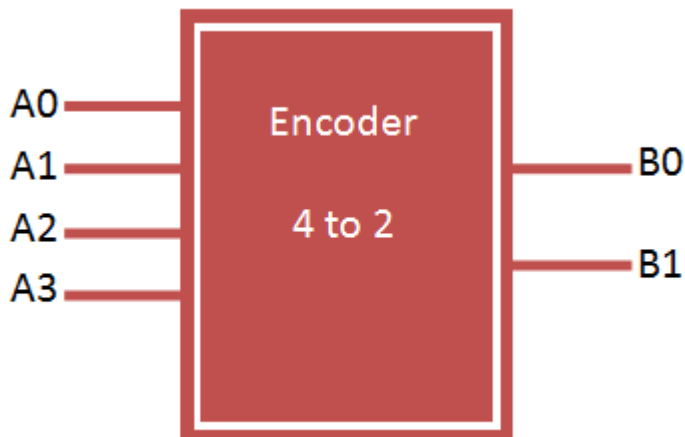


Contents

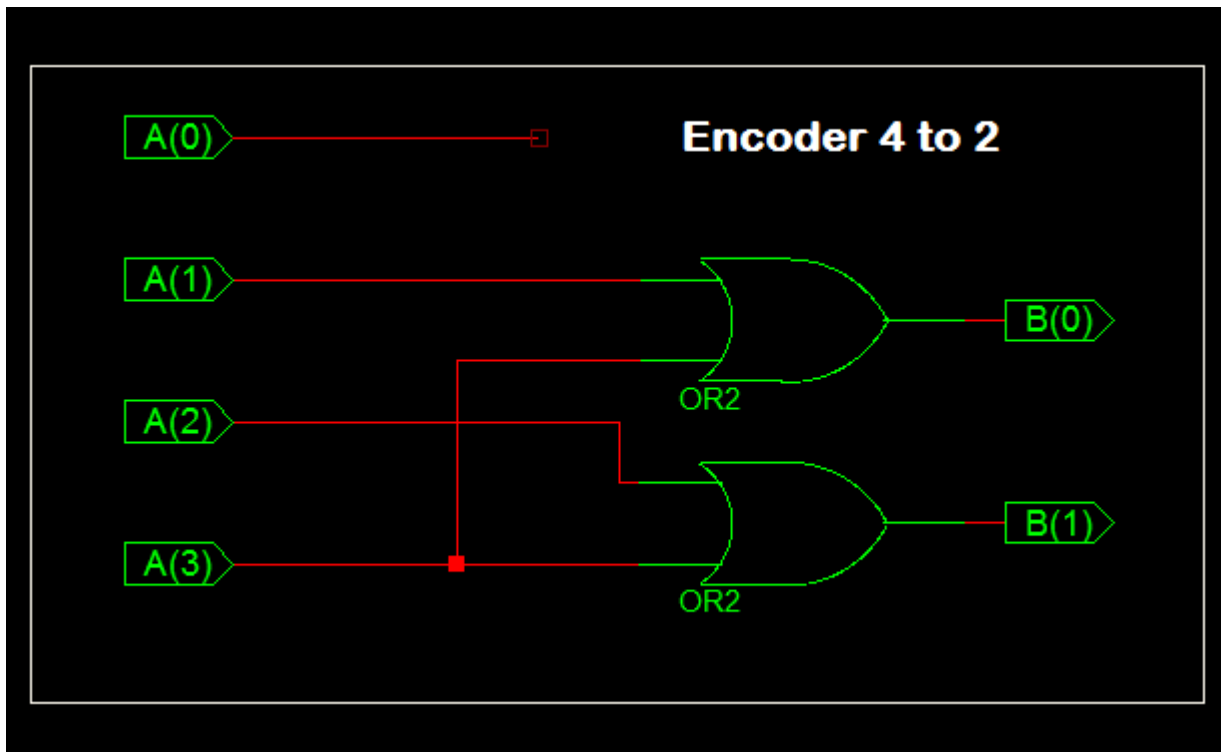
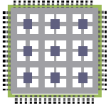
- 1 Binary Encoder
- 2 4 to 2 encoder design using logic gates
- 3 Truth Table for 4 to 2 encoder
- 4 VHDL Code for 4 to 2 encoder using case statement
- 5 VHDL Code for 4 to 2 encoder using if else statement
- 6 VHDL Code for 4 to 2 encoder using logic gates
- 7 TestBench VHDL Code for 4 to 2 encoder
- 8 TestBench wave for 4 to 2 encoder

