

An EHW Architecture for the Design of Unconstrained Low-Power FIR Filters for Sensor Control Using Custom-Reconfigurable Technology

Evangelos F. Stefatos, Tughrul Arslan
School of Engineering & Electronics
The University of Edinburgh, King's Buildings,
Mayfield Rd,
EH9 3JL, Scotland, UK
Evangelos.Stefatos@ee.ed.ac.uk

Didier Keymeulen, Ian Ferguson
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
PASADENA, CA 91109
Didier.Keymeulen@jpl.nasa.gov

Abstract

This paper presents a power-optimized evolvable hardware (EHW) architecture that employs custom-reconfigurable technology. It comprises a preliminary research work towards the implementation of filtering circuits associated with the JPL-Boeing micro-machined gyroscope. Our scope is to implement a low-power, autonomously reconfigurable architecture that is tailored for the realization of arbitrary response FIR filters. For the purpose of this paper the hardware substrate comprises a reconfigurable 4x12 array, which consists of heterogeneous, configurable, arithmetic-logic units (CALUs). The implementation of the design is based on the primitive operator filter (POF) technique in order to evolve all the parts of a filter (unconstrained filter). Furthermore, a hybrid arithmetic approach is employed in order for the architecture to cope with overflow events. The paradigms of both lowpass and highpass filters are produced, using two different strategies of evolution. The obtained results demonstrate the physical characteristics of the reconfigurable substrate and the performance of the genetic algorithm (GA) in successfully designing FIR filters. Finally, the power results of the reconfigurable architecture (RA) are compared with these of the AT6000 series FPGAs and an algorithmically power-optimized, custom reprogrammable FIR core.

1 Introduction

FIR filters are employed in the majority of digital signal processing (DSP) based electronic systems. The emergence of demanding applications (image, audio/video processing and coding, sensor filtering, etc.) in terms of power, speed performance, system compatibility and reusability make it imperative to design hybrid structures, which close the gap between the ASIC and FPGA technologies. Moreover, in

the case of aerospace applications, there is the added concern for fault-tolerant FIR filtering fabrics, which are capable to respond to various malfunctions caused by endogenous or exogenous factors.

This paper presents an EHW architecture that targets to meet all the objectives (low-power consumption, autonomous adaptability/reconfigurability, fault-tolerance, etc.), which are set by the JPL-Boeing micro-machined gyroscope [1]. Our architecture is specifically tailored for the design of FIR filters that comprise the main electronic components of the gyroscope's control and filtering tasks, including the automatic-gain-control (AGC) for the drive loop, the linear filter for the sense-rebalance loop and the demodulation of the rebalance signal with respect to the input signal to the AGC.

The design is characterized by a multiplier-less architecture that employs the POF technique through which digital filters are realized using signal flow graphs comprising low complexity operations [2]. POF is particularly attractive for autonomous filter design using EHW, as it does not require any initial encoding scheme, such as canonic signed digit (CSD) [3]. Furthermore, [4] shows that FIR filters designed using POF are smaller in area than those designed using the CSD approach. In addition to this, the heterogeneous nature of the CALUs make the design very competitive in terms of power, without lacking of flexibility and other features, like overflow prevention, which are requirements in industrial, general-purpose FPGAs. Specifically, POF architectures use hybrid arithmetic representation that mainly acts as fixed-point but also employs the simplest form of the floating point one. The authors in [5] demonstrate the evolution of an 8-tap filter by using a Xilinx Virtex XCV1000 FPGA, while [6] presents the complete hardware evolution of an adaptive filter. In comparison with these research investigations our design presents a more fine-grained approach that offers superiority in terms of power over general-purpose FPGAs or multiplier-based solutions.

In contrast with other fine-grain implementations [7], this architecture does not apply any constraints in the form (direct, transposed, etc.) of the evolved filter. This means that the entire filter, including the multiply accumulate (MAC) unit and the number of the used delays, is realized within the RA. Hence, this approach takes full advantage of evolution that may guide to the realization of arbitrary filters, which present equivalent frequency response by using fewer resources. Specifically, [8] shows that digital filters implemented by the utilization of the POF technique and evolutionary algorithms present reduced power consumption.

In this paper two different genetic approaches are presented. Each is employed to design a lowpass and a highpass filter. The former approach evolves the coefficients directly, while in the latter a radix-4, 256-point FFT is used by the GA, in order to transform the time domain of the input and output of the RA, in the frequency domain and compare the evolved magnitude with the ideal one.

2 System architecture

For the purpose of this paper, the architecture of the overall system consists of a reconfigurable hardware substrate, which is based on the POF design principle and a GA module, which is responsible to provide the best solution for the realization and the autonomously adaptation of the FIR filters. The substrate comprises an array structure that consists of 4×12 CALUs. There are two kinds of CALUs. The first performs addition/subtraction (A/S-CALU) of 20-bit numbers and the second left/right shifting (L/R-CALU) operations. Each CALU contains a programmable delay chain, which can generate delays that vary from 0 to 3 clock cycles. The interconnection scheme between two adjacent columns is controlled by six 4:1 multiplexers. Moreover, every 5 columns it has been attached in the array a column that consists of six registers, in order to prevent timing violations in the case of configurations that do not insert the proper number of delays. Figure 1 depicts the structure of the RA. In the right end of it, there is also a single A/S-CALU that gathers all the outputs. This CALU can only create a delay of one clock cycle. Therefore, it can be deduced that the minimum path for a data sample to go inside the RA can be 3 clock cycles, while the maximum path can be $3 + 12 * 3 = 39$ clock cycles. Hence, theoretically this specific array is capable to realize a FIR filter with maximum of 36 taps. In figure 2 is shown the architecture of the two CALUs. It is apparent that the A/S CALU needs 3 bit to get configured, while the L/R needs four. Moreover, in the A/S CALU there is 1 bit that determines whether the block will perform addition or subtraction and 2 bit that define the delay. Respectively, in the L/R CALU there is 2-bit programming the delay chain

and 2-bit that define the kind of shifting (shift left by 1, shift right by 1 or 2, no shift). Hence, it is apparent from figures 1, 2 that it is a pipelined architecture capable to convolve the different samples by creating paths with different delay times.

