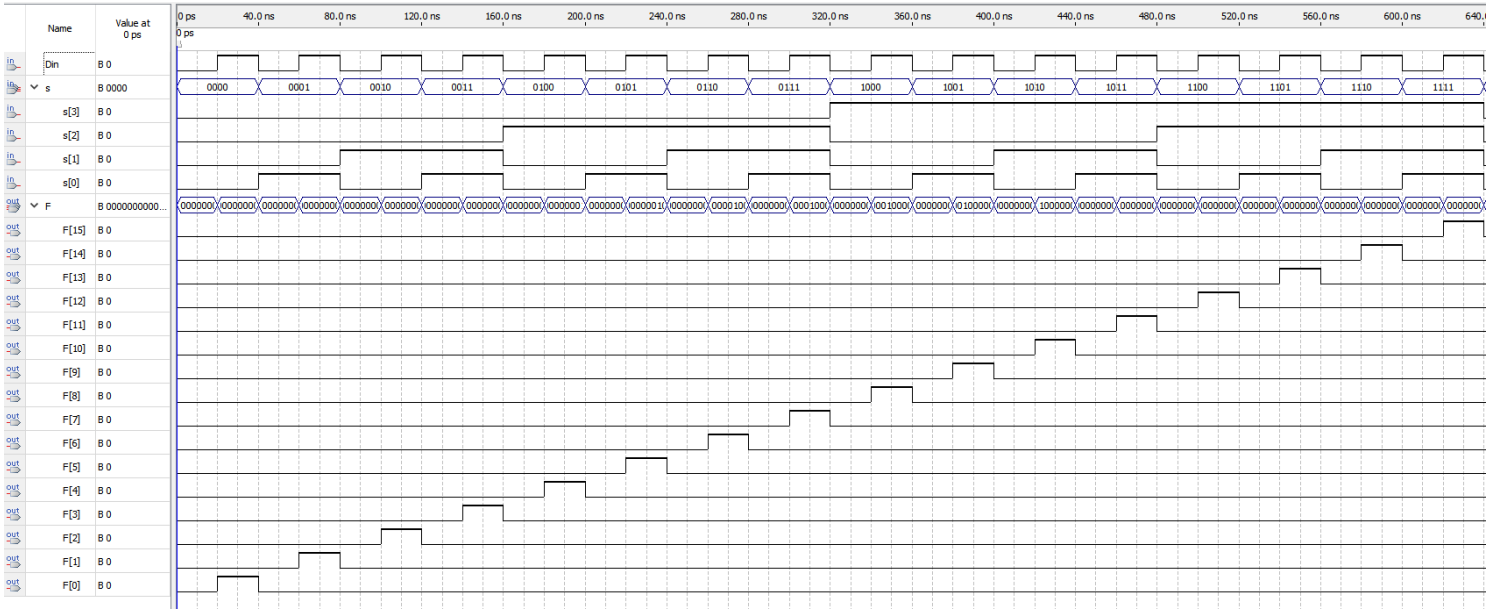
1.

程式碼:

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

entity mux is
    port(
        s: in std_logic_vector(3 downto 0);
        Din: in std_logic;
        F: out std_logic_vector(15 downto 0)
    );
end mux;

architecture mux of mux is
    signal D: std_logic_vector(3 downto 0);
    signal mux1, mux2, mux3, mux4: std_logic_vector(3 downto 0);

begin
    D(0) <= Din when s(3) = '0' and s(2) = '0' else '0';
    D(1) <= Din when s(3) = '0' and s(2) = '1' else '0';
    D(2) <= Din when s(3) = '1' and s(2) = '0' else '0';
    D(3) <= Din when s(3) = '1' and s(2) = '1' else '0';

    mux1 <= "0000" when D(0) = '0' else
            "0001" when s(1) = '0' and s(0) = '0' else
            "0010" when s(1) = '0' and s(0) = '1' else
            "0100" when s(1) = '1' and s(0) = '0' else
            "1000" when s(1) = '1' and s(0) = '1';

