ELEC 457

Assignment-5

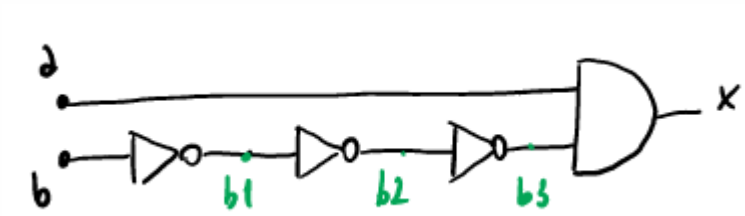
Sait Berk VARIMLI

141024054

I prepared this homework in 7 hours.

Problem 1

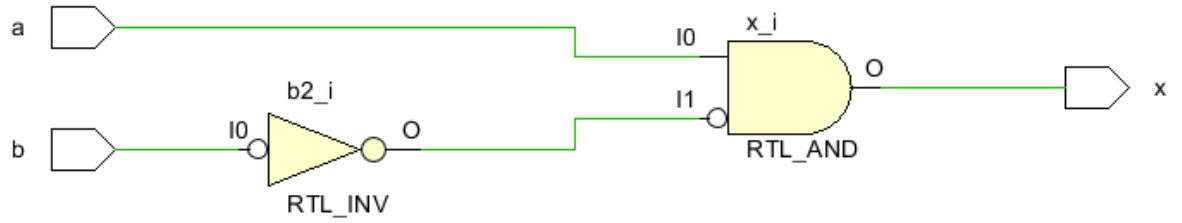
Implement the following logic circuit **as is** in your synthesis tool.



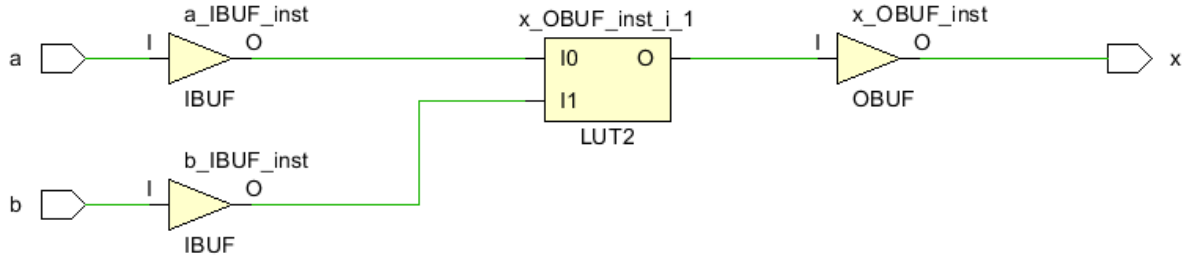
Code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity invertand is
Port (
a : in std_logic;
b : in std_logic;
x : out std_logic );
end invertand;
architecture Behavioral of invertand is
signal b1,b2,b3 : std_logic;
begin
b1 <= not b;
b2 <= not b1;
b3 <= not b2;
x <= a and b3;
end Behavioral;
```

Elaborated Schematic:



Implementation Schematic:



Report Utizilation:

Name	Slice LUTs (53200)	Slice (13300)	LUT as Logic (53200)	Bonded IPADs (2)
N invertand	1	1	1	3

Problem 2

Design and simulate a **shift register**. You should have a generic that defines how many flip-flop stages your input should be delayed.

Code:

```
library ieee;
use ieee.std_logic_1164.all;
entity shiftr is
    generic(
        N : integer := 32
    );
    port(
        clk : in std_logic;
        input  : in std_logic;
        output  : out std_logic
```

```
    );
end shiftr;
architecture rtl_a of shiftr is
    signal s_i : std_logic_vector(N - 1 downto 0);
begin
    process(clk) is
    begin
        if rising_edge(clk) then
            s_i <= s_i(N - 2 downto 0) & input;
        end if;
    end process;

    output <= s_i(N);

end rtl_a;

