**Date:** 02 September 2022

# Implement AND Gate using VHDL for FPGA on Vivado Design Suite.

## Code:

**library** **IEEE**;

**use** **IEEE.STD\_LOGIC\_1164.ALL**;

**entity** **Problem\_1** **is**

**Port** ( a,b : **in** **STD\_LOGIC**;

c : **out** **STD\_LOGIC**);

**end** **Problem\_1**;

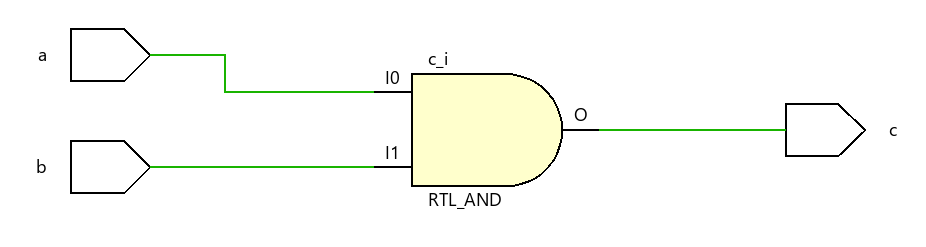
**architecture** **Behavioral** **of** **Problem\_1** **is**

**begin**

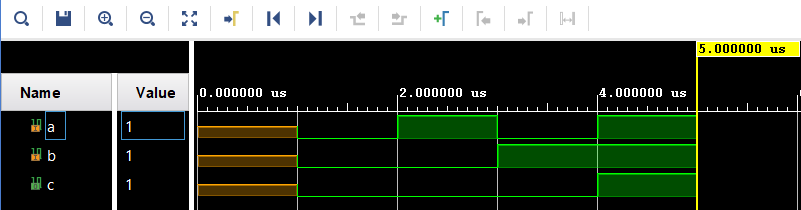
c <= a **and** b;

**end** **Behavioral**;

## RTL Schematic:



## Simulation Result:



# Implement all logic gates using VHDL for FPGA on Vivado Design Suite.

## Code:

**library** **IEEE**;

**use** **IEEE.STD\_LOGIC\_1164.ALL**;

**entity** **Probem\_2** **is**

**Port** ( a,b : **in** **STD\_LOGIC**;

y1,y2,y3,y4,y5,y6, y7 : **out** **STD\_LOGIC**);

**end** **Probem\_2**;

**architecture** **Behavioral** **of** **Probem\_2** **is**

**begin**

y1 <= a **and** b;

y2 <= a **or** b;

y3 <= a **nand** b;

y4 <= a **nor** b;

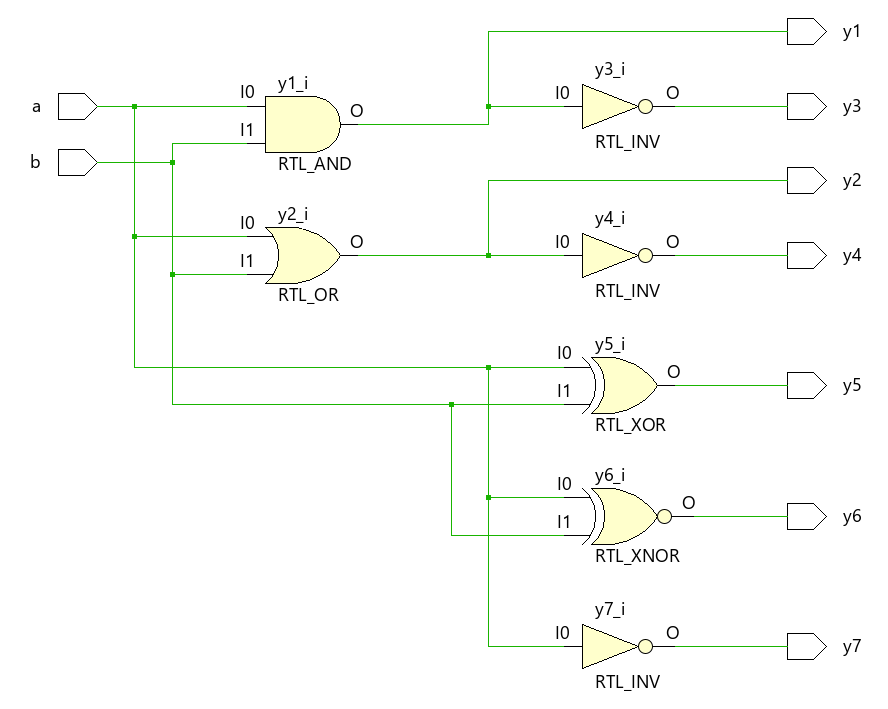
y5 <= a **xor** b;

