PERFORMANCE ANALYSIS OF SQUARE ROOT EVALUATOR USING ANCIENT INDIAN MATHEMATICS

PiyushRaut[1], SubodhMeshram[2], SonaliGuddade[3]

[1]Electronics Department, Dr. Babasaheb Ambedkar College of Engineering and Research
[2]Electronics Department, Dr. Babasaheb Ambedkar College of Engineering and Research
[3]Electronics Department, Dr. Babasaheb Ambedkar College of Engineering and Research

Abstract

Square root is vital operation in many computer graphics applications and scientific calculation processors. The growing demand for faster calculation of arithmetic operations and the power consumption leads to the analysis of square root using ancient Indian techniques and also implementation on hardware. This paper focuses on designing and analysis report of square root algorithm based on ancient Indian mathematics i.e. Vedic mathematics. Vedic mathematics is well known for its finding faster results, both for mental calculation and hardware design. The square root for 8-bit input has been implemented on Xilinx Spartan-6 FPGA. The synthesis report shows number LUTs used and combinational delay which is efficient in term of area and speed as compared to conventional methods.

Keywords: *Vedic Mathematics, Duplex Method, Square root.*

--***--

1. INTRODUCTION

Square root is a very useful and necessary operation in scientific computation. It also holds importance in graphics uses in computer applications and games and also signal processing applications [4]. Many researcher already designed square root evaluator using different techniques, which causes longer delay and utilizes more number of LUTs. The traditional methods are also used for computing the square root such as Babylonian method, digit by digit calculation, exponential identity, Bakhshali approximation, taylor series method etc.

This research uses Duplex method based on ancient Indian literature and Vedic Mathematics for evaluating square root. Vedic Mathematics, written by Sri Bharati Krishna Tirthaji Maharaja, contain sixteen simple mathematical formulae from the Vedas**.**

Proposed Vedic square root technique

We are evaluating square root of 8-bit input using duplex method.

2. VEDIC DUPLEX METHOD

Duplex method are be used to calculate square root of bigger numbers. It also offer bit by bit algorithm for evaluating square root of number. It is called *dwandwa yoga* in Veda's or duplex method which is used by bit by bit algorithm[6].

The method for finding duplex is given below:

D denotes Duplex.

Results of examples are binary numbers which are of base 2.

For a single digit 'a'

- **D**= a^2.

E.g. **D**(1) = 1
For a 2-digit number of the form 'ab',

- **D** = 2(a x b)
  E.g. **D**(11) = 2(1x1) = (10) base 2.

For a 3-digit number like 'abc',

- **D** = 2(a x c) + b2.

E.g. **D**(101) = 2(1x1) + 0 = (10) base 2.
For a 4-digit number 'abcd',

- **D** = 2(a x d) + 2(b x c)

E.g. **D**(1011) = 2(1x1) + 2(0x1) = (10) base 2.
and so on.

Let take the number whose square root is to be calculated is 150 and in binary is (10010110), by duplex method first we have to divide the bits in pairs of two bits starting from LSB. We have to use the last two bit MSB and take the nearest square root of MSB bits 10 is 1. And write the 1 in first place of quotient bit which is MSB of square root.

```
          1   0 : 0   1   0   1   1   0
 :
1 :
```

Now take the twice of MSB quotient bit which will became 10 as divisor for rest of the process and kept in left most of middle row. And subtract the square of 1 from MSB 10 and write remainder on right side of colon of middle row.

```
1   0 : 0   1   0   1   1   0
 1   0   |        : 1

 1  :
```

Concatenate remainder 1 with next left input bit which become the new dividend. Now divide 10 by divisor. This gives 1 as quotient and 0 as remainder in which 2nd quotient bit of square root is 1, and remainder concatenate to next input bit 1.