architecture rtl_b of shiftr is
    signal s_i : std_logic_vector(N - 1 downto 0);
begin
    process(clk) is
    begin
        if rising_edge(clk) then
            for i in 0 to N - 2 loop
                s_i(i+1) <= s_i(i);
            end loop;
            s_i(0) <= input;
        end if;
    end process;
    output <= s_i(N - 1);
end rtl_b;
```

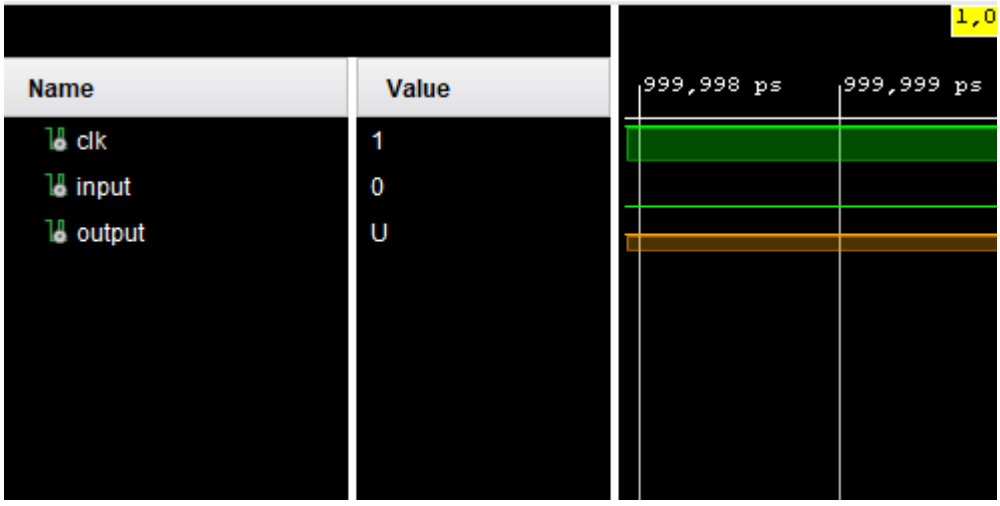
Testbench code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity tb_shiftr is
end tb_shiftr;
architecture Behavioral of tb_shiftr is
component shiftr is
generic(
        N : integer := 32
    );
    port(
        clk : in std_logic;
        input  : in std_logic;
        output  : out std_logic
    );
end component;
signal clk,input,output : std_logic;
begin
h0: shiftr port map(
clk=>clk, input=>input, output=>output);
process
begin
wait for 50ns;
clk <= '0';
input <= '0';
wait for 10ns;
clk <= '1';
input <= '1';
wait for 10ns;
clk <= '0';
input <= '1';
wait for 10ns;
```

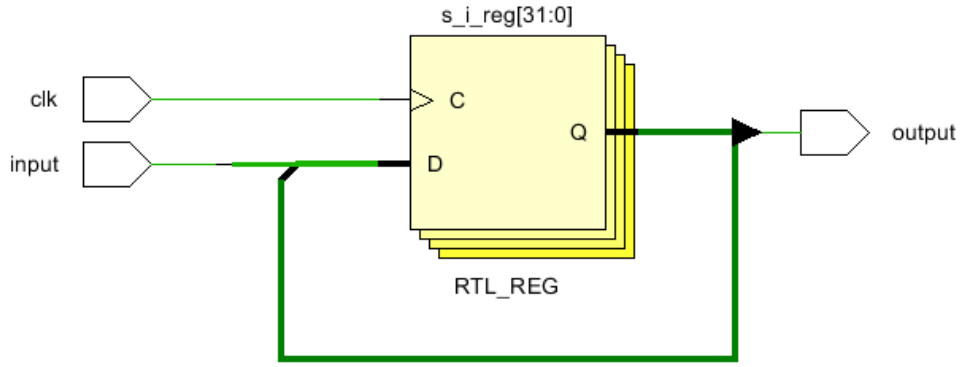
```
clk <= '1';
input <= '0';
wait for 10ns;
clk <= '0';
input <= '1';
wait for 10ns;
clk <= '1';
input <= '0';
wait;
end process;
end Behavioral;
```

Simulation:

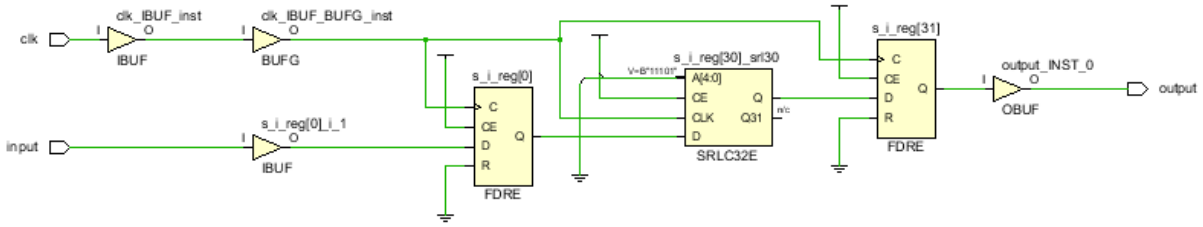
I got problem in simulation, i couldn’t fix it.



