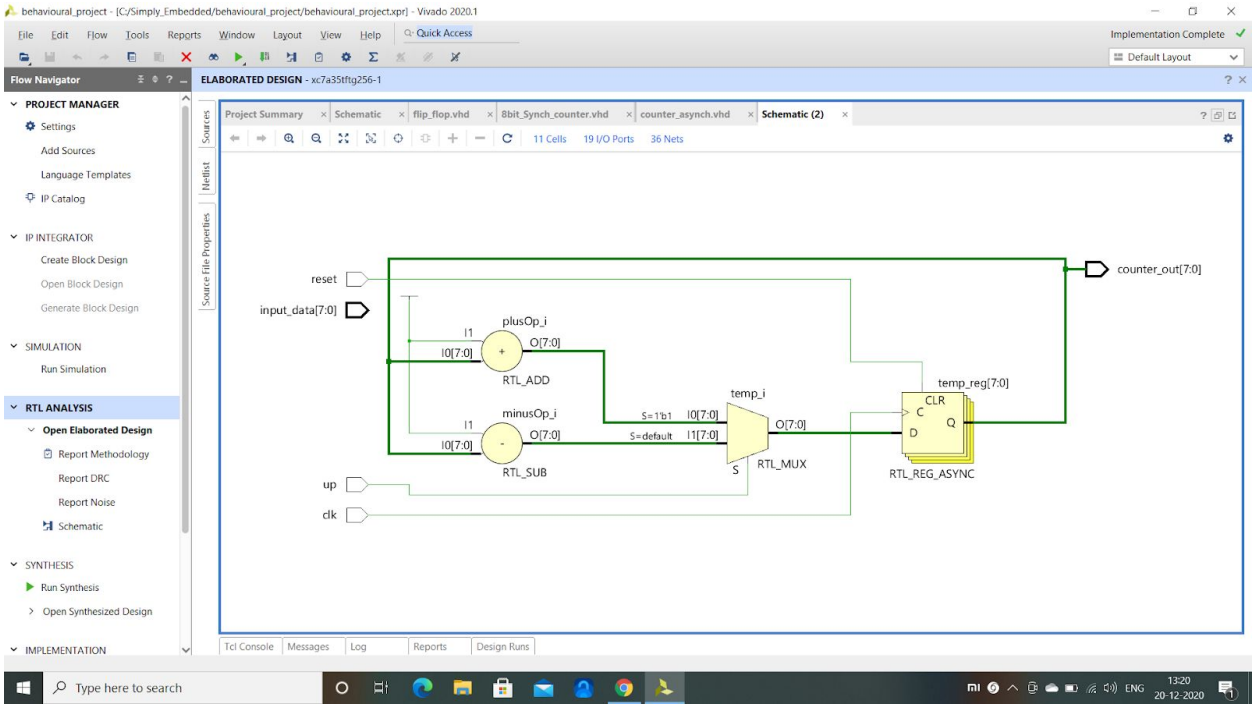
COUNTER AND LATCH
ASYNCHRONOUS

Code :

