NOVEMBER, 6,2013

BINARY MULTIPLIER

BY :
ANUVRATA CHITRANSHI : 120123052
MRIDUL KAVIDAYAL : 120123050
K.VENKATA NEEHAR : 120101038

Binary Multiplication :

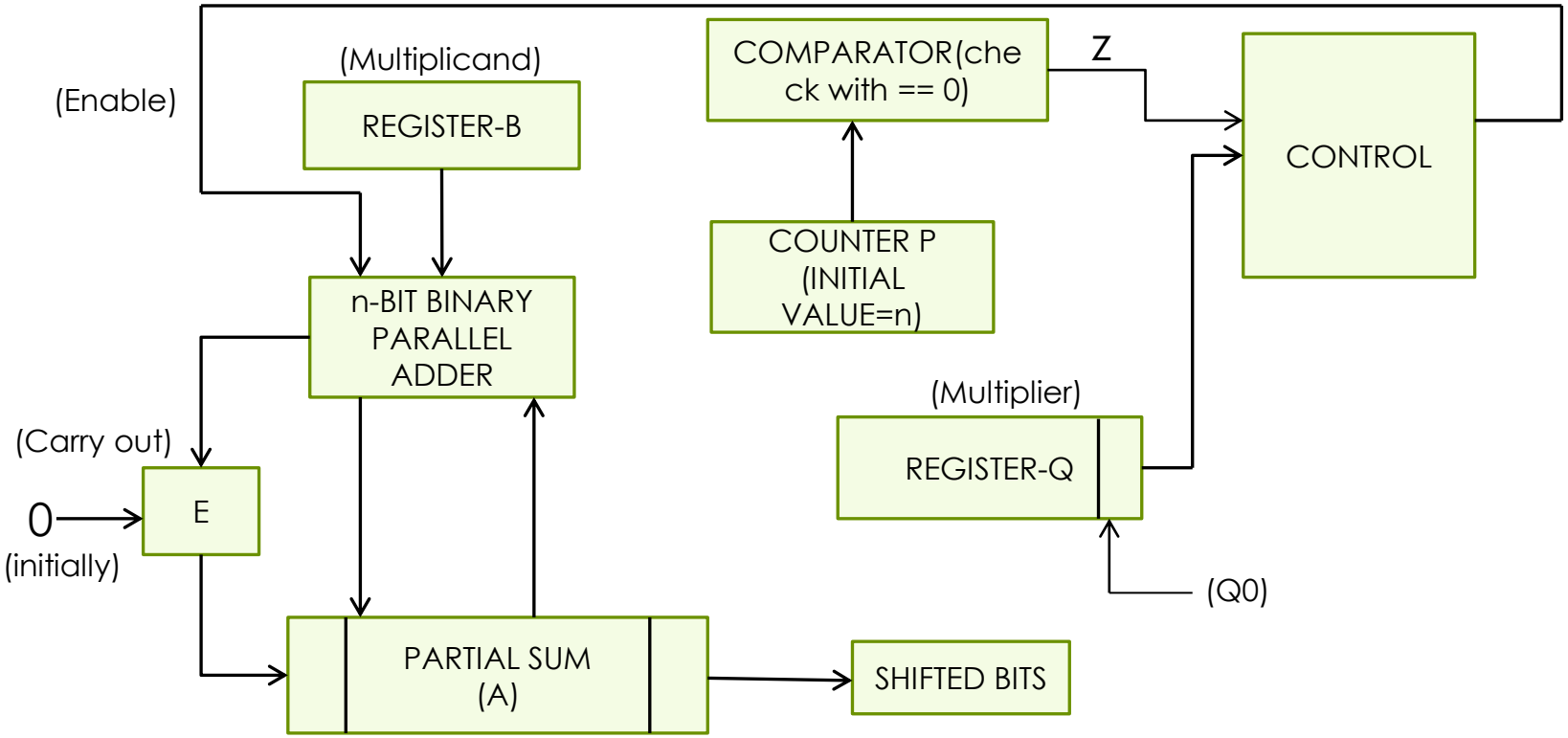
- The multiplication of two binary numbers is done with paper and pencil by successive (i.e. sequential) additions and shifting.
- The process is best illustrated with a numerical example. Let us multiply the two binary numbers :

  10111 and 10011:

```
    10111   (Multiplicand)
    10011   (Multiplier)
  ---------
    10111
   10111
  00000
 00000
10111
---------
110110101
```

- The process consists of successively adding and shifting copies of the multiplicand.
- Successive bits of the multiplier are examined, least significant bit first. If the multiplier bit is 1, the multiplicand is copied down; otherwise, 0's are copied down.
- The numbers copied in successive lines are shifted one position to the left from the previous number.
- Finally, the numbers are added and their sum forms the product. The product obtained from the multiplication of two binary numbers of *n* bits each can have up to 2n bits.

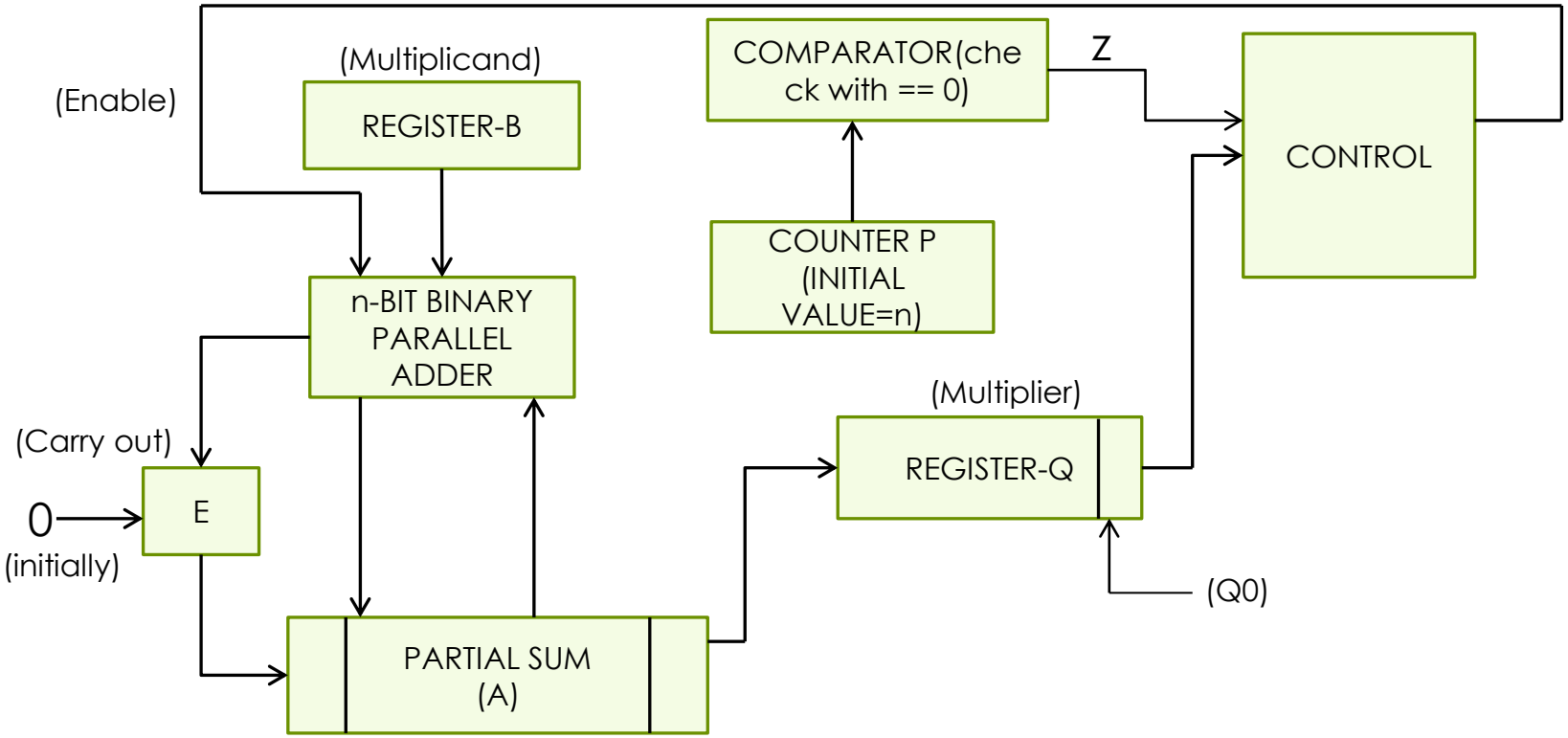
Digital Design of Binary Multiplier :



Explanation :

- The multiplicand is stored in register *B,* the multiplier is stored in register *Q,* and the partial product is formed in register A.
- A parallel adder is used to add the contents of register *B* to register A. The *E* flip-flop stores the carry after the addition.
- The *P* counter is initially set to hold a binary number equal to the number of bits in the multiplier.
- This counter is decremented after the formation of each partial product.
- The LSB of the partial product is transferred to the Shifted Bits Register.
- When the content of the counter reaches zero, the comparator gives z such that enable from control stops further processing ,the product is formed in the Shifted Bits Register and the process stops.

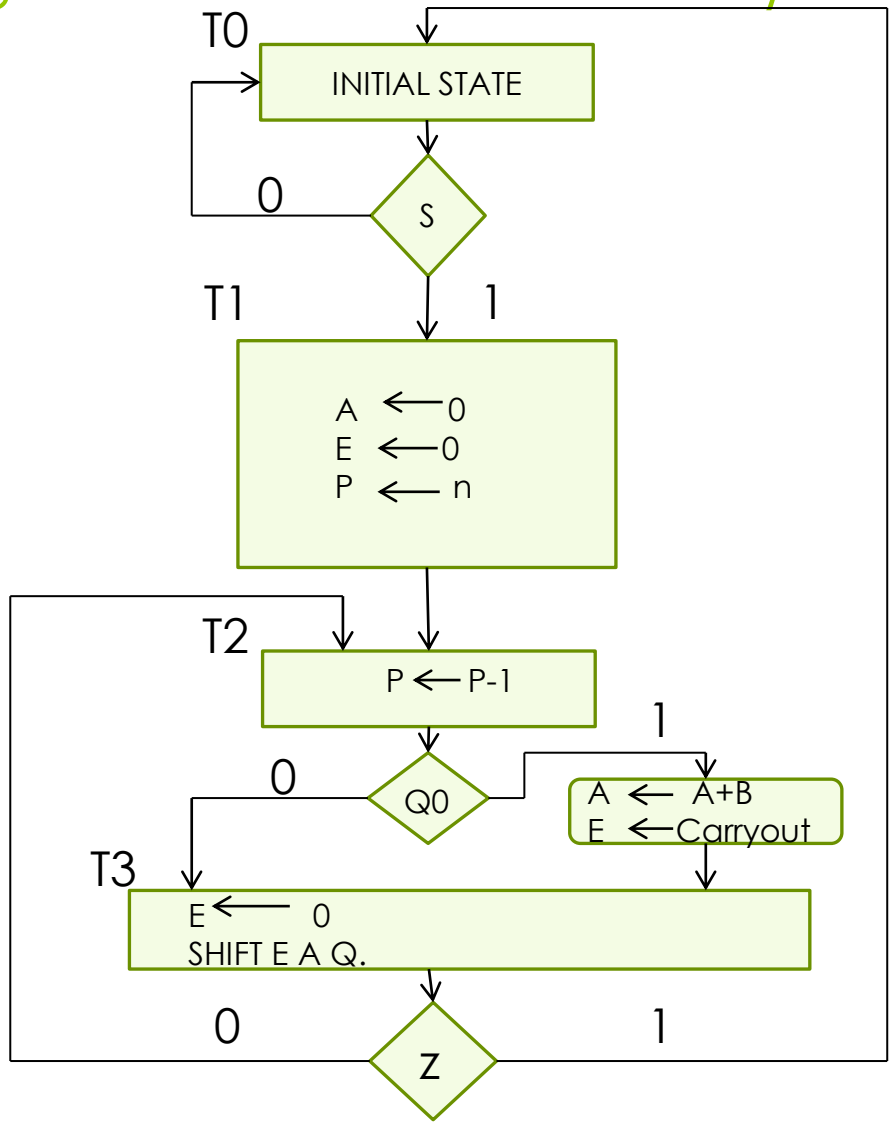
Optimization Of The Design :



Explanation :

- The sum of A and *B* forms a partial product, which is transferred to *A*.
- The output carry from the addition, whether 0 or I, is transferred to *E*.
- Both the partial product in A and the multiplier in Q are shifted to the right. The least significant bit of *A* is shifted into the Register Q.
- The carry from *E* is shifted into the most significant position of *A* and 0 is shifted into *E*.
- After the shift-right operation, one bit of the partial product is transferred into Q while the multiplier bits in Q are shifted one position to the right. In this manner, the rightmost bit of register Q, designated by Q0, always holds the bit of the multiplier that must be inspected next.

ASM (Algorithmic State Machine) Chart :



Explanation :

- Initially, the multiplicand is in *B* and the multiplier in *Q*. The multiplication process is initiated when *S* = 1.
- Register *A* and flip-flop *E* are cleared and the sequence counter *P* is set to a binary number *n*, which is equal to the number of bits in the multiplier.
- Next we enter a loop that keeps forming the partial products. The multiplier bit in Q0 is checked, and if it is equal to 1, the multiplicand in *B* is added to the partial product in *A*.
- The carry from the addition is transferred to *E*. The partial product in *A* is left unchanged if Q0= 0. The *P* counter is decremented by 1 regardless of the value of *Q*0.
- Registers *E, A,* and *Q* are combined into one composite register *EAQ,* which is then shifted once to the right to obtain a new partial product.

- The value in the *P* counter is checked after the formation of each partial product. If the content of *P* is not zero, control input Z is equal to 0 and the process is repeated to form a new partial product.
- The process stops when the *P* counter reaches 0 and the control input Z is equal to 1.
- Note that the partial product formed in *A* is shifted into Q one bit at a time and eventually replaces the multiplier. The final product is available in *A* and Q, with *A* holding the most significant bits and Q the least significant bits.

THANK YOU!! ☺