Elaborated Schematic:



Implementation Schematic:



Report Utizilation:

Name	Slice LUTs (53200)	Slice Registers (106400)	Slice (13300)	LUT as Memory (17400)	Block RAM Tile (140)	Bonded IPADs (2)	BUFIO (16)
N shiftr	1	2	2	1	2	3	1

Problem 4

Design and simulate a 32-bit count-down timer.

- Timer should count down from the given M value, and whenever it reaches to 0, it should generate a `done` flag.
- Timer should be free-running meaning it should keep running as long as the `enable` port is held high.
- Timer should generate the `done` flag when the `M`th value is reached. `M` is given as `generic`.

Code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity counter is
generic ( M: integer := 2**20);
Port (
clk, rst, i_en : in std_logic;
o_tick_s, o_tick_v : out std_logic
 );
end counter;

architecture rtl of counter is
    signal led_int : std_logic := '0';
    signal tick : std_logic := '0';
    signal s_cnt, v_cnt : unsigned(31 downto 0);
begin
process (clk) is
begin
if rising_edge(clk) then
if rst = '1' then
s_cnt <= to_unsigned(M-1, 32);
elsif i_en ='1' then
```

```
if s_cnt=0 then
s_cnt <= to_unsigned(M,32);
else
s_cnt <= s_cnt -1;
end if;
end if;
end if;
end process;
process(clk) is
begin
if rising_edge(clk) then
if rst = '1' then
v_cnt <= to_unsigned(M-1,32);
elsif i_en ='1' then
if v_cnt=0 then
v_cnt<=to_unsigned(M,32);
else
v_cnt<=v_cnt-1;
end if;
end if;
end if;
end process;

o_tick_s <= '1' when s_cnt = 0 else '0';
o_tick_v <= '1' when v_cnt = 0 else '0';

end rtl;
```

Testbench code:

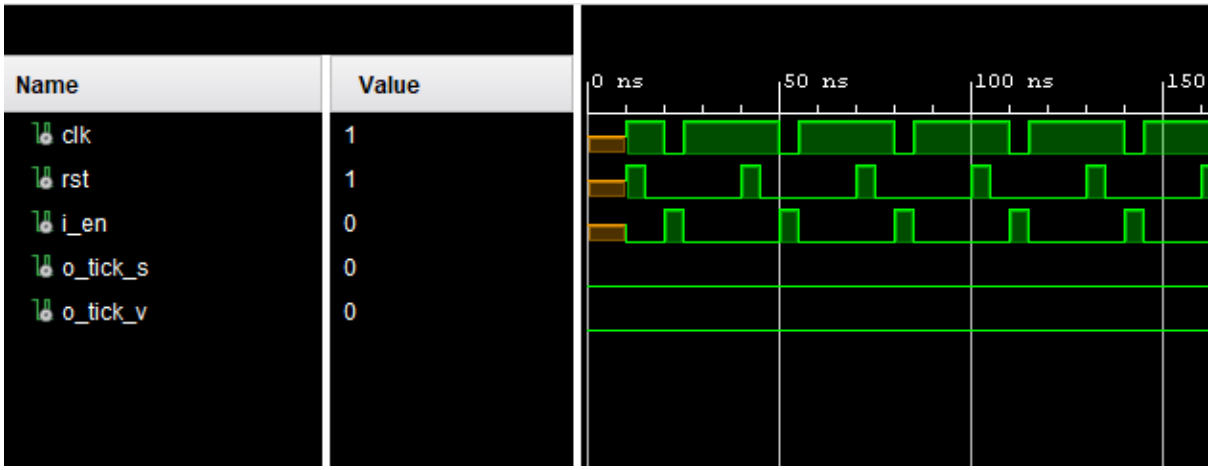
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
```

```
entity tb_counter is
end tb_counter;

architecture Behavioral of tb_counter is
component counter is
generic(M: integer := 20);
port( clk, rst, i_en : in std_logic;
o_tick_s, o_tick_v : out std_logic);
end component;

signal clk,rst,i_en,o_tick_s,o_tick_v : std_logic;

begin
h0: counter port map(
clk=>clk, rst=>rst, i_en=>i_en, o_tick_s=>o_tick_s, o_tick_v=>o_tick_v);
process
begin
wait for 10ns;
clk <= '1';
rst <= '1';
i_en <= '0';
wait for 5 ns;
clk <= '1';
i_en <= '0';
rst <= '0';
wait for 5 ns;
clk <= '0';
i_en <= '1';
rst <= '0';
wait for 5 ns;
clk <= '1';
i_en <= '0';
```