    mux2 <= "0000" when D(1) = '0' else
            "0001" when s(1) = '0' and s(0) = '0' else
            "0010" when s(1) = '0' and s(0) = '1' else
            "0100" when s(1) = '1' and s(0) = '0' else
            "1000" when s(1) = '1' and s(0) = '1';

    mux3 <= "0000" when D(2) = '0' else
            "0001" when s(1) = '0' and s(0) = '0' else
            "0010" when s(1) = '0' and s(0) = '1' else
            "0100" when s(1) = '1' and s(0) = '0' else
            "1000" when s(1) = '1' and s(0) = '1';

    mux4 <= "0000" when D(3) = '0' else
            "0001" when s(1) = '0' and s(0) = '0' else
            "0010" when s(1) = '0' and s(0) = '1' else
            "0100" when s(1) = '1' and s(0) = '0' else
            "1000" when s(1) = '1' and s(0) = '1';
    F <= mux4 & mux3 & mux2 & mux1;

end mux;
```

模擬結果:

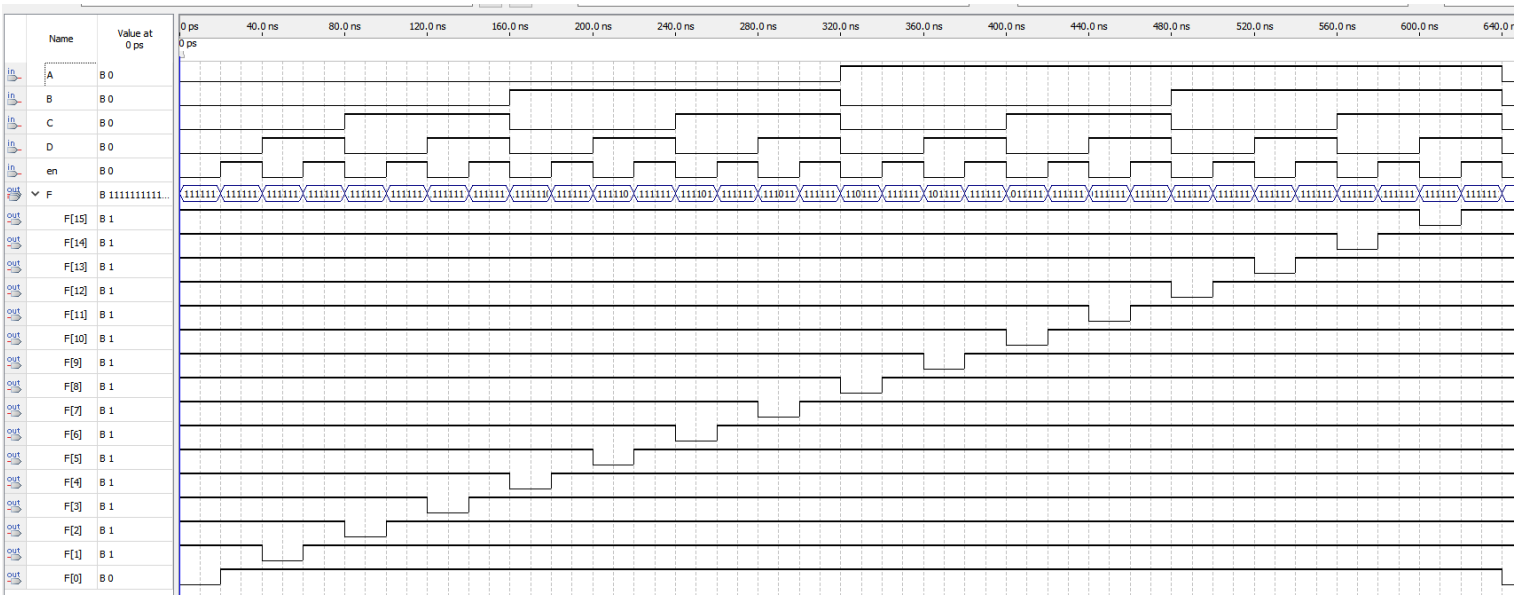

Name
Value at 0 ps
Din
B 0
s
B 0000
s[3]
s[2]
s[1]
s[0]
F
F[15]
F[14]
F[13]
F[12]
F[11]
F[10]
F[9]
F[8]
F[7]
F[6]
F[5]
F[4]
F[3]
F[2]
F[1]
F[0]
0 ps
40.0 ns
80.0 ns
120.0 ns
160.0 ns
200.0 ns
240.0 ns
280.0 ns
320.0 ns
360.0 ns
400.0 ns
440.0 ns
480.0 ns
520.0 ns
560.0 ns
600.0 ns
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

2.

程式碼:

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity decoder2to4_en is
5     port(
6        A, B: in std_logic;
7        en: in std_logic;
8        C, D: in std_logic;