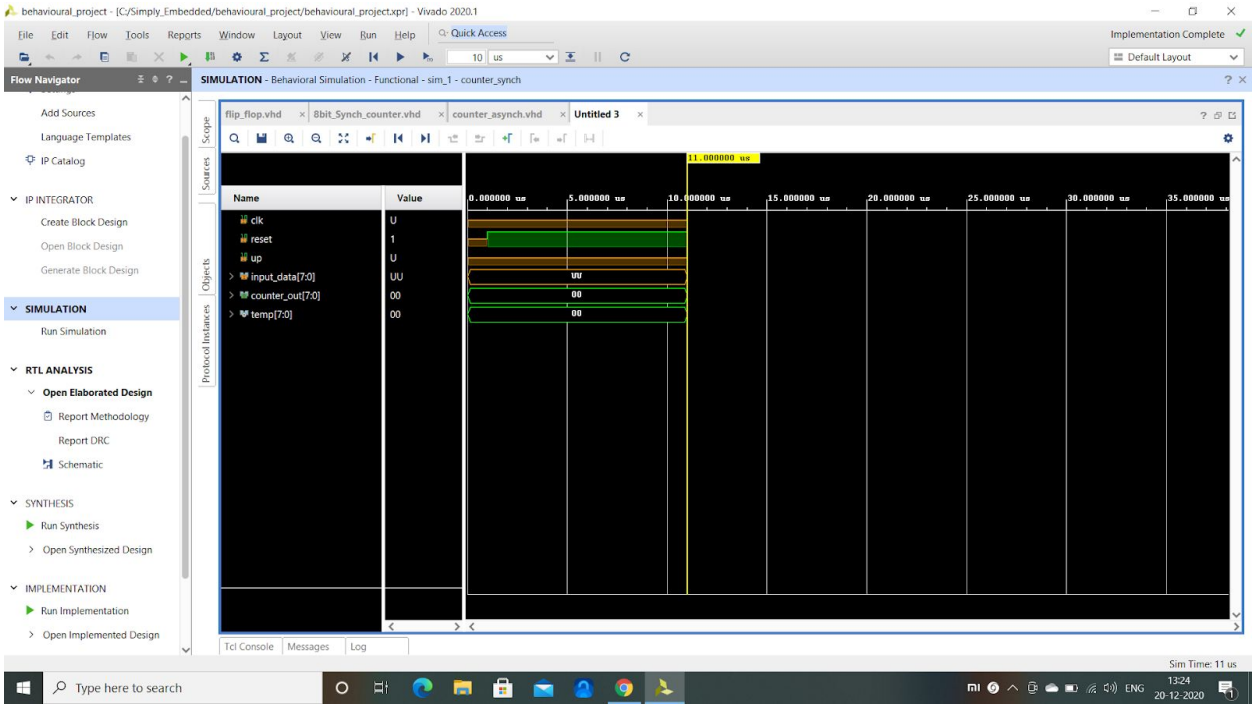
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.std_logic_unsigned.all;
use ieee.numeric_std.all;
entity counter_synch is
Port (
clk,reset,up : in std_logic;
input_data: in std_logic_vector (7 downto 0);
counter_out  : out std_logic_vector (7 downto 0)
 );
end counter_synch;

