*EE 210 P – Digital Systems and Design Practicum*

*Assignment 1*

*Name : Narmit Kumar*

*Roll No. : B20218*

*Branch : Electrical Engineering*

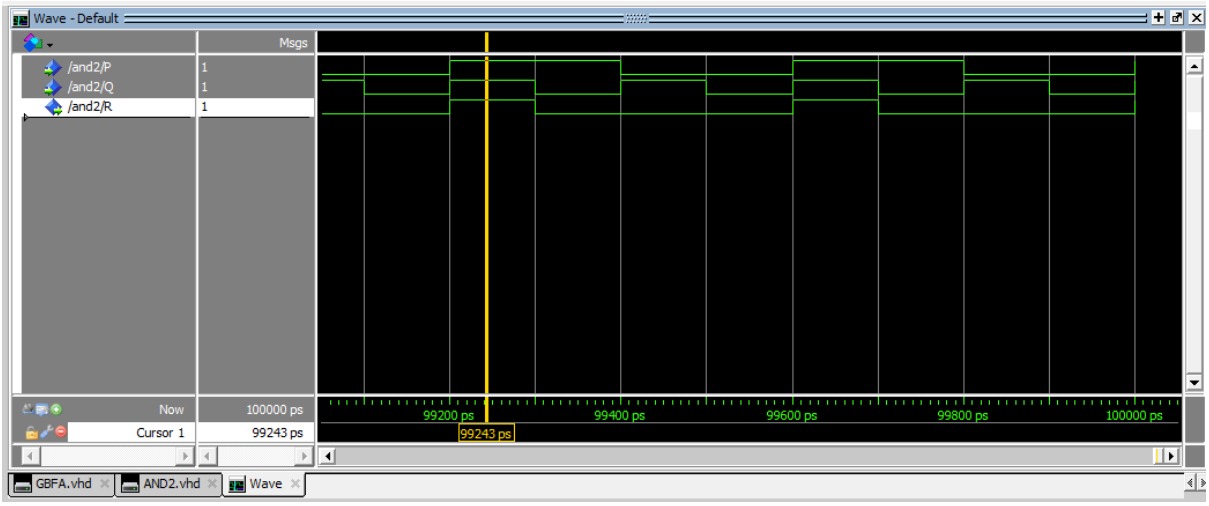
Q1.

Ans. Code for AND2 gate:

C:/intelFPGA/20.1/AND2.vhd (/mult4x8/a4/a7/a1) - Default *

```
Ln#
 1  library IEEE;
 2  use IEEE.std_logic_1164.all;
 3
 4  entity AND2 is
 5   port(P,Q : in std_logic;
 6         R : out std_logic);
 7  end AND2;
 8
 9  architecture df of AND2 is
10   begin
11      R <= P and Q;
12  end df;
13
```

Simulation :



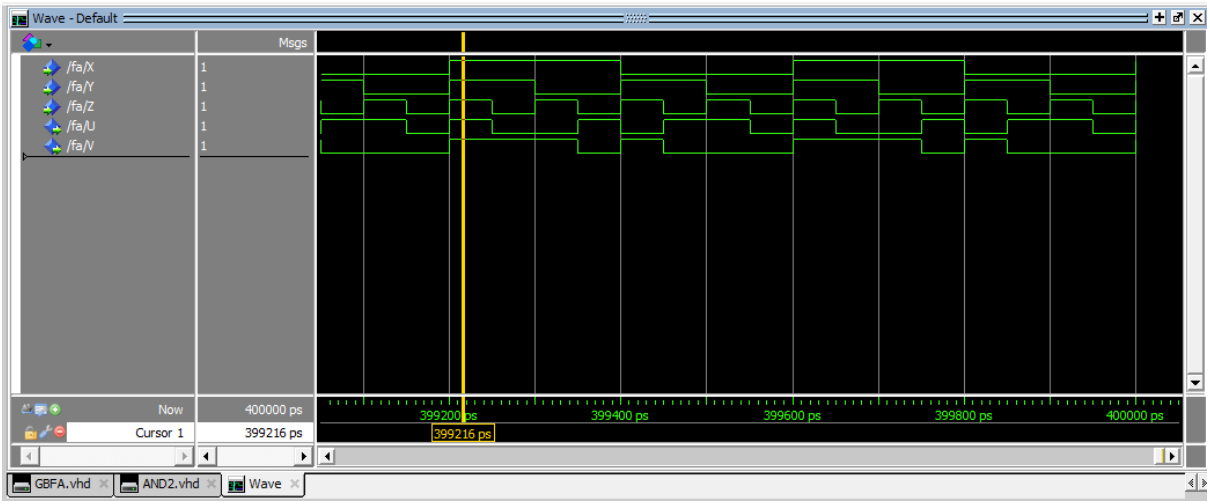

Code for Full Adder circuit:

C:/intelFPGA/20.1/FullAdder.vhd (/mult4x8/a4/a7/a2) - Default

```
Ln#
 1  library IEEE;
 2  use IEEE.std_logic_1164.all;
 3
 4  entity fa is
 5  port(X,Y,Z : in std_logic;
 6       U,V : out std_logic);
 7  end fa;
 8
 9  architecture behavior of fa is
10  begin
11      U <= X xor Y xor Z;
12      V <= (X and Y) or ((X xor Y) and Z);
13  end behavior;
14
```

\

Simulation :

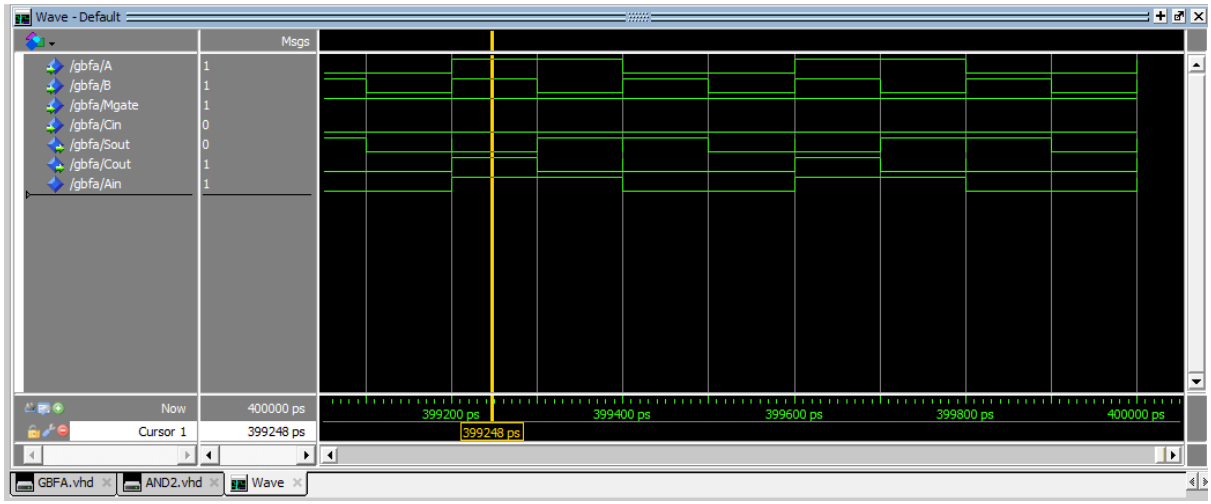


Code for Gated Binary Full Adder (GBFA) :

C:/intelFPGA/20.1/GBFA.vhd (/mult4x8/a4/a7) - Default

```
Ln#
 1  library IEEE;
 2  use IEEE.std_logic_1164.all;
 3
 4  entity GBFA is
 5  port(A,B,Mgate,Cin : in std_logic;
 6        Sout,Cout : out std_logic);
 7  end GBFA;
 8
 9  architecture struct of GBFA is
10  signal Ain: std_logic ;
11  component AND2 is
12    port(P,Q : in std_logic;
13          R : out std_logic);
14  end component;
15  component fa is
16   port( X,Y,Z : in std_logic;
17         U,V : out std_logic);
18  end component;
19  begin
20  a1:AND2 port map(A,Mgate,Ain);
21  a2:fa port map(Ain,B,Cin,Sout,Cout);
22  end struct;
23
```

Simulation:

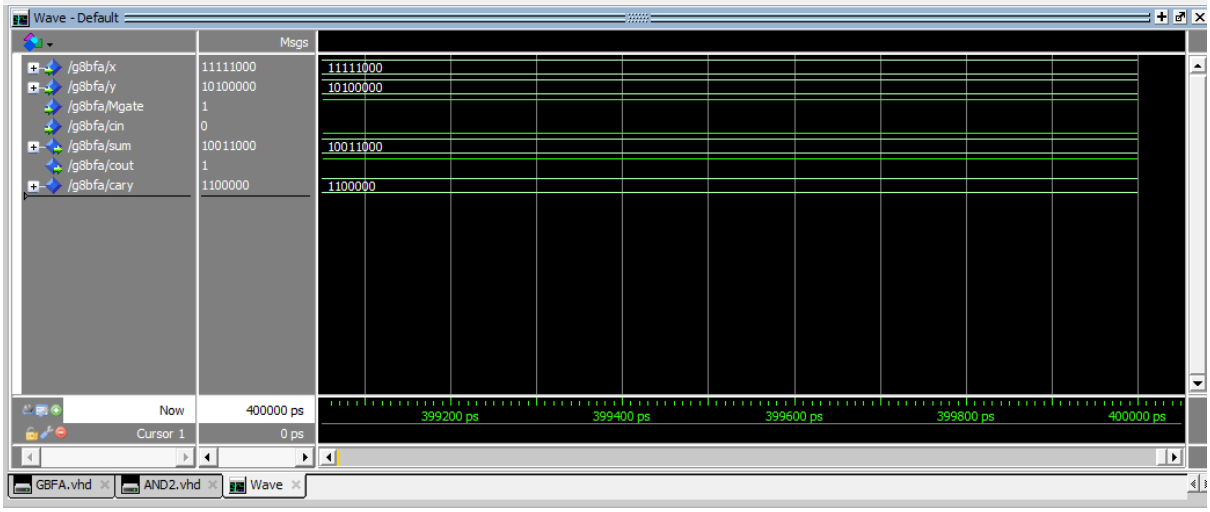


Q2.

Ans. Code for 8-bit Gated binary full adder (G8BFA):

```
Ln#
 1  library ieee;
 2  use ieee.std_logic_1164.all;
 3  entity G8BFA is
 4  port(x,y : in std_logic_vector(7 downto 0);
 5  Mgate : in std_logic;
 6  cin : in std_logic;
 7  sum : out std_logic_vector(7 downto 0);
 8  cout : out std_logic);
 9  end G8BFA;
10
11  architecture data1 of G8BFA is
12  signal cary : std_logic_vector(6 downto 0);
13
14
15  component GBFA is
16  port(A,B,Mgate,Cin : in std_logic;
17       Sout,Cout : out std_logic);
18  end component;
19
20  begin
21  a0 : GBFA port map (x(0),y(0),Mgate, cin,sum(0),cary(0));
22  a1 : GBFA port map (x(1),y(1),Mgate, cary(0),sum(1),cary(1));
23  a2 : GBFA port map (x(2), y(2),Mgate, cary(1), sum(2), cary(2));
24  a3 : GBFA port map (x(3), y(3),Mgate, cary(2), sum(3), cary(3));
25  a4 : GBFA port map (x(4), y(4),Mgate, cary(3), sum(4), cary(4));
26  a5 : GBFA port map (x(5), y(5),Mgate, cary(4), sum(5), cary(5));
27  a6 : GBFA port map (x(6), y(6),Mgate, cary(5), sum(6), cary(6));
28  a7 : GBFA port map (x(7), y(7),Mgate, cary(6), sum(7), cout);
29  end data1;
30
```

Simulation:



Here, we have added two 8- bit binary numbers 11111000 and 10100000 which gives result 10011000 with a carry 1.

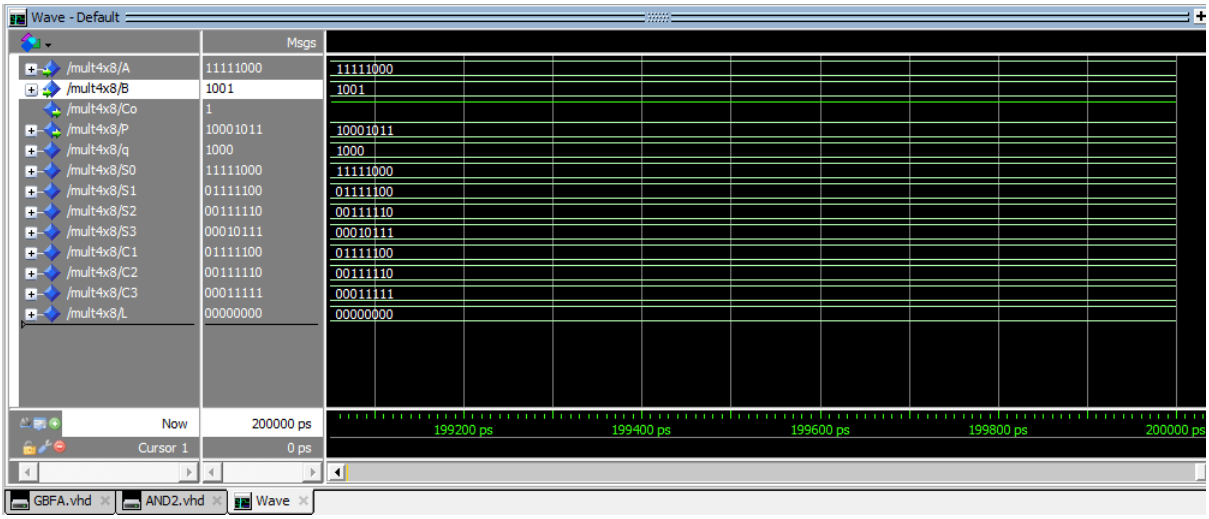
Q3.

Ans. Code :

C:/intelFPGA/20.1/Multiplier.vhd (/mult4x8) - Default

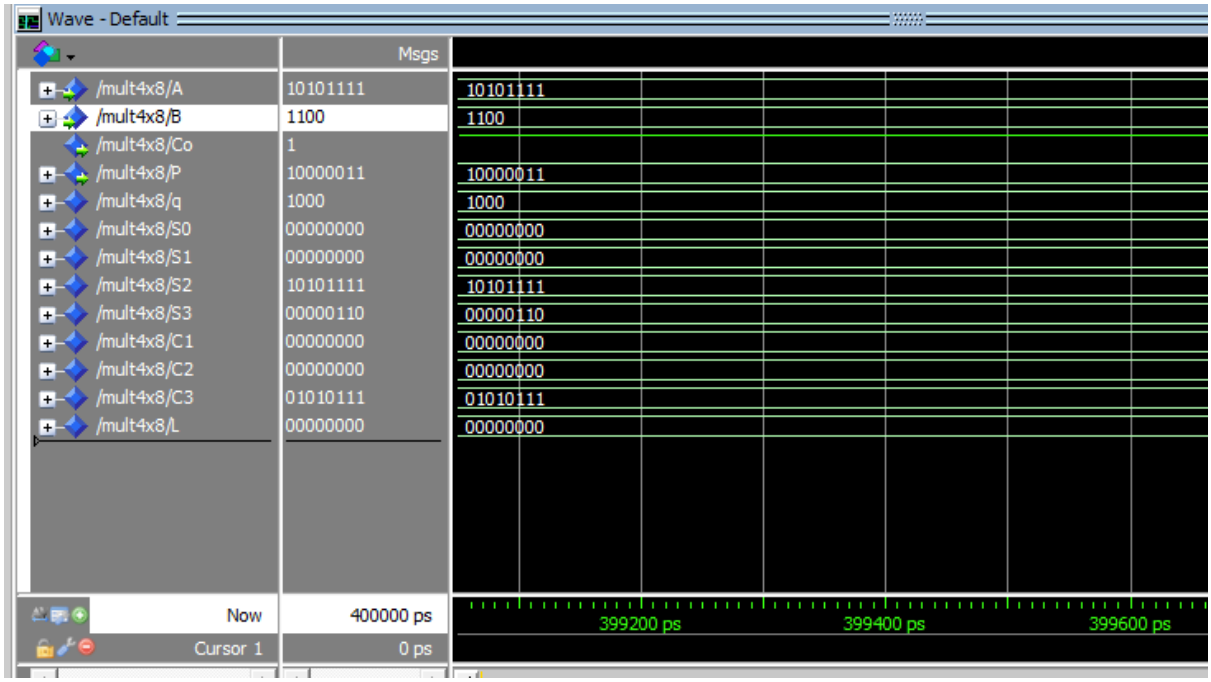
```
Ln#
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity Mult4X8 is
5   port(A: in std_logic_vector(7 downto 0);
6   B : in std_logic_vector(3 downto 0);
7   Co : out std_logic;
8   P:out std_logic_vector(7 downto 0));
9   end Mult4X8;
10
11  architecture way1 of Mult4X8 is
12  signal q: std_logic_vector(3 downto 0);
13  signal S0,S1,S2,S3,C1,C2,C3,L: std_logic_vector(7 downto 0);
14
15
16
17  component G8BFA is
18  port(x,y: in std_logic_vector(7 downto 0);
19  Mgate : in std_logic;
20  cin : in std_logic;
21  sum : out std_logic_vector(7 downto 0);
22  cout : out std_logic);
23  end component;
24
25  begin
26  L <= "00000000";
27  a1: G8BFA port map(A,L,B(0),'0',S0,q(0));
28  C1 <= q(0)&S0 (7 downto 1);
29  a2: G8BFA port map(A,C1,B(1),'0',S1,q(1));
30  C2 <= q(1)&S1 (7 downto 1);
31  a3: G8BFA port map (A,C2,B(2),'0',S2,q(2));
32  C3 <= q(2)&S2 (7 downto 1);
33  a4: G8BFA port map (A,C3,B(3),'0',S3,q(3));
34
35  P <= (q(3) & "0000000") or ("0" & S3(7 downto 1));
36  Co <= q(3);
37  end way1;
38
```

Simulation :



Here , I have multiplied 11111000 with 1001 which gives result 10001011 which is true. Hence , simulation is right.

Another simulation:




Here , I have multiplied 10101111 with 1100 which gives the true result 10000011. Hence it is correct.