Volume 5, Issue 4, April 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Implementing Booth Multiplication Algorithm Using Software Virtual Hardware

[1]Siddharth Srivastava*, [2]Shikha Srivastava
[1]Lecturer, Department of Computer Science, Central Institute of Plastics Engg. and Technology, Lucknow, U.P., India
[2]Student, Department of Computer Science (B.C.A), National P.G. College, Lucknow, U.P., India

Abstract— This journal entitled "Implementing Booth Multiplication Algorithm using software virtual hardware" is basically presents a new way of testing hardware's using our home made software entitled "VIRTUAL HARDWARE". We have developed this software for the purpose of generating feasibility study report of any hardware. This software checks whether the fabrication of any hardware is feasible or not. This software uses in-built basic hardware in order to simulate new logical hardware's and generate the execution report of new logical hardware. Due to this we can easily figure out whether the given logical hardware under consideration is feasible or not.

VIRTUAL HARDWARE is basically the software works on the concept of RTL (Register Transfer Language) and hardware implementation in order to fabricate feasibility study report. Based on this report hardware developers can easily find whether the hardware is feasible or not. This software basically virtually simulates the logical hardware under consideration and generates the execution report or feasibility report which shows the feasibility of logical hardware since it is virtually executing the logical hardware hence we have given it the name "virtual hardware". Using this software logical hardware can easily be tested before their actual implementation starts or more correctly before fabrication of actual hardware starts. In this journal we will analyse Booth Multiplication Algorithm for its feasibility. We will show how we are going to implement booth using our software and how to analyse its feasibility study report. This software also presents a unique feature using which we can get feasibility study report in form of operational table.

Keywords— Virtual Hardware, Booth Multiplication Algorithm, RTL, ASM, Micro-operations.

I. INTRODUCTION

Booth Multiplication Algorithm is basically most widely used algorithm in the world. In present day Scenario most of the systems implement Booth algorithm for performing multiplication. So we can say that Booth hardware is one of the most widely used hardware present in the market. We are going to find the feasibility of booth multiplication hardware using our software entitled "VIRTUAL HARDWARE". In this journal first of all we will describe the basic fundamentals around which booth algorithm is fabricated then we will give the complete hardware implementation of booth multiplication algorithm and after that we will give the core booth multiplication algorithm along with its ASM (Algorithmic state machine) chart. When this much is done then we will implement booth multiplication algorithm in our software and generate its feasibility study report. By this report we can observe that Booth hardware is feasible and can easily be fabricated.

II. UNDERSTANDING BASICS OF BOOTH MULTIPLICATION ALGORITHM

Booth multiplication algorithm provides the procedure for multiplying two numbers, where negative numbers are represented using signed 2's complement notation. Booth multiplication algorithm gives a procedure for multiplying binary integer in signed 2's complement representation. It operates on the fact that strings of 0's in the multiplier require no addition but just shifting and strings of 1's in the multiplier from bit weighted 2^k to weight 2^m can be treated as $2^{(k+1)}$- 2^m. Suppose binary number $(001110)_2$ represents +14 as a string of 1's for 2^3 to 2^1. This implies k=3 & m= 1.

The number can be represented as $2^{(k+1)} - 2^m$. $2^4 - 2^1$ =16-2=14. Therefore the multiplication m*14 where m is multiplicand & 14 is multiplier, can be done as $m*2^4 - m*2^1$. Thus, the products can be obtained by shifting binary multiplicand 'm' four times to the left and subtracting m shifted one's left.

As in all multiplication schemes booth algorithm requires examination of the multiplier bits and shifting of partial products prior to the shifting. The multiplicand may be added to partial product, separated from partial product or left unchanged all to the following rule which are given below.

Rule 1

The multiplicand is subtracted from the partial product upon encountering the first least significant 1 in the strings of 1's in the multiplier.

© 2015, IJARCSSE All Rights Reserved

Rule 2

The multiplicand is added to the partial product upon encountering the first 0 (provided that there was a previous one) in a string of zeroes in the multiplier.

Rule 3

The partial product does not change when the multiplier bit is identical to the previous multiplier bit. The algorithm works for positive or negative multipliers in 2's compliment representation. This is because a negative multiplier ends with a string of 1 and the last operation will be a subtraction of appropriate weight.

This algorithm is implemented on paper first and then the hardware is designed. But there are chances of error in doing calculations manually which will result in loss if the hardware has been designed considering that calculation. Due to this we have developed software using which result is obtained immediately before the physical implementation of circuit.

