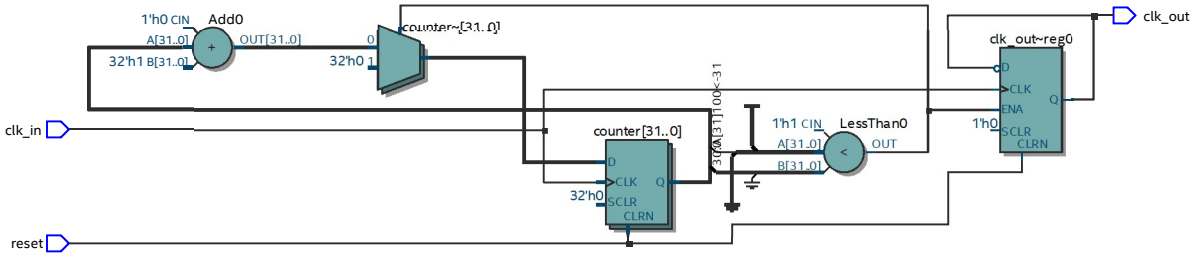
```
1   module clockDivider #(
2       parameter INPUT_FREQ = 50000000,
3       parameter OUTPUT_FREQ = 1
4   )(
5       input wire clk_in,
6       input wire reset,
7       output reg clk_out
8   );
9
10      localparam integer halfPeriod = INPUT_FREQ / (2 * OUTPUT_FREQ);
11
12      integer counter;
13
14      always @(posedge clk_in or posedge reset) begin
15          if (reset) begin
16              counter <= 0;
17              clk_out <= 0;
18          end else begin
19              counter <= counter + 1;
20              if (counter >= halfPeriod) begin
21                  clk_out <= ~clk_out;
22                  counter <= 0;
23              end
24          end
25      end
26
27  endmodule
```


1'h0 CIN
Add0
A[31..0]
OUT[31..0]
32'h1 B[31..0]
0
32'h0 1
counter~[31..0]
clk_in
reset
counter[31..0]
D
CLK
Q
SCLR
CLRN
32'h0
1'h1 CIN
LessThan0
A[31..0]
B[31..0]
OUT
clk_out~reg0
D
CLK
ENA
Q
1'h0
SCLR
CLRN
clk_out