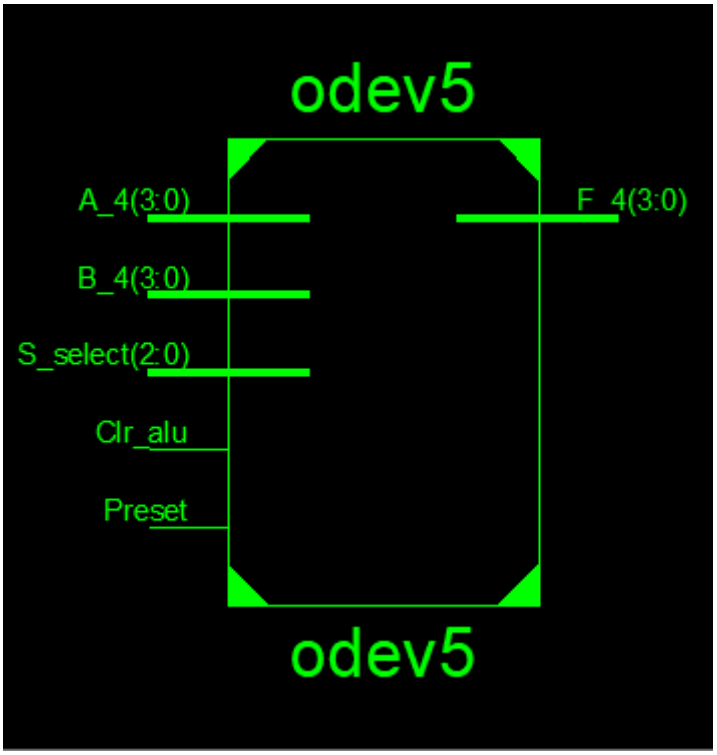
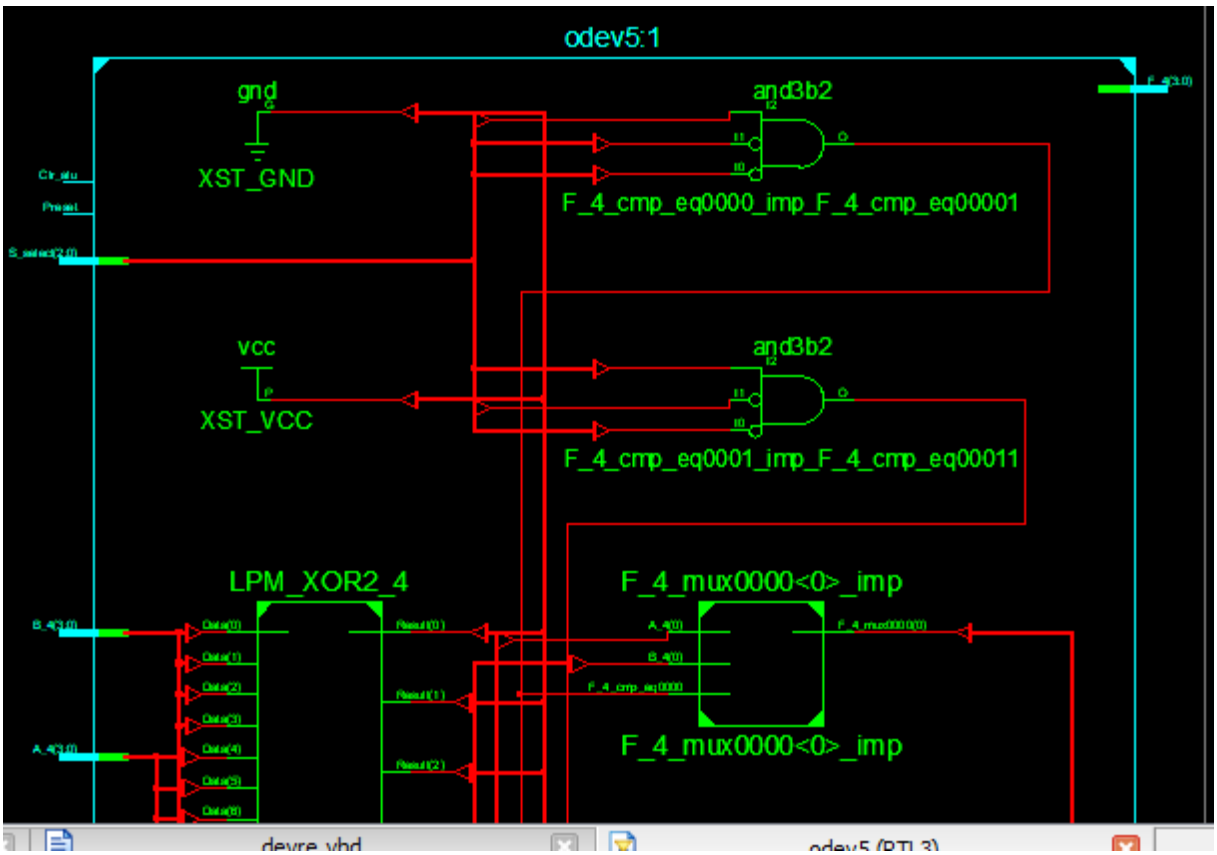
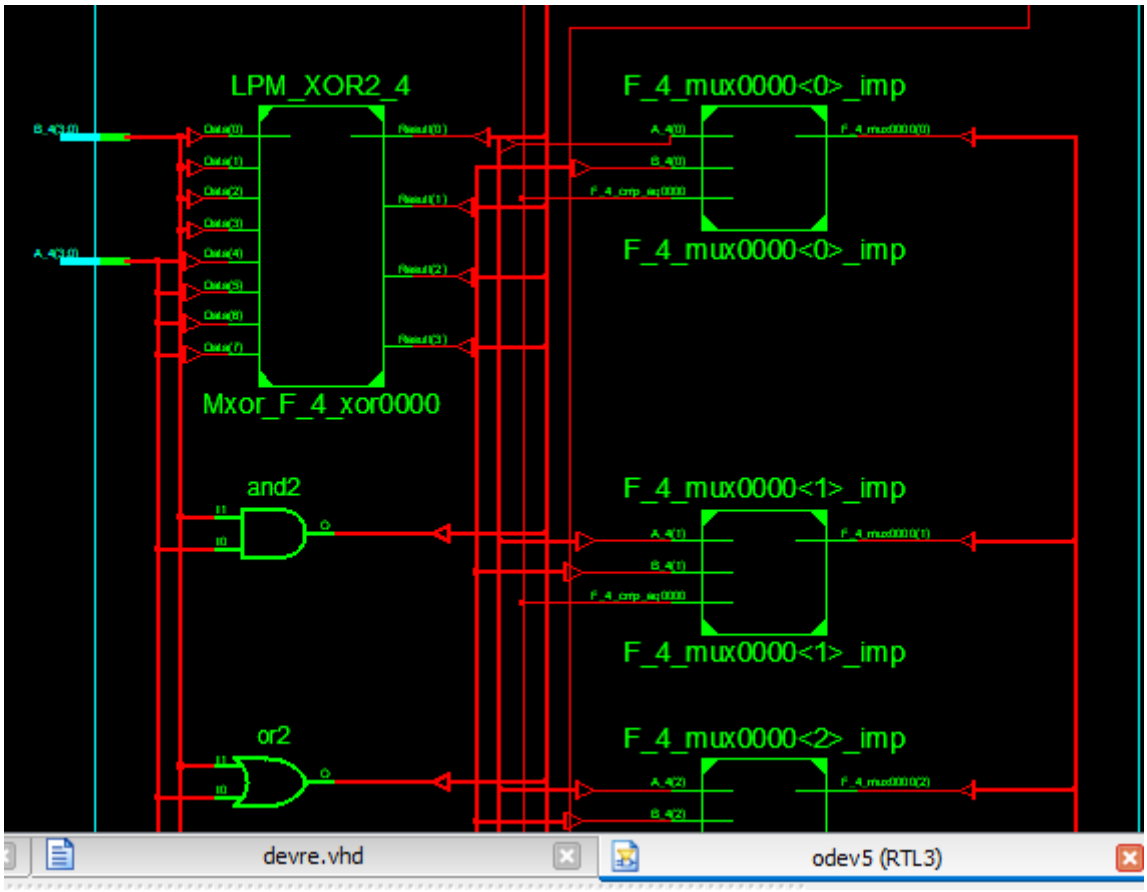
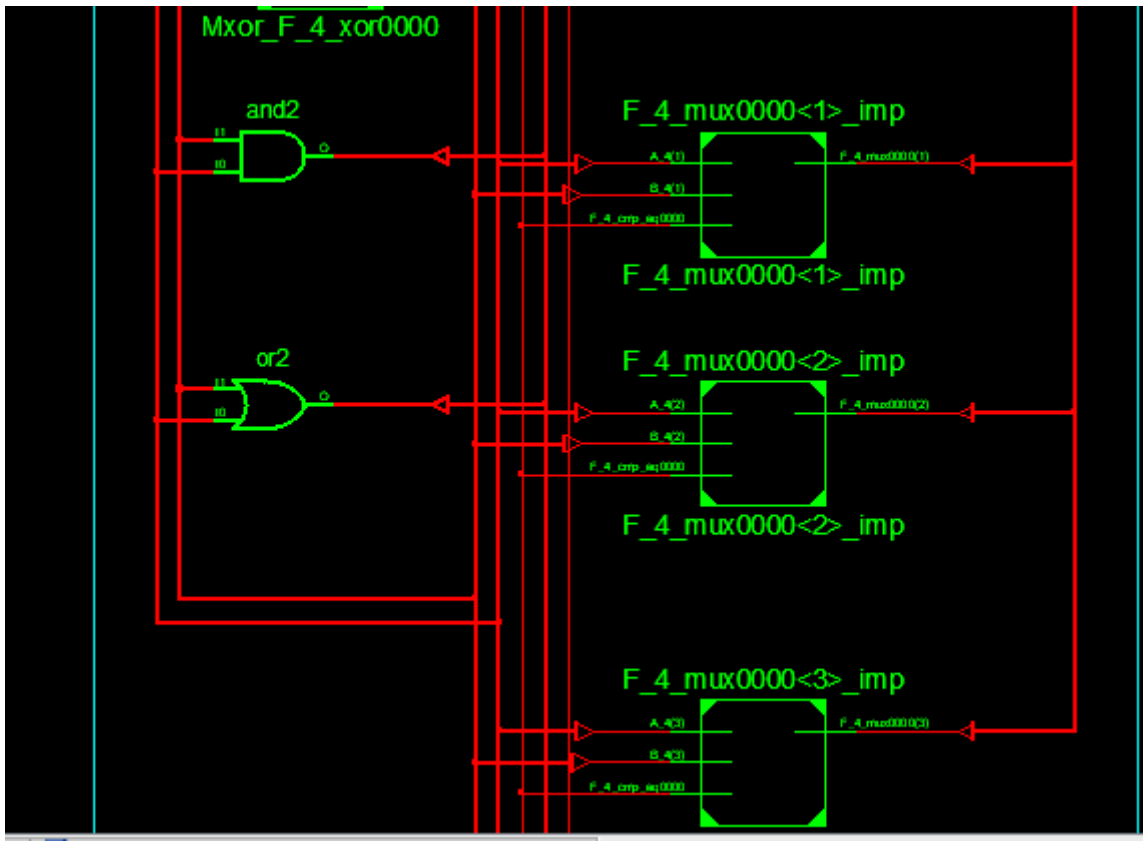
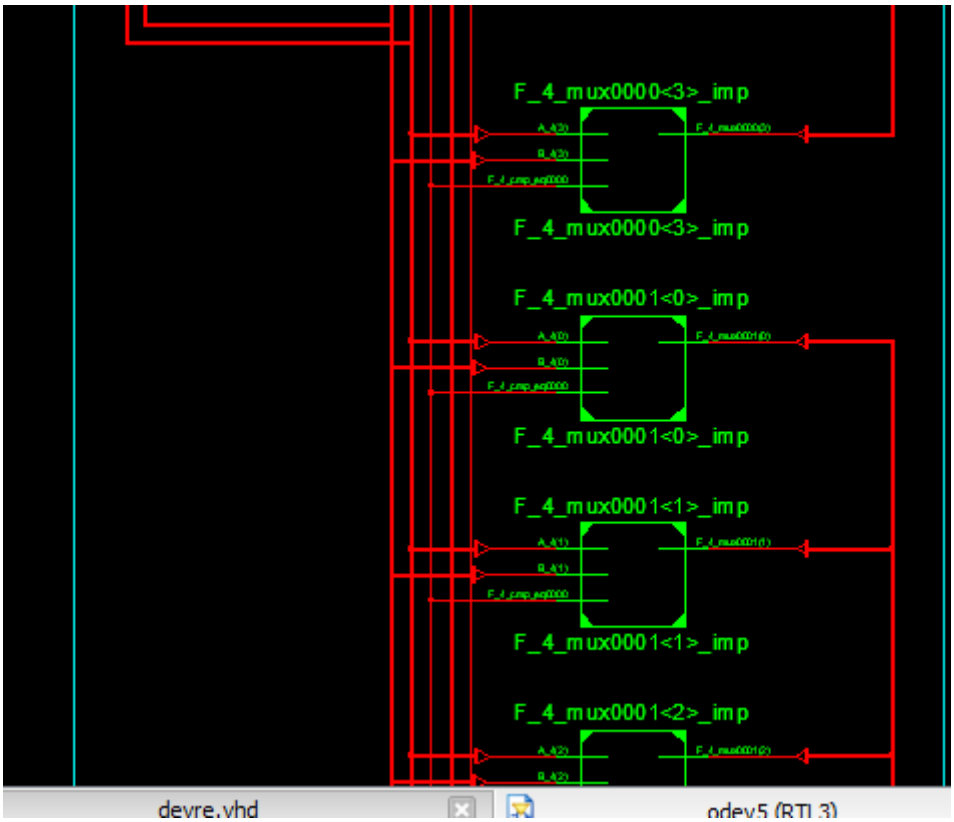
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity odev5 is
	port(	A_4: in STD_LOGIC_VECTOR (3 downto 0);
		B_4: in STD_LOGIC_VECTOR (3 downto 0);
		Clr_alu, Preset: in STD_LOGIC;
		S_select: in STD_LOGIC_VECTOR (2 downto 0);
		F_4: out STD_LOGIC_VECTOR (3 downto 0));
end odev5;
architecture Behavioral of odev5 is
begin
	process(A_4, B_4, Clr_alu, Preset, S_select)
	begin
		case S_select is
			when "000" => F_4<="0000";
			when "001" => F_4<=B_4 - A_4;
			when "010" => F_4<=A_4 - B_4;
			when "011" => F_4<=A_4 + B_4;
			when "100" => F_4<=A_4 XOR B_4;
			when "101" => F_4<=A_4 OR B_4;
			when "110" => F_4<=A_4 AND B_4;
			when "111" => F_4<="1111";
			when others => NULL;
		end case;
	end process;
end Behavioral;
```

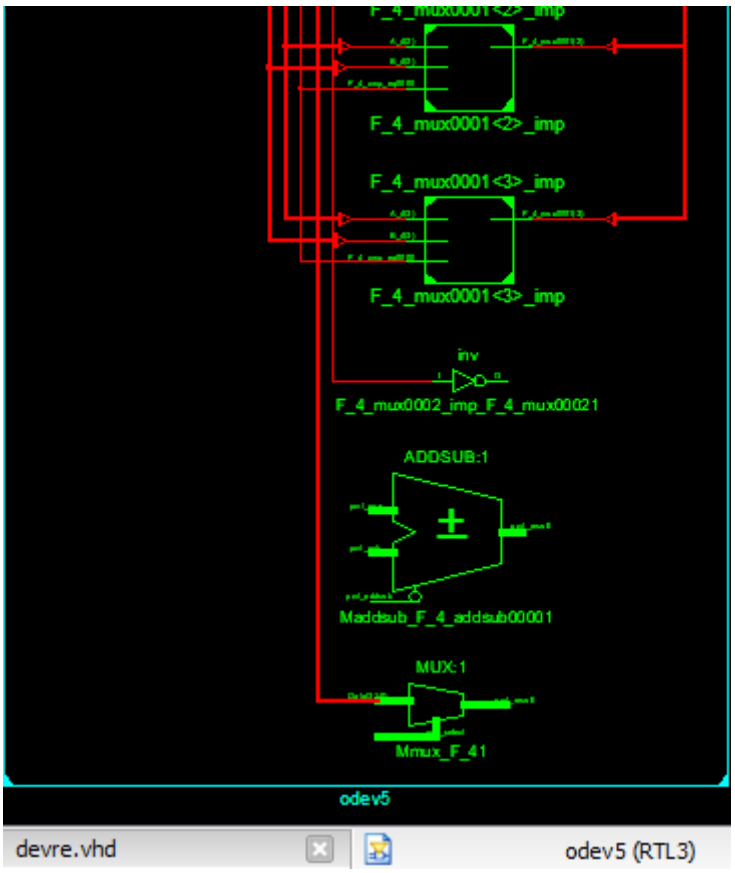

odev5
A_4(3:0)
B_4(3:0)
S_select(2:0)
Clr_alu
Preset
F_4(3:0)
odev5

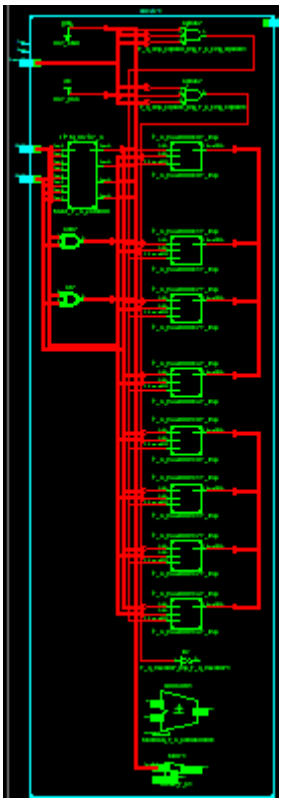

odev5:1
gnd
XST_GND
and3b2
F_4_cmp_eq0000_imp_F_4_cmp_eq00001
vcc
XST_VCC
and3b2
F_4_cmp_eq0001_imp_F_4_cmp_eq00011
LPM_XOR2_4
F_4_mux0000<0>_imp
F_4_mux0000<0>_imp


LPM_XOR2_4
F_4_mux0000<0>_imp
F_4_mux0000<0>_imp
Mxor_F_4_xor0000
and2
F_4_mux0000<1>_imp
F_4_mux0000<1>_imp
or2
F_4_mux0000<2>_imp
devre.vhd
odev5 (RTL3)

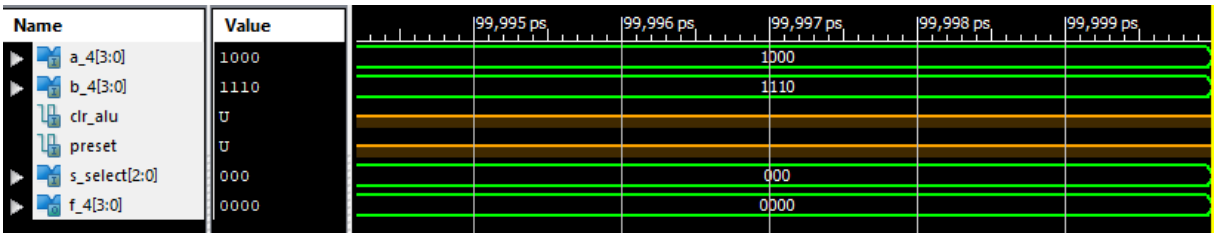

Mxor_F_4_xor0000
and2
F_4_mux0000<1>_imp
F_4_mux0000<1>_imp
or2
F_4_mux0000<2>_imp
F_4_mux0000<2>_imp
F_4_mux0000<3>_imp


F_4_mux0000<3>_imp
F_4_mux0000<3>_imp
F_4_mux0001<0>_imp
F_4_mux0001<0>_imp
F_4_mux0001<1>_imp
F_4_mux0001<1>_imp
F_4_mux0001<2>_imp
devre.vhd
odev5 (RTL3)

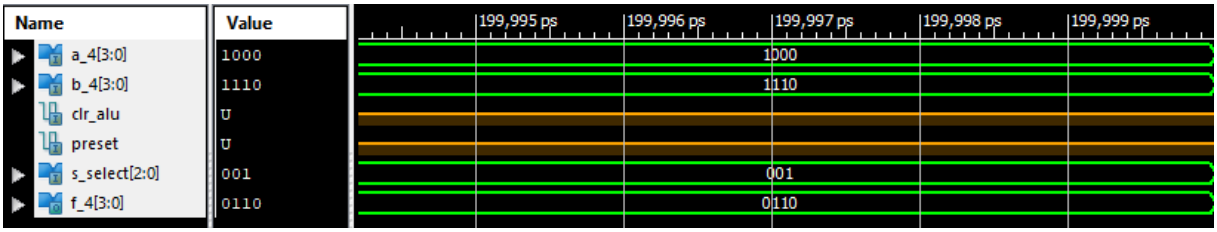

F_4_mux0001<2>_imp
F_4_mux0001<3>_imp
F_4_mux0001<3>_imp
inv
F_4_mux0002_imp_F_4_mux00021
ADDSUB:1
Maddsub_F_4_addsub00001
MUX:1
Mmux_F_41
odev5
devre.vhd
odev5 (RTL3)



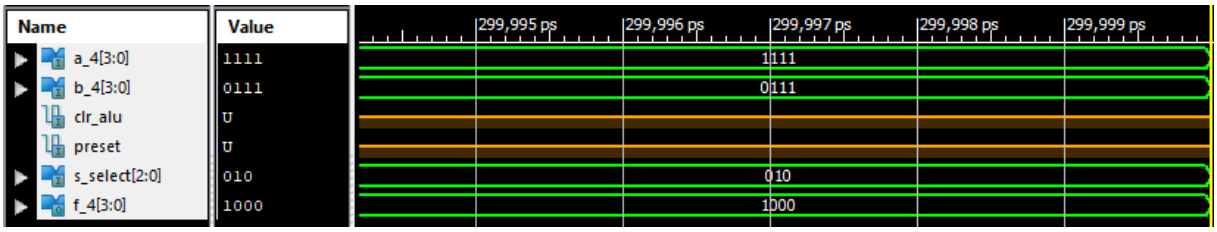
Operation	Inputs S2 S1 S0	Outputs F
Clear	0 0 0	0 0 0 0
B - A	0 0 1	B - A
A – B	0 1 0	A – B
ADD	0 1 1	A + B
XOR	1 0 0	A XOR B
OR	1 0 1	A OR B
AND	1 1 0	A AND B
Preset	1 1 1	1 1 1 1



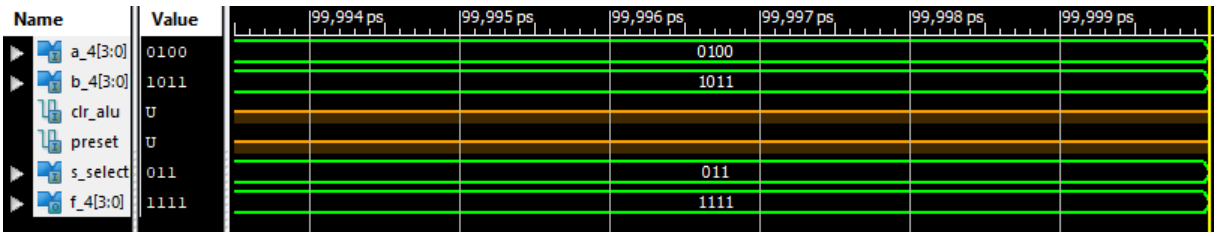
Operation	Inputs S2 S1 S0	Outputs F
Clear	0 0 0	0 0 0 0
B - A	0 0 1	B - A
A – B	0 1 0	A – B
ADD	0 1 1	A + B
XOR	1 0 0	A XOR B
OR	1 0 1	A OR B
AND	1 1 0	A AND B
Preset	1 1 1	1 1 1 1



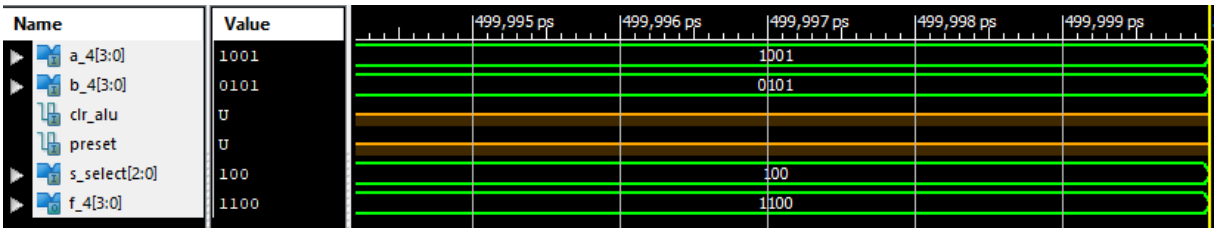
Operation	Inputs S2 S1 S0	Outputs F
Clear	0 0 0	0 0 0 0
B - A	0 0 1	B - A
A – B	0 1 0	A – B
ADD	0 1 1	A + B
XOR	1 0 0	A XOR B
OR	1 0 1	A OR B