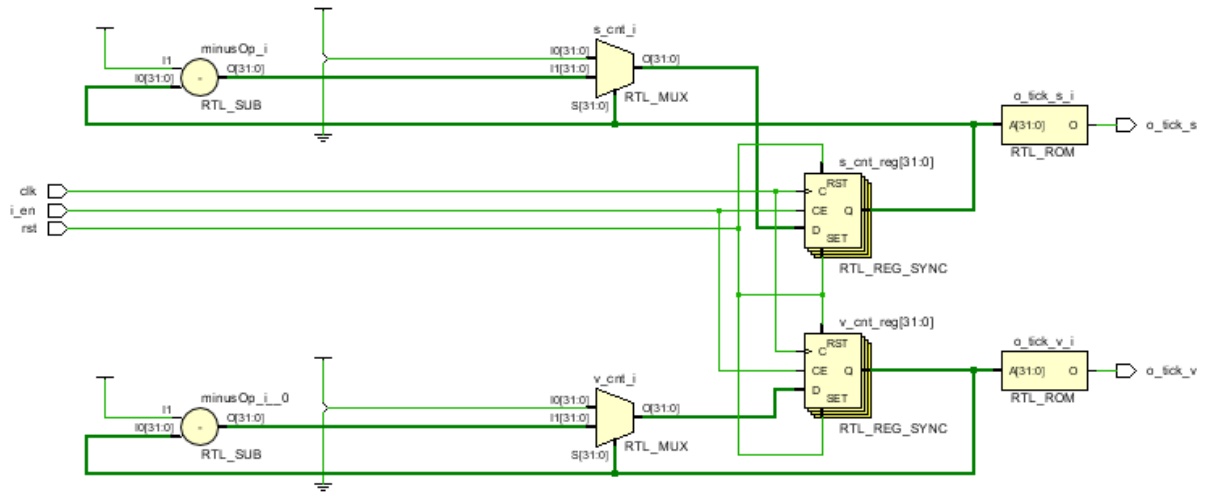
```
rst <= '0';

wait for 5 ns;

end process;


end Behavioral;
```

Simulation:

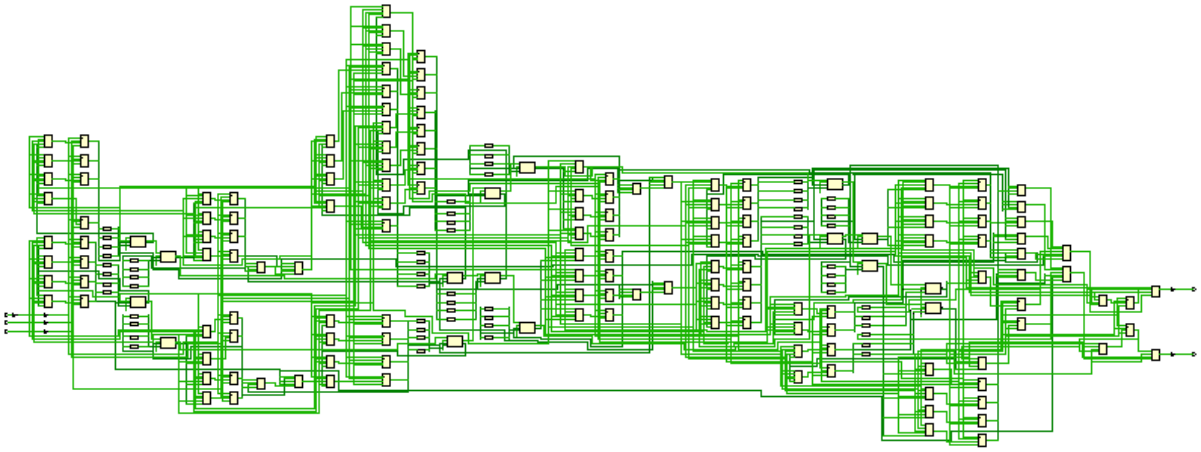
I had to leave from last lecture’s lab so i couldn’t prepare working simulation.



