

```
1  module ula_8b (
2      input [7:0] a, b,
3      input x, y,
4      output reg [7:0] s,
5      output reg c, ov, z, n
6  );
7
8      wire [8:0] soma_ext;    // Para detectar carry
9      assign soma_ext = a + b;
10
11     always @(a, b, x, y) begin
12         // Operação principal
13         case ({x, y})
14             2'b00: s = a + b;
15             2'b01: s = a & b;
16             2'b10: s = a | b;
17             2'b11: s = ~a;
18         endcase
19
20         // Flags
21         if ({x, y} == 2'b00) begin // Caso {soma}
22             c = soma_ext[8];
23             ov = (~a[7] & ~b[7] & s[7]) | (a[7] & b[7] & ~s[7]); // Carry out do bit 8 // Overflow detectado
24         end else begin
25             c = 0;
26             ov = 0;
27         end
28
29         z = (s == 8'b0); // Zero
30         n = s[7]; // Negativo (bit mais significativo)
31     end
32
33 endmodule
```