Binary Encoder

Binary encoder has 2^n input lines and n -bit output lines. It can be 4-to-2, 8-to-3 and 16-to-4 line configurations. VHDL Code for 4 to 2 encoder can be designed both in structural and behavioral modelling.



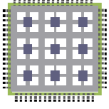
4 to 2 encoder design using logic gates



Truth Table for 4 to 2 encoder

INPUT				OUTPUT	
A3	A2	A1	A0	B1	B0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

VHDL Code for 4 to 2 encoder can be done in different methods like using case statement, using if else statement, using logic gates etc. Here we provide example code for all 3 method for better understanding of the language.



VHDL Code for 4 to 2 encoder using case statement

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity encoder is
  port(
    a : in STD_LOGIC_VECTOR(3 downto 0);
    b : out STD_LOGIC_VECTOR(1 downto 0)
  );
end encoder;

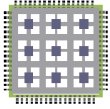
architecture bhv of encoder is
begin

  process(a)
  begin
    case a is
      when "1000" => b <= "00";
      when "0100" => b <= "01";
      when "0010" => b <= "10";
      when "0001" => b <= "11";
      when others => b <= "ZZ";
    end case;
  end process;

end bhv;
```

VHDL Code for 4 to 2 encoder using if else statement

```
library IEEE;
```



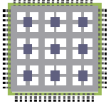
```
use IEEE.STD_LOGIC_1164.all;

entity encoder1 is
  port(
    a : in STD_LOGIC_VECTOR(3 downto 0);
    b : out STD_LOGIC_VECTOR(1 downto 0)
  );
end encoder1;

architecture bhv of encoder1 is
begin

  process(a)
  begin
    if (a="1000") then
      b <= "00";
    elsif (a="0100") then
      b <= "01";
    elsif (a="0010") then
      b <= "10";
    elsif (a="0001") then
      b <= "11";
    else
      b <= "ZZ";
    end if;
  end process;

end bhv;
```



VHDL Code for 4 to 2 encoder using logic gates

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity encoder2 is
  port(
    a : in STD_LOGIC_VECTOR(3 downto 0);
    b : out STD_LOGIC_VECTOR(1 downto 0)
  );
end encoder2;

architecture bhv of encoder2 is
begin

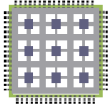
  b(0) <= a(1) or a(2);
  b(1) <= a(1) or a(3);

end bhv;
```

TestBench VHDL Code for 4 to 2 encoder

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY tb_encoder IS
END tb_encoder;
```



```
ARCHITECTURE behavior OF tb_encoder IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT encoder
PORT(
a : IN std_logic_vector(3 downto 0);
b : OUT std_logic_vector(1 downto 0)
);
END COMPONENT;

--Inputs
signal a : std_logic_vector(3 downto 0) := (others => '0');

--Outputs
signal b : std_logic_vector(1 downto 0);

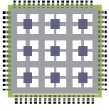
BEGIN

-- Instantiate the Unit Under Test (UUT)
 uut: encoder PORT MAP (
  a => a,
  b => b
 );

-- Stimulus process
 stim_proc: process
 begin
  -- hold reset state for 100 ns.
  wait for 100 ns;

  a <= "0000";

  wait for 100 ns;
```



```
a <= "0001";  
  
wait for 100 ns;  
  
a <= "0010";  
  
wait for 100 ns;  
  
a <= "0100";  
  
wait for 100 ns;  
  
a <= "1000";  
  
wait;  
end process;  
  
END;
```

TestBench wave for 4 to 2 encoder

