1.

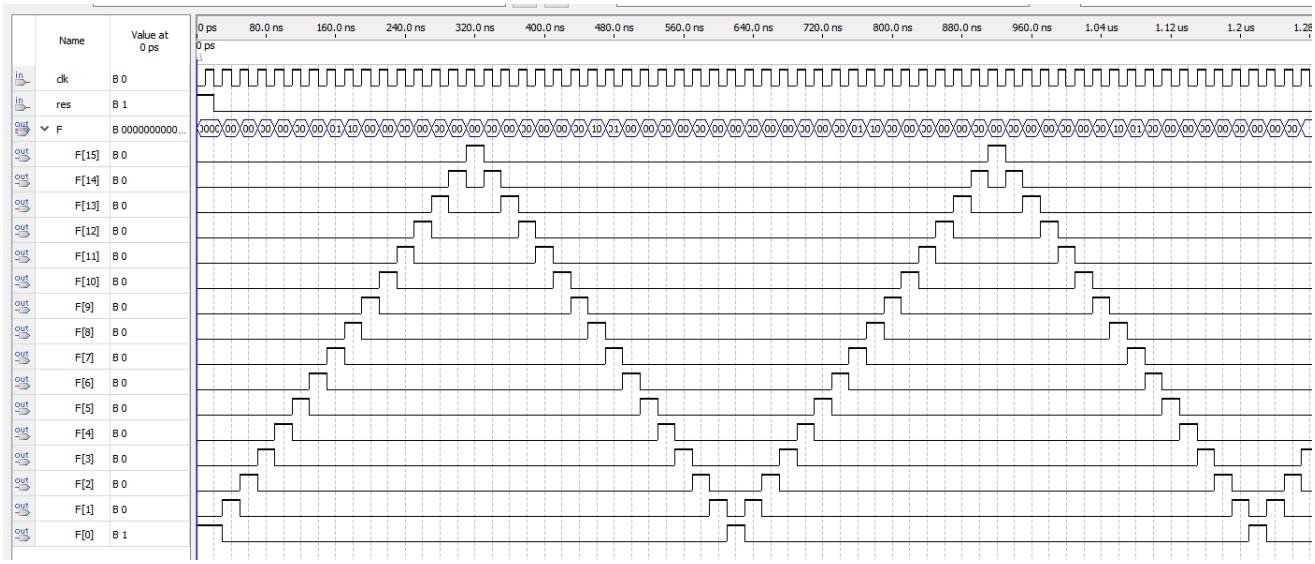
Code:

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3   use ieee.std_logic_unsigned.all;
4
5   entity cnt is
6      port(
7         clk, res: in std_logic;
8         Q: out std_logic_vector(3 downto 0)
9      );
10  end cnt;
11
12  architecture cnt of cnt is
13     signal nowt: std_logic_vector(3 downto 0);
14     signal m: std_logic;
15     begin
16        process(clk, res)
17        begin
18           if res = '1' then
19              m <= '1'; nowt <= "0000";
20           elsif clk'event and clk = '1' then
21              if nowt = "1111" and m = '1' then
22                 nowt <= nowt - 1;
23                 m <= '0';
24              elsif nowt = "0000" and m = '0' then
25                 nowt <= nowt + 1;
26                 m <= '1';
27              elsif m = '1' then nowt <= nowt + 1;
28              elsif m = '0' then nowt <= nowt - 1;
29              end if;
30           end if;
31        end process;
32        Q <= nowt;
33     end cnt;
34
```

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity decoder is
    port(
        input: in std_logic_vector(3 downto
        F: out std_logic_vector(15 downto 0)
    );
end decoder;

architecture decoder of decoder is
begin

    F <= x"0001" when input = x"0" else
         x"0002" when input = x"1" else
         x"0004" when input = x"2" else
         x"0008" when input = x"3" else
         x"0010" when input = x"4" else
         x"0020" when input = x"5" else
         x"0040" when input = x"6" else
         x"0080" when input = x"7" else
         x"0100" when input = x"8" else
         x"0200" when input = x"9" else
         x"0400" when input = "1010" else
         x"0800" when input = "1011" else
         x"1000" when input = "1100" else
         x"2000" when input = "1101" else
         x"4000" when input = "1110" else
         x"8000" when input = "1111" else
         x"0000";

end decoder;
```

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity cnt_decoder is
    port(
        clk, res: in std_logic;
        F: out std_logic_vector(15 downto 0)
    );
end cnt_decoder;

architecture cnt_decoder of cnt_decoder is
    signal cntout: std_logic_vector(3 downto 0);
    component cnt is
        port(
            clk, res: in std_logic;
            Q: out std_logic_vector(3 downto 0)
        );
    end component;

    component decoder is
        port(
            input: in std_logic_vector(3 downto 0);
            F: out std_logic_vector(15 downto 0)
        );
    end component;