architecture Behavioral of counter_synch is
signal temp : std_logic_vector (7 downto 0) := X"00";
begin
process(clk)
begin
  if(reset = '1') then
      temp <= X"00";
   elsif(rising_edge(clk)) then
      if( up = '1') then
      temp <= temp +1;
      else
       temp <= temp -1;
       end if;
    end if;
end process;
counter_out <= temp;
end Behavioral;
```

RTL schematic:

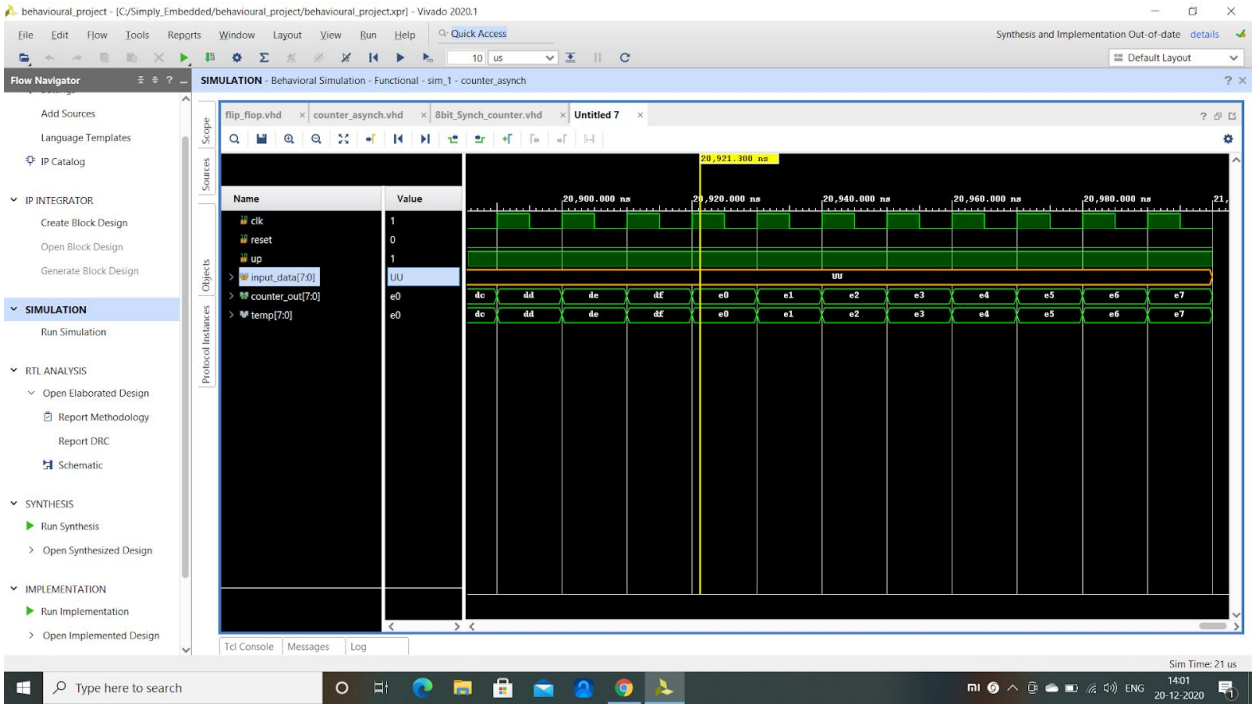


Waveform:

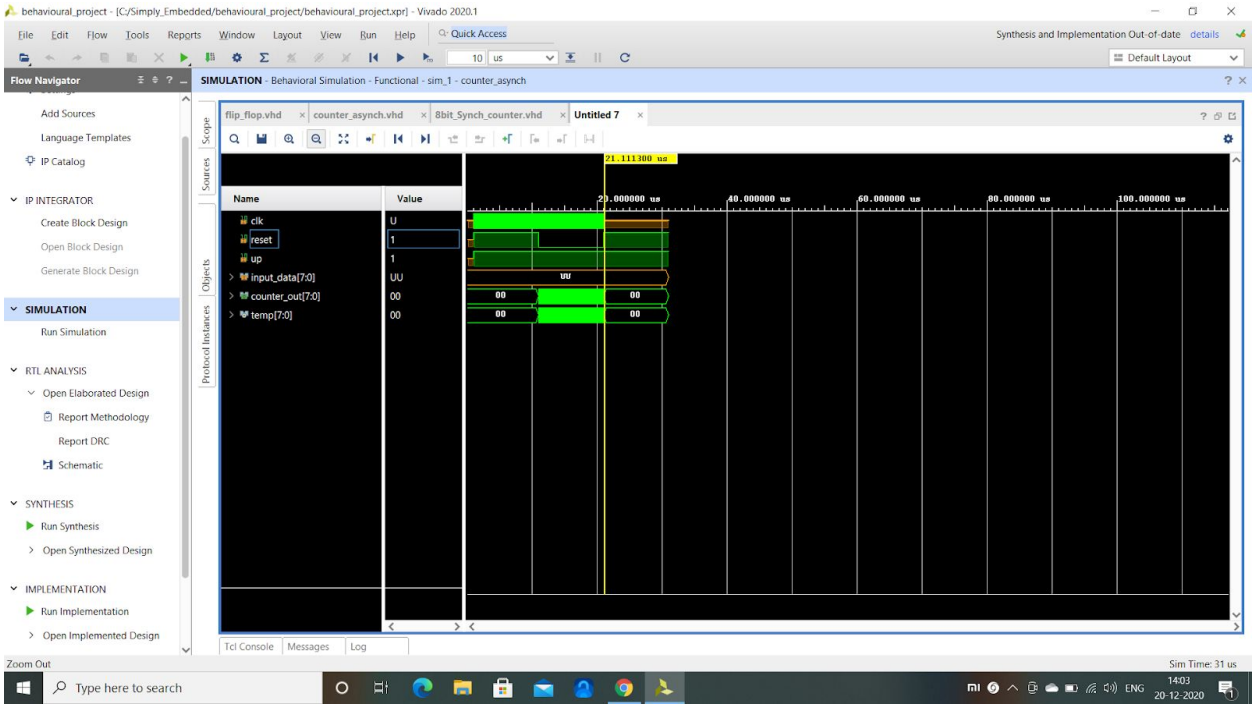
Reset = ‘1’ and clock = ‘0’;



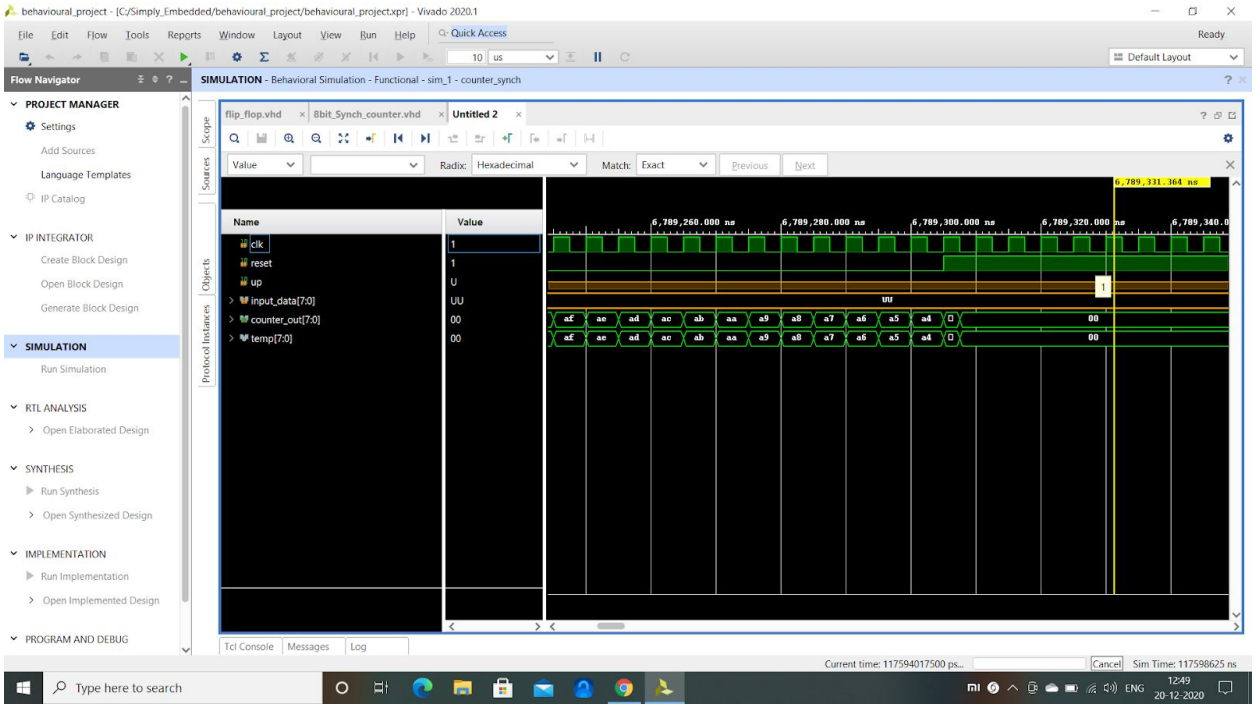
