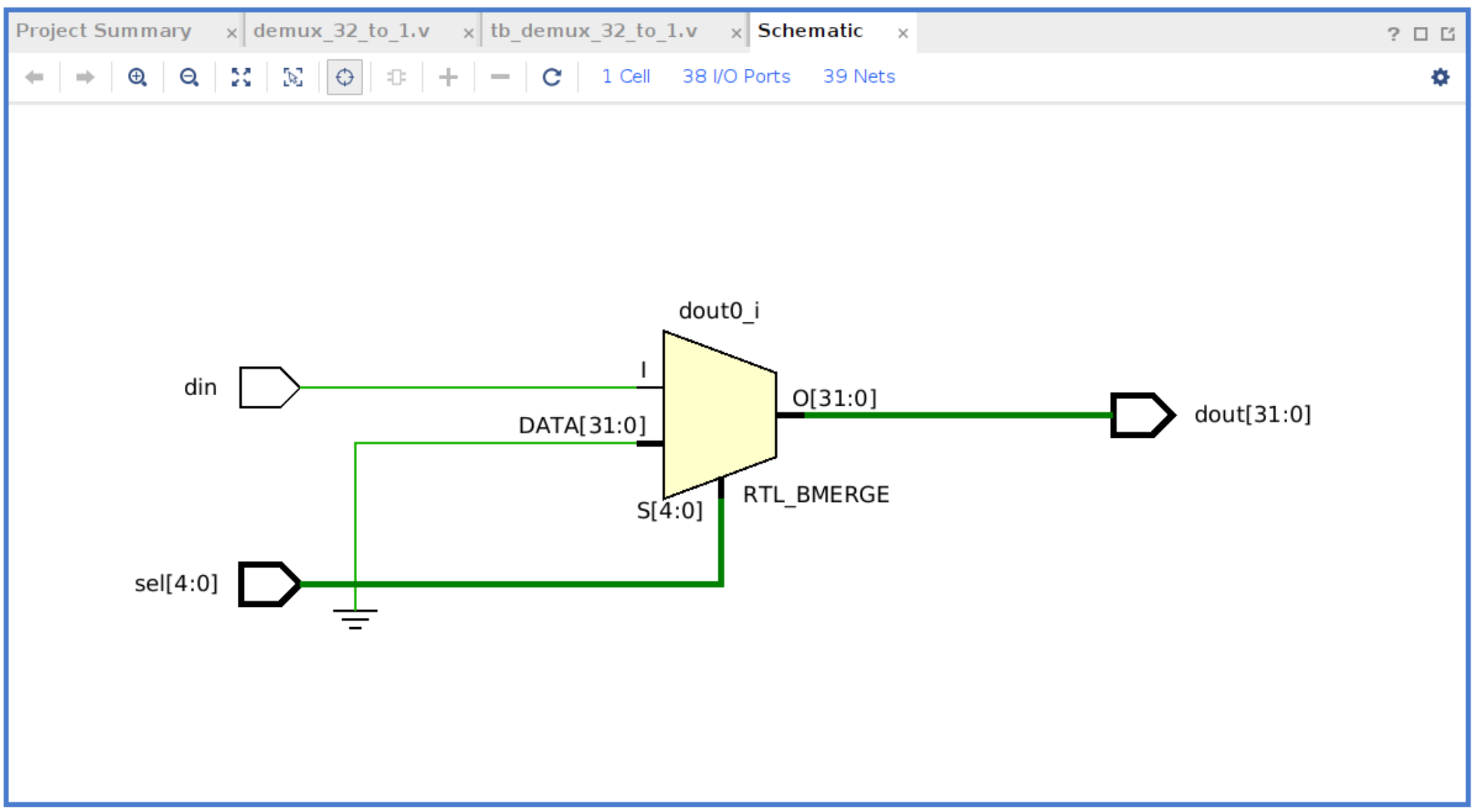


The figure is a timing diagram for a 32-bit shift register. The horizontal axis represents time in nanoseconds (ns), ranging from 0 to 300,000 ns with major ticks every 20,000 ns. The vertical axis lists the signals: sel[4:0], dout[31:0], and din. The signal sel[4:0] is a 5-bit input that changes at various points in time. The signal din is a 1-bit input that is high for the first 10,000 ns and then low. The signal dout[31:0] is the 32-bit output, which is initially 00000001 and then shifts right as sel[4:0] changes. A yellow vertical line is drawn at 5.575 ns. The output dout[31:0] is shown as a staircase pattern, indicating a shift register operation. The output is 0 for most of the time, except for the first 10,000 ns where it is 00000001. The output is 0 for most of the time, except for the first 10,000 ns where it is 00000001. The output is 0 for most of the time, except for the first 10,000 ns where it is 00000001.



/home/itzzinfinity/Cozy Drive/100daysofRTL/day\_030/project\_1/project\_1.srscs/sources\_1/new/demux\_32\_to\_1.v



```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////
3  // Engineer: Anjan Prasad
4  // Create Date: 10/21/2024 06:13:39 AM
5  // Module Name: demux_32_to_1
6  //////////////////////////////////////
7
8
9  module demux_32_to_1(
10     input wire [4:0] sel,
11     input wire din,
12     output reg [31:0] dout
13 );
14     always @(*) begin
15         dout = 32'b0;
16         dout[sel] = din;
17     end
18 endmodule
19
```

/home/itzzinfinity/Cozy Drive/100daysofRTL/day\_030/project\_1/project\_1.srscs/sim\_1/new/tb\_demux\_32\_to\_1.v



```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////
3  // Engineer: Anjan prasd
4  // Create Date: 10/21/2024 06:16:36 AM
5  // Module Name: tb_demux_32_to_1
6  //////////////////////////////////////
7
8  module tb_demux_32_to_1;
9
10     reg [4:0] sel;
11     reg din;
12     wire [31:0] dout;
13
14     demux_32_to_1 uut (
15         .sel(sel),
16         .din(din),
17         .dout(dout)
18     );
19     initial begin
20         din = 1;
21         for (sel = 5'd0; sel <= 5'd31; sel = sel + 1) begin
22             #10;
23         end
24         $stop;
25     end
26
27 endmodule
28
```