International Journal of LNCT
Volume-2 Issue-8 October-2018
ISSN: (Online):2456-9895

An Extensive Survey on Fault Tolerant Parallel FFTs Using Error Correction Codes and Parseval Checks

Neha Singh[1], Prof. Prashant Purohit[2]

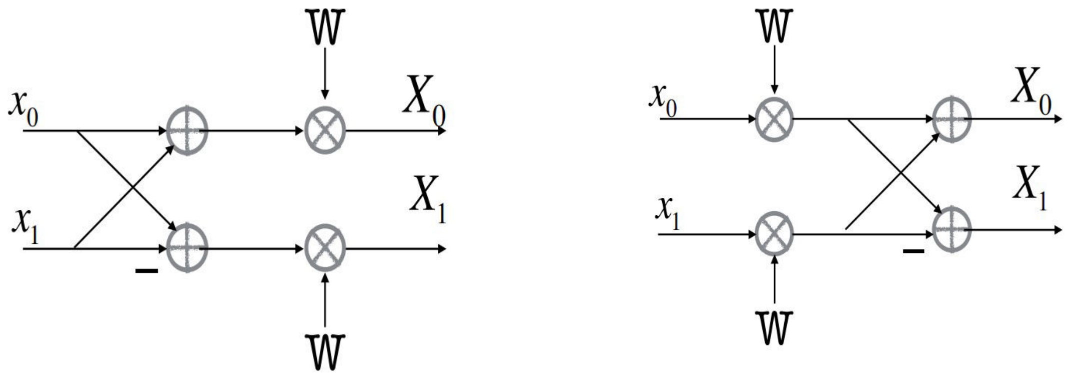
M.Tech Scholar[1,] Research Guide[2], Department of ECE, LNCT College, Bhopal

Abstract-The Fourier Transform or FFT is a mathematical operation generally utilized in many fields like image processing, digital signal processing etc. It is an important tool which is used to execute complicated calculations in many processors. In image processing applications FFT is utilized for image restoration, image filtering, image reconstruction and image analysis. High performance computing is undergoing a significant transformation, in the sense that performance is no longer the sole consideration while developing applications. Fault- tolerance, which is defined as providing dependable service in spite of faults occurring or having occurred, is becoming equally important as performance. The primary goal of this examination is to analyze a computation for the Fast Fourier Transform so it will figure the Fourier Transform significantly faster and efficient for fixed length data.

Keywords- FFT, Parallel Processing, Error Correction Codes, Fault Tolerant Design.

1. Introduction

The Discrete Fourier Transform (DFT) is one of the most widely used digital signal processing (DSP) algorithms. DFTs are never registered specifically, yet rather are computed utilizing the Fast Fourier Transform (FFT), which involves an accumulation of algorithms that efficiently ascertain the DFT of a succession. The quantity of utilizations for FFTs keeps on growing and incorporates such different areas as: interchanges, signal processing, instrumentation, biomedical designing, sonics and acoustics, numerical strategies, and connected mechanics.Fast Fourier transform (FFT) is an efficient algorithm to compute the discrete Fourier transform (DFT) and its inverse. There are numerous FFT algorithms including an extensive variety of calculation. A DFT deteriorates a succession of qualities into various frequencies parts. This task is helpful in numerous fields however figuring it coordinate registering from its definition is too ease back to possibly be reasonable. An FFT computation gives same result more quickly by computing a DFT of N points in the naive way, using the definition, takes N2 complex multiplications and N(N-1) complex additions, while an FFT can compute the same result in only N/2 log N complex multiplications and N log N complex additions. The difference in speed can be substantial for long data sets where data may be in the thousands or millions.Various higher radix FFT algorithms like Radix-4, Radix-8, Radix-16, Radix-2k, and Split-Radix FFT algorithms have been reported for increasing the performance of FFT cores of small and large data sequences.Before utilizing the radix-2 calculation, the hardware acknowledgment of the 32-point FFT is parallel-in parallel-out, or, in other words the execution, due to the huge measure of the utilization of the adders and the multipliers. The quantity of the info ports for the entire architecture is in excess of 32 ports, i.e. For the chip manufacturing, the pins of the chip would increase at the same time.Typically, the multiplication coefficients in Cooley - Tukey algorithms are called twiddle factors. For the FFT calculation a radix-2 DIF butterfly arrangement is utilized. Another setup is radix-2 DIT butterfly, the distinction between these two algorithms is the region of the twiddle factors. Figure 1.1 illustrates the different position of the twiddle factors respectively.As semiconductor technologies move toward finer geometries, both the available performance and the functionality per die increase. Unfortunately, the power consumption of processors manufactured in propelling advancements additionally keeps on growing. This power increment has brought about the present circumstance, in which potential FFT applications, once in the past restricted by accessible performance, are currently as often as possible constrained by accessible power saving schemes. The recent dramatic increase in the number of portable and embedded applications has contributed significantly to this growing number of power-limited opportunities.