AND	1 1 0	A AND B
Preset	1 1 1	1 1 1 1



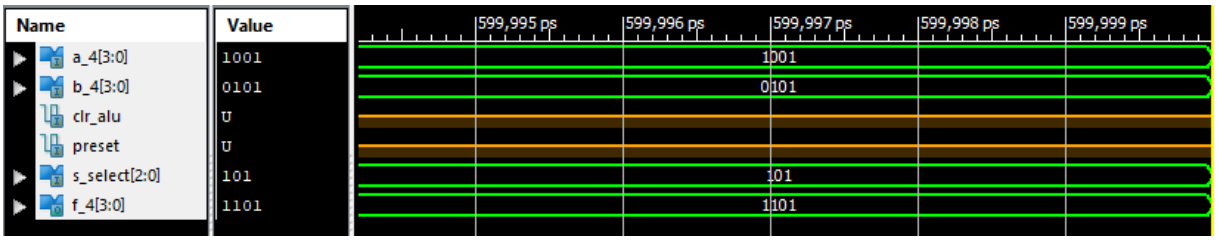
Operation	Inputs S2 S1 S0	Outputs F
Clear	0 0 0	0 0 0 0
B - A	0 0 1	B - A
A – B	0 1 0	A – B
ADD	0 1 1	A + B
XOR	1 0 0	A XOR B
OR	1 0 1	A OR B
AND	1 1 0	A AND B
Preset	1 1 1	1 1 1 1



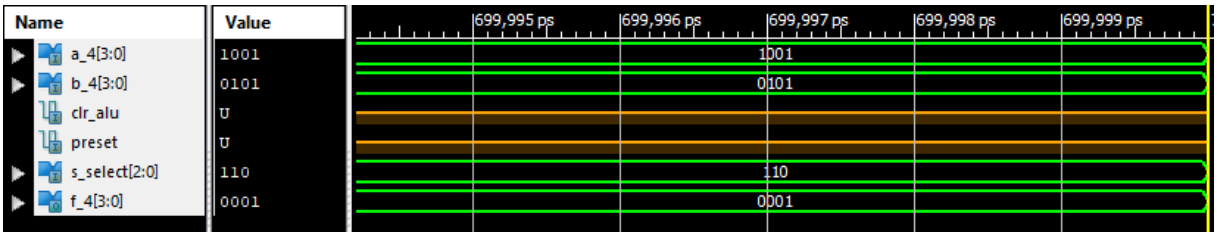
Operation	Inputs S2 S1 S0	Outputs F
Clear	0 0 0	0 0 0 0
B - A	0 0 1	B - A
A – B	0 1 0	A – B
ADD	0 1 1	A + B
XOR	1 0 0	A XOR B
OR	1 0 1	A OR B
AND	1 1 0	A AND B
Preset	1 1 1	1 1 1 1



Operation	Inputs S2 S1 S0	Outputs F
Clear	0 0 0	0 0 0 0
B - A	0 0 1	B - A
A – B	0 1 0	A – B
ADD	0 1 1	A + B
XOR	1 0 0	A XOR B
OR	1 0 1	A OR B
AND	1 1 0	A AND B
Preset	1 1 1	1 1 1 1



Operation	Inputs S2 S1 S0	Outputs F
Clear	0 0 0	0 0 0 0
B - A	0 0 1	B - A
A – B	0 1 0	A – B
ADD	0 1 1	A + B
XOR	1 0 0	A XOR B
OR	1 0 1	A OR B
AND	1 1 0	A AND B
Preset	1 1 1	1 1 1 1



Operation	Inputs S2 S1 S0	Outputs F
Clear	0 0 0	0 0 0 0
B - A	0 0 1	B - A
A – B	0 1 0	A – B
ADD	0 1 1	A + B
XOR	1 0 0	A XOR B
OR	1 0 1	A OR B
AND	1 1 0	A AND B
Preset	1 1 1	1 1 1 1