Design and Implementation of Run-time reconfigurable multi-precision FP multiplier

M V Sarala Rani¹, Pradeep S V² and Dr Siva Yellampalli³
¹,²Lecturer, VTU Extension Centre, UTL Technologies Ltd, Bangalore
³Principal, VTU Extension Centre, UTL Technologies Ltd, Bangalore

Abstract—Floating Point (FP) multiplication is used based on the need of the application and this work has showed the reconfigurable FP multiplier with six different modes. The different modes give different precision as the modes are divided based on the number of mantissa bits. The complex part of FP multiplication is the Mantissa multiplication that is done with the efficient use of both the Karatsuba and Urdhva algorithm. Here, we had targeted for the decrease in the delay and area and it has been achieved successfully with usage of simple carry select and save adders instead of the ripple carry adders. The fully fledged Double precision multiplier actually consumes more area and also delay is more, here we have reduced the delay by 47 percent approximately in comparison to the reference.

I. INTRODUCTION

FP multiplication is mostly used in many of the multimedia applications and complex computing systems such as DSP, signal processing, image processing etc. The research has started many years back for the FP multiplication many years back as it had involved more area and more power. The area consumption and power consumption increases with the increase of accuracy and also this increases the complexity. The small error in the precision also may result in the major failures of many of the systems that could cause failures which can never be recovered. The standard IEEE format used for floating point multiplication is IEEE-754 and this may not be suitable for all kinds of applications. Even a minute mistake in the precision can cause immense effects. These mistakes are possible in the units of the floating point mainly since the discrete behavior of these IEEE 754[1] as the floating point representation, where the fixed number of the respective bits are herein used. The precision can be represented in the single and double format. The precision varies for various applications.

Some applications does not require high precision but some require low precision. Some applications may use single, some double and some quadruple precision but they all vary on applications and they consume more area and power. For the devices, which are portable or the wearable type devices wherein requirement of the precision does varies with the various known applications and even power is the significant factor, practical applications of the greater precision FP in multipliers cannot be the better option. In such of the cases, the variable precision multiplier does be better option which does immensely save power and even time when the respective application does not need the greater precision. There are lot of such models like [2][3] and [4]. Most of the represented designs utilizes already available IPs which constitute the units of the DSP and 18*18 multiplier type units.

In this paper, we herein present the design of the power efficient FP multiplier with the different modes as part of selection of precision. With various modes of the precision, we do choose the mode where there is an appropriate for the required applications. With precision requirements, the width of the multiplier increases and thus increases the area.

Fig 1: Multiplier Unit
II. SUGGESTED MODEL

The suggested model can be represented as the reconfigurable FP multi precision multiplier which does be operated in six of the different respective modes in accordance with the precision requirements. It can perform the format of the FP in multiplication of the various sizes of the mantissa reckoning on the requirement of the precision. The fundamental recognized unit is the double precision FP of the unit. In accordance to the precision chosen, the size of the respective mantissa is herein varied. Fig3 does exhibits the FP multiplication formats, which are considered in the suggested model.

The block diagram of this work is as below,

![Fig 2: Block Diagram](image)

From the above block diagram, we could see that the main blocks are:
- Input Registers
- Mode Select
- Normalization (Truncation and Rounding)
- Floating Point Multiplier

The multiplier accepts two of the respective inputs wherein there is sixty seven of wide. The first three bits are utilized for the selection of the mode. The respective inputs to the multiplier can be provided in the double precision format of the FP as the respective format with three bits as the mode of the chosen bits.

![Fig 3: FP format considered](image)

III. KARATSUBA URDHVA TIRYAGHYAM TYPE ALGORITHM

In the FP multiplication, most significant and complex part is the mantissa multiplication. Multiplication operation does require more time in comparison to the summation. As the number of bits increases, it herein increases more area and also more time. In the format of double precision, we need 53x53 of the bit multiplier and in the respective single precision format we do need 24x24 bit as the multiplier. It does require much time to perform the respective workability and it is the major known contributor to the delay of the FP multiplier. To make the multiplication in operation more of area efficient and also faster, the suggested type model utilizes the combination of the karastuba and also the Tiryaghyam algorithm.

Karatsuba algorithm utilizes a divide and conquer approach, where it does breaks down the respective inputs into most of the significant half and also the least significant half and this process is continued until the respective operands are of the eight bit wide. This
algorithm can be best suited for the respective operands of the greater bit length. But at lower bit lengths, where it is not as efficient as it is of the greater bit lengths, where it is not found efficient as it is at the greater bit lengths. To eliminate this respective problem, Urdhva Tiryabhyam algorithm is considered at the lower recognized stages. This is the best known for the respective binary multiplication which in terms of the area and delay. But as the number of respective bits increases, delay also increases as the partially represented products, which are summed in the ripple manner. For an instance, four bit multiplication, where it does requires six of the adders associated in the ripple manner. And eight bit multiplication does require fourteen of the adders and following. Compensating the respective delay, it does cause increase in the respective area. Hence, this Urdhva Tiryagbhyam type cannot be choicest if the respective numbers of bits are more. If we consider Karatsuba at the greater stages and this type in the lower stage, it does manages to atone the restrictions in both and thus multiplier becomes more productive. The circuits are further optimized by considering the carry select and carry save adders instead of ripple carry adders. Hence, this minimizes the delay to the great extent with the minimal increase in the respective hardware.

Urdhva Tiryagbhyam sutra is an ancient Vedic mathematics algorithm for the multiplication [6]. It is the general known formula which is applicable to all the respective cases of the multiplication. It is the clearly regulated formula, which is applicable to all the known cases of the multiplication. The whole formula is very short and comprises of only one of the compound word and it does mean vertically and crosswire. In this Urdhva Tiryagbhyam type, various number of steps are required for the multiplication.

Fig 5: Line type notation
IV. KARATSUBA ALGORITHM FOR MULTIPLICATION

Karatsuba multiplication type is basically used here for multiplying very large numbers. This whole method is divide and conquer method, wherein we divide the numbers into their most of the significant half and least of the significant half and then multiplication is extensively performed. Karatsuba algorithm decreases the number of the required multipliers by eliminating the multiplication workability by the adders. Adders are faster than the multipliers. As the number of the respective bits of those inputs is increased, Karatsuba algorithm does become an efficient. This algorithm is known to be best if widths of those inputs are more than the sixteen bits.

V. RESULTS NORMALIZATION

Results being normalized and have hidden bit in the corresponding mantissa respectively, which does always has the value one and hence it is not reserved in the memory to save one of the bit. A leading one in the mantissa is known to be unseen bit where in it is one just immediate the left of the respective decimal point. Usually normalization is performed by basically shifting hence the MSB of the respective mantissa does becomes non zero and it is in the radix two, non-zero does means one. The decimal point in the mantissa multiplication once shifted, the value of the exponent is raised by one. This is represented as the normalization of the number, since the value of the unseen bit is always equal to one.

VI. ENACTMENTS AND RESULTS

The main objective of this respective work is basically to design and implement FP in variable precision of the circuit which the device does reconfigure self regularly in congruence to the respective requirements of the precision and does operates at greater agility which is irrespective of the precision and also consumes less power where the precision is not an issue.
Since the mantissa type multiplication is the most complex part in the respective FP multiplier, does operate at the higher speed and decreases the delay and corresponding area is remarkably less than the augment number of the bits. The FP multipliers herein of the various modes with the IEEE-754 standard recognized format and the custom type precision format is implemented in the separate manner utilizing the Verilog HDL and even tested further.

The binary units of the multiplier are herein further optimized by basically replacing simple adders with efficient adders like carry select adders and also the carry save adders. The suggested model is implemented and simulated with Xilinx synthesis (ISE 14.7) which is targeted on Virtex4 of the family. The model which does operates in the chosen mode only and also during the respective operation, only the chosen multiplier unit does be at state of ON. Hence, if there is lesser precision of the mode which is chosen, the area and also power does be lesser.

The below figure shows the simulation of fully fledged FP multiplier as usually the fully fledged consumer more area and power and more delay also.

The synthesis results of the above simulation are shown in the figure below:

![Fig 8: Simulation of Fully Fledged Double Precision Multiplier](image)

The comparison of the results achieved in this work with the reference is shown as below:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Reference[17]</th>
<th>This work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slices</td>
<td>3877</td>
<td>469</td>
</tr>
<tr>
<td>LUTs</td>
<td>4033</td>
<td>862</td>
</tr>
<tr>
<td>IOBs</td>
<td>193</td>
<td>204</td>
</tr>
<tr>
<td>Delay(ns)</td>
<td>18.966ns</td>
<td>10.13ns</td>
</tr>
<tr>
<td>fmax(MHz)</td>
<td>173.952MHz</td>
<td>55.3MHz</td>
</tr>
</tbody>
</table>

![Fig 10: Comparison of the Results of Fully Fledged Double Precision Multiplier](image)

VII. CONCLUSION AND FUTURE WORK

This project gives an opportunity to effectively adjust the delay and consumption of the power for the different precision in the requirements depending on the application. Also, the whole project does shows how to effectively decrease the percentage increase in the delay and the area of the respective FP multiplier with an increase in the number of the respective bits by considering very efficient combination of the Karatsuba and Urdhva Tiryagbyham type. The whole model can be furthered optimised respectively in terms of the delay by considering the concepts like pipelining and the precision of the results can be increased by adding an
efficient methods of normalization and the methods of rounding.

REFERENCES