Reset = 0



Reset = 1 clock =0;



Reset = ‘1’;



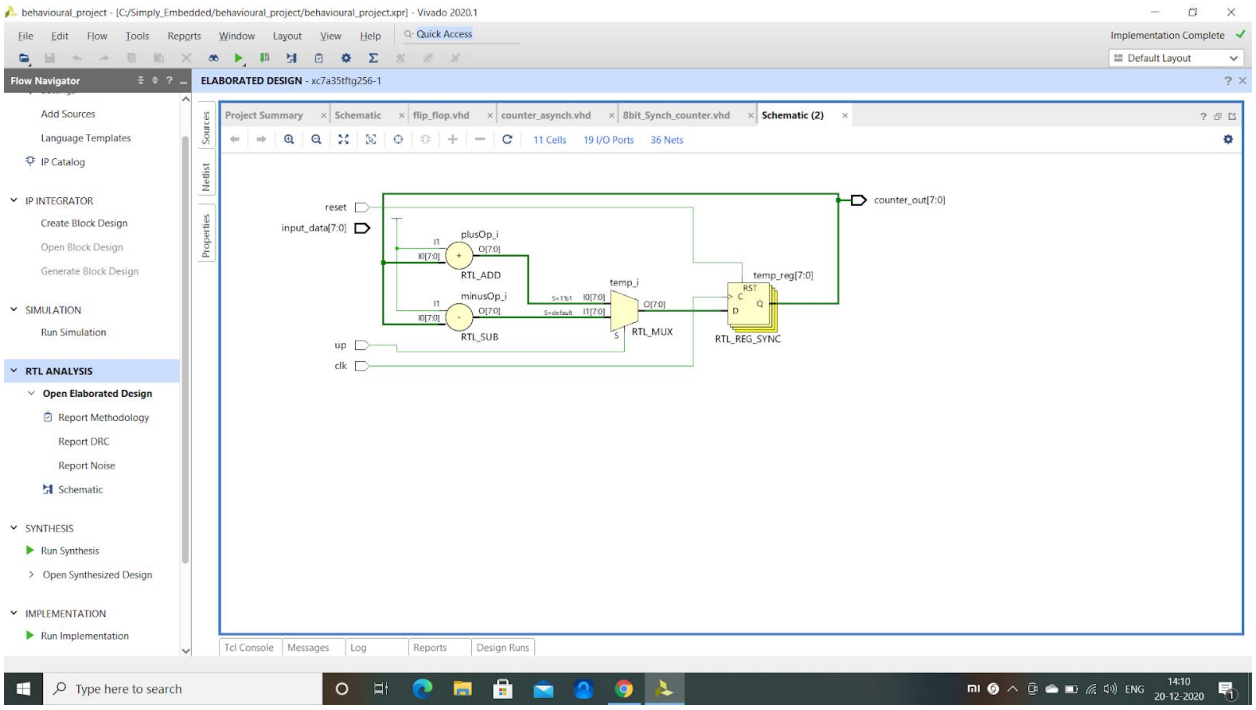
SYNCHRONOUS

code :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.std_logic_unsigned.all;
use ieee.numeric_std.all;
entity counter_synch is
Port (
clk,reset,up : in std_logic;
input_data: in std_logic_vector (7 downto 0);
counter_out  : out std_logic_vector (7 downto 0)
  );
end counter_synch;

architecture Behavioral of counter_asynch is
signal temp : std_logic_vector (7 downto 0) := X"00";
begin
process(clk)
begin
```