1 0 : 0 1 0 1 1 0
1 0 | : 1 0
1 : 1

Now 01 become the new gross dividend, but which is not actual dividend we have to subtract duplex of quotient bit from it. So the duplex of 1, D(1)= 1 and after subtraction actual dividend become is 00. Now divide 00 by 10 gives 0 as quotient 0 as remainder, in which quotient of it is the 3rd quotient bit of square root. Again concatenate remainder with next bit which will become 00.

1 0 : 0 1 0 1 1 0
1 0 | : 1 0 0
1 : 1 0

Now 00 is not also actual dividend we have to subtract duplex of 2 bit quotient (1 0). D(1 0)=2*1*0=0 after subtract duplex from 00 gives 00 which will became new actual gross dividend. Now divide it by divisor 10.

1 0 : 0 1 0 1 1 0
1 0 | : 1 0 0
1 : 1 0 0

Division gives 0 as quotient and 0 as remainder. Now we are finding square root of 8 bit input so the square root obtain will be of 4 bit .thus, the forth quotient bit is the last bit square root. The square root of input 150 (binary 10010110) as 12 (binary 1100).

In this way we can obtain square root for different input bits.

3. MODIFIED DUPLEX METHOD

This method differs from basic duplex method the flow chart shows below for the modified method [4].

This method divides the input bits into groups of 4-bits and group of 2-bits, and then used the duplex method for finding the square root of group and whole.

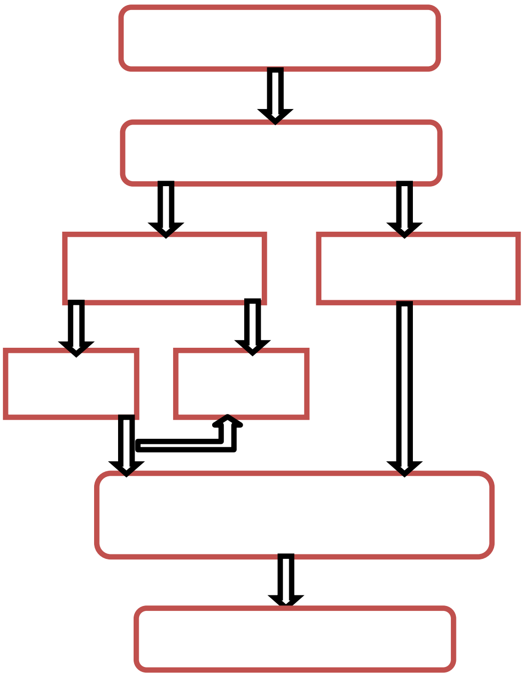
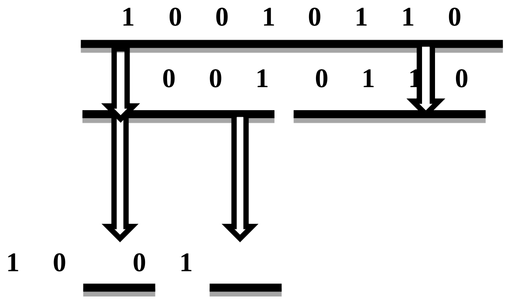


Fig. 1 Modified duplex method

The modified method for evaluating square root for number 150 (binary 10010110) and the process is shown below[4]. First divide the 8-biy input into group of two 4-bit number upper and lower 4-bit.then again upper 4-bit is divided into two 2-bits each.



So let's take the 10 pair of bits, take the nearest square root of it that is 1 write in the 1st quotient bit of square root of upper 4-bit. The square root of upper 4-bit is of 2 bit. After do the same process as for the duplex method.

The square root of upper 4-bit is of 2 bit. After do the same process as for the duplex method.

1 0 : 0 1
1 0 | : 1 0
1 : 1

Now the square root of 1001 is 11, and it is used to find the square root of 8-bits. The upper 4-bit write in to left side of column and remaining lower 4-bits on right side of it.

