

Homework # 2 Solutions

1). XOR implementation with AND-OR-NOT gates:-

Code:-

```
entity xor1 is
port(
  x : in bit;
  y : in bit;
  z : out bit);
end xor1;

architecture xor1_arch of xor1 is

signal int1,int2,int3,int4 : bit;

begin

int1 <= not x after 1 ns;
int2 <= not y after 1 ns;
int3 <= x and int2 after 1 ns;
int4 <= y and int1 after 1 ns;
z  <= int3 or int4 after 1 ns;

end;
```

Simulation results in LIST format:-

ps	delta	/xor1/x	/xor1/y	/xor1/z
0	+0	0	0	0
10000	+0	0	1	0
12000	+0	0	1	1
20000	+0	1	1	1
23000	+0	1	1	0
30000	+0	1	0	0
33000	+0	1	0	1
40000	+0	0	0	1
42000	+0	0	0	0

2). XOR implementation with NAND gates:-**Code:-**

```

entity xor2 is
port(
  x : in bit;
  y : in bit;
  z : out bit);
end xor2;

architecture xor2_arch of xor2 is

signal int1,int2,int3 : bit;

begin

int1 <= x nand y after 1 ns;
int2 <= x nand int1 after 1 ns;
int3 <= y nand int1 after 1 ns;
z  <= int2 nand int3 after 1 ns;

end;

```

Simulation results in LIST format:-

ps	delta	/xor2/x	/xor2/y	/xor2/z
0	+0	1	0	0
1000	+0	1	0	1
2000	+0	1	0	0
3000	+0	1	0	1
10000	+0	1	1	1
13000	+0	1	1	0
20000	+0	0	1	0
23000	+0	0	1	1
30000	+0	0	0	1
32000	+0	0	0	0

The above solutions are using the behavioral modeling in VHDL. However, implementation can be done using the structural modeling. An example of this is given below. Structural modeling is preferred as any synthesis tool would implement the same hardware structure (as designed) for a given structural model (except for the lowest level in the hierarchy as this level uses behavioral modeling). There is no guarantee that two synthesis tools will implement the same hardware for a given behavioral model.

Example of structural modeling:-

```
entity nand2 is
port(
    x : in bit;
    y : in bit;
    z : out bit);
end nand2;
```

```
architecture nand2_arch of nand2 is
```

```
begin
```

```
z <= x nand y after 1ns;
```

```
end;
```

```
-----
entity xor2 is
port(
    x : in bit;
    y : in bit;
    z : out bit);
end xor2;
```

```
architecture xor2_arch of xor2 is
```

```
component nand2
port(
    x : in bit;
    y : in bit;
    z : out bit);
end component;
```

```
signal int1,int2,int3 : bit;
```

```
begin
```

```
N1 : nand2 port map(x,y,int1);
N2 : nand2 port map(x,int1,int2);
N3 : nand2 port map(y,int1,int3);
N4 : nand2 port map(int2,int3,z);
```

```
end;
```