

```
module registro_salida_incondicional(  
    input wire clk,  
    input wire write,  
    input wire [7:0] dato_entrada,  
    output wire [7:0] dato_salida  
);  
  
    reg [7:0] valor;  
  
    always @( posedge clk )  
        if ( write )  
            valor <= dato_entrada;  
  
    assign dato_salida = valor;  
  
endmodule
```

```
module registro_salida_triestado(  
    input wire clk,  
    input wire write, read,  
    input wire [7:0] dato_entrada,  
    output wire [7:0] dato_salida  
);  
  
    reg [7:0] valor;  
  
    always @( posedge clk )  
        if ( write )  
            valor <= dato_entrada;  
  
    assign dato_salida = read ? valor : 'bZ;  
  
endmodule
```

```
module registro_bus_bidireccional(  
    input wire clk,  
    input wire write, read,  
    inout wire [7:0] dato  
);  
  
    reg [7:0] valor;  
  
    always @( posedge clk )  
        if ( write )  
            valor <= dato;  
  
    assign dato = read ? valor : 'bZ;  
  
endmodule
```

```

module ram_sincrona_bus_datos_bidireccional(
    input wire clk,
    input wire write, read,
    inout wire [7:0] dato,
    input wire [9:0] direccion
);

    reg [7:0] valor [1023:0];

    always @( posedge clk )
        if ( write )
            valor[direccion] <= dato;

    assign dato = read ? valor[direccion] : 'bZ;

endmodule

// 'bZ en este contexto equivale a 8'bZZZZZZZZ,
// que son 8 bits a HI (alta impedancia)

```