III. HARDWARE IMPLEMENTATION OF BOOTH MULTIPLICATION ALGORITHM

Let us assume that multiplier is stored in BR and multiplicand in QR and number of bits, are stored in register SC that is sequence counter .Register AC and BR are connected with each other using complement and parallel adder circuits. The result of addition or subtraction executed between AC and BR are stored in register AC. Register AC,QR and flip-flop QR+1 holds the least significant bit of QR when arithmetic shift operation is performed on register AC,QR, QR+1 respectively. The final result is stored in register AC and QR. The hardware implementation of Booth multiplication algorithm is shown below.

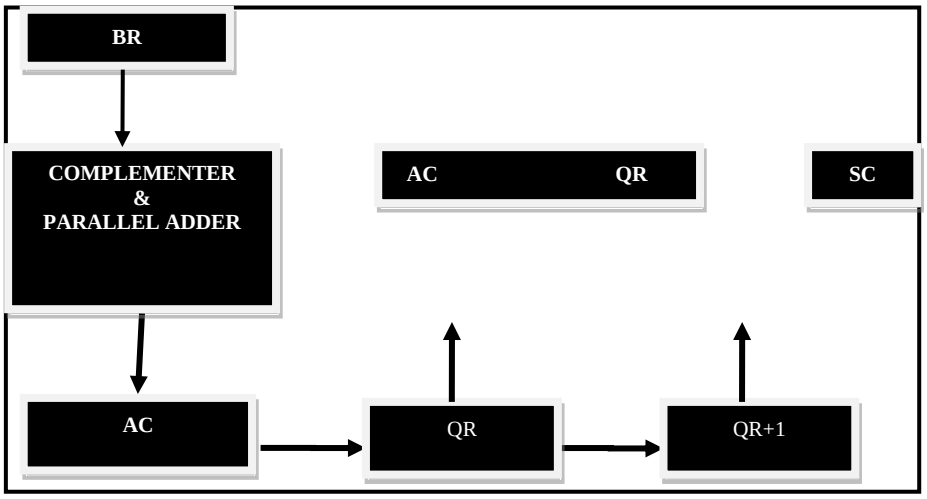

Figure 1. Showing Hardware Implementation of Booth Multiplication Algorithm.

Booth multiplication algorithm along with its ASM chart is given below. Its explanation is already given above in section.

```
Booth_Multiplication_Algorithm()
{Step 1:
 Set (Sequence Counter) SC=011 (which is number of bits present in BR or QR).
 Set AC=000 and QR+1=0;

 Step 2:
 If((QR=0 and QR+1=0) OR(QR=1 and QR+1=1))
 {ASHR (AC,QR,QR+1);}
 Else if((QR=0 and QR+1=1))
 {AC=AC+BR;
  ASHR (AC,QR,QR+1);
 }
 Else if((QR=1 and QR+1=0))
 {AC=AC+(BR)';
  ASHR (AC,QR,QR+1);
 }
 SC=SC-1;

 Step 3:
 Repeat step 2 until SC!=0;

 Step4:
 Result is stored in AC and QR.
}
```

