Enhanced Fast Fourier transform Using Approximate Radix 8 Booth Multiplier

Boojith Krishna V.
2018103016

Subramanian V
2018103606

Deepak V.M
2018103522

INTRODUCTION

The Fast Fourier Transform (FFT) is an efficient algorithm to calculate the Discrete Fourier Transform (DFT), which is often employed in Digital Signal Processing (DSP) and communication. Fourier Analysis converts a signal from its origin domain to frequency domain. Fourier transform has a wide range of applications. But calculating it directly from the the definition is very slow. FFT reduces the complexity of computing Discrete Fourier transform from $O(N^2)$ to $O(NlogN)$ where N is the data size. The difference in speed can be enormous, especially for long data sets where N may be in the thousands or millions. The applications often do not require precision but demands high speed computation. In such cases multiplications in FFT can be accelerated with approximate computing using probabilistic radix 8 booth multiplier. The proposed architecture is to be implemented in FPGA and the performance is to be demonstrated using image filtering in frequency domain.

OVERALL OBJECTIVES

Approximate computing is a concept applied to error-tolerant applications in which the accuracy of an operation is reduced to improve other measures of circuit performance.Accuracy are typically acceptable in applications such as digital signal processing, image processing, data mining, and pattern recognition Use of approximate circuits allow for improvements in performance measures such as power, area, delay. The objectives include:

1. To design approximate booth multiplier with improved speed and reduced area
2. Enhance the speed in Fast Fourier Transform using the Approximate Booth Multiplier designed
3. Improved image filtering using Enhanced Fast Fourier transform and Approximate radix-8 Booth multiplier.

LITERATURE SURVEY

To reduce hardware complexity of multipliers, truncation is widely employed in fixed-width multiplier designs. Then a constant or variable correction term is added to compensate for the quantisation error introduced by the truncated part. Approximation techniques in multipliers focus on accumulation of partial products, which is crucial in terms of power consumption. Broken array multiplier is implemented in, where the least significant bits of inputs are truncated, while forming partial products to reduce hardware complexity. The proposed multiplier in saves few adder circuits in partial product accumulation[1].

Truncation error in existing fixed width multipliers is high. better error compensation bias is derived to reduce truncation error by properly choosing generalised index. A lower error fixed-width multiplier, for VLSI realisation is constructed in [2]. FIR for filter for speech processing is designed using the proposed model. The performance for consonant part is better than that using other fixed-width multipliers.

The mean error in truncation is not distributed symmetrically. An effective error compensation function that makes the error distribution more symmetric and centralised in the error equal to zero, leading the fixed-width modified Booth multiplier to very small mean and mean-square errors is proposed in [4]. The proposed error compensation circuits offer at least 12.3% and 6.3% reductions in mean-square error over the previous circuits, respectively

Reliability is analysed using sequential probability transition matrices (SPTMs) in [5]. New metrics are proposed for evaluating the reliability as well as the power efficiency of approximate and probabilistic adders.

The error detection, compensation and recovery circuits of the approximate multiplier is simplified.in [5] The proposed design outperforms the other approximate multipliers in the FIR filter operation, thus this design may be useful for low-power and imprecise operations in error-resilient systems.

An accuracy-adjustment fixed-width Booth multiplier that compensates the truncation error using a multilevel conditional probability is proposed. Multiplication technique is mathematically modelled to allow various levels of approximation and performance trade-offs. Approximation is implemented such that the 6-input lookup table (LUT) and the associated carry chains of the FPGAs are fully utilised in [8]. The proposed multipliers are applied to the application of Sobel edge detection. The proposed solution detected 98.45% edges with energy savings of 26.41%.

AREA AND DELAY ENHANCEMENT

- The usual multiplication involves
    - Recoding of multiplier bits
    - Generation of partial products
    - Partial Product reduction tree
    - Vector merge addition

- Partial products are generated using AND gates. An n x n multiplier requires AND gates. Approximation in this stage reduces the area requirement.
- Partial product reduction (n - 2 Compressors) stage introduces huge delay of order O(n) where n is the number of bits in the multiplier.

