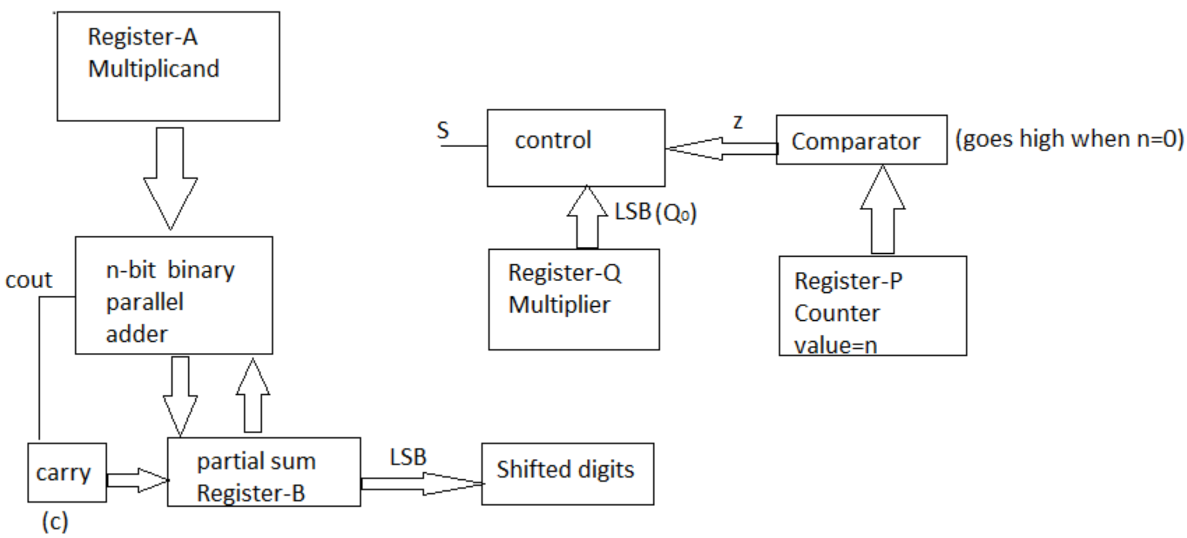
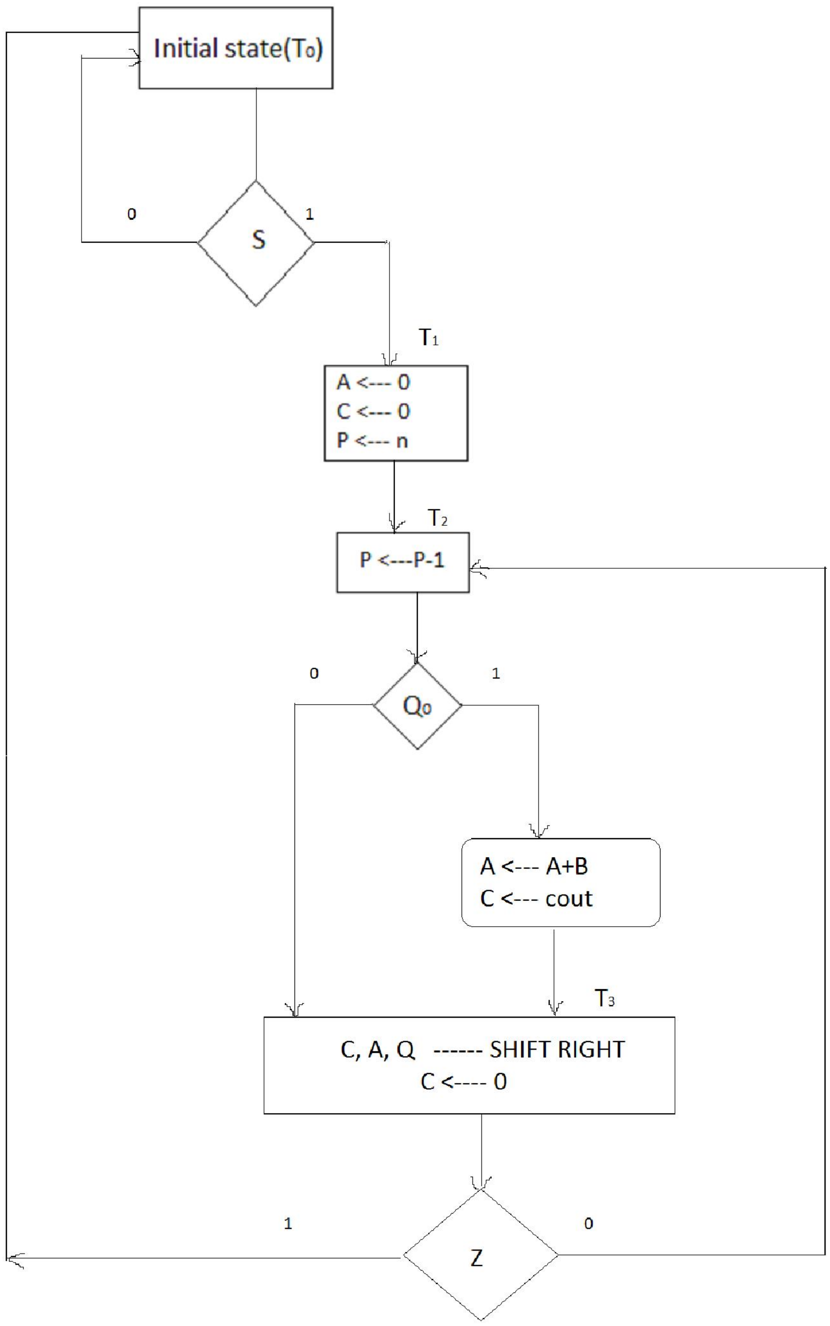
BINARY MULTIPLIER

- Binary multiplication :
    - Inputs
        - 1.Multiplicand
        - 2.Multiplier
- Algorithm:
    - We start from the least significant bit of the multiplier, if the encountered bit is one, the partial output is the multiplicand; else if the encountered bit is zero ,the partial output is zero.
    - We keep shifting the partial outputs as they come out by one bit left w.r.t the previous partial output and we add the partial outputs.
    - Registers are used to store the data i.e. multiplier , multiplicand, partial outputs
    - During addition of partial output to the previous partial sum,the last digit of the sum directly gets to the final answer, so we add a n-bit and (n-1)bit number each time.Therefore, we require a n-bit binary adder for this addition



- Process starts when S=1.
- Each time we check the value of Q_0 and carry out the appropriate action based on the value.
- After each check we keep shifting the multiplier to right by 1-bit.

- Q_0 ----- 0 : only right shift the partial sum by 1-bit.
  ----- 1 : Add the partial output to existing sum and then shift the new sum by 1-bit to right.
- We start with counter value at ‘n’.After every single operation we decrement the value of counter by 1 and check if the value of counter has gone to ‘0’.
- If counter value goes to ‘0’, we end the process, else continue.



- Optimization : An optimization would be to merge shifted digits register with the multiplier register, as multiplier register loses a bit each time while shifted bits register gains one.

Done by,

Abhishek Singh (120101025)
Shubhakar Reddy (120101074)
Gopichand Vasireddy (120101076)