© 2015, IJARCSSE All Rights Reserved

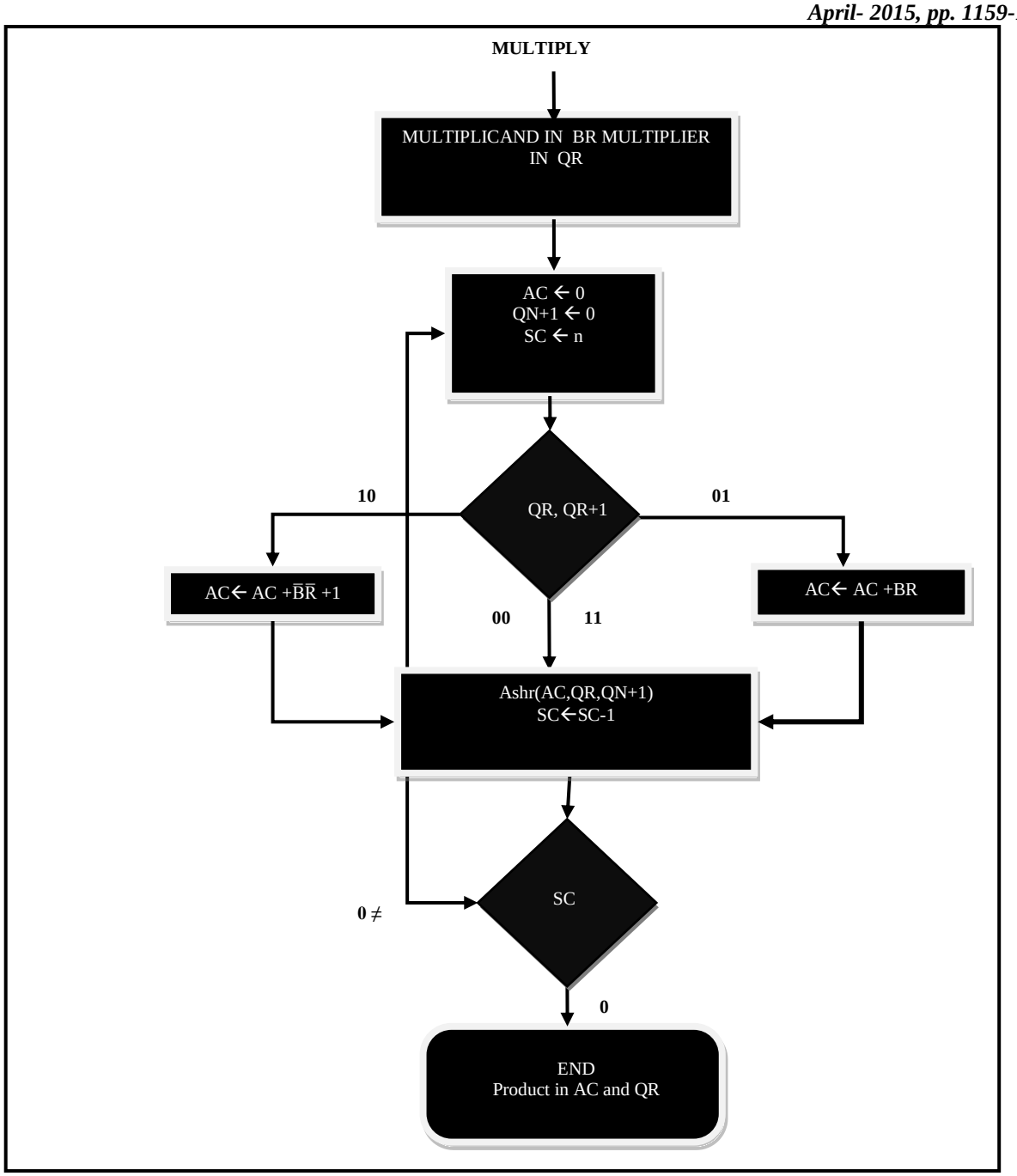

Figure 2. Showing ASM chart of Booth Multiplication Algorithm.

IV. IMPLEMENTATION OF BOOTH MULTIPLICATION ALGORITHM ON SOFTWARE VIRTUAL HARDWARE

Now we are going to show how we can implement booth multiplication algorithm in our software and how our software generates the feasibility study repot which proves the consistency of booth multiplication algorithm .But before implementing booth in our software we must have an overview of tags which are required by our software for implementing booth multiplication algorithm. These tags along with their meaning are described below in tabular form.

Table 1. Showing Basic tags of Software Virtual Hardware.

TAGS	MEANING
Register””	Use to create virtual registers
<PRINT>	Use to display messages
<PRINTLN>	Use to display messages with new line
ADD destination ,souce1 ,source2	Use to perform add operation on registers
NOT register1	Use to perform not operation on register1
MERGE	Use to perform merge operation

© 2015, IJARCSSE All Rights Reserved

CHOP	Use to perform chop operation
<clean_table>	Use to clean feasibility table
<create_table>	Use to create new feasibility table
<start_row></start_row>	Use to insert new row in feasibility table
<start_column></start_column>	Use to insert new column in feasibility table
<insert>	Use to insert data infeasibility table

