

Low-Power Reconfigurable VLSI Architecture for the Implementation of FIR Filters

Evangelos F. Stefatos, Han Wei, Tughrul Arslan, Robert Thomson

School of Engineering & Electronics

*The University of Edinburgh, King's Buildings, Mayfield Rd, Edinburgh, EH9 3JL,
Scotland, UK*

{Evangelos.Stefatos, W.Han, Tughrul.Arslan, Robert.Thomson}@ee.ed.ac.uk

Abstract

This paper presents a custom reconfigurable VLSI architecture that is tailored for the implementation of low-power, medium/high order, digital finite impulse response (FIR) filters. These are realized within a reconfigurable array that consists of heterogeneous, programmable, arithmetic-logic units. The reconfigurable design is based on the primitive operator design (POF) technique. The concept of a genetic algorithm (GA) is introduced, which utilizes a radix-4, 256-point fast-fourier-transform (FFT) to calculate the frequency response of the evolved filters. The results related to the performance, physical-area and power consumption make this architecture very competitive in comparison with other industrial, general purpose FPGAs.

1. Introduction

Nowadays, FIR filters constitute the back-bone of most digital signal processing (DSP) systems. Based on the nature of the targeted application, there are several issues that must be considered like performance, physical area, power consumption and systems robustness. General purpose DSPs, such as TMS320 series from Texas Instruments do not provide adequate throughput and lack of reliability since they do not provide any redundancy. On the other hand, general purpose FPGAs present high flexibility in terms of design reusability but are not suitable for low-power applications. Therefore, there is a need in the industrial, electronic market for hybrid reconfigurable fabrics that constitute a trade-off solution between general-purpose DSPs and FPGAs [4]. This paper presents a reconfigurable multiplier-less architecture that is suitable for the realization of robust, reprogrammable medium/high order FIR filters. In this architecture, the POF [3] technique is used, which presents superiority over canonic signed digit code

(CSD). It utilizes directed graphs to optimize filter coefficients by using add/subtract and shift operations in order to generate reduced complexity filters. Therefore, the dedicated, power-hungry multipliers are substituted with a series of bit-shifts, additions and subtractions [2], [6], [7]. In [1] a number of configurable arithmetic macro-structures, which utilize the POF technique, have been investigated and the obtained results showed that these structures present advantages both in terms of speed and physical area. Previous research work has been done in [5], which is relative with the implementation of multiplier-less, fault-tolerant reconfigurable FIR filter architecture. However, our work goes a step further by efficiently tackling some issues related with FIR filter implementation. Firstly, since our architecture does not follow the folded transposed or direct form as [5] does, the multiply accumulate (MAC) unit is realized within the reconfigurable array itself and all the coefficients are retrieved from the same arithmetic/logic unit. Therefore for high-order filters, there is not any need to realize huge arrays in order to take one coefficient by each unit. Additionally, this increases the reliability of the architecture because all the parts of a FIR filter can be evolved within the reconfigurable array and hence there is more than one possible unit to act as input or output.

2. System Architecture

The architecture that is presented in this paper consists of a 13x13 configurable arithmetic/logic units (CALUs) array. Unlike the more macro-based approach taken in [1], this architecture is a more fine-grained perspective. It constitutes a heterogeneous array since it is composed of three different kinds of CALUs. The aim behind this is to reduce the complexity of each CALU, in order to increase its performance and decrease its physical area and power consumption, respectively. Almost all of these are configurable and able to implement *right/left shifting* or

addition/subtraction operations. The output of each CALU is fed into a synchronous register, in order to create a pipelined architecture, which increases data throughput. Also, there are a few blocks (switch boxes) that provide data routing in the horizontal/vertical level and clock the output in a register. This implies that the total number of registers is 169 and the number of taps that can be achieved is:

$$Taps = 169 - (width + height - 1) \quad (1)$$

As it was mentioned in the introduction, this design is a multiplier-less architecture that employs the POF technique, in order to produce low-complexity, reduced area and low-power FIR filters. Both approaches are based on the generation of partial products, which can be re-used in order to generate coefficient terms. In figure 1 is depicted the topology of the reconfigurable array and how the CALUs are connected to each other. It can be deduced from figure 1 and 4 that based on a specific configuration some CALUs, which belong in the same row but are not adjacent, can communicate within a single clock cycle (fast interconnections). This approach gives more flexibility for configuration and also increases the maximum number of taps than can be achieved, according to (1). Moreover, from the topology of this architecture, it can be deduced that based on the configuration scheme, different number of taps can be achieved by creating data paths with different delay times. The switch boxes denoted with “F” define the critical path in the design and fix the timing violations that may occur when invalid configurations are applied on the architecture. Each one of the three different types of CALUs within this architecture apart from their primary scope, they can also act as an output from which the coefficient can be obtained. However, it is obvious that as we move in the top-right corner of this architecture, the number of order of the filter increases. Concerning the configuration process, the implemented design needs in total 464 bits to get configured. The load of the configuration string happens in parallel and so it takes to the architecture only one clock cycle to apply a configuration string. This may be a drawback in terms of physical area in comparison with a bit-serial loaded architecture, but decreases tremendously the power consumption during configuration time. Specifically, in this paper that the applied configuration is guided by a GA, the power reduction is of great importance, since there will be applied many configuration schemes until the GA convergences and thus the process must be as fast as possible and the switching activity during configuration must be reduced to the lowest possible. Figure 2 depicts the overall architecture of our system that consists of the reconfigurable array and the GA, which is responsible for the evolutionary realization of medium/high order FIR filters. The GA uses a R4SDC, 256-point, pipelined FFT

in order to calculate the frequency response of the evolved filter and compare it with the ideal one. Finally, the fitness function of the GA also considers the number of saturation events, which occur for a specific configuration, in order to prevent the evolution of non-linear filters.

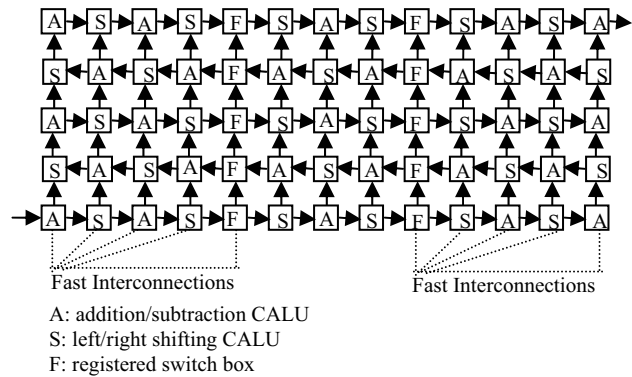


Figure 1. Topology of CALUs

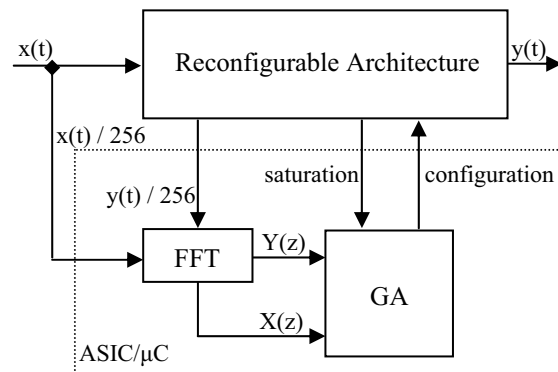


Figure 2. System architecture

2.1. Addition/Subtraction (AS) CALU

The AS-CALU consists of both combinational and sequential logic. The combinational part includes a unit that performs either 20-bit addition or subtraction. The inputs of AS-CALU are determined by two multiplexers. The vertical output is fed into a synchronous register that stores the output of the addition/subtraction unit. Moreover, the horizontal output can take the value of the register or one of the two inputs of the CALU. Thus, each AS-CALU is able to have different output value in each of the two output-ports. Furthermore, AS-CALU is associated with some additional logic that copes with overflow phenomena by using a new arithmetic scheme. It is a combination of fixed and floating point arithmetic, truncation, saturation and scaling. The detailed mechanism is described later.

2.2. Left/Right Shifting (L/R-S) CALU

Similarly to the AS-CALU, the L/R-S CALU has identical combinational logic but different arithmetic unit. The shifter itself is configured by 1-bit and based on its value the unit shifts on the right or left by 1 position the input value. Furthermore, the L/R-S CALU has two multiplexers that select the input and output of the CALU. The multiplexer attached in the horizontal output can either select to forward the value stored in the register or pass automatically the selected input to the output, without performing any shifting operation.