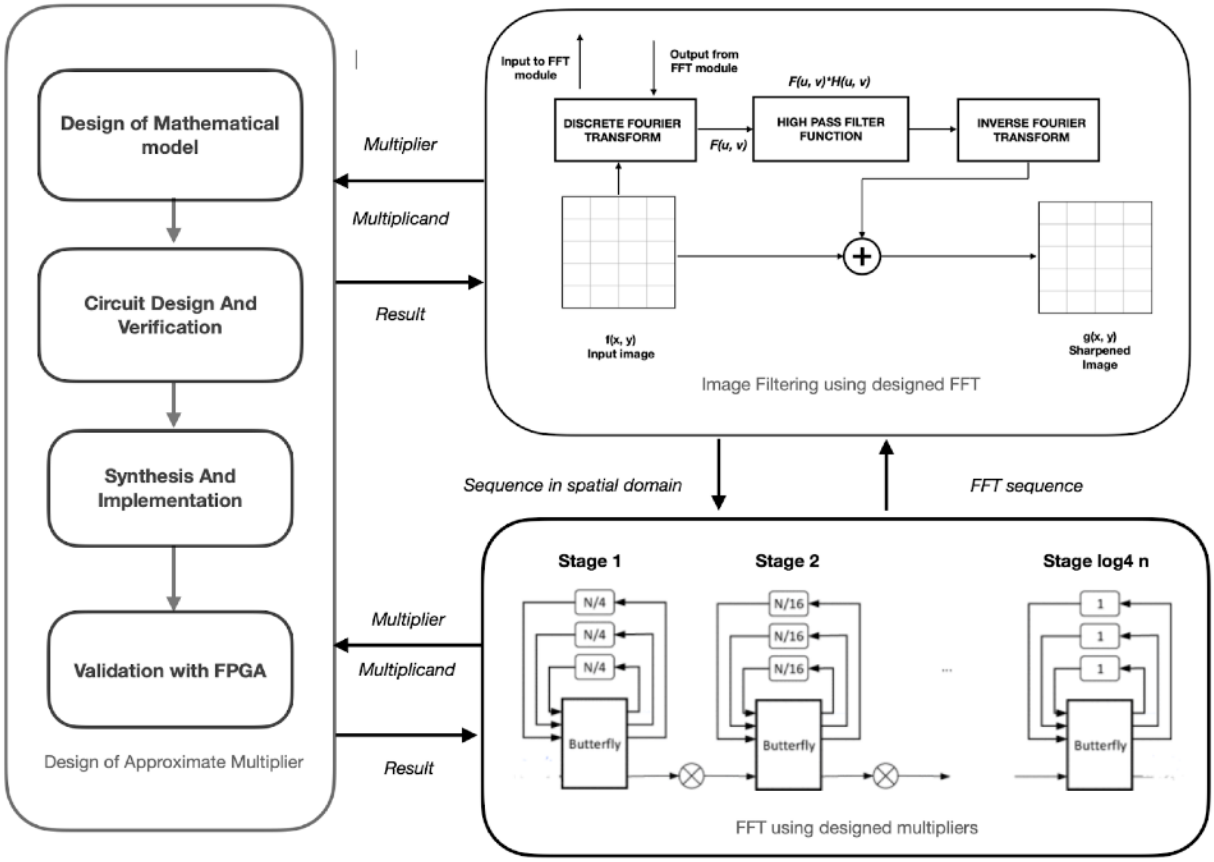
- Partial product reduction using Wallace tree algorithm can reduce this complexity to O(logn). Approximation in this stage can reduce the constant factor.

BOOTH ENCODING AND BIT PAIR RECODING

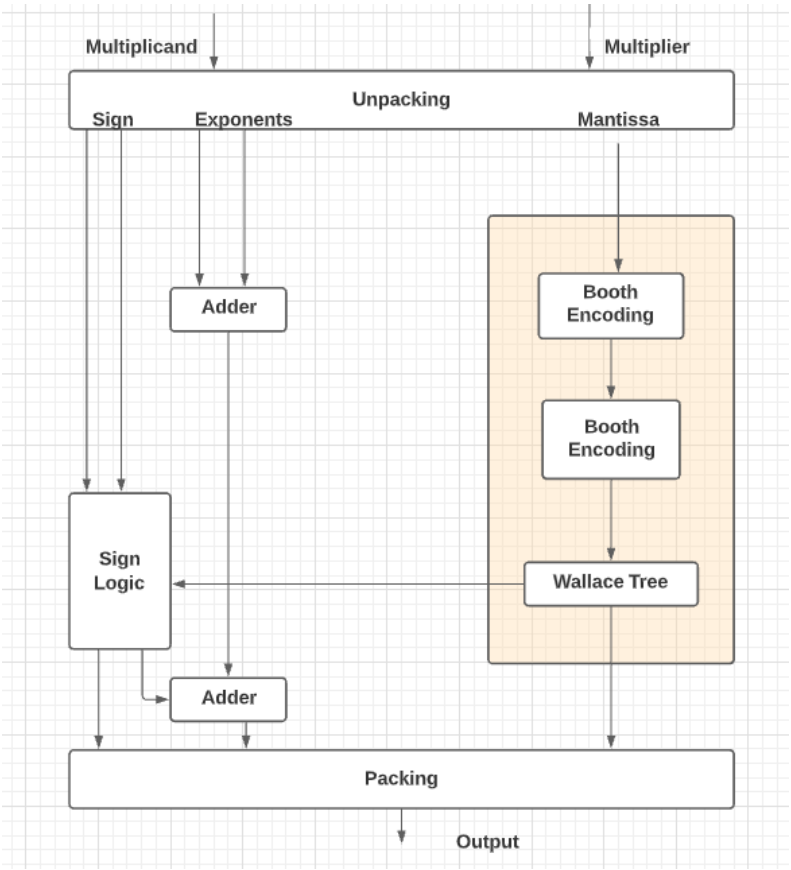
- In Booth multipliers, a higher radix corresponds to a decrease in the number of rows of the partial product matrix.