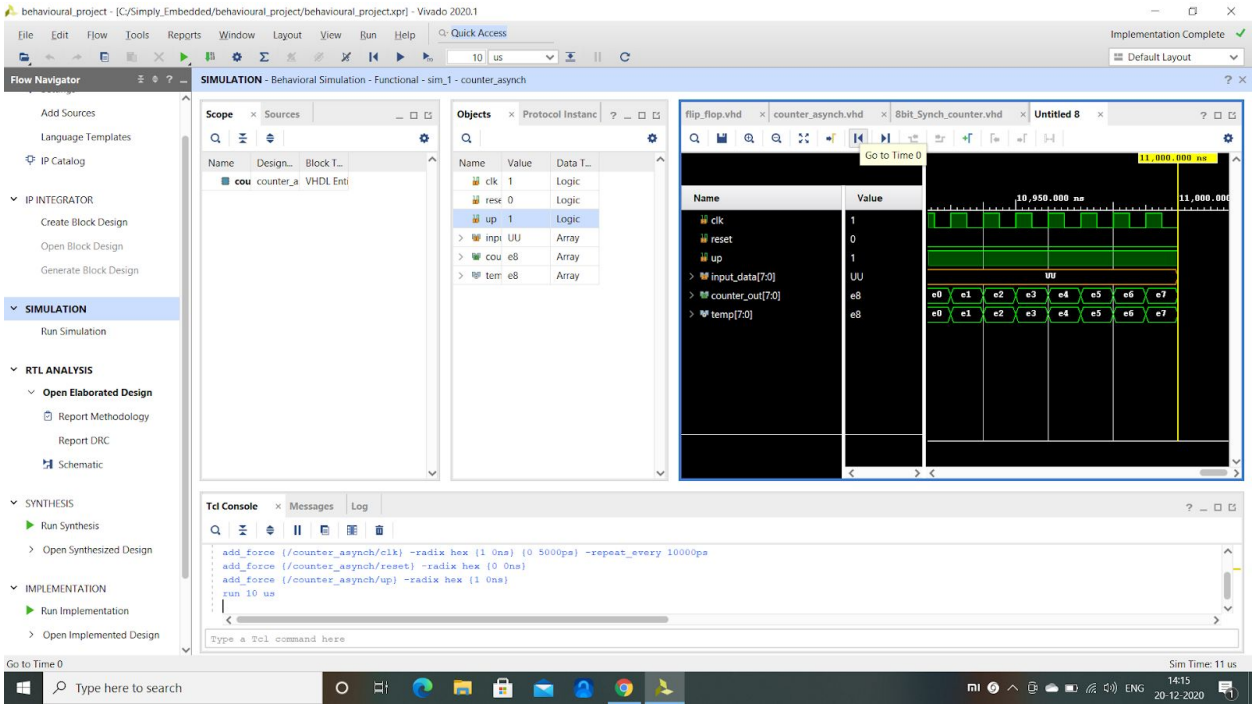
```
  if(rising_edge(clk)) then
     if(reset = '1') then
     temp <= X"00";
     elsif( up = '1') then
     temp <= temp +1;
     else
      temp <= temp -1;
      end if;
   end if;
end process;
counter_out <= temp;
end Behavioral;
```

