Parallel
Multiplier–Accumulator Based on Radix-2
Modified Booth Algorithm

Team details

Name	USN
Nidhi Darak	01FE20BEC004
Avinash V V	01FE20BEC018

Theoretical Background:

The Booth algorithm was invented by A. D. Booth for Signed number multiplication. These have the potential to speed up signed multiplication. Booth's algorithm is based upon recoding the multiplier, to a recoded leaving the multiplicand unchanged. It uses encoders that generates half the partial products in parallel. By extending sign bit of the operands and generating an additional partial product the SUMBE multiplier is obtained. The Carry Save Adder (CSA) tree and the final Carry Look ahead (CLA) adder used to speed up the multiplier operation.

Working of Modified Booth Algorithm:

- Set the Multiplicand and Multiplier binary bits.
- Initially, we set the accumulator value to 0.
- The sequence counter that is continuously decremented till equal to the number of bits reached to 0.
- On each cycle of the booth algorithm, Y_n and Y_{n+1} bits will be checked on the following parameters as follows:
    i. When two bits Y_n and Y_{n+1} are 00 or 11, we simply perform the arithmetic shift right operation to the partial product. And the bits of Yn and Y_{n+1} is incremented by 1 bit.

ii. If the bits of Y_n and Y_{n+1} is shows to 01, the multiplicand bits (M) will be added to the Accumulator register. After that, we perform the right shift operation to the accumulator.

- The operation continuously works till we reached n - 1 bit in the booth algorithm.
- Results of the Multiplication binary bits will be stored in the accumulator and the registers.

The Booth recoding procedure is as follows:

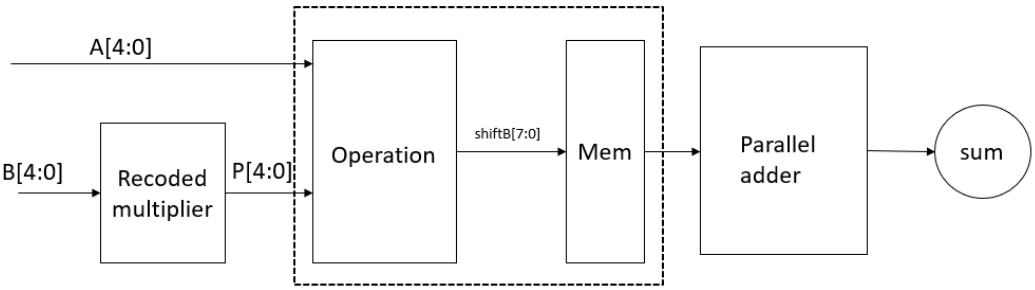
1. Working from LSB to MSB, replace each 0 digit of the original number with a 0 in the recoded number until a 1 is encountered.

2. When a 1 is encountered, insert a 1 at that position in the recoded number, and skip over any succeeding I's until a 0 is encountered.

3. Replace that 0 with a 1 and continue.

Y_i	Y_{i-1}	Partial Product
0	0	0×Multiplicand
0	1	1×Multiplicand
1	0	-1×Multiplicand
1	1	0×Multiplicand

Architecture:

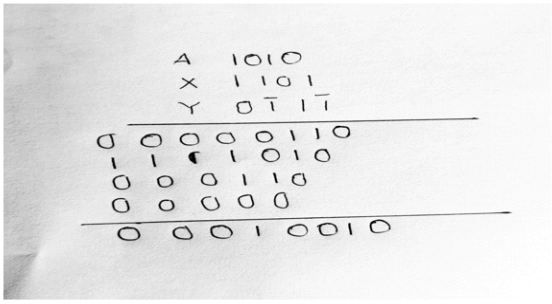


Example:

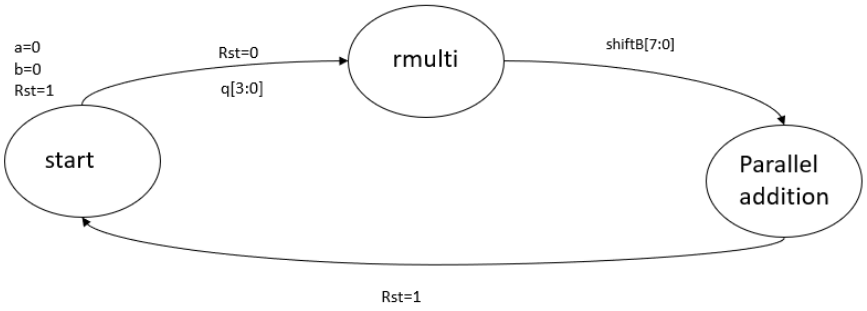
Booth Algorithm

A		1	0	1	0			-6
X	×	1	1	0	1			-3
Y		0	$\bar{1}$	1	$\bar{1}$			recoded multiplier
Add -A		0	1	1	0			
Shift		0	0	1	1	0		
Add A		1	0	1	0			
		1	1	0	1	0		
Shift		1	1	1	0	1	0	
Add -A		0	1	1	0			
		0	1	0	0	1	0	
Shift		0	0	1	0	0	1	0

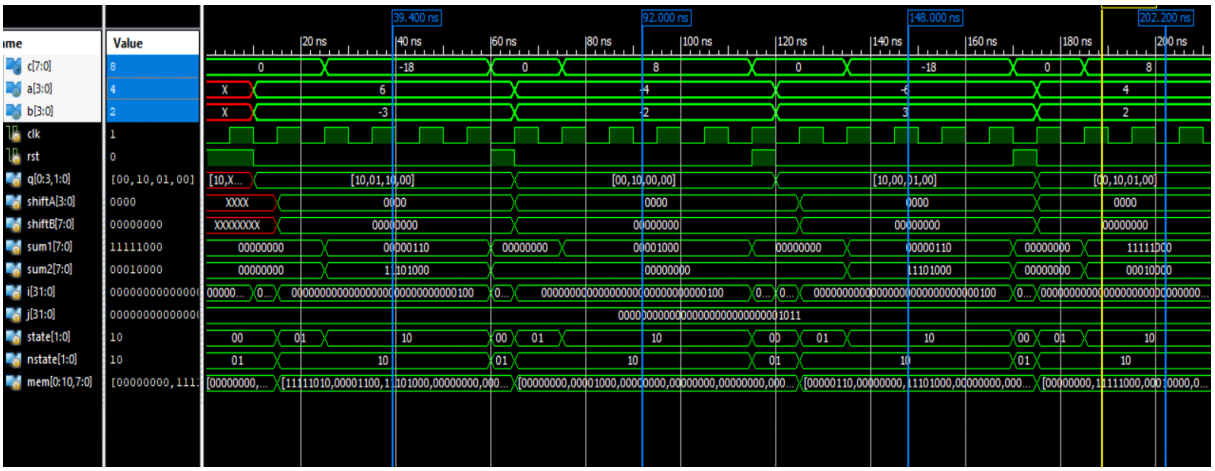
Modified Booth Algorithm



FSM:



Results:



Conclusion:

In this project, the Architecture and FSM to implement parallel multiplier–accumulator based on radix-2 modified booth algorithm is proposed. The project is successfully implemented using Verilog code in Xilinx software.