(b)

(a)

Fig. 1.1 the positions of twiddle factors in (a) radix-2 DIF butterfly and (a) radix-2 DIT butterfly.

2. Error Correction Codes

The exponential growth in the sophistication and processing power of computer chips, brought about by miniaturization and increase in CMOS integration density, has been the driving force in technological advancements in the past decades. In addition, today, computing clusters and virtualization provide for significant parallelism possibilities in comparison to conventional desktop multicore systems. These systems, together with the corresponding parallel algorithms that run on them, find numerous applications in multimedia processing systems, especially with the increase in the availability of large datasets and the requirement for real-time data processing. However, despite the throughput acceleration offered by multi-core and multi-processor systems, modern computing clusters are now beginning to exhibit unreliable operation even under nominal conditions. Soft errors are random, non-destructive short lived disturbances in semi- conductor devices, usually caused by electrical noise, fluctuations in signal voltage, inductive coupling effects, particle strikes, clock skew effects, CMOS process variations and other external factors beyond the system designer's control. For example, when charged particles (typically stemming from radioactive materials in chip packaging), cosmic rays and other high energy particles interact with semiconductor memory elements [48, 49], single or multiple bits of the stored value at the memory location are flipped. When these flips go undetected, subsequent computations with these erroneous data elements are likely to propagate silently to the application layer thereby producing erroneous outputs or degrading system performance (cf. Fig.1 of [50]) and in severe cases, failures. Thus, although a soft error may not be damaging to the device in itself, its proliferation as silent data corruption (a.k.a, fail continue failures) poses great reliability challenges for end users. Previously, it was assumed that soft errors were specifically problematic only for space and aviation-based electronic devices since they are continuously exposed to high energy particles (e.g. neutrons from cosmic rays) that generate electron-hole pairs when they come in contact with semiconductor de- vices. These electron-hole pairs result in the appearance of large number of charges such that when a given critical threshold is exceeded, the operating circuit responds by erroneously flipping the state of a logic or memory at the location of the high energy particle impact.

Error Correction Codes (ECC) are all based on the principal of adding extra symbols to a data word to form a code word. These symbols are based on the symbols in the data word and thus provide redundancy such that errors in the data word can be detected or corrected. There are two main categories of error correction codes:

- Block codes: These codes are applied to information on a block-by-block basis. The different blocks are independent of each other.
- Convolutional codes: These codes use the current input information as well as the previous inputs and outputs.

For the memory cells in a softcore, the used symbols are bits and only block codes can be applied. Convolutional codes are difficult/impossible to use because the values in the register file and data memory are (mostly) independent of each other and instructions can also be considered independent from each other although there is some form of dependency between consecutive instructions.

