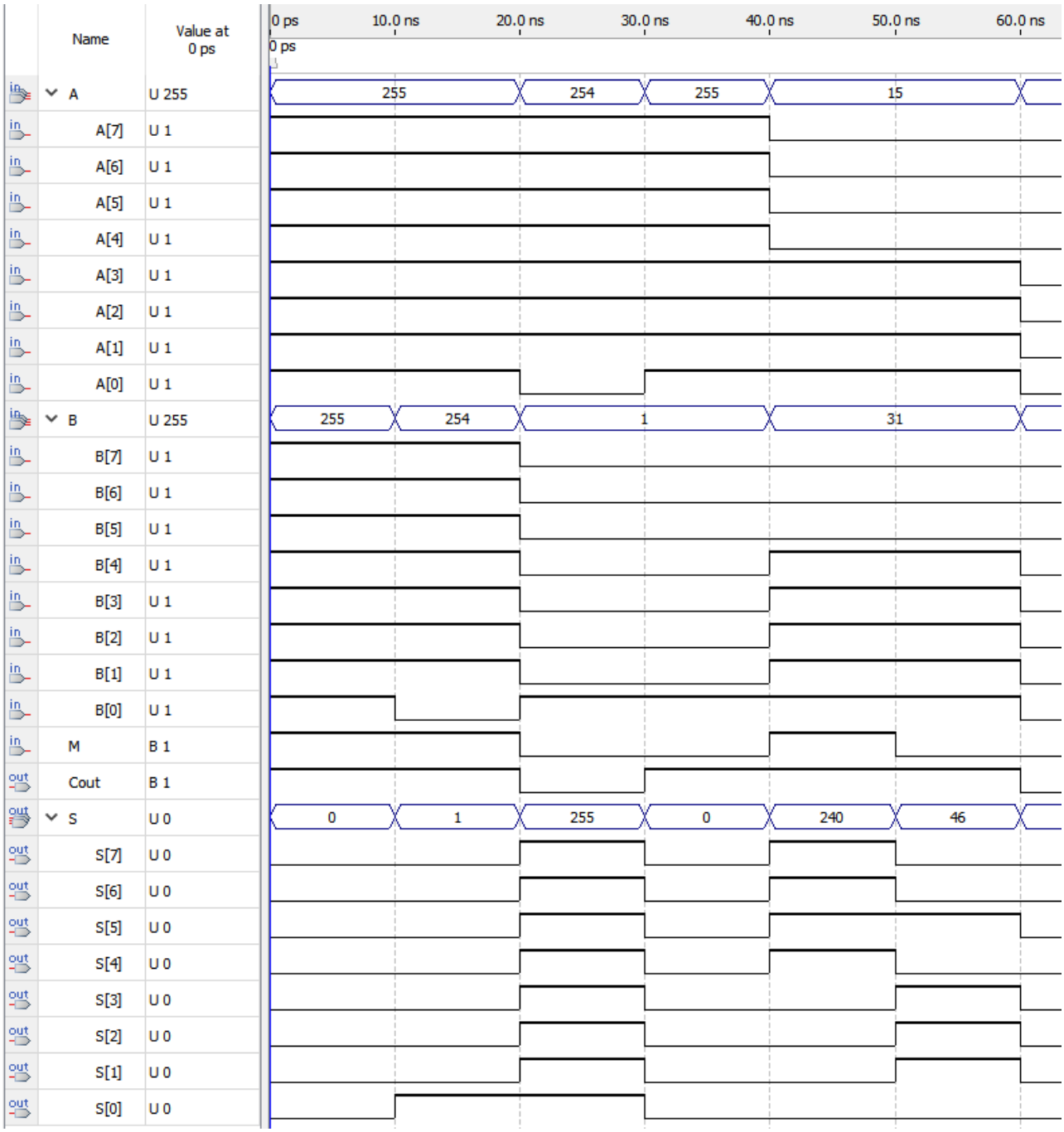
1.

