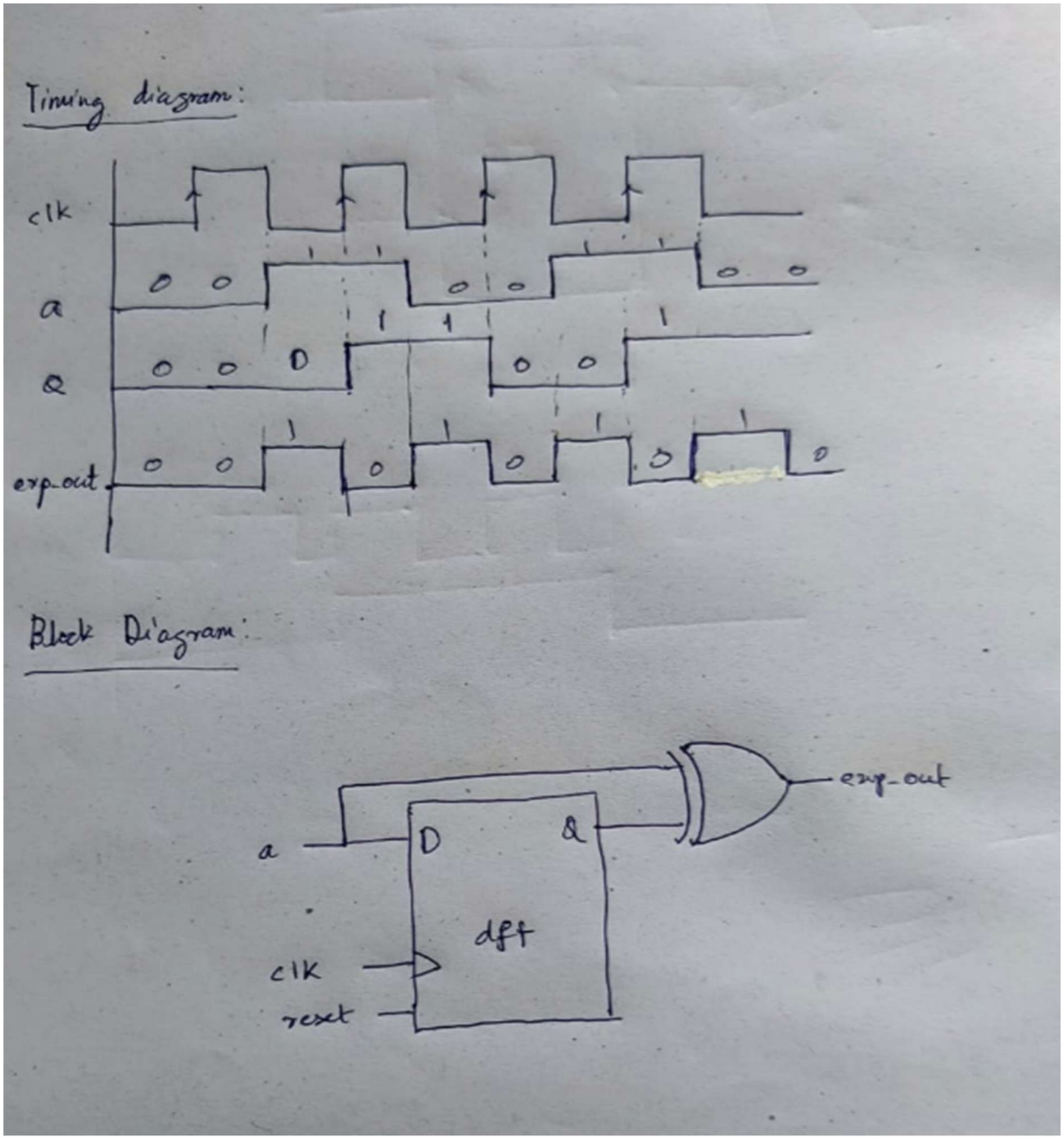
Both positive and negative edge Detector

Here input a is synchronous with clk.