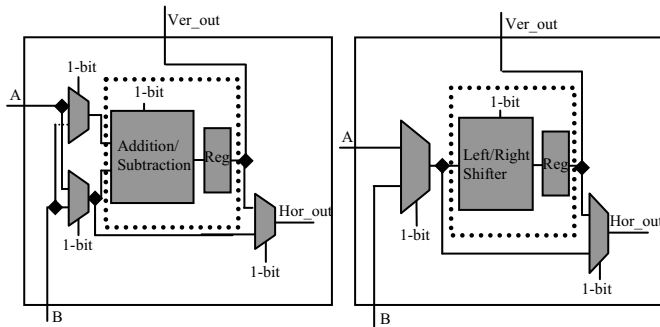


Figure 4. Addition/Subtraction & Shifter CALUs

3. Overflow Protection Mechanism

As the design consists of many arithmetic units, which perform additions and left shifts, overflow is very likely to happen. Therefore, reconfigurable designs must be carefully designed and incorporate additional logic in order to secure the quality of their operation. Therefore, it has been added in the 20-bit data bus a protection bit that sets whenever overflow is to occur. All the arithmetic CALUs have an overflow detection mechanism and whenever it is to happen, the data are shifted on the right by 3 bits. After the shift, the sign bit is extended and the 3 less significant bits are truncated. This can happen only once for a specific data sample and the protection bit is latched logic high in order to indicate that there is a binary exponent of 3. Using this protection bit, the architecture can safely add and subtract data samples, which have different binary exponents. Hence, it can be deduced that the employed arithmetic representation is a kind of hybrid one that mainly acts like fixed-point but also utilizes the simplest form of the floating point one. In addition to this, each CALU asserts to logic 1 a register that indicates that saturation has been occurred. This is a further way to handle the overflow according to which if the result is out of the range that can be properly represented in the given data size, then it takes the closest value within the representable range. This happens whenever the protection bit has already been latched to logic high and an overflow is going to occur again.

Consequently, the GA that controls the configuration of the system considers the “saturation” signals and adds the equivalent penalty in the fitness function in order such configurations to be avoided. Moreover, the role of the right shifts can also be other than performing a conventional division by 2 on a single number. The division of the coefficients by a constant scale factor inserts attenuation into the frequency response but it does not affect its shape. This is an alternative way to prevent overflow that is known as scaling. However, it is very important to figure out how much scaling is necessary to avoid overflow because it results in loss of effective number of digits that increases the quantization error. The process of scaling in our design will be controlled by the GA, which will be responsible to find out a configuration with the most acceptable frequency response.

4. Synthesis Results

For the purpose of this paper a design that consists of 13x13 CALUs has been synthesized using Synopsys synthesis tool and UMC 0.18 technology cell library. The results depicted in table 1, are very promising in terms of physical area comparing with industrial general-purpose FPGAs. Moreover, the timing analysis showed that the clock frequency of this architecture can be 100MHz that is quite competitive even compared with ASIC applications.

Table 1. Area results

Area (sq.mm)	AS CALU	L/R-S CALU	SBox CALU	Array (16x16)
Total Cell	0.0074	0.0042	0.0025	1.139
Net	0.0015	0.0008	0.0004	0.326
Total	0.0089	0.0050	0.0029	1.465

5. Power Analysis

The power consumption of this architecture has been calculated using Prime-Power tool (Synopsys), considering the standard-delay-format (SDF) information, which has been extracted after synthesis. Figure 5 depicts the results of the top module of the reconfigurable design that contains 169 CALUs, from which the 63 perform addition/subtraction operations, another 63 perform left/right shifting by 1 operations and the rest 41 are used to switch the data samples. It illustrates the power in mW, which is consumed per MHz of operation. It is important to mention here that the operational voltage of the design is 1.8V, which implies that the design pulls current equal to 1mA/MHz. In figure 6 is depicted the maximum power consumed by each CALU separately. The obtained

numbers in figure 6 are very significant because since each CALU is clocked through an AND gate, it can be switched off in the case it does not participate on a certain configuration. Hence the total consumption depicted in figure 5 can be further decreased under certain circumstances. Table 2 summarizes the power consumption of our architecture compared to AT6000 series FPGAs [8]. It can be deduced that our design consumes less power even if we accept that the design utilizes the total amount of its hardware resources.