Code:

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity FA_8bits is
5     port (
6       A, B : in std_logic_vector(7 downto 0);
7       M : in std_logic;
8       S : out std_logic_vector(7 downto 0);
9       Cout : out std_logic
10    );
11  end FA_8bits;
12
13  architecture FA_8bits of FA_8bits is
14    signal C1, C2, C3, C4, C5, C6, C7: std_logic;
15    signal tB : std_logic_vector(7 downto 0);
16  begin
17    tB(0) <= B(0) xor M;
18    tB(1) <= B(1) xor M;
19    tB(2) <= B(2) xor M;
20    tB(3) <= B(3) xor M;
21    tB(4) <= B(4) xor M;
22    tB(5) <= B(5) xor M;
23    tB(6) <= B(6) xor M;
24    tB(7) <= B(7) xor M;
25
26    S(0) <= A(0) xor tB(0) xor M;
27    S(1) <= A(1) xor tB(1) xor C1;
28    S(2) <= A(2) xor tB(2) xor C2;
29    S(3) <= A(3) xor tB(3) xor C3;
30    S(4) <= A(4) xor tB(4) xor C4;
31    S(5) <= A(5) xor tB(5) xor C5;
32    S(6) <= A(6) xor tB(6) xor C6;
33    S(7) <= A(7) xor tB(7) xor C7;
34
35    C1 <= ((A(0) xor tB(0)) and M) or (A(0) and tB(0));
36    C2 <= ((A(1) xor tB(1)) and C1) or (A(1) and tB(1));
37    C3 <= ((A(2) xor tB(2)) and C2) or (A(2) and tB(2));
38    C4 <= ((A(3) xor tB(3)) and C3) or (A(3) and tB(3));
39    C5 <= ((A(4) xor tB(4)) and C4) or (A(4) and tB(4));
40    C6 <= ((A(5) xor tB(5)) and C5) or (A(5) and tB(5));
41    C7 <= ((A(6) xor tB(6)) and C6) or (A(6) and tB(6));
42    Cout <= ((A(3) xor tB(3)) and C3) or (A(3) and tB(3));
43  end FA_8bits;