9        F: out std_logic_vector(15 downto 0)
10    );
11 end decoder2to4_en;
12
13 architecture decoder2to4_en of decoder2to4_en is
14    signal AB: std_logic_vector(1 downto 0);
15    signal DD: std_logic_vector(3 downto 0);
16    signal tmp1, tmp2, tmp3, tmp4: std_logic_vector(3 downto 0);
17    signal CD: std_logic_vector(1 downto 0);
18
19 begin
20    AB <= A & B;
21    CD <= C & D;
22
23    DD <= "1111" when en = '1' else
24          "1110" when AB = "00" else
25          "1101" when AB = "01" else
26          "1011" when AB = "10" else
27          "0111";
28
29    tmp1 <= "1111" when DD(3) = '1' else
30            "1110" when CD = "00" else
31            "1101" when CD = "01" else
32            "1011" when CD = "10" else
33            "0111";
34
35    tmp2 <= "1111" when DD(2) = '1' else
36            "1110" when CD = "00" else
37            "1101" when CD = "01" else
38            "1011" when CD = "10" else
39            "0111";
40
41    tmp3 <= "1111" when DD(1) = '1' else
42            "1110" when CD = "00" else
43            "1101" when CD = "01" else
44            "1011" when CD = "10" else
45            "0111";
46
47    tmp4 <= "1111" when DD(0) = '1' else
48            "1110" when CD = "00" else
49            "1101" when CD = "01" else
50            "1011" when CD = "10" else
51            "0111";
52
53    F <= tmp1 & tmp2 & tmp3 & tmp4;
54
55 end decoder2to4_en;
```