y6 <= a **xnor** b;

y7 <= **not** a;

**end** **Behavioral**;

## RTL Schematic:



## Simulation Result:

# Implement all logic gates using vector representation of bit-string in VHDL for FPGA on Vivado Design Suite.

## Code:

**library** **IEEE**;

**use** **IEEE.STD\_LOGIC\_1164.ALL**;

**entity** **Problem\_3** **is**

**Port** ( a : **in** **STD\_LOGIC\_VECTOR** (**6** **downto** **0**);

b : **in** **STD\_LOGIC\_VECTOR** (**5** **downto** **0**);

c : **out** **STD\_LOGIC\_VECTOR** (**6** **downto** **0**));

**end** **Problem\_3**;

**architecture** **Behavioral** **of** **Problem\_3** **is**

**begin**

c(**0**) <= a(**0**) **or** b(**0**);

c(**1**) <= a(**1**) **and** b(**1**);

c(**2**) <= a(**2**) **nor** b(**2**);

c(**3**) <= a(**3**) **nand** b(**3**);

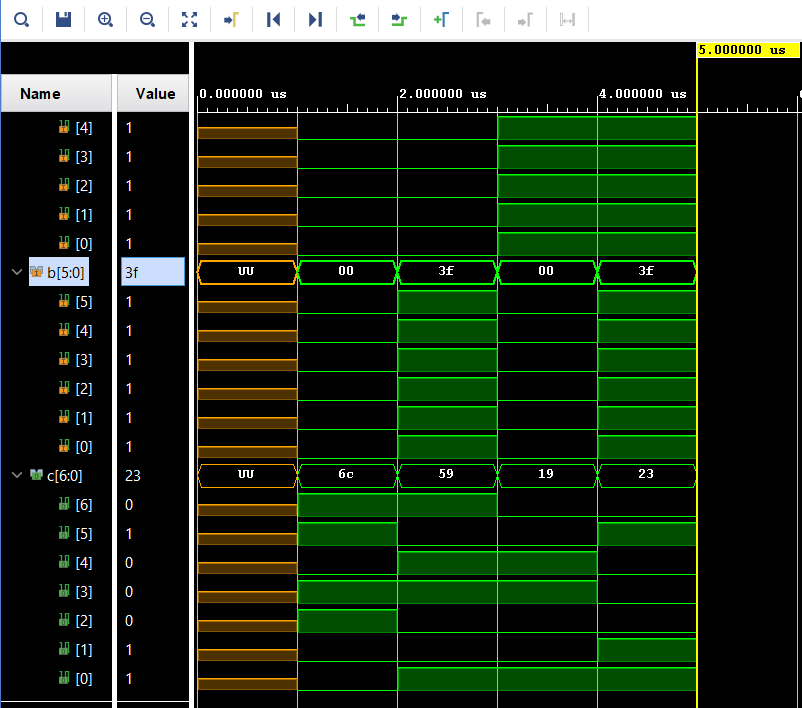
c(**4**) <= a(**4**) **xor** b(**4**);

c(**5**) <= a(**5**) **xnor** b(**5**);

c(**6**) <= **not** a(**6**);

**end** **Behavioral**;

## RTL Schematic:



## Simulation Result:

# Implement given logical expressions in VHDL for FPGA on Vivado Design Suite.

A = [3:0];

B = [3,0];

s1 = A’;

y1 = A + B’;

y2 = s1.B’;

y3 = y2 + B’;

## Code:

**library** **IEEE**;

**use** **IEEE.STD\_LOGIC\_1164.ALL**;

**entity** **Problem\_4** **is**

**Port** ( a, b : **in** **STD\_LOGIC\_VECTOR** (**3** **downto** **0**);

y1, y3 : **out** **STD\_LOGIC\_VECTOR** (**3** **downto** **0**);

y2 : **inout** **STD\_LOGIC\_VECTOR** (**3** **downto** **0**));

**end** **Problem\_4**;

**architecture** **Behavioral** **of** **Problem\_4** **is**

**signal** s1: **STD\_LOGIC\_VECTOR**(**3** **downto** **0**);

**begin**

s1 <= **not** a;

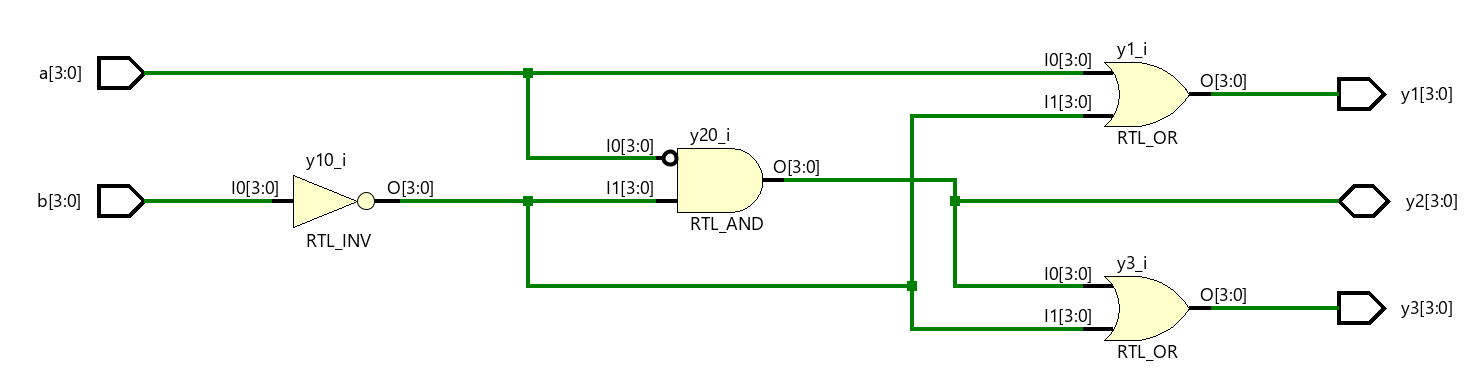
y1 <= a **or** **not** (b);

y2 <= s1 **and** **not** (b);

y3 <= y2 **or** **not** (b);

**end** **Behavioral**;

## RTL Schematic:



## Simulation Result:

# Implement given logical expressions in VHDL for FPGA on Vivado Design Suite.

A = [7:0];

B = [7,0];

C = [3:0]

y1 = A(6:0)&B(7);

y2 = C(3:0)&B(7:4);

y3 = “00”&C(1:0) . a(7:6)&”11”;

## Code:

**library** **IEEE**;

**use** **IEEE.STD\_LOGIC\_1164.ALL**;

**entity** **Problem\_5** **is**

**Port** ( a, b, c : **in** **STD\_LOGIC\_VECTOR** (**7** **downto** **0**):="00000000";

y1, y2 : **out** **STD\_LOGIC\_VECTOR** (**7** **downto** **0**):="00000000";

y3 : **out** **STD\_LOGIC\_VECTOR** (**3** **downto** **0**):="0000");