3. Prior Work

SR. NO.	Title	Authors	Year	Approach
1	Fault Tolerant Parallel FFTs Using Error Correction Codes and Parseval Checks	Z. Gao et al.,	2016	Algorithmic-based fault tolerance (ABFT) techniques that try to exploit the algorithmic properties to detect and correct errors.
2	A novel low-overhead fault tolerant parallel-pipelined FFT design	Y. Xie, C. Yang, C. Mao, H. Chen and Y. Xie	2017	A novel low-overhead fault tolerant FFT design, combining modified reduced precision redundancy (RPR) method and error correction codes (ECCs).
3	Parallel pipelined FFT architecture for real valued signals using radix-2	S. K. Mali and M. C. Lakkannavar,	2016	Reported parallel pipelined architecture (PPA) for the RFFT by using a canonical-signed-digit multiplier (CSDM) to optimize the area.
4	Design of parallel FFT architecture using Cooley Tukey algorithm	R. Shirbhate, T. Panse and C. Ralekar,	2015	A parallel FFT architecture is Reported to give an efficient throughput and less energy consumption with the help of Cooley Tukey algorithm for radix 8
5	FFT-based parallel encoding algorithm for structured LDPC codes	Yanan Fan, Xin Meng, Xiujuan Yao and Yi Yan	2015	A new method for the encoding of the structured LDPC codes has been Reported, namely FFT-based parallel encoding algorithm
6	VL-ECC: Variable Data-Length Error Correction Code for Embedded Memory in DSP Applications	J. Park, J. Park and S. Bhunia,	2014	Reported a variable data-length ECC (VL-ECC) for the embedded memory devices of digital signal processor
7	Software-Based Hardening Strategies for Neutron Sensitive FFT Algorithms on GPUs	L. L. Pilla et al.,	2014	Reported specific software-based hardening strategies to reduce its failure rate

Z. Gao et al., [1] Soft errors represent a reliability threat to current electronic circuits. This makes security against soft errors a necessity for some applications. Communications and signal processing systems are no exceptions to this trend. For a few applications, a fascinating alternative is to utilize algorithmic-based fault tolerance (ABFT) strategies that endeavor to abuse the algorithmic properties to distinguish and redress errors. Signal processing and correspondence applications are appropriate for ABFT. One precedent is Fast Fourier Transform(FFTs) that are a key building block in numerous frameworks. A few insurance plans have been accounted for to identify and amend errors in FFTs. Among those, likely the utilization of the Parseval or sum of squares check is the most broadly known.

In present day correspondence frameworks, it is progressively basic to discover a few blocks working in parallel. As of late, a procedure that adventures this reality to execute adaptation to non-critical failure on parallel channels has been accounted for. In this concise, this system is first connected to secure FFTs. Then, two improved protection schemes that combine the use of error correction codes and Parseval checks are reported and evaluated. The results show that the reported schemes can further reduce the implementation cost of protection.

Y. Xie, C. Yang, C. Mao, H. Chen and Y. Xie, [2] As soft errors become a significant threat to modern electronic systems, the first priority of protection against soft errors should be decreasing resource consumption. This brief proposes a novel low-overhead fault tolerant FFT design, combining modified reduced precision redundancy (RPR) method and error correction codes (ECCs). RPR can lower the hardware overhead when compared with traditional full-precision redundancy techniques, especially when resource of the original design is huge. ECCs are cost-efficient for achieving fault tolerance on our parallel-pipelined FFT. As an example, an FPGA implementation of a four-channel 16K-point FFT is presented, which demonstrates that the reported scheme can further reduce the overhead of fault tolerance designs.

S. K. Mali and M. C. Lakkannavar,[3] The pipelined design has gained fame because of its capacity to accomplish high throughput and low hardware multifaceted nature, low power consumption. Thus, these architectures are widely used in many applications, mainly for the real-time applications. The (FFT) offers enhanced plan for the mind boggling tests, i.e., complex esteemed FFT (CFFT) yet not for the real input tests, i.e., real-valued FFT (RFFT). This examination displays the four parallel pipelined architecture (PPA) for the RFFT by utilizing an accepted signed-digit multiplier (CSDM) to advance the area. The acquired outcomes are contrasted and the pipelined design of creator Salehi et al. [1] and is analyzed that the revealed engineering is productive in area enhancement.