模擬結果:

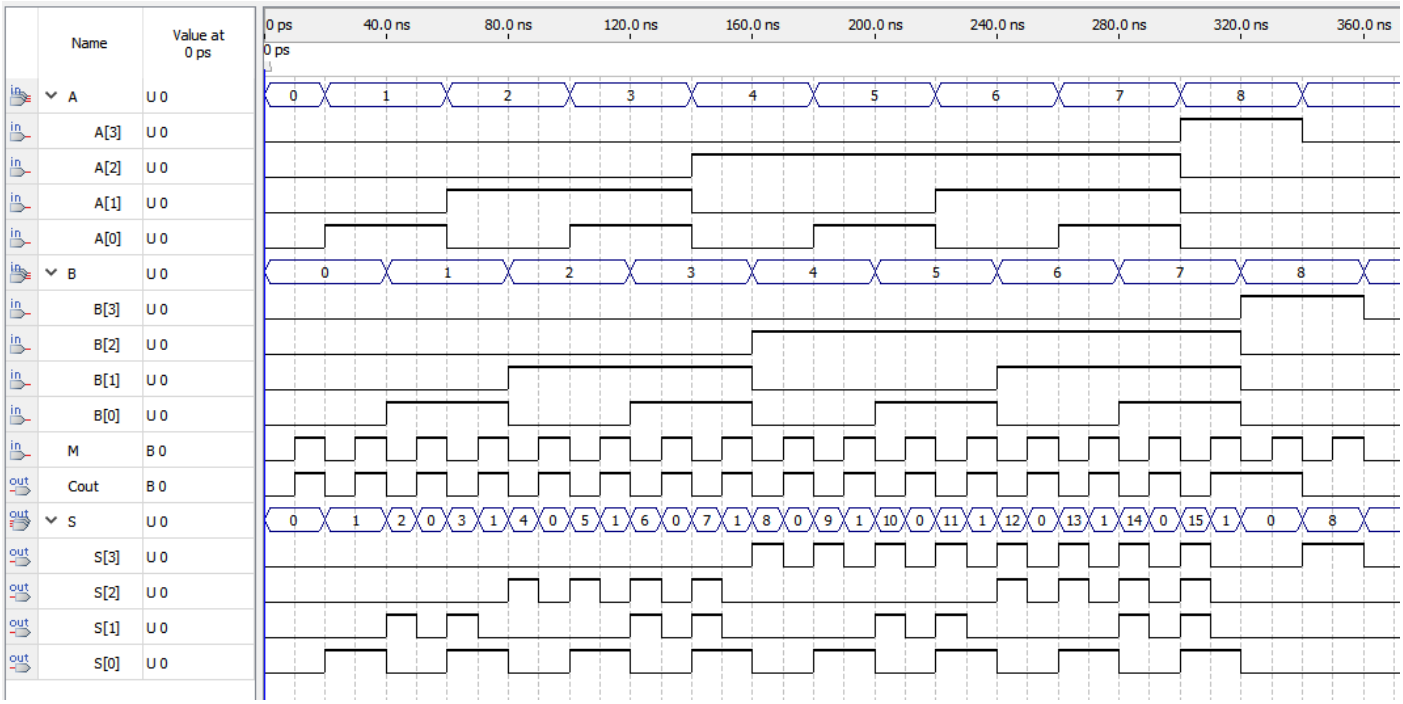

Name
Value at 0 ps
A
B
C
D
en
F
F[15]
F[14]
F[13]
F[12]
F[11]
F[10]
F[9]
F[8]
F[7]
F[6]
F[5]
F[4]
F[3]
F[2]
F[1]
F[0]
0 ps
40.0 ns
80.0 ns
120.0 ns
160.0 ns
200.0 ns
240.0 ns
280.0 ns
320.0 ns
360.0 ns
400.0 ns
440.0 ns
480.0 ns
520.0 ns
560.0 ns
600.0 ns

3.

程式碼:

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

entity FA_4bit is
  port (
    A, B : in std_logic_vector(3 downto 0);
    M : in std_logic;
    S : out std_logic_vector(3 downto 0);
    Cout : out std_logic
  );
end entity;

architecture FA_4bit of FA_4bit is
  signal C1, C2, C3 : std_logic;
  signal tB : std_logic_vector(3 downto 0);
begin
  tB(0) <= B(0) xor M;
  tB(1) <= B(1) xor M;
  tB(2) <= B(2) xor M;
  tB(3) <= B(3) xor M;

  S(0) <= A(0) xor tB(0) xor M;
  S(1) <= A(1) xor tB(1) xor C1;
  S(2) <= A(2) xor tB(2) xor C2;
  S(3) <= A(3) xor tB(3) xor C3;

  C1 <= ((A(0) xor tB(0)) and M) or (A(0) and tB(0));
  C2 <= ((A(1) xor tB(1)) and C1) or (A(1) and tB(1));
  C3 <= ((A(2) xor tB(2)) and C2) or (A(2) and tB(2));
  Cout <= ((A(3) xor tB(3)) and C3) or (A(3) and tB(3));
end architecture;
```

模擬結果:


Name
Value at 0 ps
A
A[3]
A[2]
A[1]
A[0]
B
B[3]
B[2]
B[1]
B[0]
M
Cout
S
S[3]
S[2]
S[1]
S[0]
0 ps
40.0 ns
80.0 ns
120.0 ns
160.0 ns
200.0 ns
240.0 ns
280.0 ns
320.0 ns
360.0 ns