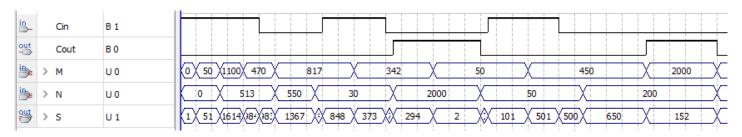
1.

```
Code:
  1
       library ieee;
  2
       use ieee.std logic 1164.all;
  3
       use ieee.std logic unsigned.all;
  4
  5
     ⊟entity NM_FA is
  6
     generic(
  7
           N: integer range 1 to 32 := 8;
  8
           M: integer range 1 to 32 := 8
  9
 10
           port (
     11
           A: in std logic vector(N - 1 downto 0);
 12
           B: in std logic vector(M - 1 downto 0);
 13
           C: in std logic;
 14
           S: out std logic vector(M - 1 downto 0);
 15
           Co: out std logic
 16
           );
 17
       end NM FA;
 18
 19
     ⊟architecture NM FA of NM FA is
 20
           signal tp: std logic vector(M downto 0);
 21
     ⊟begin
           tp \ll ('0' \& A) + B + C;
 22
 23
           S \le tp(M - 1 downto 0);
 24
           Co \leftarrow tp(M);
 25
       end NM FA;
     library ieee;
 1
     use ieee.std_logic_1164.all;
 2
     use ieee.std logic unsigned.all;
 5 Hentity NM FA Gen is
         port (
 7
         A: in std_logic_vector(7 downto 0);
         B: in std_logic_vector(7 downto 0);
 8
         C: in std_logic;
 9
```

```
S: out std_logic_vector(7 downto 0);
10
         Co: out std_logic
11
12
         );
    end NM_FA_Gen;
13
14
15 Harchitecture NM FA Gen of NM FA Gen is
         component NM FA is
16 ⊟
17 ⊟
             generic(
18
                 N: integer range 1 to 32;
19
                 M: integer range 1 to 32
20
             );
             port(
21
                 A: in std_logic_vector(N - 1 downto 0);
22
23
                 B: in std_logic_vector(M - 1 downto 0);
24
                 C: in std logic;
                 S: out std_logic_vector(M - 1 downto 0);
25
26
                 Co: out std_logic
27
             );
28
         end component;
29
         U1: NM FA generic map(8, 8) port map(A, B, C, S, Co);
30
31
     end NM FA Gen;
```

## 模擬:



2.