Elaborated Schematic:



Implementation Schematic:



Report Utizilation:

Name	Slice LUTs (53200)	Slice Registers (106400)	Slice (13300)	LUT as Logic (53200)	Block RAM Tile (140)	Bonded IPADs (2)	BUFIO (16)
N counter	140	64	43	140	64	5	1

Problem 5

Design and simulate a 6-bit up / down counter.

- There should be an input to choose between up and down modes.
- The counter value should be the output.
- Use your circuit from problem 1 for slowing down your counter for human interaction (you can choose a big `M` to slow down).
- Simulate & verify circuit operation.
- You may implement this on the boards. Choose RGB Leds as outputs, a switch to enable, and another switch to choose the direction of the counter.

Code:

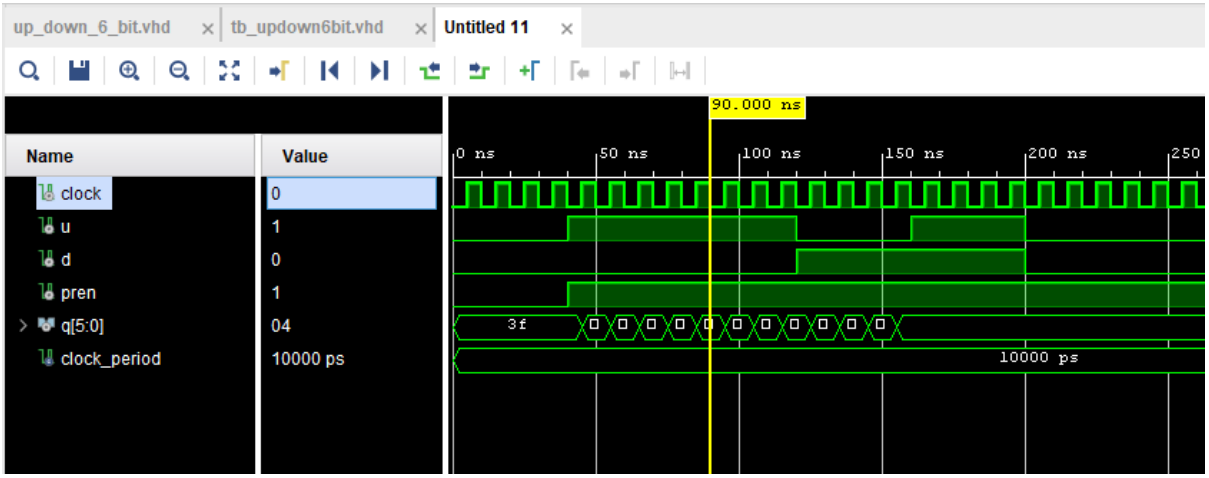
```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

use ieee.std_logic_unsigned.all;

use ieee.std_logic_arith.all;

entity up_down_6_bit is

Port(clock : in std_logic;

u : in std_logic;

d : in std_logic;

pren : in std_logic;

q : out std_logic_vector(5 downto 0));

end up_down_6_bit;

architecture Behavioral of up_down_6_bit is

signal reg: std_logic_vector (5 downto 0);

begin

process(pren, clock)

begin

if(pren='0') then

reg<="111111";

elsif(clock'event and clock='1') then

if(u='1' and d='0') then

reg<=reg + '1';

elsif(u='0' and d='1') then

reg<=reg-'1';
```

```
else
reg<=reg;
end if;
end if;
end process;
q <= reg;
end Behavioral;


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity tb_updown6bit is
--  Port ( );
end tb_updown6bit;

architecture Behavioral of tb_updown6bit is
component up_down_6_bit
port(
clock : in std_logic;
u : in std_logic;
d : in std_logic;
pren : in std_logic;
q : out std_logic_vector(5 downto 0));
end component;

signal clock,u,d,pren : std_logic :='0';
signal q : std_logic_vector(5 downto 0);
constant clock_period : time := 10ns;
begin
u0: up_down_6_bit port map(
clock=>clock, u=>u, d=>d, pren=>pren, q=>q);
clock_process : process
```