RTL schematic:

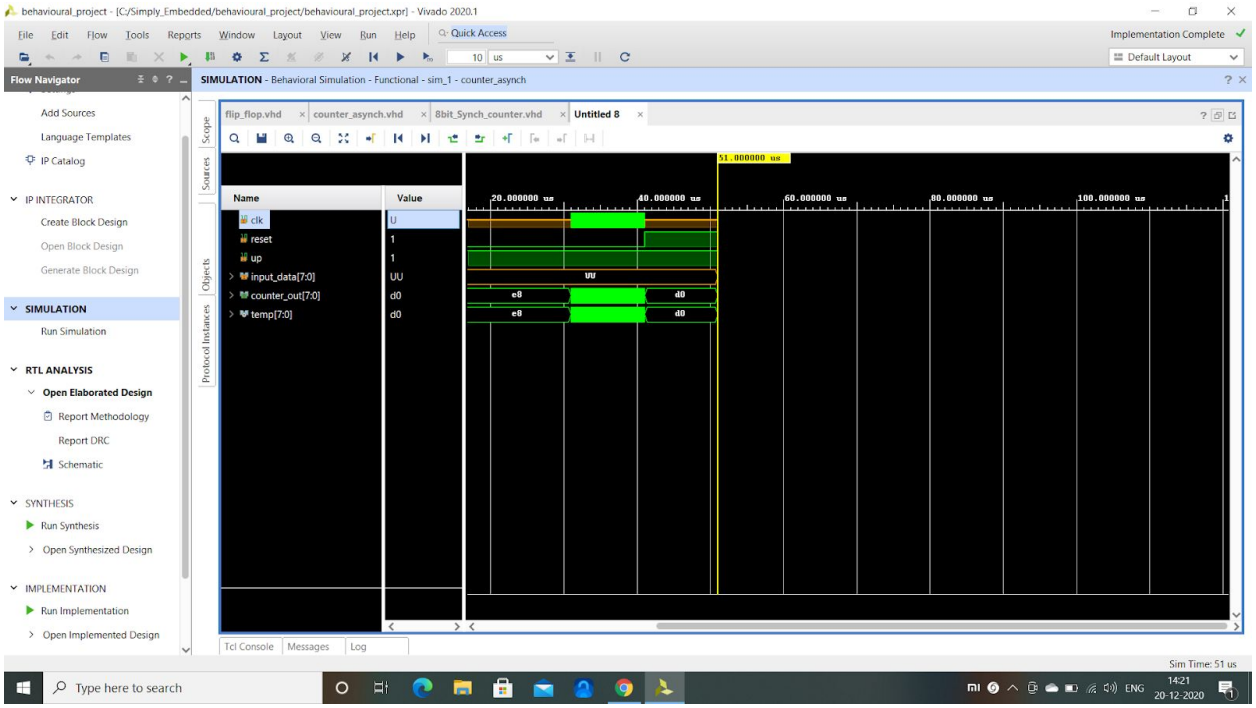


Waveform:

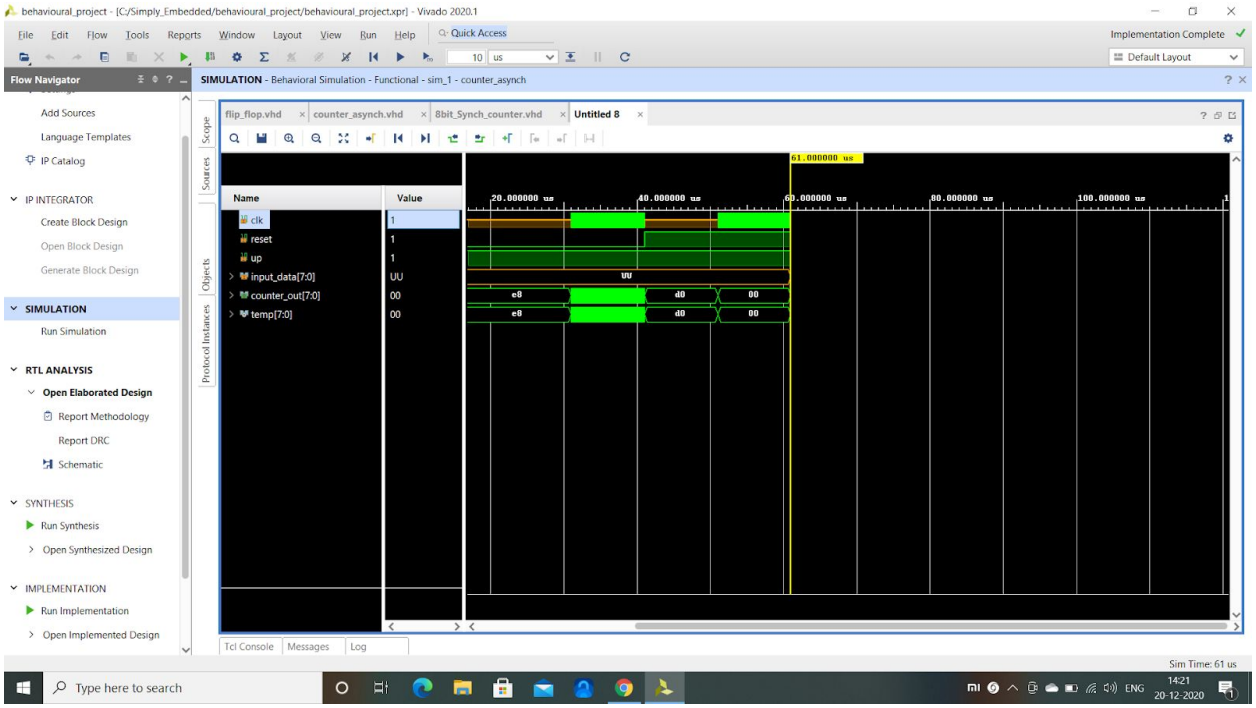
Reset = ‘1’ clock = enabled



Reset = ‘1’ clock = disabled



Reset = ‘1’ clock = enabled

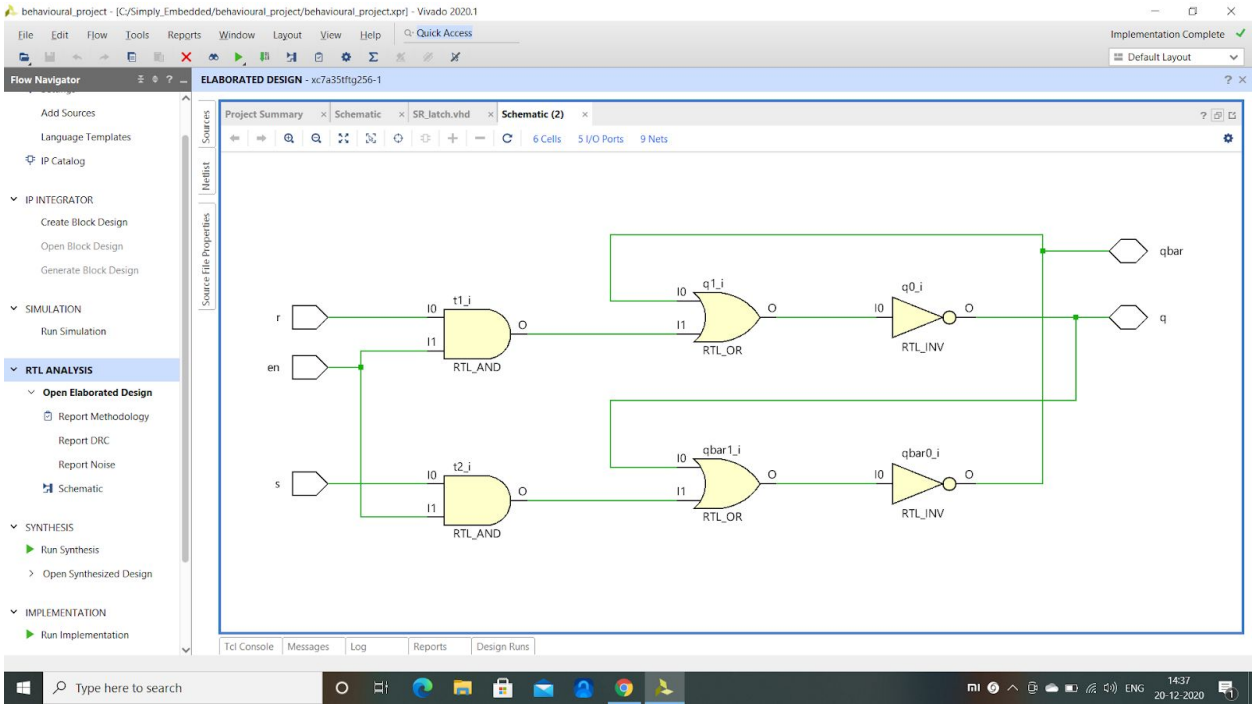


SR LATCH

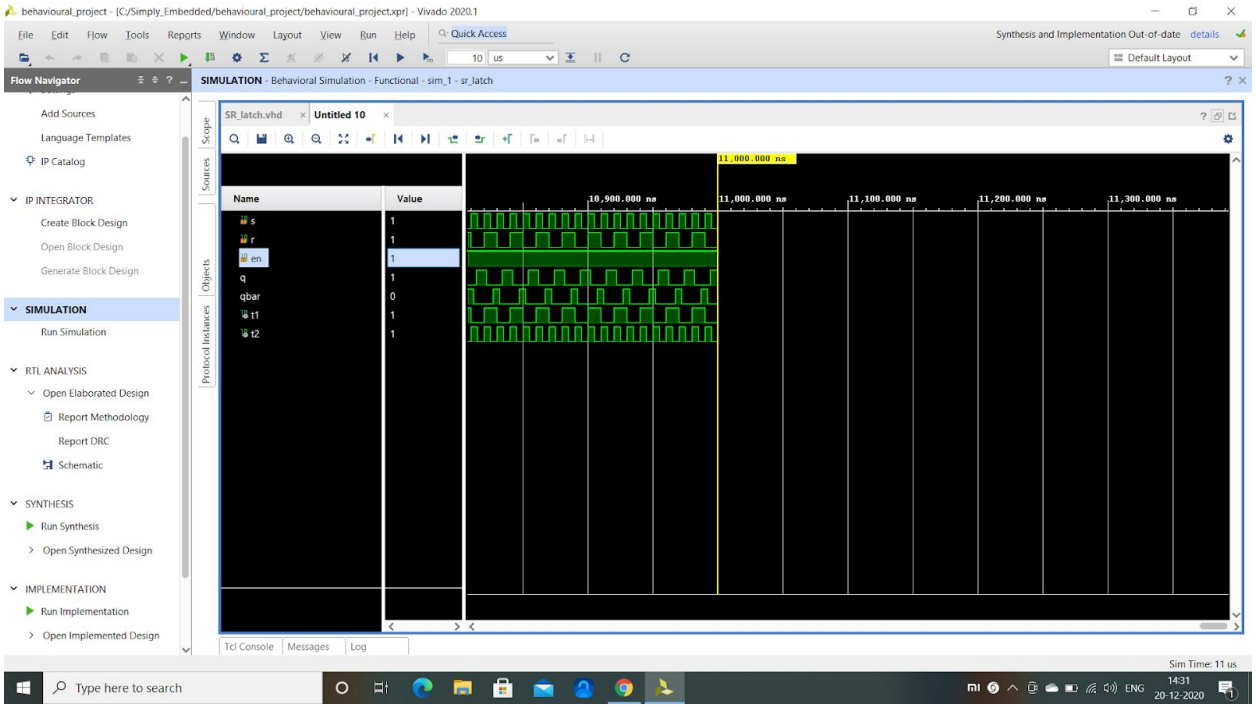
Code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity sr_latch is
    Port ( s,r,en : in STD_LOGIC;
           q,qbar : inout STD_LOGIC);
end sr_latch;

architecture Behavioral of sr_latch is
signal t1,t2 : std_logic;
begin
t1 <= r and en;
t2 <= s and en;
Q <= Qbar nor t1 after 2 ns;
Qbar <= Q nor t2 after 2 ns;
end Behavioral;
```

RTL schematic:


behavioural_project - [C:/Simply_Embedded/behavioural_project/behavioural_project.xpr] - Vivado 2020.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Implementation Complete
Default Layout
Flow Navigator
ELABORATED DESIGN - xc7a35tftg256-1
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
Report Methodology
Report DRC
Report Noise
Schematic
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Project Summary
Schematic
SR_latch.vhd
Schematic (2)
6 Cells
5 I/O Ports
9 Nets
r
en
s
t1_i
t2_i