R. Shirbhate, T. Panse and C. Ralekar [4] In this examination, a parallel FFT architecture is reported to give an efficient throughput and less energy consumption with the assistance of Cooley Tukey calculation for radix 8. In this calculation the DFT of N measure is isolated into littler sizes of N/2 and rehashed until last DFT scalars are found. It isolates the DFT in even list and odd list term. The computation time which is calculated by the pre defined formula (Nlog₂(N)) is lessened by the utilization of parallel design. Energy is characterized as power utilized per unit time. Parallel design performs number of tasks at the same time. As less time is required, the energy is proficiency is expanded. The aim of this examination is to check throughput and productivity utilizing Cooley Tukey calculation for higher radix. The ongoing patterns of this calculation is advancement of FPGA that is Field Programmable Gate Array as it can perform signal processing errands in parallel, execute pipeline structure and additionally accelerate the calculation of dull algorithms. The principle preferred standpoint of Cooley Tukey calculation is that it diminishes arithmetic calculations and in addition quick processing. As this calculation separates the DFT into littler DFTs, it very well may be joined with some other calculation at the same time.

Yanan Fan, Xin Meng, Xiujuan Yao and Yi Yan,[5] Structured LDPC, such as RA, IRA, ARA, QC-LDPC etc., are some critical LDPC codes, these codes normally have great performance and encourage the usage of encoding and unraveling for the auxiliary highlights they have. Another technique for the encoding of the organized LDPC codes has been displayed, to be specific FFT-based parallel encoding algorithm, which uses FFT to realize the parallel encoding of LDPC codes by achieving the circular convolution in the frequency domain. The examine demonstrates that the detailed calculation has bring down computational unpredictability than the already announced encoding calculation. Despite the fact that changing to the frequency domain requires complex tasks in real time domain which expanding the quantity of the quantization focuses, the throughput of the encoder has greatly improved because of the increase of the degree of parallelism. The simulation results demonstrate that the computational multifaceted nature of the FFT-based encoding calculation has a roughly direct association with the extent of sub-circle network, and when this size goes bigger, the recently revealed calculation will have a lower computational unpredictability and a higher throughput compared with the previously reported algorithms.

J. Park, J. Park and S. Bhunia,[6] Expanding process varieties combined with forceful scaling of cell area and working voltage in the mission of higher thickness and lower control have incredibly influenced the reliability of on-chip memory. Error correction code (ECC) has been traditionally used inside memory to provide uniform protection to all bits in a code word. They experience the effects of either satisfactory protection against multibit disappointments or huge overhead because of encoding/unraveling logic and equality bits. To address this issue, present a variable information length ECC (VL-ECC) for the implanted memory elements of digital signal processors, in which the data length of ECC can be dynamically reconfigured to preferentially protect the relatively more important bits. In the reported VL-ECC, when the number of failures exceeds the error correction capability, the data length of ECC is reduced to focus on the generally more critical higher order information bit parts, subsequently limiting framework quality corruption because of bit disappointments. At the point when the revealed VL-ECC is connected to the installed memory elements of a H.264 processor, normal peak signal-to-noise ratio enhancements of up to 5.12 dB are accomplished contrasted and the customary ECC under supply voltage of 800 mV or lower. With the fast Fourier transform processor, signal-to-quantization noise ratio is improved by up to 5.2 dB.

J. L. Pilla et al.,[7] In this examination survey the neutron affectability of Graphics Processing Units (GPUs) when executing a Fast Fourier Transform (FFT) calculation, and propose particular software-based hardening procedures to diminish its disappointment rate. Our examination is roused by test results with an unhardened FFT that exhibit a larger part of different errors in the yield on account of disappointments, which are caused by information conditions. In addition, the use of the built-in error-correction code (ECC) showed a large overhead, and proved to be insufficient to provide high reliability. Experimental results with the hardened calculation demonstrate a two requests of extent disappointment rate change over the first calculation (one request of size over ECC) and an overhead 64% smaller than ECC.

5. Problem Statement

Fast Fourier Transform (FFT) is used to build various image processing systems and application specific Digital Signal Processing (DSP) hardware. At present all outlines for FFT utilize ROMs or memory for complex twiddle multiplications. While actualizing FFT cores for long information arrangement on FPGA, the quantity of fortified sources of input and outputs (IOBs) are dependably a matter of concern. Exact methods must be utilized to reduce the quantity of IOBs. The quantity of IOBs can decrease by partitioning the long information arrangement into a gathering of short information groupings of same length. By this throughput will be expanded as well as the quantity of IOBs will be decreased by an incredible factor. Various FFT algorithms are used pipelining in order to increase throughput and speed.

6. Conclusion

In this examination an extensive survey of literature on Fault Tolerant Parallel FFTs Using error correction codes and Parseval checks has been reported. The Fourier Transform is a mathematical operation that transforms one function of a real variable into another. The new function, frequently called the recurrence frequency domain of the first function, depicts which frequencies are available in the first function. The Fourier transform is like numerous different transform in mathematics which makes up the subject of Fourier investigation. In this particular case, both the areas of the original function and its frequency domain representation are consistent and unbounded. The term Fourier Transform can allude to both the frequency domain representation of a function or to the procedure that changes one function into the other.

References

[1] Z. Gao et al., "Fault Tolerant Parallel FFTs Using Error Correction Codes and Parseval Checks," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 2, pp. 769-773, Feb. 2016.

[2] Y. Xie, C. Yang, C. Mao, H. Chen and Y. Xie, "A novel low-overhead fault tolerant parallel-pipelined FFT design," 2017 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Cambridge, 2017, pp. 1-4.

[3] S. K. Mali and M. C. Lakkannavar, "Parallel pipelined FFT architecture for real valued signals using radix-2," 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, 2016, pp. 1277-1281.

[4] R. Shirbhate, T. Panse and C. Ralekar, "Design of parallel FFT architecture using Cooley Tukey algorithm," 2015 International Conference on Communications and Signal Processing (ICCSP), Melmaruvathur, 2015, pp. 0574-0578.

[5] Yanan Fan, Xin Meng, Xiujuan Yao and Yi Yan, "FFT-based parallel encoding algorithm for structured LDPC codes," 11th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2015), Shanghai, 2015, pp. 1-5.

[6] J. Park, J. Park and S. Bhunia, "VL-ECC: Variable Data-Length Error Correction Code for Embedded Memory in DSP Applications," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 61, no. 2, pp. 120-124, Feb. 2014.

[7] L. L. Pilla et al., "Software-Based Hardening Strategies for Neutron Sensitive FFT Algorithms on GPUs," in IEEE Transactions on Nuclear Science, vol. 61, no. 4, pp. 1874-1880, Aug. 2014.

[8] N. Kanekawa, E. H. Ibe, T. Suga, and Y. Uematsu, Dependability in Electronic Systems: Mitigation of Hardware Failures, Soft Errors, and Electro-Magnetic Disturbances. New York, NY, USA: Springer-Verlag, 2010.

[9] R. Baumann, “Soft errors in advanced computer systems,” IEEE Des. Test Comput., vol. 22, no. 3, pp. 258–266, May/Jun. 2005.

[10] M. Nicolaidis, “Design for soft error mitigation,” IEEE Trans. Device Mater. Rel., vol. 5, no. 3, pp. 405–418, Sep. 2005.

[11] T. Hitana and A. K. Deb, “Bridging concurrent and non-concurrent error detection in FIR filters,” in Proc. Norchip Conf., Nov. 2004, pp. 75–78.

[12] S. Pontarelli, G. C. Cardarilli, M. Re, and A. Salsano, “Totally fault tolerant RNS based FIR filters,” in Proc. 14th IEEE Int. On-Line Test Symp. (IOLTS), Jul. 2008, pp. 192–194.

[13] B. Shim and N. R. Shanbhag, “Energy-efficient soft error-tolerant digital signal processing,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 14, no. 4, pp. 336–348, Apr. 2006.