#### Code:

```
1
     library ieee;
     use ieee.std logic 1164.all;
 2
     use ieee.std_logic_unsigned.all;
 3
 4
 5
   ⊟entity FA N is
 6
   generic(
 7
           N: integer range 1 to 32 := 3
8
        );
9 ⊟
       port(
           A: in std logic vector(N - 1 downto 0);
10
11
           B: in std logic vector(N - 1 downto 0);
12
           C: in std logic;
13
           S: out std logic vector(N - 1 downto 0);
14
           Co: out std logic
15
        );
16
     end FA N;
17
18
    ⊟architecture FA N of FA N is
19
        signal tp: std logic vector(N downto 0);
20
    ⊟begin
21
        tp <= ('0' \& A) + B + C;
22
        S \ll tp(N - 1 downto 0);
23
        Co \leftarrow tp(N);
24
     end FA N;
```

```
library ieee;
 2
        use ieee.std_logic_1164.all;
 3
        use ieee.std_logic_unsigned.all;
      entity MUL NM is
 5
           generic(
 6
      П
 7
              N: integer range 1 to 32 := 8;
 R
              M: integer range 1 to 32 := 8
9
          ) ;
10
          port(
      11
             X: in std_logic_vector(N - 1 downto 0);
12
             Y: in std logic vector(M - 1 downto 0);
12
             P: out std_logic_vector(N + M - 1 downto 0)
14
           1 :
       end MUL_NM;
15
16
17
      architecture MUL_NM of MUL_NM is
18
           type CC is array (0 to M - 1) of std_logic_vector(n - 1 downto 0);
19
           type TT is array (0 to M - 3) of std_logic_vector(N downto 0);
20
           signal C: CC;
21
           signal T: TT;
22
22
          component FA_N is
      24
             generic(N: integer range 1 to 32);
25
      port(
26
                 A: in std_logic_vector(N - 1 downto 0);
27
                 B: in std logic vector(N - 1 downto 0);
28
                 C: in std_logic;
29
                 S: out std_logic_vector(N - 1 downto 0);
30
                 Co: out std_logic
21
              ) =
22
           end component;
       begin
33
34
          process(X, Y)
      35
           begin
36
             for i in 0 to M - 1 loop
      П
                 for j in 0 to N - 1 loop
37
      38
                    C(i)(j) \ll X(j) and Y(i);
                 end loop;
29
40
              end loop;
41
           end process;
42
43
      □ LP: for i in 0 to M - 2 generate
44
45
      FI: if i = 0 generate
46
                 FA: FA_N generic map(N) port map('0' & C(i)(N - 1 downto 1),
      C(i + 1), '0', T(i)(N - 1 downto 0), T(i)(N));
47
48
              end generate;
49

☐ MID: if i > 0 and i < M - 2 generate
</p>
50
                 FA: FA_N generic map(N) port map(T(i - 1)(N downto 1), C(i + 1), '0',
      51
                 T(i)(N - 1 downto 0), T(i)(N));
52
              end generate;
53
      FN: if i = M - 2 generate
54
                 FA: FA_N generic map(N) port map(T(i - 1)(N downto 1), C(i + 1), '0',
      P(N + M - 2 \text{ downto } M - 1), P(N + M - 1));
55
56
              end generate;
       end generate LP;
57
58
          P(0) \leftarrow C(0)(0);
59
60
          process(T)
      61
           begin
              for i in 1 to M - 2 loop
62
      63
                 P(i) <= T(i - 1)(0);
64
              end loop;
65
           end process;
       end MUL_NM;
66
```

```
library ieee;
 2
     use ieee.std_logic_1164.all;
 3
     use ieee.std logic unsigned.all;
 5
   ⊟entity N3 is
        port(
 6
   7
           X: in std logic vector(3 downto 0);
8
           P: out std logic vector(11 downto 0)
9
        );
10
     end N3;
11
12
   ⊟architecture N3 of N3 is
        signal tp: std logic vector(7 downto 0);
13
14
   component MUL NM is
15 ⊟
           generic(
16
              N: integer range 1 to 32;
17
              M: integer range 1 to 32
18
           );
19
   port(
20
              X: in std logic vector(N - 1 downto 0);
21
              Y: in std logic vector(M - 1 downto 0);
22
              P: out std logic vector(N + M - 1 downto 0)
23
           );
24
25
        end component;
26
     begin
27
        U1: MUL NM generic map(4, 4) port map(X, X, tp);
28
        U2: MUL NM generic map(4, 8) port map(X, tp, P);
29
     end N3;
```

## 模擬:

		Name	Value at 0 ps	760.0 ns	780.0 ns	800.0 ns	820.0 ns	840.0 ns	860.
i <b>∌</b>	>	n s	U 0	11 12 1331 1728	13 14 2197 2744	15 16	17 \ 18 \ \ 4913 \ 5832	19 20 6859 8000	21
0.0 ns	s	880.0	) ns 900,(	) ns 920	1.0 ns 94	10.0 ns	960.0 ns	980.0 ns	1.0 us
	22	23 \(\)\(\)\(\)\(\)\(\)	24 \ 25 \ 13824 \ 15625 \	26 27 17576 19683	28 29 \(21952\)\(2438		791 0	1 2 2	3 27

3.

#### 只有 top\_level 和第二題不同,其他皆相同

```
library ieee;
     use ieee.std logic 1164.all;
 2
    use ieee.std logic unsigned.all;
   ⊟entity N3 is
 5
 6
   port (
7
           X: in std logic vector(2 downto 0);
           Y: in std_logic_vector(3 downto 0);
8
9
           Z: in std logic vector(4 downto 0);
10
           P: out std_logic_vector(11 downto 0)
11
        );
12
    end N3;
13
14
   ⊟architecture N3 of N3 is
15
        signal tp: std logic vector(6 downto 0);
16 ⊟
        component MUL NM is
17
           generic(
   18
              N: integer range 1 to 32;
19
              M: integer range 1 to 32
20
           );
21
           port (
   22
              X: in std logic vector(N - 1 downto 0);
23
              Y: in std logic vector(M - 1 downto 0);
24
              P: out std logic vector(N + M - 1 downto 0)
25
           );
26
27
        end component;
28
29
        U1: MUL NM generic map(3, 4) port map(X, Y, tp);
30
        U2: MUL_NM generic map(5, 7) port map(Z, tp, P);
31
     end N3;
```

# 模擬:

		o ps		
	> k	U 0	13 14 15 16 17 18 19 20 21 22	ķ
i <b>s</b>	> m	U 10	31 2	Ż
i Be	> n	U 16	14 7	ķ
**	> s	U 0	5642 X 6076 X 6510 X 6944 X 7378 X 252 X 266 X 280 X 294 X 308	ż

23 24 25 26 27	28 29 30 31 0	$\subset$
12	22	$\subset$
3	1	$\subseteq$
828 864 900 936 972	616 638 660 682	