q1_i
qbar1_i
q0_i
qbar0_i
RTL_AND
RTL_OR
RTL_INV
qbar
q
Tcl Console
Messages
Log
Reports
Design Runs
Type here to search
ENG
14:37
20-12-2020

Waveform:


behavioural_project - [C:/Simply_Embedded/behavioural_project/behavioural_project.xpr] - Vivado 2020.1
File Edit Flow Tools Reports Window Layout View Run Help
Quick Access
Synthesis and Implementation Out-of-date details
Default Layout
Flow Navigator
SIMULATION - Behavioral Simulation - Functional - sim_1 - sr_latch
SR_latch.vhd
Untitled 10
Name
Value
s
r
en
q
qbar
t1
t2
10,900.000 ns
11,000.000 ns
11,100.000 ns
11,200.000 ns
11,300.000 ns
Tcl Console
Messages
Log
Sim Time: 11 us
Type here to search
ENG
14:31
20-12-2020

MUX 4 : 1

Code:

```
entity mux_41 is
Port (
a,b,c,d : in std_logic;
sel : in std_logic_vector (1 downto 0);
out_mux : out std_logic

 );
end mux_41;

architecture Behavioral of mux_41 is

begin
process
begin
case(sel) is
when "00" =>out_mux <= a;
when "01" =>out_mux <= b;
when "10" =>out_mux <= c;
when "11" =>out_mux <= d;
when others => out_mux <= '0';
end case;
end process;
```

RTL schematic: