Journal of Signal Processing Systems (2018) 90:1581–1582
https://doi.org/10.1007/s11265-018-1396-1

Guest Editorial: Special Section on Fast Fourier Transform (FFT) Hardware Implementations

Mario Garrido[1] · María Luisa López-Vallejo[2] · Sau-Gee Chen[3]

Published online: 20 August 2018

This year we celebrate the 250th anniversary of the birth of Jean-Baptiste Joseph Fourier in 1768, creator of the well-known Fourier transform. Despite the age of this algorithm, it is still nowadays one of the most used algorithms in signal processing.

The Fourier transform serves to determine the frequency components of a signal provided in the time domain. When it comes to digital system, the Fourier transform is calculated by its discrete version, the discrete Fourier transform (DFT). If the sampling of the signal respects the Nyquist theorem, the DFT provides a quantized version of the spectrum of the Fourier transform.

The calculation of the DFT requires simple mathematical operations: Additions and rotations in the complex plane. However, it was not until 1969 that Cooley and Tukey realized that there are redundant operations in the calculation of the output frequencies, being possible to share the computations among different frequencies. This resulted in the Cooley-Tukey algorithm, which reduces the order of operations of the DFT from $O(N^2)$ to $O(N \log_2 N)$.

The algorithm that is obtained after applying the Cooley-Tukey algorithm is called fast Fourier transform (FFT). Due to its reduced number of operations, the calculation of the FFT is generally preferred to the direct computation of the DFT.

During the last decades, the FFT has been the subject of a large amount of research. The goal for hardware designers is to provide efficient implementations aiming primarily for high performance, low usage of hardware resources (adders, rotators and memory), high accuracy or low power consumption. In order to achieve these goals, the design of FFT hardware architectures involves several interrelated research fields: Design and implementation of the architectures, data management, study of FFT algorithms, implementation of rotators and accuracy analysis.

First, regarding the design of the architectures, there exist several well-known architectures: Pipelined FFTs such as single-delay feedback (SDF), single-delay commutator (SDC), multi-path delay feedback (MDF) and multi-path delay commutator (MDC), and iterative FFTs. Most of the FFTs in the literature correspond to these architectures. However, there is still room for novel architectures such as the SC FFT or the SFF FFT proposed in one of the papers that we present later.

Second, the data management plays an important role in the design of FFT architectures. Note that the mathematical operations of the FFT (additions and rotations) are simple operations. However, it is a big challenge to determine the order in which these operations are carried out. New orders of operations is what leads to new FFT architectures. This happens very often in iterative FFTs, where multiple different memory access patterns are feasible. Furthermore, the output of an FFT hardware architecture is generally provided in the so called bit reversed order. Sorting out the output data to achieve a natural order is a challenge where efficient solutions have been proposed in the last years.

Third, the FFT algorithm admits numerous variations that lead to different radices. A large number of these variations are collected in the binary tree decomposition and in the triangular matrix decomposition. They are the result of moving the FFT rotations among different FFT stages. This allows for placing only trivial rotations in some of the FFT stages, where more complicated rotations are placed in other stages. This is the case of the radix-2^2 algorithm and, in general radix-2^k algorithms. The consequence for the hardware architecture is that rotators are only needed in some FFT stages and not in all of them.

Fourth, the implementation of the rotators is a challenge in itself. In many cases, it is beneficial to implement the rotators as shift-and-add operations. This is done by the use of single constant multiplication (SCM), multiple constant multiplication (MCM) and constant matrix multiplication (CMM) approaches. This allows to substitute complex multipliers by

✉ Mario Garrido
mario.garrido.galvez@liu.se

[1] Linköping University, 581 83 Linköping, Sweden
[2] Technical University of Madrid, 28040 Madrid, Spain
[3] National Chiao Tung University, Hsinchu 300, Taiwan

shift-and-add operations, which results in important savings in hardware resources. Alternatively, the coordinate rotation digital computer (CORDIC) also transforms rotations into shift-and-add operations. The CORDIC is based on the principle that a rotation in the complex plane by a given angle can be carried out by rotating a series of angles whose sum is the given angle.

Finally, in the last years there is an increasing interest in analyzing the accuracy of the FFT, both in terms of the accuracy of the rotators and the accuracy of the entire FFT. Some approaches allow for either increasing the accuracy without increasing the hardware cost, or reducing the hardware cost without decreasing the accuracy.

As a result, the design of an FFT architecture involves multiple variables and allows for a wide range of solutions, from hardware and power-efficient architectures for implantable devices to very high throughput designs for communication systems or radio astronomy. All depends on the requirements of the application.

To commemorate the 250 anniversary of Fourier's birth, we are pleased to present this special section on FFT hardware implementation. The special section includes three papers that cover different topics related to the FFT design.

The paper "SFF - The Single-Stream FPGA-Optimized Feedforward FFT Hardware Architecture" (https://doi.org/10.1007/s11265-018-1370-y) by C. Ingemarsson and O. Gustafsson provides new insights on the implementation of FFT architectures on FPGAs. Based on the idea that a design should take into account the resources available in the FPGA, the paper provides a new hardware architecture that makes an efficient use of those resources. This changes the typical paradigm of doing a good design first and then try to fit it on an FPGA to a more efficient approach where the structure of the FPGA is taken into account at the design phase.

The paper "Parallel Memory Accessing for FFT Architectures" (https://doi.org/10.1007/s11265-018-1387-2) by V. Kitsakis, K. Nakos, D. Reisis and N. Vlassopoulos presents an architecture where data management plays a key role. The authors propose an efficient address generation scheme that eliminates bank conflicts between FFT stages. The proposed addressing scheme provides parallel load and store of the data involved in radix-r butterfly computations and leads to an efficient architecture when r is a power of 2. This technique improves the circuits required for implementing the in-place architecture, including the data and twiddles address generation circuit, the interconnection and the control. The proposal has been validated on Xilinx FPGAs by the implementation of in-place radix-8 FFT architectures with input sizes 64 and 512 complex points.

The paper "Design Space Exploration of 1-D FFT Processor" (https://doi.org/10.1007/s11265-018-1393-4) by S. Liu and D. Liu proposes a design space exploration methodology for 1-D FFT processors that can quantitatively optimize target FFT hardware architectures. The considered design space basically covers all possible FFT solutions, including a wide-range candidate architecture collection, coarse-grained architecture selection, FFT radix and parallelism selection, and circuit level design optimizations. As such, high-performance FFT architectures can be obtained in short time. The effectiveness of the proposed design methodology has been verified with major state-of-art applications.

We would like to thank all of the authors for their contributions to this special section and the anonymous reviewers for their efforts in ensuring the paper quality. We hope that you enjoy reading this special section.

Mario Garrido received the M.S. degree in electrical engineering and the Ph.D. degree from the Technical University of Madrid (UPM), Madrid, Spain, in 2004 and 2009, respectively. In 2010 he moved to Sweden to work as a postdoctoral researcher at the Department of Electrical Engineering at Linköping University. Since 2012 he is Associate Professor at the same department. His research focuses on optimized hardware design for signal processing applications. This includes the design of hardware architectures for the calculation of transforms, such as the fast Fourier transform (FFT), circuits for data management, the CORDIC algorithm, and circuits to calculate statistical and mathematicaloperations. His research covers high-performance circuits for real-time computation, as well as designs for small area and low power consumption.