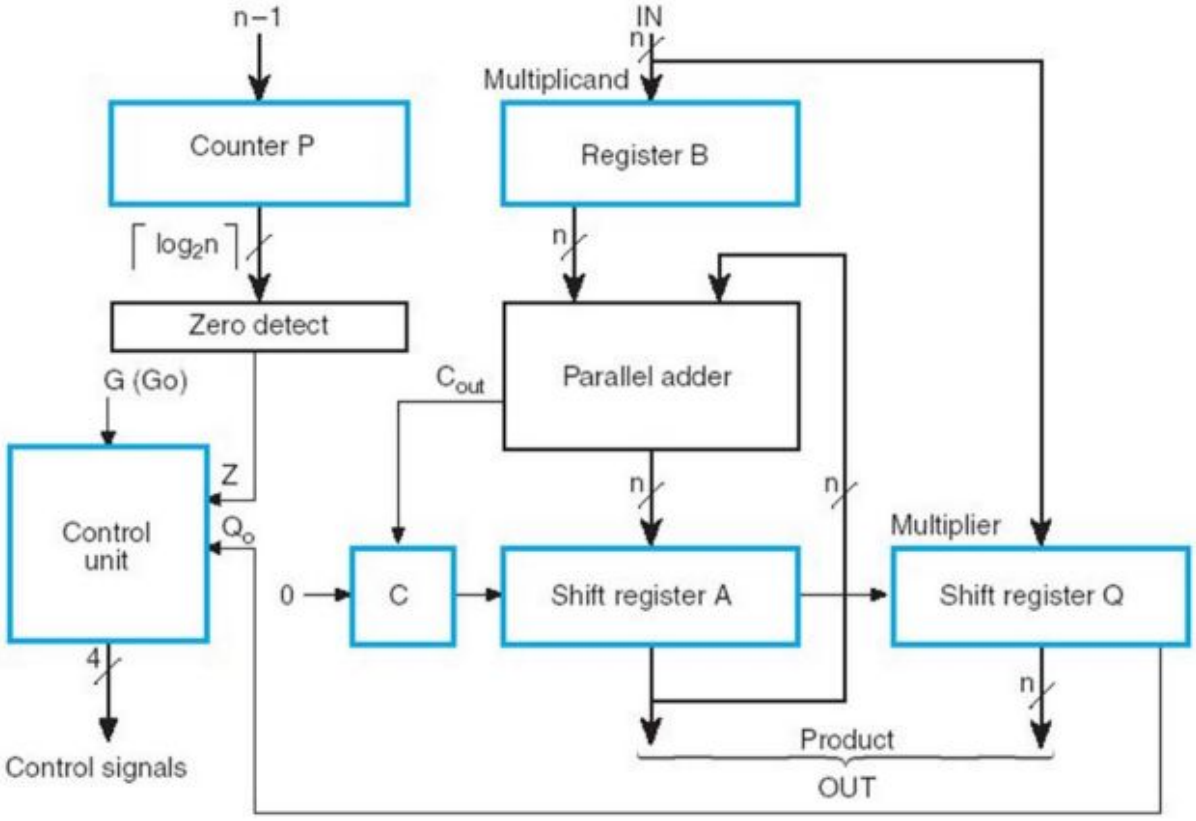
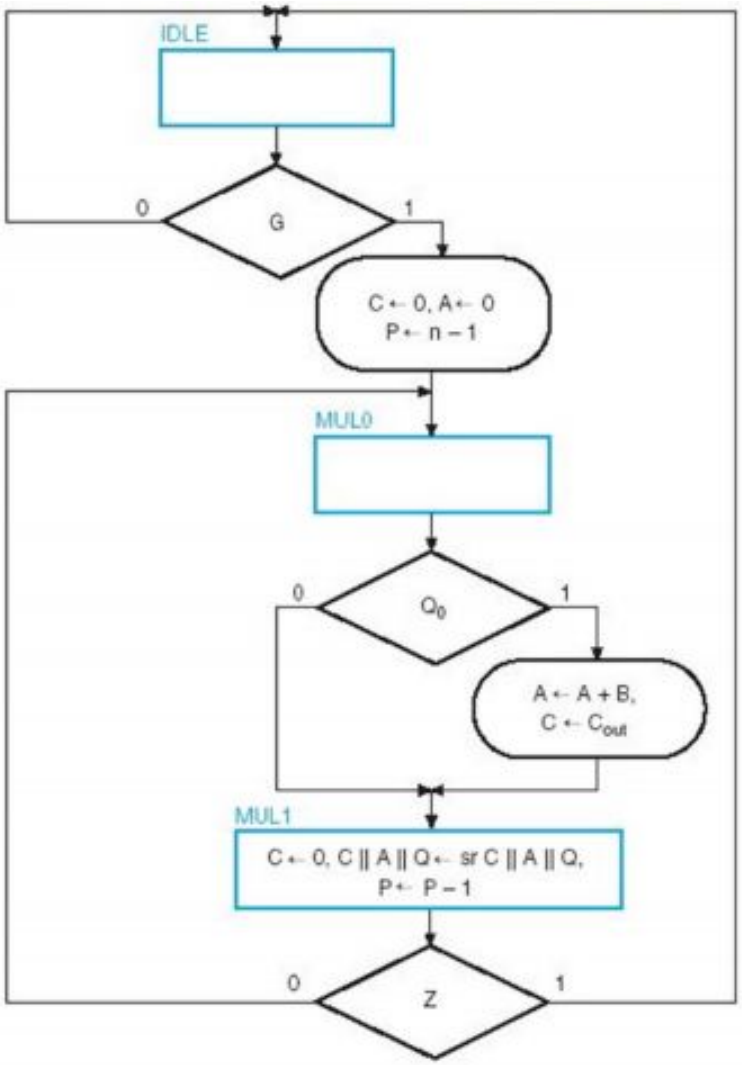

n–1
IN
n
Multiplicand
Counter P
Register B
⌈ log2n ⌉
n
Zero detect
G (Go)
Cout
Parallel adder
Z
n
n
Multiplier
Control unit
Qo
0
C
Shift register A
Shift register Q
4
n
Product
Control signals
OUT



1. IDLE - If G = ’1’, move to MUL1 state, else remain idle.
2. Set signals C = ‘1’, A = ‘0’, P = n-1 where:
    a. C = Carry
    b. A = Answer
    c. n = number of steps left?
3. MUL0 - If least significant bit of Q = ‘1’, A = A + B and C = C_{out} where:
    a. Q = Multiplier
    b. A = Answer
    c. B = Multiplicand
4. C = ‘0’, Shift C||A||Q right, P-- where:
    a. P = counter
5. If p = ‘0’, go to MUL0, else IDLE.

```
      n = 6

0)    100011 -B   Initial State
x     011001 -Q
      000000 -A

1)    100011 -B   p=6 ➡ Q(0) = 1, addition occurs ➡A=A+B
x     011001 -Q
      100011 -A

      100011 -B   p--;
x     101100 -Q   Q = A(0) || Q>>1
      010001 -A   A = C || A>>1     C=0