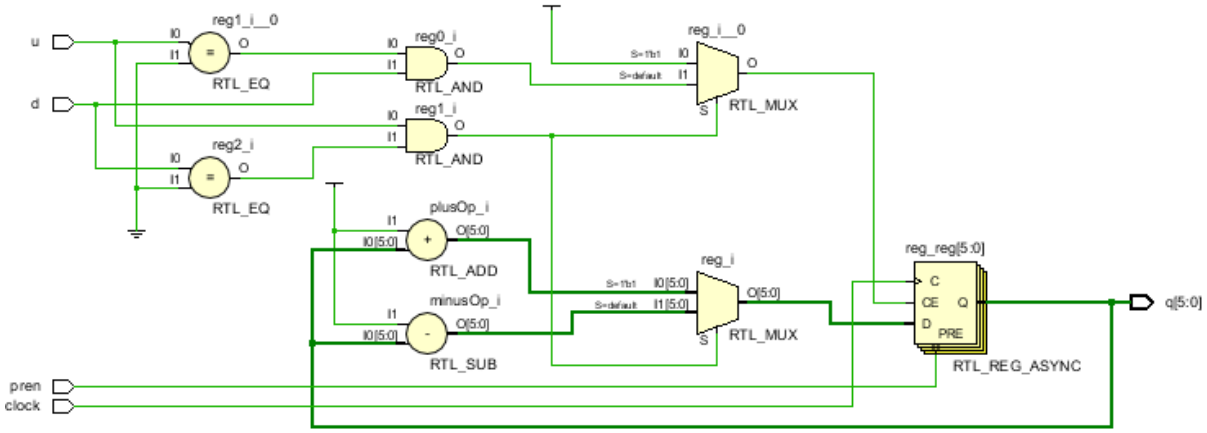
```
begin
clock <= '0';
wait for clock_period/2;
clock <='1';
wait for clock_period/2;
end process;
stim_proc: process
begin
pren <= '0';
wait for 40ns;
pren <='1';
u<='1';
d<='0';
wait for 40 ns;
pren<='1';
u<='1';
d<='0';
wait for 40 ns;
pren<='1';
u<='0';
d<='1';
wait for 40 ns;
pren<='1';
u<='1';
d<='1';
wait for 40 ns;
pren<='1';
u<='0';
d<='0';
wait for 40 ns;
pren <='1';
u<='0';
```

```
d<='0';
wait;
end process;

end Behavioral;
```

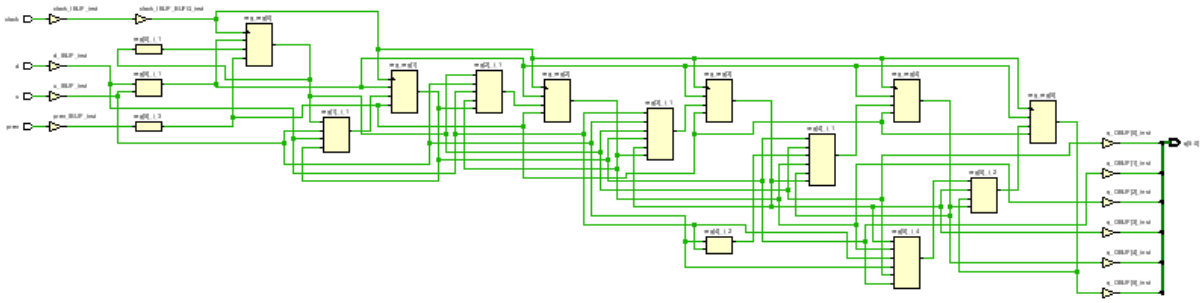
Simulation:



Elaborated Schematic:



Implementation Schematic:



Report Utizilation:

Name	Slice LUTs (53200)	Slice Registers (106400)	Slice (13300)	LUT as Logic (53200)	Block RAM Tile (140)	Bonded IPADs (2)	BUFIO (16)
up_down_6_bit	8	6	2	8	6	10	1