```

模擬圖:

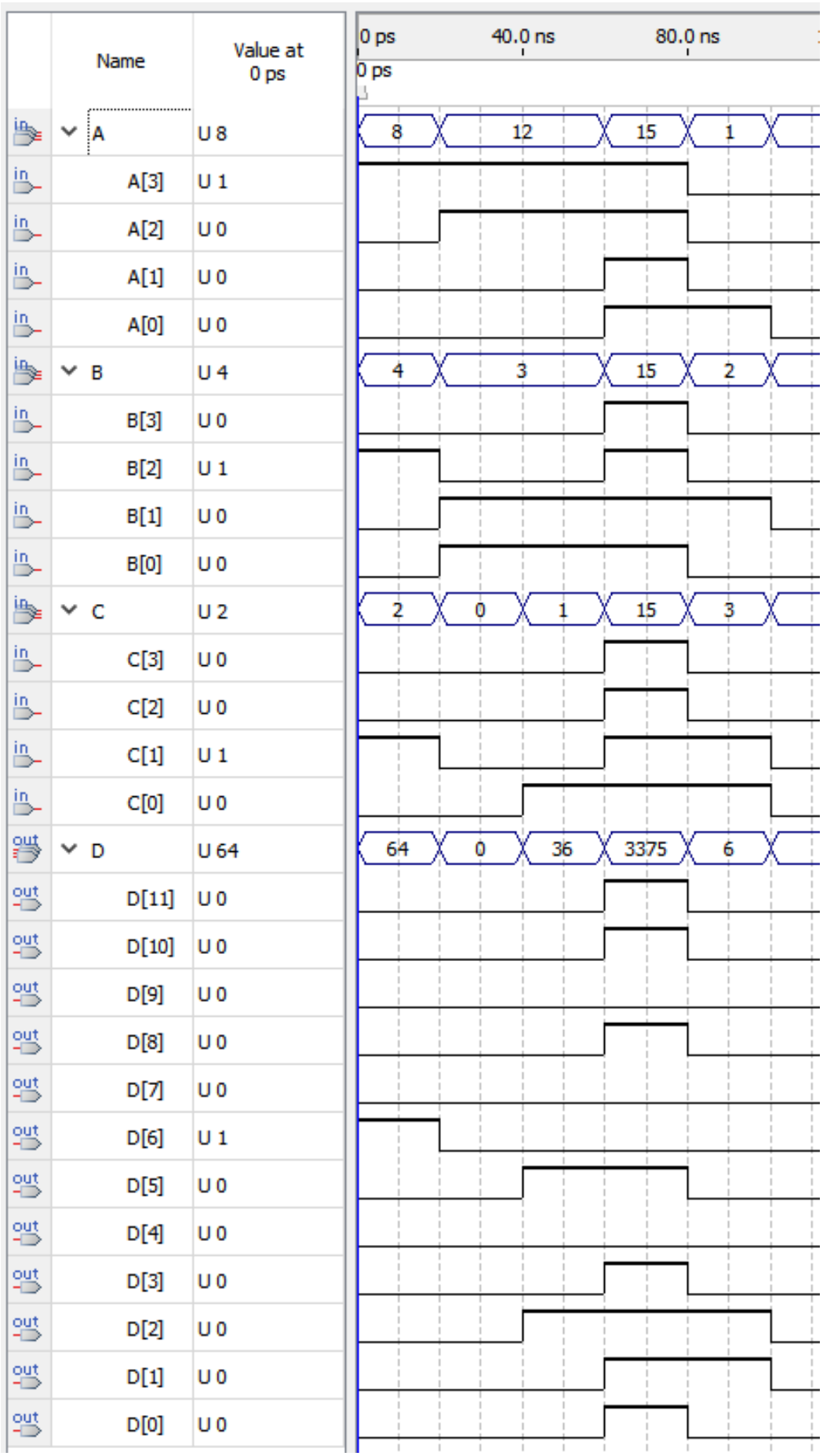

Name
Value at 0 ps
0 ps
10.0 ns
20.0 ns
30.0 ns
40.0 ns
50.0 ns
60.0 ns
A
U 255
255
254
255
15
A[7]
U 1
A[6]
U 1
A[5]
U 1
A[4]
U 1
A[3]
U 1
A[2]
U 1
A[1]
U 1
A[0]
U 1
B
U 255
255
254
1
31
B[7]
U 1
B[6]
U 1
B[5]
U 1
B[4]
U 1
B[3]
U 1
B[2]
U 1
B[1]
U 1
B[0]
U 1
M
B 1
Cout
B 1
S
U 0
0
1
255
0
240
46
S[7]
U 0
S[6]
U 0
S[5]
U 0
S[4]
U 0
S[3]
U 0
S[2]
U 0
S[1]
U 0
S[0]
U 0

2.

Code:

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_unsigned.all;
4
5  entity mul is
6     port(
7        A, B, C: in std_logic_vector(3 downto 0);
8        D: out std_logic_vector(11 downto 0)
9     );
10 end mul;
11
12 architecture mul of mul is
13    signal t1, t2, t3, t4, D1: std_logic_vector(7 downto 0);
14    signal tt1, tt2, tt3, tt4: std_logic_vector(11 downto 0);
15 begin
16
17    t1 <= "0000" & A when B(0) = '1'  else "00000000";
18    t2 <= "000" & A & '0' when B(1) = '1' else "00000000";
19    t3 <= "00" & A & "00" when B(2) = '1' else "00000000";
20    t4 <= "0" & A & "000" when B(3) = '1' else "00000000";
21    D1 <= (t1 + t2 + t3 + t4);
22
23    tt1 <= "0000" & D1 when C(0) = '1' else "000000000000";
24    tt2 <= "000" & D1 & '0' when C(1) = '1' else "000000000000";
25    tt3 <= "00" & D1 & "00" when C(2) = '1' else "000000000000";
26    tt4 <= "0" & D1 & "000" when C(3) = '1' else "000000000000";
27    D <= (tt1 + tt2 + tt3 + tt4);
28
29 end mul;
```