begin
    U1: cnt port map(clk, res, cntout);
    U2: decoder port map(cntout, F(15 downto 0));
end cnt_decoder;
```

模擬:


Name
Value at 0 ps
0 ps
80.0 ns
160.0 ns
240.0 ns
320.0 ns
400.0 ns
480.0 ns
560.0 ns
640.0 ns
720.0 ns
800.0 ns
880.0 ns
960.0 ns
1.04 us
1.12 us
1.2 us
clk
B 0
res
B 1
F
B 0000000000...
F[15]
F[14]
F[13]
F[12]
F[11]
F[10]
F[9]
F[8]
F[7]
F[6]
F[5]
F[4]
F[3]
F[2]
F[1]
F[0]
B 1

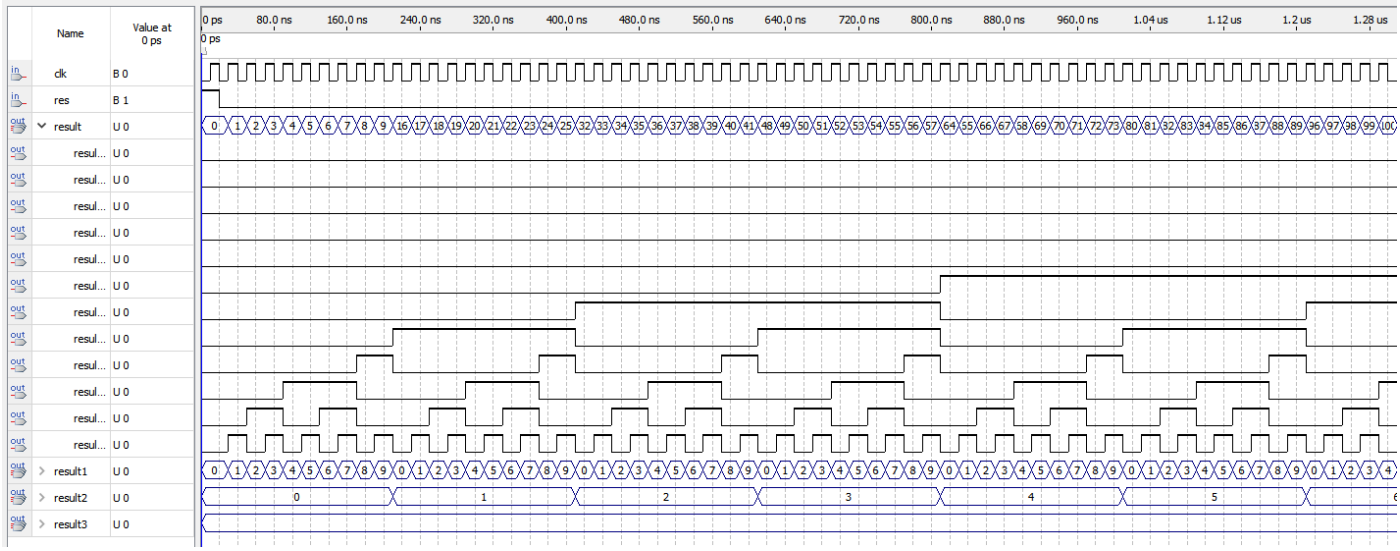
2.

Code:

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3   use ieee.std_logic_unsigned.all;
4
5   entity BCD is
6      port(
7         clk, res, Cin: in std_logic;
8         Q: out std_logic_vector(3 downto 0);
9         Cout: out std_logic
10     );
11  end BCD;
12
13  architecture BCD of BCD is
14     signal nowt: std_logic_vector(3 downto 0);
15  begin
16     process(clk, res, Cin)
17     begin
18        if res = '1' then nowt <= "0000";
19        elsif clk'event and clk = '1' and Cin = '1' then
20           if nowt = "1001" then
21              nowt <= "0000";
22           else nowt <= nowt + 1;
23           end if;
24        end if;
25     end process;
26     Q <= nowt;
27     Cout <= '1' when nowt = "1001" and Cin = '1' else '0';
28  end BCD;
```

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity BCD_counter is
    port(
        clk, res: in std_logic;
        result1, result2, result3: out std_logic_vector(3 downto 0);
        result: out std_logic_vector(11 downto 0)
    );
end BCD_counter;

architecture BCD_counter of BCD_counter is
    component BCD is
        port(
            clk, res, Cin: in std_logic;
            Q: out std_logic_vector(3 downto 0);
            Cout: out std_logic
        );
    end component;
    signal sum1, sum2, sum3: std_logic_vector(3 downto 0);
    signal Ci, Co: std_logic_vector(2 downto 0);
begin
    Ci(0) <= '1';
    Ci(1) <= Co(0);
    Ci(2) <= Co(1);

    U1: BCD port map(clk, res, Ci(0), sum1, Co(0));
    U2: BCD port map(clk, res, Ci(1), sum2, Co(1));
    U3: BCD port map(clk, res, Ci(2), sum3, Co(2));

    result1 <= sum1;
    result2 <= sum2;
    result3 <= sum3;
    result <= sum3 & sum2 & sum1;

end BCD_counter;
```

模擬:


Name
Value at 0 ps
clk
B 0
res
B 1
result
U 0
result1
result2
result3
0 ps
80.0 ns
160.0 ns
240.0 ns
320.0 ns
400.0 ns
480.0 ns
560.0 ns
640.0 ns
720.0 ns
800.0 ns
880.0 ns
960.0 ns
1.04 us
1.12 us
1.2 us
1.28 us