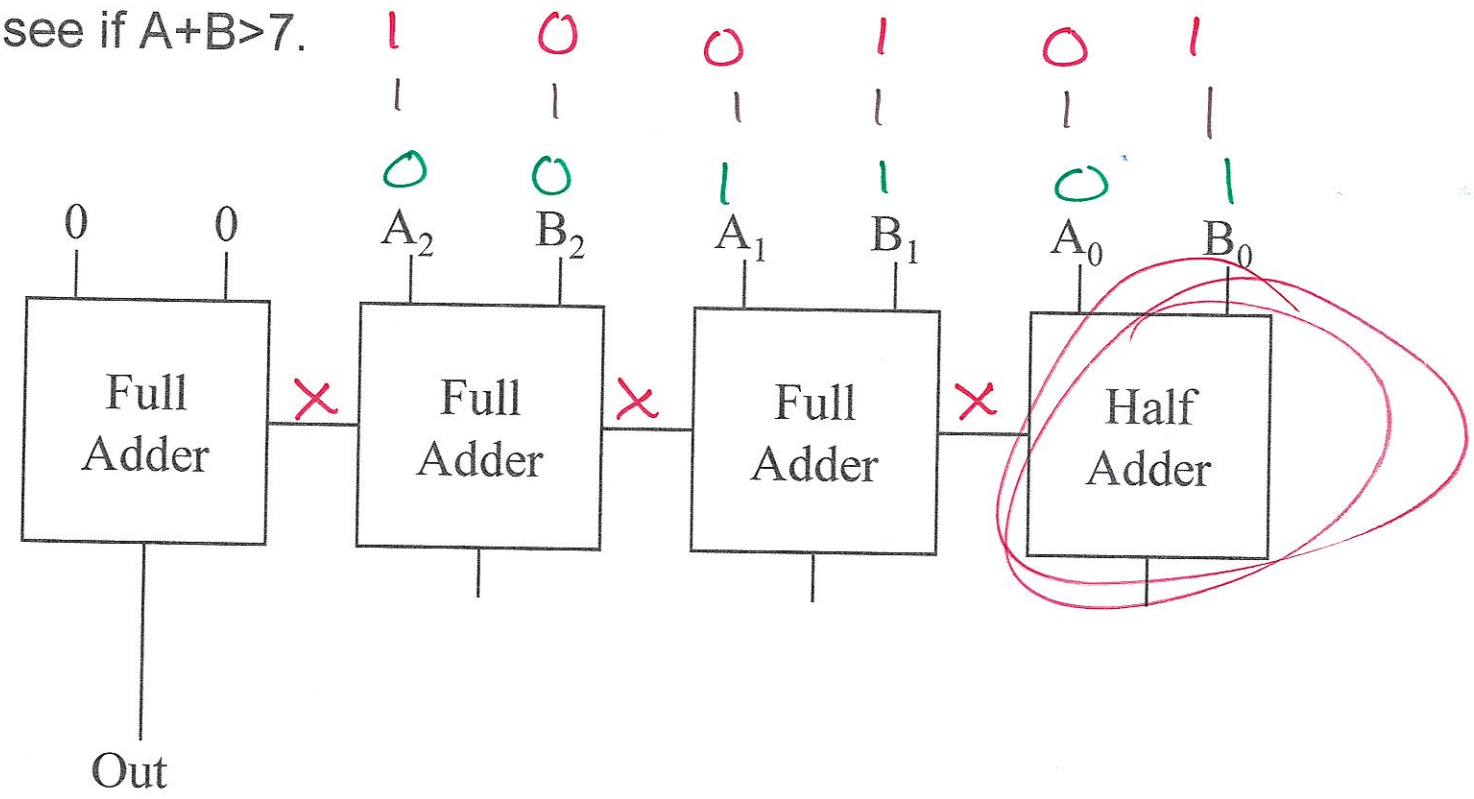
Small : try all combos
big : try all behaviors

Complex circuits require careful debugging

Rip up and retry?

try all combos of sub elements

Ex. Circuit to see if A+B>7.