2)    100011 -B   p=5 ➡ Q(0) = 0, no addition
x     101100 -Q
      010001 -A

      100011 -B   p--;
x     110110 -Q   Q = A(0) || Q>>1
      001000 -A   A = C || A>>1

3)    100011 -B   p=4 ➡ Q(0) = 0, no addition
x     110110 -Q
      001000 -A

      100011 -B   p--;
x     011011 -Q   Q = A(0) || Q>>1
      000100 -A   A = C || A>>1

4)    100011 -B   p=3; ➡ Q(0) = 1, addition ➡A=A+B, C=0
x     011011 -Q   Q = A(0) || Q>>1
      100111 -A

      100011 -B   p--;
x     101101 -Q   Q = A(0) || Q>>1
      010011 -A   A = C || A>>1

5)    100011 -B   p=2 ➡ Q(0) = 1, addition ➡A=A+B, C=0
x     101101 -Q
      110110 -A

      100011 -B   p--;
x     010110 -Q   Q = A(0) || Q>>1
      011011 -A   A = C || A>>1
```

```
6)    100011 -B   p=1 ➡ Q(0) = 0,  no addition
x     010110 -Q
      011011 -A

      100011 -B   p--;
x     101011 -Q   Q = A(0) || Q>>1
      001101 -A   A = C || A>>1

      p=0, finished answer = A||Q = 001101||101011 = 001101101011
```