Roll - 200050006

Ques - 2

In this question, we were asked to design a Booth Multiplier which takes two unsigned 4 bit numbers as input and produces an unsigned 8 bit number as output using a sequential clocked circuit.

Booth's Multiplier Logic:

Let's consider that we need to multiply two numbers A and B. So first we sill need to multiply A with couplets from B instead of single bits from B which we used to do in array multiplier. Here the two couplets are B1B0 and B3B2. As B is a 4 bit number, we will get two partial products. The number of partial products in Booth Multiplier is much less than array multiplier(half). There are some rules that need to be followed while calculating the partial products in the Booth Multiplier. The rules are -

1) If B1B0 is "00" then A multiplied by B1B0 is 0.
2) If B1B0 is "01" then A multiplied by B1B0 is A.
3) If B1B0 is "10" then A multiplied by B1B0 is 2A which is A left shifted by 1 place.
4) If B1B0 is "11" then A multiplied by B1B0 is 3A which is calculated using 4A - A, and 4A is just A left shifted by 2 bits and -A is 2's complement of A.
5) We just need to follow the same set of operations for B3B3 also.

Logic For Code:

First we give a if condition that further steps are to be performed only if there is a rising edge in the clock. Then we will check the condition of rst pin.

If rst pin = '1', we will reset the circuit and set the two partial products res1, res2 and final product result as "00000000".

I used a flag i to set the condition that the output should be delivered in two clock cycles only and not more or less. If flag i = '0' , we will calculate the partial products res1 and res2. And if flag i= '1', we will calculate the final result using res1 and res2.

If rst pin = '0' and flag i='0', we will check for different cases for the couplets of B, and put the result of the partial products as calculated from the rules mentioned above into res1 and res2 respectively for B1B0 and B3B2 and we will also set flag i = '1' for next step.

If rst pin = '0' and flag i='1', we will calculate the final result using res1 and res2.
Now to calculate the final result , we first need to shift the res2 i.e left shift res2 by two places and then add it with res1 to get the final result and we will also set flag i = '0' for the next set of inputs which can be given and result will be calculated in next two clock cycles.