0 1 X

Debugging Complex Circuits (cont.)

```
module fullAdd (Cout, S, A, B, Cin);
  output Cout, S;  input A, B, Cin;

  assign Cout = (A&B) | (A&Cin) | (B&Cin);
  assign S = A^B^Cin;
endmodule

module halfAdd (Cout, S, A, B);
  output Cout, S;  input A, B;

  fullAdd a1(.Cout, .S, .A, .B, .Cin);
endmodule

module greaterThan7 (Out, A, B);
  output Out;  input [2:0] A, B;  wire [3:0] C, S;

  halfAdd pos0(.Cout(C[0]), .S(S[0]), .A(A[0]), .B(B[0]));
  fullAdd pos1(.Cout(C[1]), .S(S[1]), .A(A[1]), .B(B[1]), .C(C[0]));
  fullAdd pos2(.Cout(C[2]), .S(S[2]), .A(A[2]), .B(B[2]), .C(C[1]));
  fullAdd pos3(.Cout(C[3]), .S(Out), .A(0), .B(0), .C(C[2]));
endmodule
```

Debugging Approach

Test all behaviors.

All combinations of inputs for small circuits, subcircuits.

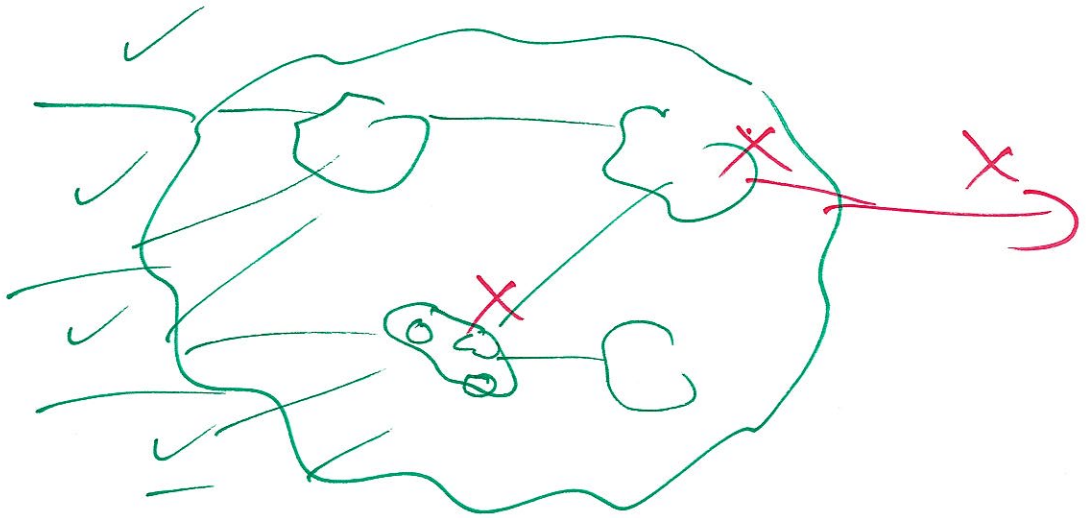
Identify any incorrect behaviors.

Examine inputs and outputs to find earliest place where value is wrong.

Typically, trace backwards from bad outputs, forward from inputs.

Look at values at intermediate points in circuit.

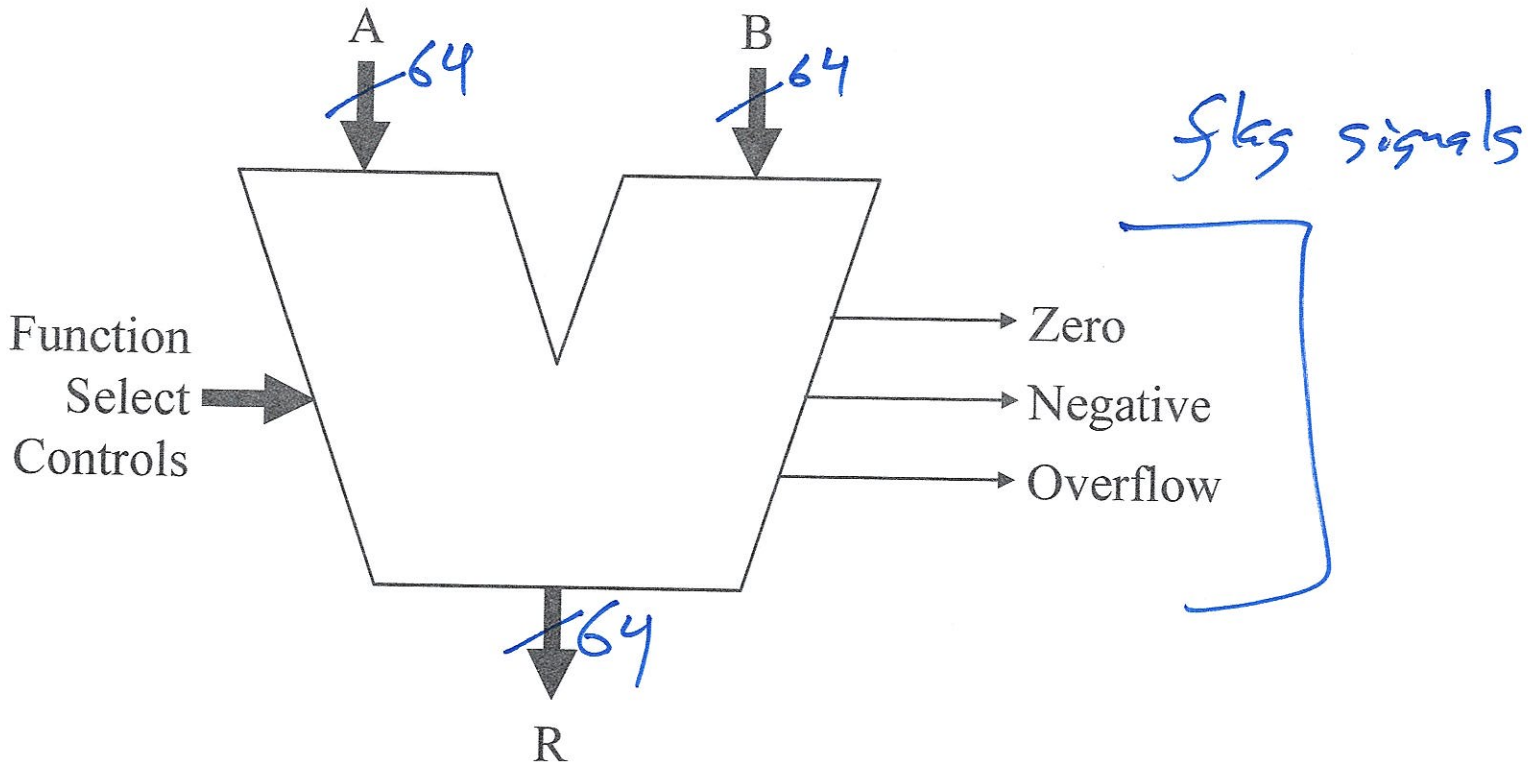
DO NOT RIP UP, DEBUG!



