AND GATE

VHDL Code:

```
Library ieee;
use ieee.std_logic_1164.all;

entity and1 is
  port(x,y:in bit ; z:out bit);
end and1;

architecture virat of and1 is
begin
  z<=x and y;
end virat;
```

OR Gate

VHDL Code:

```
Library ieee;
use ieee.std_logic_1164.all;

entity or1 is
  port(x,y:in bit ; z:out bit);
end or1;

architecture virat of or1 is
begin
  z<=x or y;
end virat;
```

NOT Gate

VHDL Code:

```
Library ieee;
use ieee.std_logic_1164.all;

entity not1 is
  port(x:in bit ; y:out bit);
end not1;
```

```
architecture virat of not1 is
begin
   y<=not x;
end virat;
```

NAND Gate

VHDL Code:

```
Library ieee;
use ieee.std_logic_1164.all;

entity nand1 is
   port(a,b:in bit ; c:out bit);
end nand1;

architecture virat of nand1 is
begin
   c<=a nand b;
end virat;
```

NOR Gate

VHDL Code:

```
Library ieee;
use ieee.std_logic_1164.all;

entity nor1 is
   port(a,b:in bit ; c:out bit);
end nor1;

architecture virat of nor1 is
begin
   c<=a nor b;
end virat;
```

XOR Gate

VHDL Code:

```
Library ieee;
use ieee.std_logic_1164.all;

entity xor1 is
  port(a,b:in bit ; c:out bit);
end xor1;

architecture virat of xor1 is
begin
  c<=a xor b;
end virat;
```

X-NOR Gate

VHDL Code:

```
Library ieee;
use ieee.std_logic_1164.all;

entity xnor1 is
  port(a,b:in bit ; c:out bit);
end xnor1;

architecture virat of xnor1 is
begin
  c<=not(a xor b);
end virat;
```