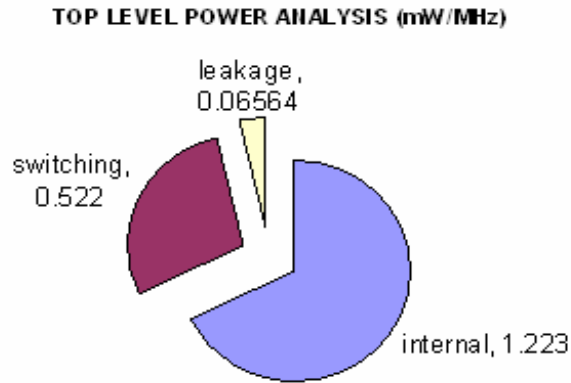


Figure 5. Top module power analysis

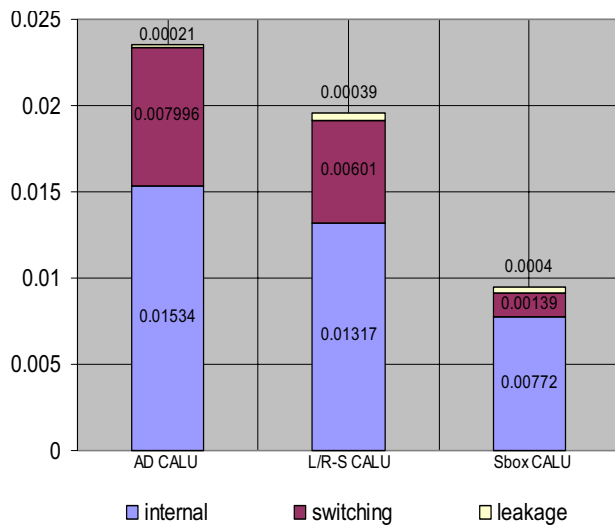


Figure 6. Power of the 3 different CALUs

Table 2. Power comparison

Symmetrical FIR filter	AT6000 series FPGA (mA/MHz)	Reconfigurable array (mA/MHz)
8-tap	1.048	1.006
16-tap	1.29	
32-tap	2.62	

6. Conclusions

In this paper a novel, low-power, custom-reconfigurable VLSI architecture is proposed that is tailored for FIR filters realization. The novelty of this architecture lies in several fields that are associated with the interconnection scheme of the 169 CALUs within a fine-grained architecture and the way the coefficients are retrieved from the same output, using paths with different delay time based on a certain configuration. Additionally, by using the POF technique and designing heterogeneous, configurable blocks, the power calculation of our design has been proved to be competitive compared with general purpose FPGAs, although the associated built-in flexibility that implies high degree of robustness and the additional logic circuitry, which is introduced in order the system to cope efficiently with overflow. The primary results, which have been obtained, are related with the hardware realization and power consumption of this architecture. Finally, our future research is going to be focused on the efficient realization of medium and high order FIR filters on this architecture by using evolutionary techniques.

7. References

- [1]. T. Arslan, H. I. Eskikurt, and D. H. Horrocks, "Configurable structures for a primitive operator digital filter FPGA, Signal Processing Systems", *SIPS 97 – Design and Implementation*, IEEE Workshop on, 1997, pp. 532-540.
- [2]. A. Avizienis, "Signed-digit number representation for fast parallel arithmetic", *IRE Trans. Electron. Comput.*, 1961, pp. 389-400.
- [3]. D. R. Bull, D. H. Horrocks, "Primitive operator digital filters", *IEE Proc-G*, 1991, pp. 401-412.
- [4]. R. A. Hawley, B. C. Wong, Thu-ji Lin, J. Laskowski, and H. Samuelli, "Design Techniques for silicon compiler implementations of high-speed FIR digital filters", *IEEE Journal. Solid-State Circuits*, Vol.31, No 5, 1996, pp. 656-667.
- [5]. B. Hounsell and T. Arslan, "A programmable multiplierless digital filter array for embedded SoC applications", *IEE Electronics Letters*, Vol. 37, No. 12, 2001, pp. 735-737.
- [6]. D. W. Redmill, D. R. Bull, E. Dagless, "Genetic synthesis of reduced complexity filters and filter banks using primitive operator directed graphs", *IEE Proc. – Circuits Devices Syst.*, Vol. 147, No. 5, Oct 2000.
- [7]. H. Samuelli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients", *IEEE Trans. Circuits Syst.*, Vol. 36, No 7, 1989, pp. 1044-1047.
- [8]. www.atmel.com/dyn/products/app_notes.asp?family_id=623&part_id=2073