Now we are going to present our sample script for booth multiplication algorithm which when we insert it in our software then we will get feasibility study report as the output of the software using which we can see whether the hardware under consideration is feasible or not.

```
Register Result<i>"00000000"</i>
Register AC<i>"0000"</i>
Register BR<i>"0010"</i>
Register BRNP1<i>"1110"</i>
Register QR<i>"0011"</i>
Register QNP1<i>"0"</i>
Register SC<i>"100"</i>
Register ZERO<i>"000"</i>
Register ONE<i>"001"</i>
<clean_table>
<start_row>
<start_column><insert>"Micro-Operations"</start_column>
<start_column><insert>"AC"</start_column>
<start_column><insert>"QR"</start_column>
<start_column><insert>"QR+1"</start_column>
<start_column><insert>"SC"</start_column>
</start_row>
LOOP
[START]
IF(QR<bit>0</bit>=='1'&&QNP1<bit>0</bit>=='0')
 ADD AC,AC,BRNP1<>
 MERGE Result,AC,QR<>
 MERGE Result,Result,QNP1<>
 ASHR Result<>
ELSE IF(QR<bit>0</bit>=='0'&&QNP1<bit>0</bit>=='1')
 ADD AC,AC,BR<>
 MERGE Result,AC,QR<>
 MERGE Result,Result,QNP1<>
 ASHR Result<>
ELSE
 MERGE Result,AC,QR<>
 MERGE Result,Result,QNP1<>
 ASHR Result<>
<BREAK>
 SUB SC,SC,ONE<>
 <PRINT>"_____"
  [STOP]
  UNTIL(CMP,SC,ZERO==false)
  MERGE Result,AC,QR<>
```

Now when we insert this script in our software then our software will produce feasibility study report in log form as well as in tabular form using which we can verify whether our hardware is feasible or not.

V. WORKING OF SOFTWARE VIRTUAL HARDWARE

In order to implement Booth multiplication hardware algorithm we insert the script given above in our software and press simulate button. By doing so feasibility study report is generated in form of micro-operation log and in form of feasibility study table which list the process of calculation, using which we can easily observe that Booth is feasible since it yields correct output using basic micro-operations. The Screen Shot of our software is given below. Moreover we also list our feasibility study reports.

© 2015, IJARCSSE All Rights Reserved

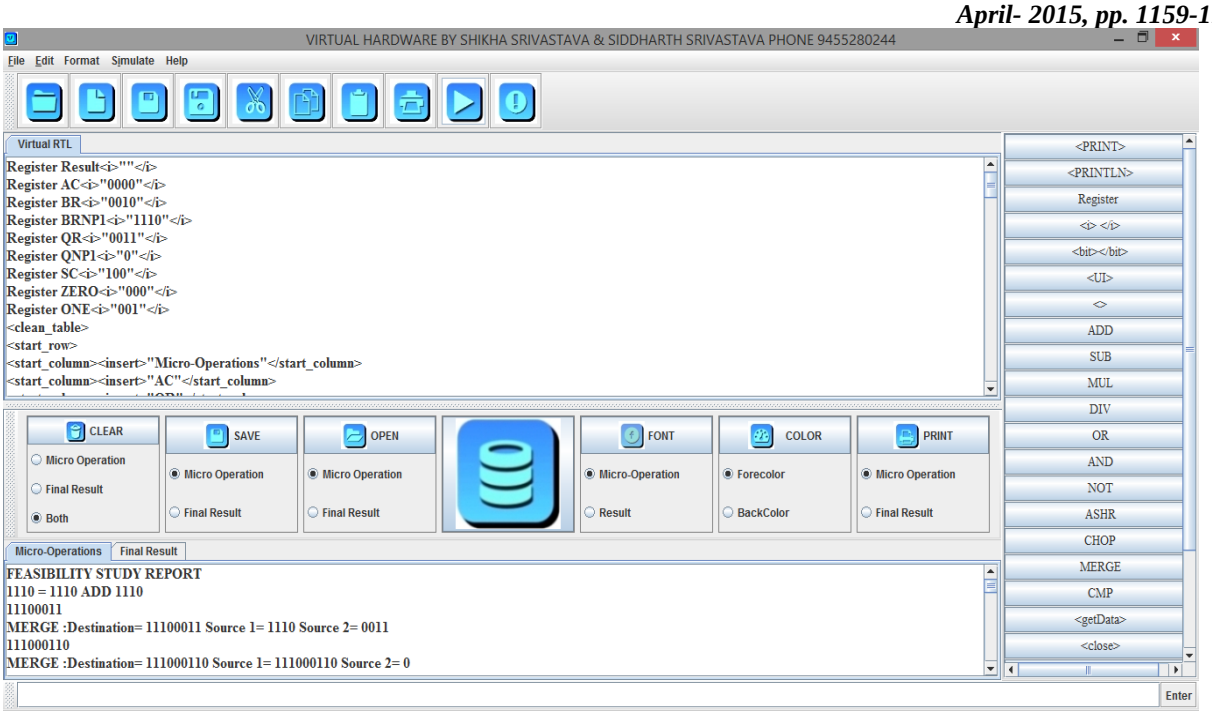

Figure 3. Showing GUI of Software Virtual Hardware Developed By Shikha Srivastava and Siddharth Srivastava.