**end** **Problem\_5**;

**architecture** **Behavioral** **of** **Problem\_5** **is**

**begin**

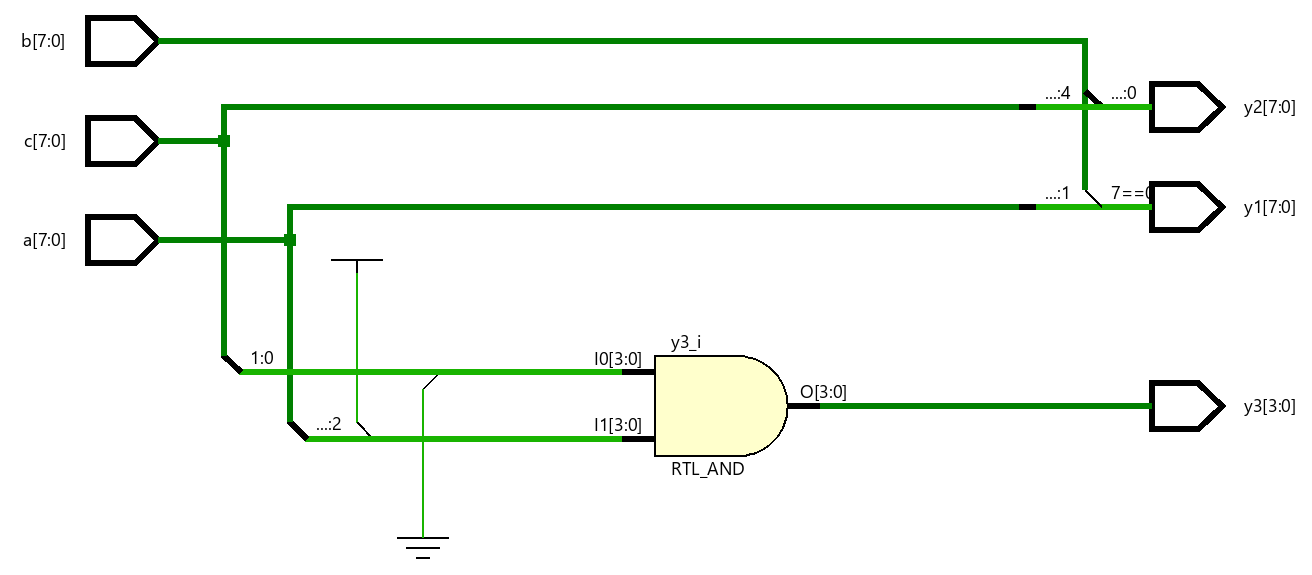
y1 <= a(**6** **downto** **0**) & b(**7**);

y2 <= c(**3** **downto** **0**) & b(**7** **downto** **4**);

y3 <= "00" & c( 1 downto 0) and a(7 downto 6) & "11";

**end** **Behavioral**;

## RTL Schematic:



## Simulation Result:

**Date:** 09 September 2022

# Implement Universal Adder / Subtractor using VHDL for FPGA on Vivado Design Suite.

## Code:

**library** **IEEE**;

**use** **IEEE.STD\_LOGIC\_1164.ALL**;

**use** **IEEE.STD\_LOGIC\_UNSIGNED.ALL**;

**entity** **Universal\_adder\_subtractor** **is**

**Port** ( a, b : **in** **STD\_LOGIC\_VECTOR** (**3** **downto** **0**) := "0000";

m : **in** **STD\_LOGIC**:='0';

br, c : **inout** **STD\_LOGIC**:='0';

y : **out** **STD\_LOGIC\_VECTOR**(**3** **downto** **0**):="0000");

**end** **Universal\_adder\_subtractor**;

**architecture** **Behavioral** **of** **Universal\_adder\_subtractor** **is**

**signal** s1: **STD\_LOGIC\_VECTOR**(**4** **downto** **0**) := "00000";

**signal** s2: **STD\_LOGIC\_VECTOR**(**3** **downto** **0**) := "0000";

**signal** i: **STD\_LOGIC**:='0';

**begin**

s1 <= ('0' & a) + ('0' & (b **xor** (m & m & m & m))) + ("0000" & m);

c <= s1(**4**);

br <= **not** s1(**4**);

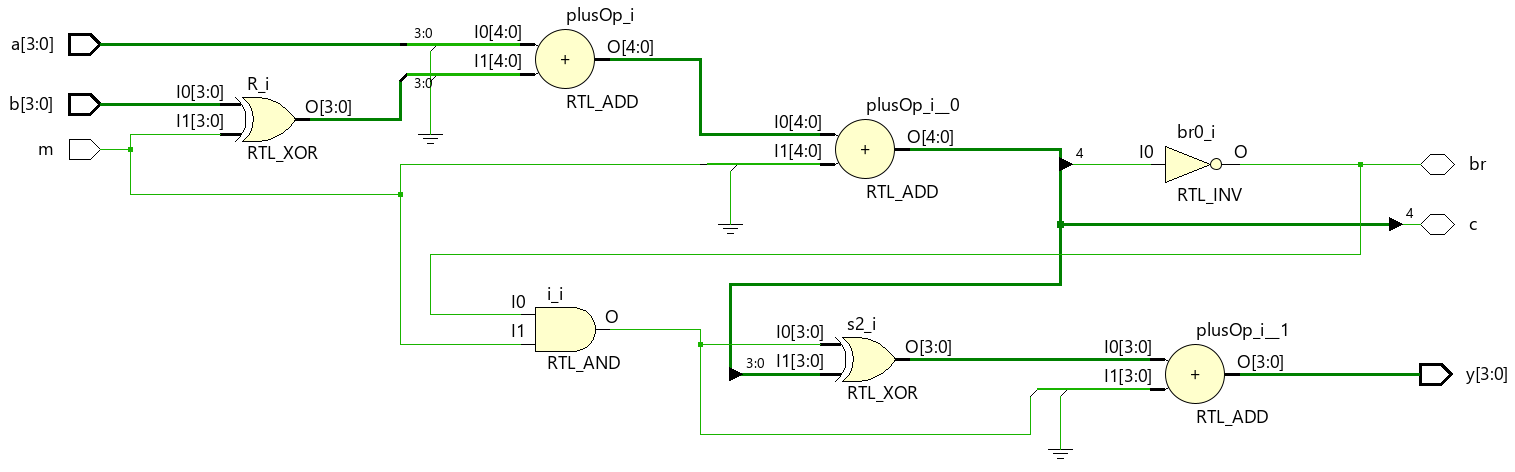
i <= br **and** m;

s2 <= (i & i & i & i) **xor** (s1(**3** **downto** **0**));

y <= s2 + ("000" & i);

**end** **Behavioral**;

## RTL Schematic:



## Simulation Result:

# Implement MUX (4:1) for y = ∑ (M0 + M9 + M11 – M15) using VHDL for FPGA on Vivado Design Suite.

## Code:

**library** **IEEE**;

**use** **IEEE.STD\_LOGIC\_1164.ALL**;

**entity** **Process\_MUX** **is**

**Port** ( a, b, c, d: **in** **STD\_LOGIC** :='0';

y: **out** **STD\_LOGIC** := '0' );

**end** **Process\_MUX**;

**architecture** **Behavioral** **of** **Process\_MUX** **is**

**signal** I0, I1, I2, I3: **STD\_LOGIC**:='0';

**signal** cd: **STD\_LOGIC\_VECTOR**:="00";

**begin**

cd <= c&d;

P0: **Process** (a, b)

**begin**

I0 <= '1';

I1 <= **not** (a) **or** **not** (b);

I2 <= '0';

I3 <= a **and** b;

**end** **Process**;

P1: **Process** (cd, I0, I1, I2, I3)

**begin**

**case** (cd) **is**

**when** "00" => y <= I0;

**when** "01" => y <= I1;

**when** "10" => y <= I2;

**when** "11" => y <= I3;

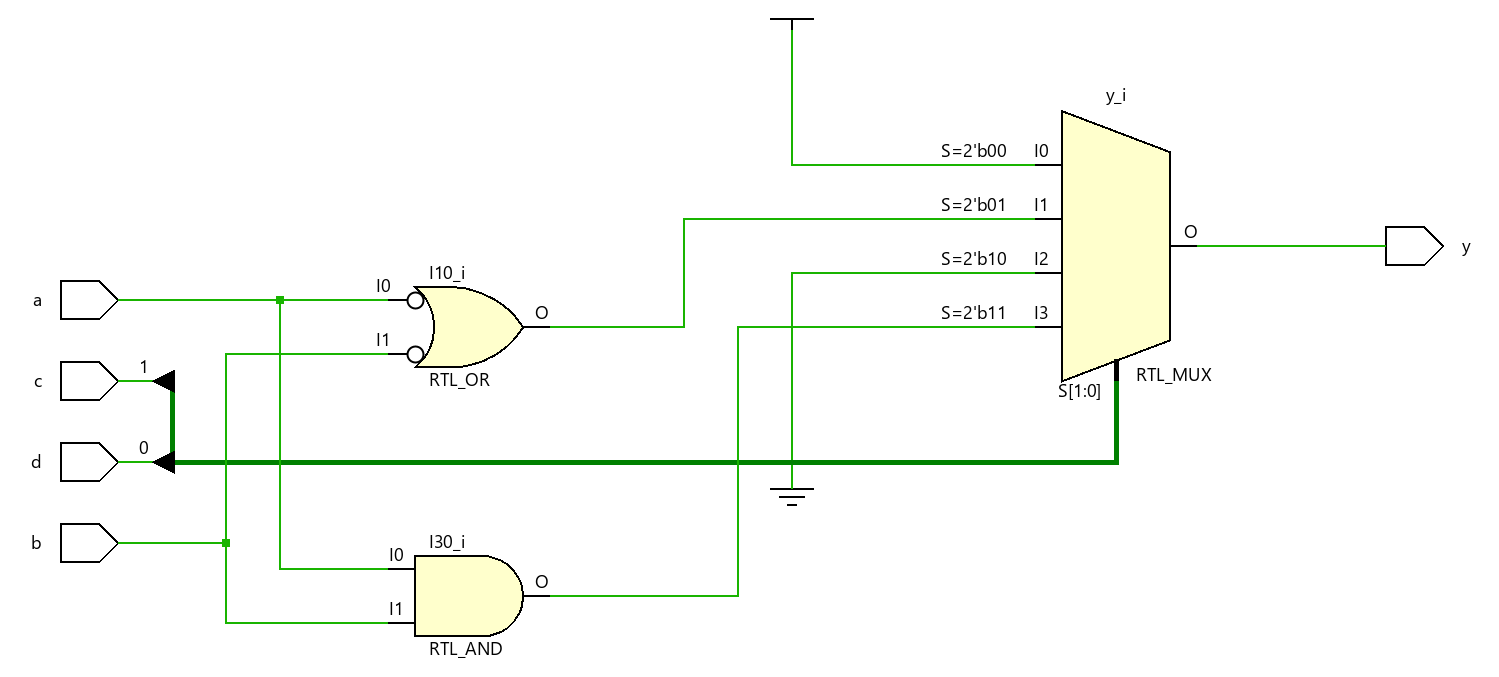
**when** **others** => y <= '0';

**end** **case**;

**end** **Process**;

**end** **Behavioral**;

## RTL Schematic:



## Simulation Result:

# Implement Rising Edge triggered LED pattern along with level triggered control using VHDL for FPGA on Vivado Design Suite. (y1 = “1000 0000”; y2 = “0000 0001”)

Ctrl = 0

Ctrl = 1

y1 = ↓

y1 = ↑

y2 = ↑

y2 = ↓

## Code:

**library** **IEEE**;

**use** **IEEE.STD\_LOGIC\_1164.ALL**;

**entity** **Rising\_Edge\_LED** **is**

**Port** ( a, ctrl : **in** **STD\_LOGIC**:='0';

y1: **inout** **STD\_LOGIC\_VECTOR** (**7** **downto** **0**):="10000000";

y2 : **inout** **STD\_LOGIC\_VECTOR** (**7** **downto** **0**):="00000001");

**end** **Rising\_Edge\_LED**;

**architecture** **Behavioral** **of** **Rising\_Edge\_LED** **is**

**begin**

P0: **Process** (a, ctrl)

**begin**

**if** rising\_edge(a) **then**

**if** ctrl = '0' **then**

y1 <= y1(**0**) & y1(**7** **downto** **1**);

y2 <= y2(**6** **downto** **0**) & y2(**7**);

**else**

y1 <= y1(**6** **downto** **0**) & y1(**7**);

y2 <= y2(**0**) & y2(**7** **downto** **1**);

**end** **if**;

**end** **if**;

**end** **Process**;

**end** **Behavioral**;

## RTL Schematic:

## Simulation Result: