|  |  |
| --- | --- |
| **Name : Amir Shaikh** | **Date :** |
| **Roll No : 18102A0053** | **Signature :** |
| **Class : SE CMPN - A** | **Grade :** |
| **Batch :** |  |
| **Experiment No: 5** |  |

**Title:** Booths Multiplication

**Estimated time to complete this experiment:** 2 hours

**Objective:**

Understanding behaviour of Booth’s Multiplier from module designed by the student as part of the experiment .

Designing Booth’s multiplier with a controller and data path.

Understanding the advantage of Booth’s Multiplier.

**Books/ Journals/ Websites referred:**

Books:

Digital Logic and Computer Design - M. Morris Mano. Pearson Education - Prentice Hall.

Digital Principles Foundation of Circuit Design and Application - Arun Kumar Singh. New Age Publishers.

The Art of Electronics - Paul Horowitz and Winfield Hill (1989). Cambridge University Press

Modern Dictionary of Electronics - Rudolf F. Graf (1999). Newnes

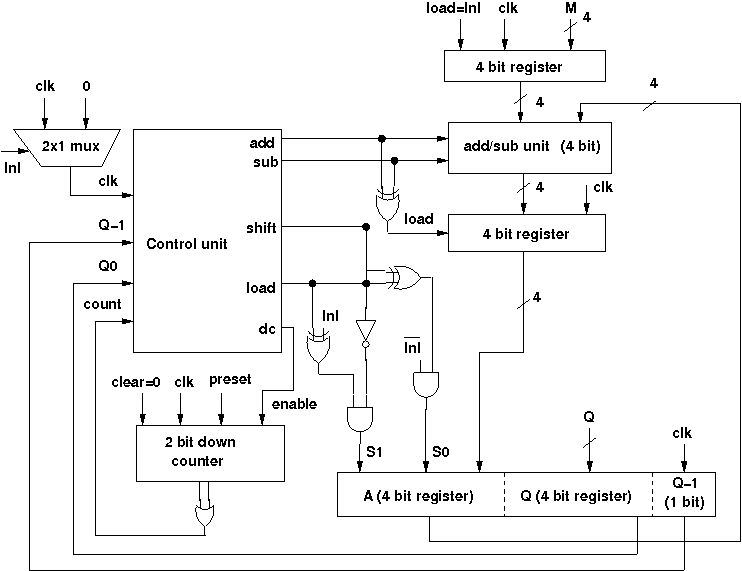
Web Sites:

* https://en.wikipedia.org/wiki/Booth%27s\_multiplication\_algorithm
* [NPTEL (e-learning courses from IITs and IISC)](http://nptel.iitm.ac.in/courses.php?disciplineId=106)

**Requirements:** Virtual simulator.

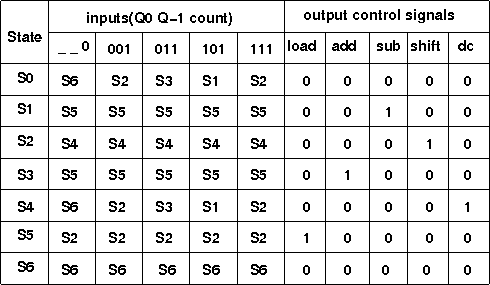
* **Components :-**
* Controller
* BoothM Datapath (.i.e. Registers, adder / subtractor unit, logic gates , down counter used to construct the datapath)
* 9 Bit display’s to obtain the output.
* 2 Digital displays to check input value .
* Bit Switch
* 1 Clock Input.
* Wires

**Circuit Diagram:**

****

**Procedure:**

1. Start the simulator as directed.
2. To design the circuits we need, 9 Bit displays(to get the answer and the ), a clock input, bit switch (to give input),wires.
3. The pin configuration of a component is shown whenever the mouse is hovered on any canned component of the palette. Pin numbering starts from 1 and from the bottom left corner(indicating with the circle) and increases anticlockwise.
4. The behaviour of a control unit can be represented in a state table. The controller in the simulator accepts the Moore type state chart and must contain an end state.
5. The controller will generate 5 output control signals. After entering these informations, the second form will appear where you can set the names of the inputs and outputs. Here inputs are the Q0, Q-1, count. Outputs control signals are load, add, sub, shift, dc. The order of given input/outputs are maintained while creating terminals of the controller. for example, the first output signal will appear in the left most output terminal (lower terminals), second output will appear in the second left most bit and so on. In case of input terminals, the left most bit is for clock input, so the first input appears in the second terminal, and then the order is maintained. Then the third form will appear where you actually specify the state chart i.e. state, outputs of that state and transition conditions . The fields of the chart will be generated dynamically according to previously given information on states, inputs and outputs.After entering the following state chart, click on the controller component in the palette of the simulator then click on the position of the design editor where you want to put the component (no drag and drop, simple click will serve the purpose



1. To connect any two components select the Connection menu of Palette, and then click on the Source terminal and click on the target terminal. According to the circuit diagram connect all the components. Start the clock input using the input provided in the top palette the output from each flip-flop is taken into a bit display which displays the bit output.
2. At first initialize the multiplier by giving the specified inputs specified earlier, this will load the multiplier and multiplicand, then start the multiplication operation by giving the specified inputs specified earlier. At the end state (S6), the multiplication result will be seen through ports FQ0 to FA3 (FQ0 is LSB, FA3 is MSB). The current state of the controller is shown in the left pane as it transits from one state to another. The controller can be reset by clicking the reset controllerbutton in the top toolbar, to start with a new input.

**Conclusion:** Booth’s Multiplier can handle

* + It can handle signed integers in 2's complement notion
  + It decreases the number of addition and subtraction
  + It requires less hardware than combinational multiplier
  + It is faster than straightforward sequential multiplier

Hence making it very versatile

**Real Life Application:**

1. Low power VLSI techniques.
2. Modified Booth Multiplier is used in MAC architecture.

**Post Lab Questions:**

Q. Which sequence of 1 & 0 gives best performance with Booth's algorithm?

Q. Give the Flow Chart for Booth’s Algorithm

Q. Why Booth's multiplier is faster than other approaches like shift and add multiplier?

Q. Booth's algorithm increases space in case of fixed point signed multiplication?