模擬圖:

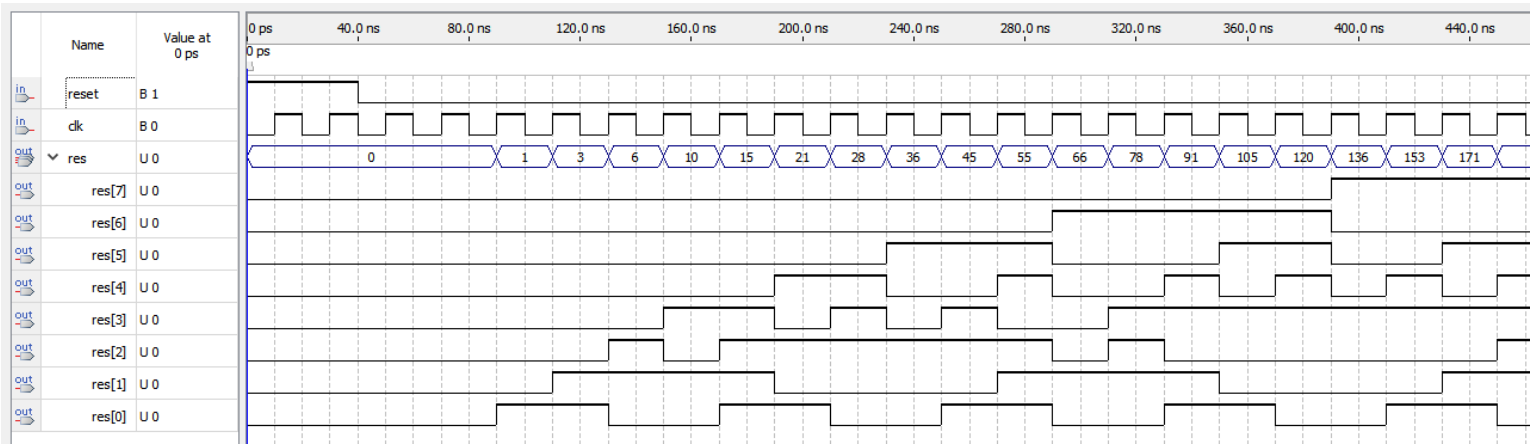

Name
Value at 0 ps
0 ps
40.0 ns
80.0 ns
0 ps
A
U 8
8
12
15
1
A[3]
U 1
A[2]
U 0
A[1]
U 0
A[0]
U 0
B
U 4
4
3
15
2
B[3]
U 0
B[2]
U 1
B[1]
U 0
B[0]
U 0
C
U 2
2
0
1
15
3
C[3]
U 0
C[2]
U 0
C[1]
U 1
C[0]
U 0
D
U 64
64
0
36
3375
6
D[11]
U 0
D[10]
U 0
D[9]
U 0
D[8]
U 0
D[7]
U 0
D[6]
U 1
D[5]
U 0
D[4]
U 0
D[3]
U 0
D[2]
U 0
D[1]
U 0
D[0]
U 0

3.

Code:

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_unsigned.all;
4
5  entity FA_8bit_process is
6     port(
7        reset, clk: in std_logic;
8        res: out std_logic_vector(7 downto 0)
9     );
10 end FA_8bit_process;
11
12 architecture FA_8bit_process of FA_8bit_process is
13    signal n, tmp: std_logic_vector(7 downto 0);
14 begin
15    process(reset, clk)
16    begin
17       if reset = '1' then
18          tmp <= "00000000";
19          n <= "00000000";
20       elsif rising_edge(clk) then
21          res <= tmp;
22          tmp <= tmp + n;
23          n <= n + "00000001";
24       end if;
25    end process;
26 end FA_8bit_process;
```

模擬圖:


Name
Value at 0 ps
reset
B 1
clk
B 0
res
U 0
res[7]
U 0
res[6]
U 0
res[5]
U 0
res[4]
U 0
res[3]
U 0
res[2]
U 0
res[1]
U 0
res[0]
U 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
0 ps
0
1
3
6
10
15
21
28
36
45
55
66
78
91
105
120
136
153
171