When we simulate the script then software generates the feasibility study report which is given below.

FEASIBILITY STUDY REPORT (Micro Log)
1110 = 1110 ADD 1110
11100011
MERGE :Destination= 11100011 Source 1= 1110 Source 2= 0011
111000110
MERGE :Destination= 111000110 Source 1= 111000110 Source 2= 0
ASHR=111100011
1
011 = 011 SUB 001

11110001
MERGE :Destination= 11110001 Source 1= 1111 Source 2= 0001
111100011
MERGE :Destination= 111100011 Source 1= 111100011 Source 2= 1
ASHR=111110001
010 = 010 SUB 001

0001 = 0001 ADD 0010
00011000
MERGE :Destination= 00011000 Source 1= 0001 Source 2= 1000
000110001
MERGE :Destination= 000110001 Source 1= 000110001 Source 2= 1
ASHR=000011000

00001100
MERGE :Destination= 00001100 Source 1= 0000 Source 2= 1100
000011000
MERGE :Destination= 000011000 Source 1= 000011000 Source 2= 0
ASHR=000001100

MERGE :Destination= 00000110 Source 1= 0000 Source 2= 0110
FINAL RESULT: 00000110

When we analyze this report then we find that BR=0010; QR=0011; initially and the result produced is 00000110 which is produced using basic micro-operations hence making booth multiplication hardware feasible. Using this software we can easily analyze the feasibility of any hardware.

We can also see the tabular feasibility study report using which we can see the feasibility in one go just by checking the final answer.

© 2015, IJARCSSE All Rights Reserved

RESULT | Table

HEAD1	HEAD2	HEAD3	HEAD4	HEAD5
Micro-Operations	AC	QR	QR+1	SC
Initial Values	0000	0011	0	100
AC=AC+BRNP1	1110	0011	0	100
ashr AC,QR,Qn+1	1111	0001	1	100
ashr AC,QR,Qn+1	1111	1000	1	011
AC=AC+BR	0001	1000	1	010
ashr AC,QR,Qn+1	0000	1100	0	010
ashr AC,QR,Qn+1	0000	0110	0	001
Result	0000	0110	0	000

Figure 4. Showing Feasibility Table Generated by Software Virtual Hardware Developed by Shikha Srivastava and Siddharth Srivastava.

From table shown above we can see that result is 00000110 and initial value of BR=0010; QR=0011; since this table is formed using log generated above so it shows tabular form of feasibility study report.

VI. CONCLUSIONS

This software entitled "Virtual Hardware" can be used to find feasibility status of any logical hardware before actual fabrication of hardware starts. Using this software consistency of any hardware can easily be checked. Moreover using this software we can easily reduce the cost of hardware fabrication since its feasibility study can be done without making the actual hardware prototype. So the cost of developing hardware prototype gets eliminated. Since software generates feasibility study report prior to the fabrication of any hardware prototype so the hardware project development time is reduced.

ACKNOWLEDGMENT

God did all for us and make us the nimit matra of it. This journal is also the outcome of this line. Readers can contact Mr. Siddharth Srivastava at xenonion@hotmail.com and Miss. Shikha Srivastava at shikha3081996@gmail.com. We want to say that with the grace of God all students of National P.G. College supported each other in field of studies and they did their very best in order to publish good Journals.

REFERENCES

[1] M. Morris Mano, Digital Design, Third Edition, Prentice Hall of India, Private limited.
[2] M Morris Mano, Computer System and Architecture, third edition,EEE.
[3] James Adward Keogh, J2ME the Complete Referance, TMH
[4] Herbert Schield, Swing the beginners guide, TMH.
[5] Herbert Schield, java the complete reference, TMH.
[6] Herbert Schield, C++ the complete reference, TMH.
[7] Siddharth Srivastava, "Virtual Technology", 2012, Lambert Academic Publication.
[8] Siddharth Srivastava and Umesh Chandra Jaiswal; (November 2011) Virtually Modified Multi-Protocol Label Switching, International Journal of Engineering Research &Indu. Appls. (IJERIA). ISSN 0974-1518, Vol.4, No. IV (November 2011), pp. 165-192
[9] Siddharth Srivastava and Umesh Chandra Jaiswal; (July 2011) Study of State Load Dispatch Center and EDI based Software Tree List, International Journal of Engineering Science and Technology (IJEST), ISSN : 0975-5462, Vol. 3 No. 7 July 2011, pages 5818-5824
[10] Yashwant Karnetkar, Let Us C, BPB.

© 2015, IJARCSSE All Rights Reserved