Block diagram:



Truth table for the exp_out – XOR gate:

a	Q	exp_out
0	0	0
0	1	1
1	0	1
1	1	0

Verilog code:

```
module both_edgeDetector(
      input clk,reset,a,
      output exp_out,
      output wire x1
      );

      dff d1(clk,reset,a,x1);
      xor(exp_out,a,x1);

  endmodule

  module dff(input clk,reset,D,output reg Q);
  always @(posedge clk) begin
  if (reset)
     Q <=1'b0;
  else
     Q<=D;
  end
  endmodule
```

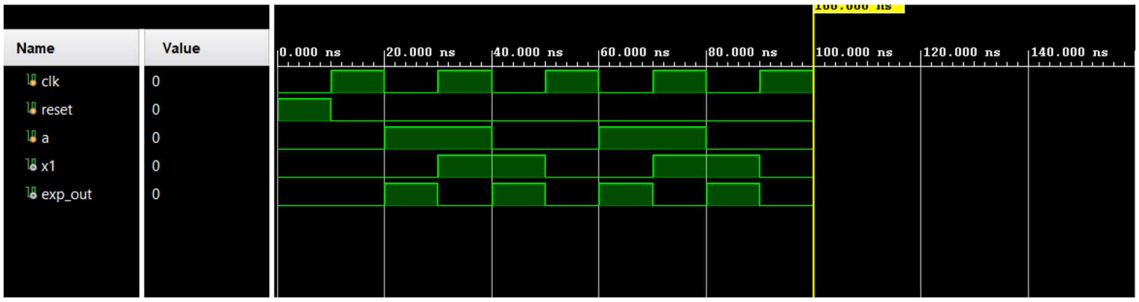
Testbench code:

```
module both_edgeDetector_tb;
reg clk;
reg reset;
reg a;
wire x1;
wire exp_out;
```

```
both_edgeDetector uut(.clk(clk),.reset(reset),.a(a),.exp_out(exp_out),.x1(x1));


initial begin
   clk = 0;
   forever #10 clk = ~clk;
end


initial begin
   reset = 1; a = 0;
   #10;
   reset = 0;
   #10; a = 1;
   #10; a = 1;
   #10; a = 0;
   #10; a = 0;
   #10; a = 1;
   #10; a = 1;
   #10; a = 0;
   #10; a = 0;
   #10;
   $finish;
end


initial begin
   $monitor("time = %0t | clk = %b | reset = %b | a = %b | exp_op = %b | Q = %b",$time, clk,
reset, a, exp_out,x1);
end
```

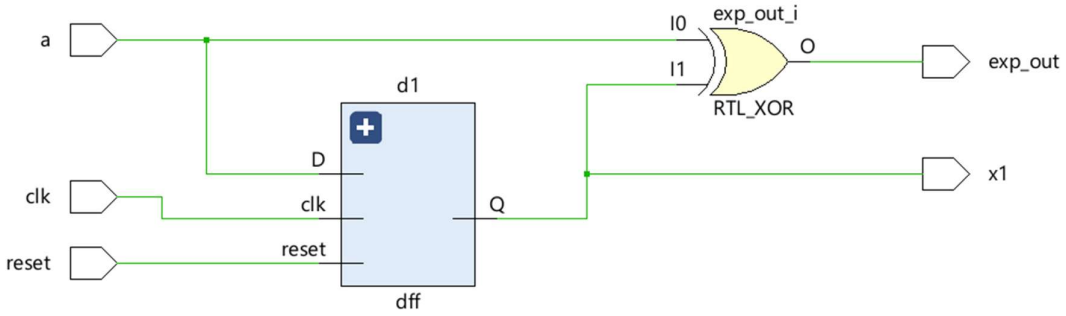
endmodule

Timing diagram:

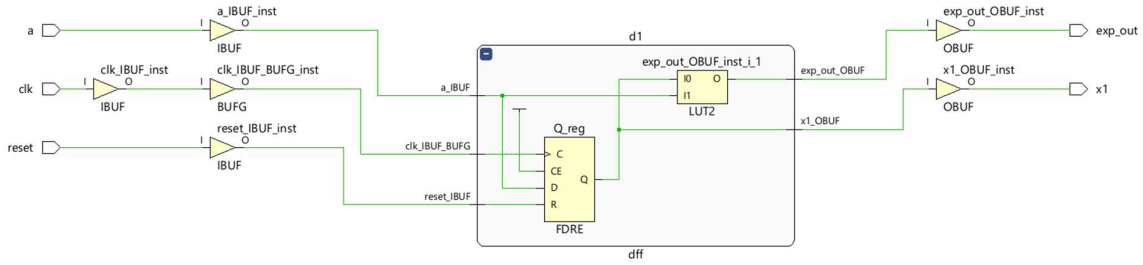


Schematics:

1. **RTL schematic:**



2. **Synthesized schematic:**



LUT2:

exp_out_OBUF_inst_i_1

I1	I0	O=I0 & !I1 + !I0 & I1
0	0	0
0	1	1
1	0	1
1	1	0