- For instance, in radix-4 Booth multipliers, partial product generation produces values of 0, 1, and 2 multiplicand and reduces the size of the partial product matrix by nearly half.

ARCHITECTURE DIAGRAM



ARCHITECTURE OF THE MULTIPLIER

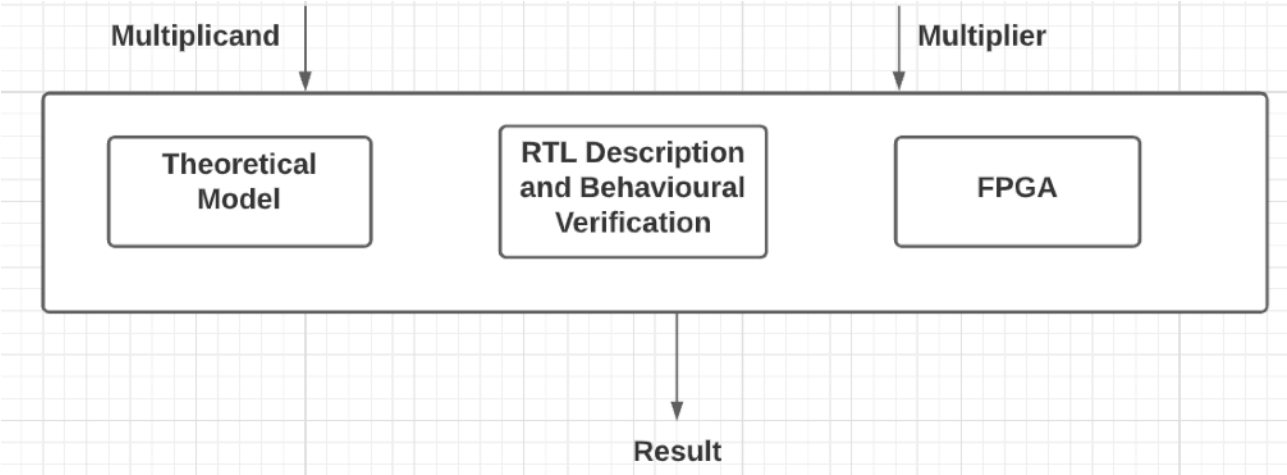


MODULES

I. Design and Analysis of Multipliers
II. FFT using Designed Multipliers
III. Filtering Using FFT

MODULE 1: Design and Analysis of Multipliers

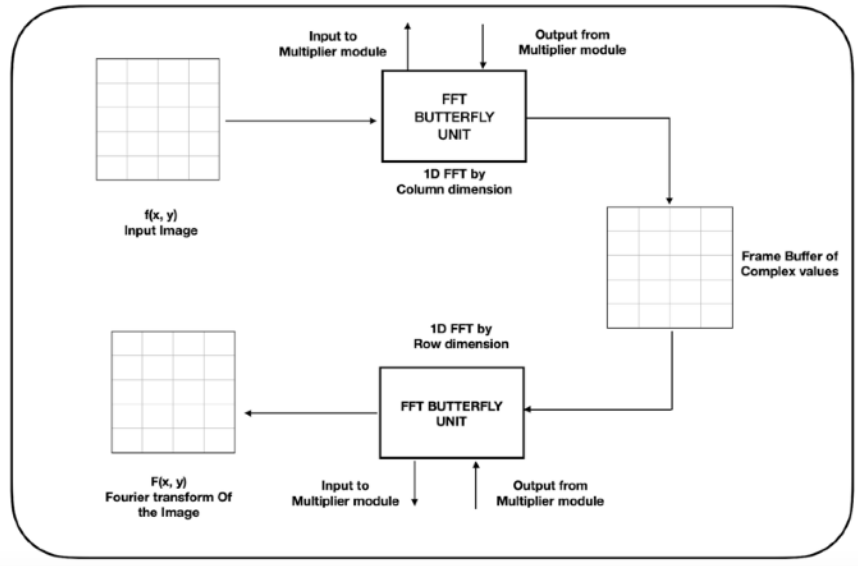
Theoretical modelling of the approximate multiplier architecture using python is done. RTL description of multiplier is carried out using VHDL. It is followed by behavioural simulation and verification of the HDL code. Finally validation is done using FPGA.



INPUT: MULTIPLIER AND MULTIPLICAND
OUTPUT: RESULT

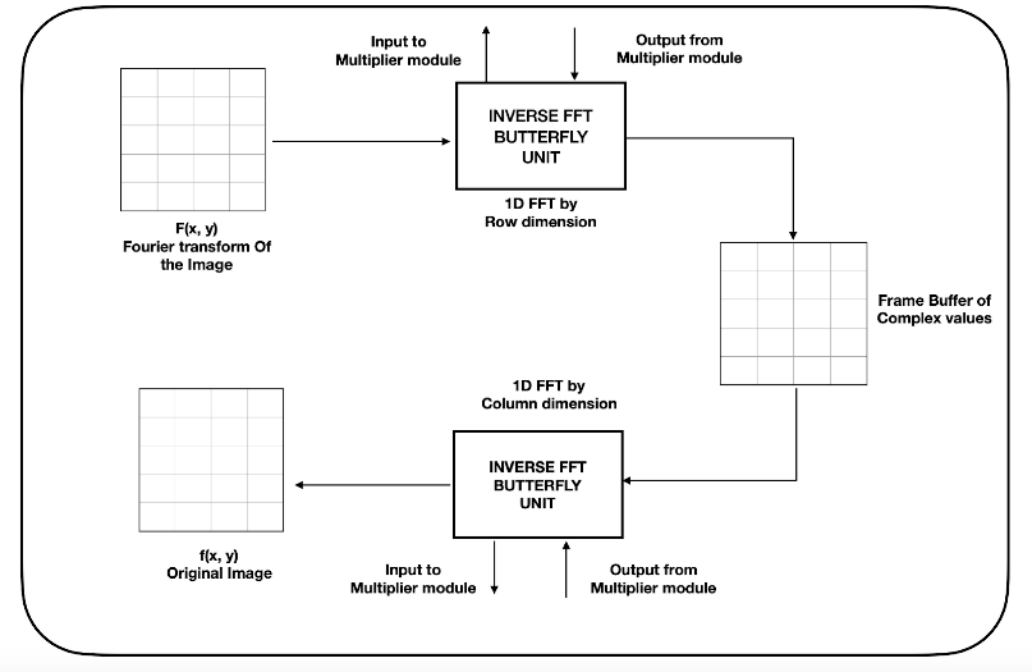
MODULE 2: FFT Using Designed Multipliers

The hardware implementation of parallel FFT is directly mapped from the butterfly scheme. 1D FFT is calculated using butterfly unit along column dimension followed by row dimension to obtain 2D FFT.



FFT Using Approximate Multipliers

INPUT: IMAGE
OUTPUT: DFT OF THE IMAGE



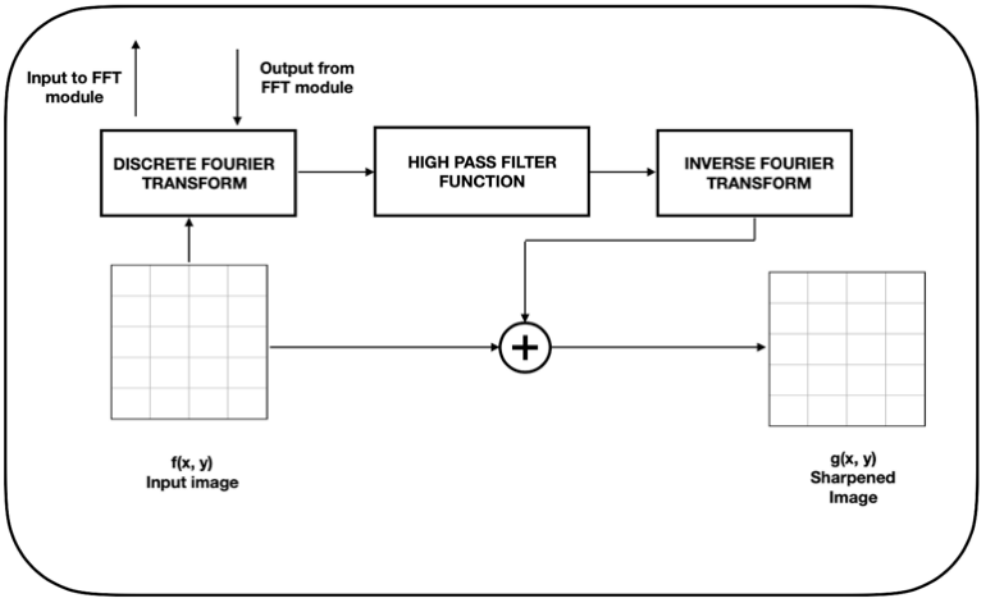
Inverse FFT Using Approximate Multipliers

INPUT: DFT OF THE IMAGE
OUTPUT: ORIGINAL IMAGE

MODULE 3: Filtering Using FFT

The image is converted into frequency domain using the proposed Fast Fourier transform using approximate multipliers. A high pass filter is applied to enhance the edges. The image is converted back to spatial domain using Inverse Fast Fourier transform using approximate multipliers, which is then added with the original image to obtain the sharpened image.

INPUT: IMAGE
OUTPUT: SHARPENED IMAGE



PERFORMANCE MEASURES

Reliability is analysed using the so-called sequential probability transition matrices (SPTMs). Error distance (ED) is initially defined as the arithmetic distance between an erroneous output and the correct output for a given input. The mean error distance (MED) and normalised error distance (NED) are then proposed as unified figures that consider

the averaging effect of multiple inputs and the normalisation of multiple-bit adders.

Following metrics are to be calculated for accurate and approximate radix 8 Booth multipliers and a detailed comparison of the architectures is to be analysed.

Area
Power
Error Measures

ERROR MEASURES

1. Pass rate

$$passrate = \frac{number\,of\,correct\,results}{total\,results}$$

2. Accuracy for amplitude data

$$ACC_{amp} = 1 - \frac{R_c - R_e}{R_e}$$

$$where\,R_e\,is\,the\,result\,of\,proposed\,multiplier$$
$$R_c\,is\,the\,result\,of\,accurate\,multiplier$$

3. Accuracy for information data - measures error significance in terms of hamming distance

$$ACC_{inf} = \frac{1 - B_e}{B_w}$$

$$Where\,B_e - number\,of\,error\,bits$$

$$Where\,B_w - bit\,width\,of\,data$$

REFERENCES

1. Y. C. Lim, "Single-precision multiplier with reduced circuit complexity for signal processing applications," in IEEE Transactions on Computers, vol. 41, no. 10, pp. 1333-1336, Oct. 1992, doi: 10.1109/12.166611.

2. Lan-Da Van, Shuenn-Shyang Wang and Wu-Shiung Feng, "Design of the lower error fixed-width multiplier and its application," in IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 47, no. 10, pp. 1112-1118, Oct. 2000, doi: 10.1109/82.877155.

3. S. -J. Jou, Meng-Hung Tsai and Ya-Lan Tsao, "Low-error reduced-width Booth multipliers for DSP applications," in IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 50, no. 11, pp. 1470-1474, Nov. 2003, doi: 10.1109/TCSI.2003.817779.3.

4. J. Wang, S. Kuang and S. Liang, "High-Accuracy Fixed-Width Modified Booth Multipliers for Lossy Applications," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 19, no. 1, pp. 52-60, Jan. 2011, doi: 10.1109/TVLSI.2009.2032289.

5. J. Liang, J. Han and F. Lombardi, "New Metrics for the Reliability of Approximate and Probabilistic Adders," in IEEE Transactions on Computers, vol. 62, no. 9, pp. 1760-1771, Sept. 2013, doi: 10.1109/TC.2012.146.

6. H. Jiang, J. Han, F. Qiao and F. Lombardi, "Approximate Radix-8 Booth Multipliers for Low-Power and High-Performance Operation," in IEEE Transactions on Computers, vol. 65, no. 8, pp. 2638-2644, 1 Aug. 2016, doi: 10.1109/TC.2015.2493547.

7. S. Venkatachalam, E. Adams, H. J. Lee and S. Ko, "Design and Analysis of Area and Power Efficient Approximate Booth Multipliers," in IEEE Transactions on Computers, vol. 68, no. 11, pp. 1697-1703, 1 Nov. 2019, doi: 10.1109/TC.2019.2926275.

8. H. Waris, C. Wang, W. Liu and F. Lombardi, "AxBMs: Approximate Radix-8 Booth Multipliers for High-Performance FPGA-Based Accelerators," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 68, no. 5, pp. 1566-1570, May 2021, doi: 10.1109/TCSII.2021.3065333.