1 0 0 1 : 0 1 1 0
1 1 0 | : 0 0 0
1 1 : 0 0

As the square root of 1001 is 11, then we have to take twice of it and make divisor for all process that is 110. Square of11 is subtracted from 1001 upper 4-bit, remainder of subtraction is 0, and this 0 is concatenated with next bit and form the new dividend. Now this dividend 00 is divide by divisor 110 get the quotient as 00 and remainder as 0. As we require the square root of 8-bit input we got 4-bit output for it.

The square root by using modified duplex method of 150 (binary 10010110) is 12 (binary 1100).

4. RESULT

Table 1.FPGA design summary of duplex method

Number of Slice Registers	10
Number of Slice LUTs	19
Number of bonded IOBs	12
Delay	7.119 ns
Power consumption	0.155 W

Table 2. FPGA design summary of modified duplex method

Number of Slice Registers	10
Number of Slice LUTs	20
Number of bonded IOBs	12
Delay	6.553 ns
Power consumption	0.154

5. SIMULATION RESULT

The simulation has been done using the ISE simulator. The simulation result ofthe square root of 150 (binary 10010110) is shown in Fig. 2 and Fig.3

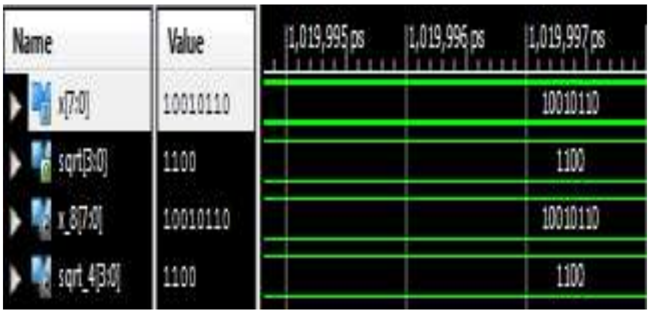


Fig. 2. Simulation result of duplex method

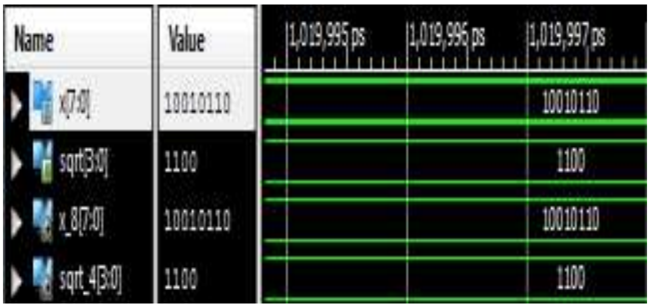


Fig. 3. Simulation result of modified duplexmethod

6. CONCLUSION

Fast circuit are required to achieve parallel processing in digital system hence fast square root evaluator are required. Simulation result shows that delay required for modified duplex method i.e. 6.553ns is less than that of duplex method i.e. 7.119ns. Hence the modified duplex method is quite faster than duplex method. However number of LUT's and input output blocks utilization is similar, and the power consumption is also similar.

REFERENCES

[1]. Banerjee, A., Ghosh, A. and Das, M. (2015) "High Performance Novel Square Root Architecture Using Ancient Indian Mathematics for High Speed Signal Processing". Advances in Pure Mathematics, 5, 428-441.

[2]. ToleSutikno, AimanZakwanJidin, AuzaniJidin and NikRumziNikIdris "Simplified VHDL Coding of Modified Non-Restoring Square Root Calculator", ISSN: 2089-4864 Vol. 1, No. 1, March 2012, pp. 37~42.

[3]. International Journal of Industrial Electronics and Electrical Engineering, ISSN: 2347-6982.

[4]. JaspreetKaur ,Nirmal Singh Grewal "Design and FPGA Implementation of a Novel Square Root Evaluator based on Vedic Mathematics", ISSN 0974-2239 Volume 4, Number 15 (2014), pp. 1531-1537.

[5]. PradnyaHarde, R.V. Kshirsagar"Performance analysis of various multipliers using vhdl" P.G.9-12 VCAN proceeding,2015.

[6]. www.vedantree.com

[7]. www.xilinx.com