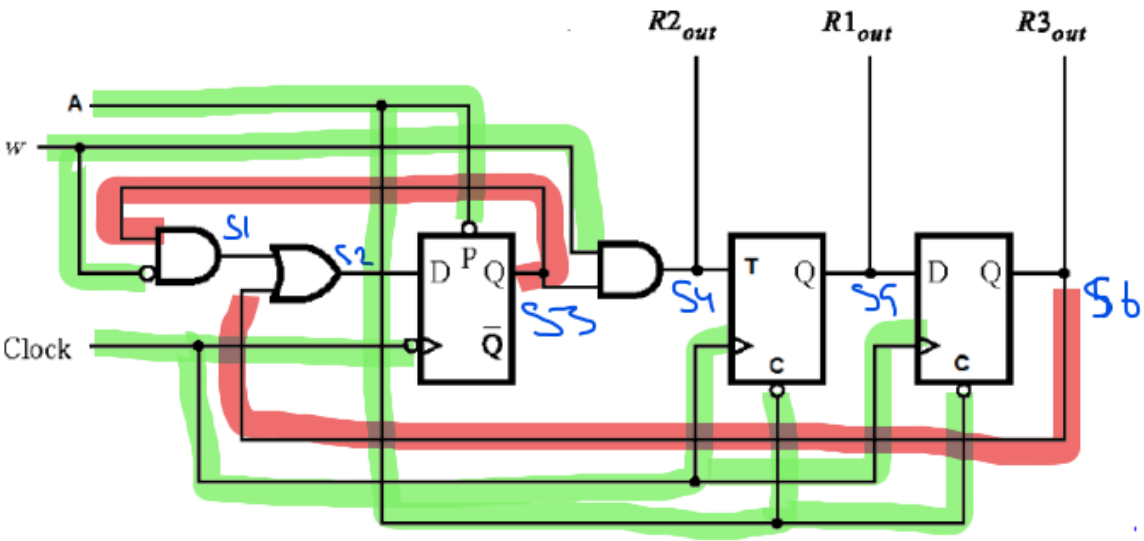
Assignment 2

Name: Nour-aldin Ibrahim Ahmed Elbadawy
ID: 2018-13394

Q5)

Question 5: (Assignment)

Consider the circuit diagram shown in figure:



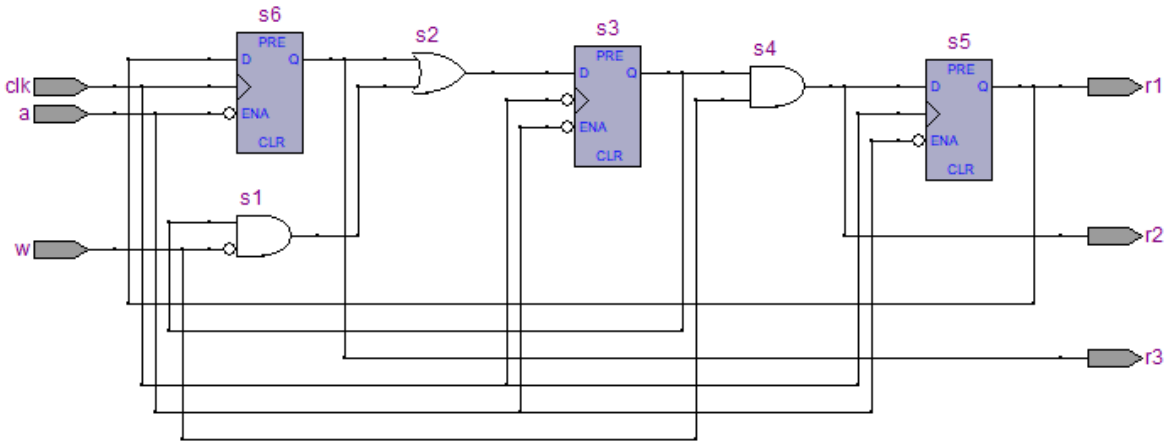
Behavioral VHDL

Main Code

```
library ieee;
use ieee.std_logic_1164.all;




entity Q5 is
port(
                a,w,clk : in std_logic;
                r1,r2,r3: out std_logic );
end entity;
```

```
architecture behave of Q5 is

	signal s1,s2,s3,s4,s5,s6: std_logic;

begin
	s1 <= not(w) and s3;
	s2 <= s6 or s1;

	process (clk,a)
	begin
		if ( falling_edge (clk) ) then
			if(a = '0') then
				s3 <= s2;
			end if;
		end if;
	end process;

	s4 <= s3 and w;
	r2 <= s4;

	process (clk,a)
	begin
		if ( rising_edge (clk) ) then
			if(a = '0') then
				s5 <= s4;
			end if;
		end if;
	end process;

	process (clk,a)
	begin
		if ( rising_edge (clk) ) then
			if(a = '0') then
				s6 <= s5;
			end if;
		end if;
	end process;

	r1 <= s5;
	r3 <= s6;



end behave;
```

RTL



note that I didn't use D and T flip flop and I replaced them with register.

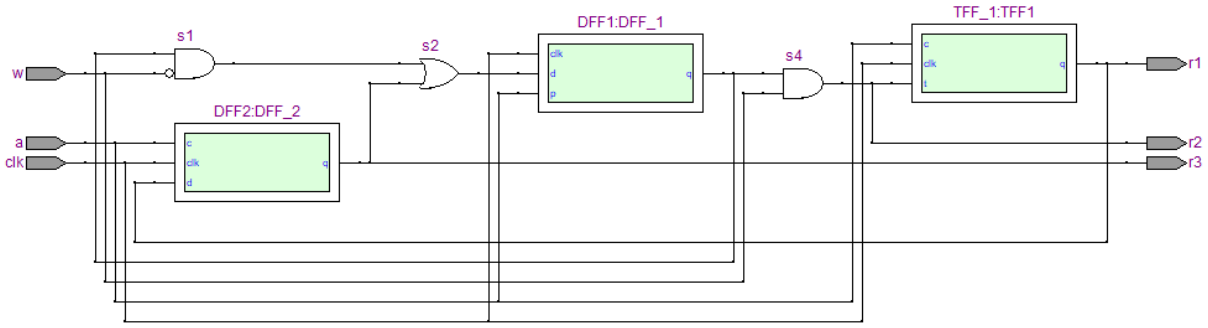
Structural VHDL

Main code

```
library ieee;
use ieee.std_logic_1164.all;




entity Q5 is
port(
                a,w,clk : in std_logic;
                r1,r2,r3: out std_logic );
end entity;



architecture behave of Q5 is

        signal s1,s2,s3,s4,s5,s6: std_logic;

        component DFF1 is
        port(
                        p,d,clk : in std_logic;
                        q : out std_logic
                );
        end component;

        component DFF2 is
        port(
                        c,d,clk : in std_logic;
                        q: out std_logic
                );
        end component;

        component TFF_1 is
        port(
                        c,t,clk : in std_logic;
                        q: out std_logic
                );
        end component;


begin
```

```
        s1 <= not(w) and s3;
        s2 <= s6 or s1;



        DFF_1: DFF1 port map( d => s2, q => s3, clk => clk, p => a);


        s4 <= s3 and w;
        r2 <= s4;

        TFF1: TFF_1 port map( c => a, t => s4, clk => clk, q => s5 );

        DFF_2: DFF2 port map( d => s5, q => s6, clk => clk, c => a);

        r1 <= s5;
        r3 <= s6;



end behave;
```

RTL



TFF Component

```
library ieee;
use ieee.std_logic_1164.all;




entity TFF_1 is
port(
                c,t,clk : in std_logic;
                q: out std_logic );
end entity;



architecture behave of TFF_1 is


begin

q <= c and t and clk; -- just to get the rtl shape


end behave;
```

DFF1 Component

```
library ieee;
use ieee.std_logic_1164.all;




entity DFF1 is
port(
                p,d,clk : in std_logic;
                q: out std_logic );
end entity;



architecture behave of DFF1 is


begin

q <= p and d and clk; -- just to get the rtl shape


end behave;
```

DFF2 Component

```
library ieee;
use ieee.std_logic_1164.all;




entity DFF2 is
port(
                c,d,clk : in std_logic;
                q: out std_logic );
end entity;



architecture behave of DFF2 is


begin

q <= c and d and clk; -- just to get the rtl shape


end behave;
```