ALU: Arithmetic Logic Unit

Computes arithmetic & logic functions based on controls

- Add, subtract
- XOR, AND, NAND, OR, NOR
- ==, <, overflow, …

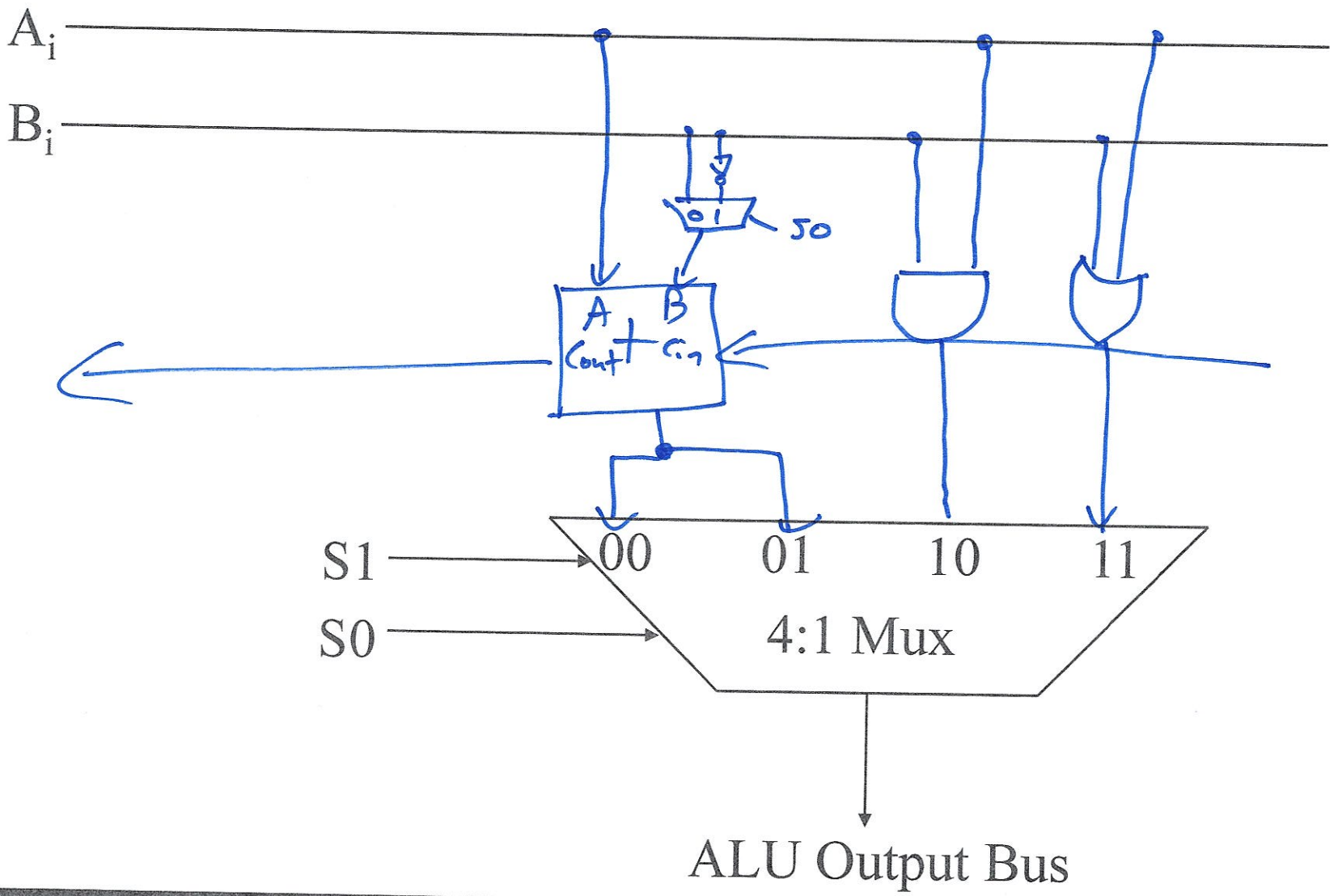


i+h bit

Bit Slice ALU Design

00 01 10 11

Add, Subtract, AND, OR



Bit Slice ALU Design (cont.)

Route Carries

Overflow, zero, negative