```
1
2   ----------------------------------------2)   INTEGER MULTIPLIER---------------------------------
3
4   library IEEE;
5   use IEEE.STD_LOGIC_1164.ALL;
6   use IEEE.NUMERIC_STD.ALL;
7   --ENTITY
8   entity int_mul is
9      generic (IN_S:NATURAL:=8-1);
10
11     port(num1,num2: in unsigned(IN_S downto 0);
12
13        prod:out unsigned(IN_S downto 0);
14        suc_flag: out std_logic);
15  end int_mul;
16  --ARCHITECTURE
17  architecture Behavioral of int_mul is
18  signal sig:unsigned(IN_S downto 0);
19  begin
20
21  process(num1,num2)
22     variable a,m:unsigned(IN_S-1 downto 0);
23     variable mul,sum:unsigned(IN_S downto 0):="00000000";
24     variable carry: std_logic;
25     begin
26  --INITAILISING VARIABLES
27        suc_flag<='1';
28        a:=num1(IN_S-1 downto 0);
29        m:= num2(IN_S-1 downto 0);
30        for k in 0 to ((IN_S-1)) loop
31           if (m(k)='1') then
32              carry:='0';
33  --Adding each summand
34              for n in 0 to (IN_S-1) loop
35                 sum(n):=a(n) xor (mul(n) xor carry);
36                 carry:=(a(n) and carry) or (mul(n) and carry) or (a(n) and mul(n));
37              end loop;
38           end if;
39  --storing the sum of each new summand
40           mul:=sum;
41           if carry='1' then
42              suc_flag<='0';
```

```
43            end if;
44    --SHIFTING THE OPERAND
45            a:= a sll 1;
46          end loop;
47          sig<=mul;
48    --GIVING RESULT TO OUTPUT
49          prod(IN_S-1 downto 0)<=mul((IN_S-1) downto 0);
50          prod(IN_S)<=num1(IN_S) xor num2(IN_S);
51
52       end process;
53    end Behavioral;
54
55
```