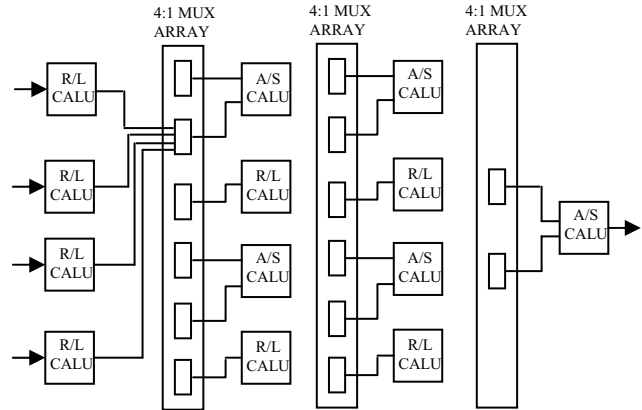


Figure 1: Example of the reconfigurable platform

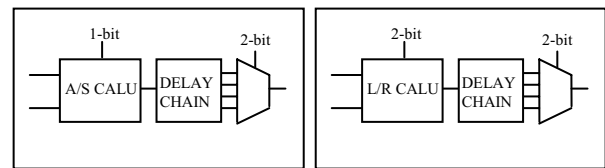


Figure 2: Programmable CALUs for FIR filtering

Figure 3 depicts the overall system architecture. The GA in the case of the frequency response evolution employs a radix-4, 256-point, pipelined FFT module, in order to meet the functional specification. As it was mentioned in the introduction the CALUs have an overflow prevention mechanism. It can be seen in figure 3 that the RA informs the GA for the number of saturations, which occur after each applied configuration. As overflow is of great importance in the realization of non-noisy filters, real input samples are applied in the RA during configuration process. The overflow mechanism is based on the addition of 1-bit (protection bit) in the 20-bit data bus that sets to logic 1, whenever overflow is to occur. When this happens the specific sample is shifted right by 3 bit. After the shift the sign bit is extended and the 3 less significant bit are truncated. This can happen only once for a specific data sample and the protection bit is latched high in order to indicate that there is a binary exponent of 3. Using this protection bit, the system can safely add and subtract data samples, which have different binary exponents. Moreover, when overflow is to occur in a data sample, which has used the protection bit already, the system saturates by representing the value of this sample with the closest value within the representable range. It is very important the fact that the RA supports right shifts, which can scale

coefficients by a constant factor and add attenuation in the frequency response without affecting the actual shape.

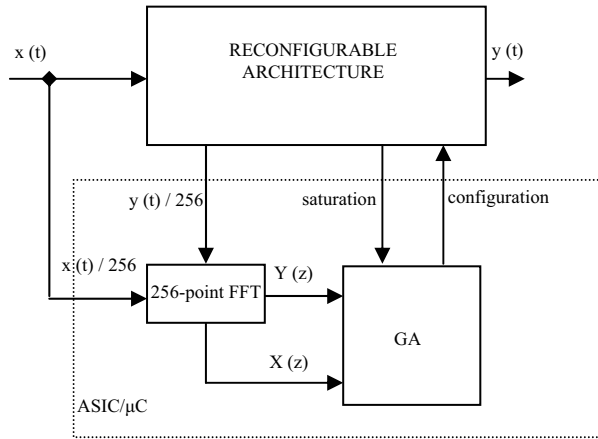


Figure 3: Overall system architecture

3 Genetic algorithm

The two GAs that are used for the evolution of the coefficients and the frequency response, utilize the $(\mu+\lambda)$ methodology. According to this, the initial population consists of 50 chromosomes (parents) and is then extended to 100 by the addition of another 50 chromosomes (children). The children chromosomes are generated by crossover and mutation operations, which are applied on randomly selected chromosomes taken from the initial population. The rate of crossover is 80%, while the rate of mutation is 0.014% per gene. After evaluation and ranking the best chromosome is selected through elitism and in conjunction with another 49 survivors, which are selected through tournament selection, they compose the new parent chromosomes (population). Each chromosome consists of a 309-bit binary string, which encodes the operation of 25 A/S-CALUs, 24 R/L-CALUs and 70 4:1-multiplexers.

3.1 Evolution of coefficients

The fitness function of the first GA is expressed as the sum of the difference in square between the ideal and the evolved coefficients. Moreover, there is a penalty applied on configurations that realize filters with different number of taps. Finally, there is an additional penalty associated with configurations that cause extensive saturation. Hence, the mathematical formula that expresses the final fitness-score is given below:

$$Fitness = \sum_{i=0}^{taps-1} (coef_{ideal_i} - coef_{evolved_i})^2 - penalty_i - penalty_{sat_i} \quad (1)$$

Because the RA is capable to deal with fractional coefficients as well, both coefficients in formula (1) are multiplied first by a constant number ($\times 10^3$), in order to achieve the appropriate accuracy.

3.2 Evolution of frequency response

The fitness function of this GA is expressed by the sum of the difference of the evolved and ideal squared magnitudes, respectively ($magn^2=10^{10}$). The ideal squared magnitude is pre-computed according to the applied specification, while the evolved one (amplitude spectral density $|F(j\omega)|^2$) is calculated by dividing the square of the output $Y^2(z)$ with the square of the input $X^2(z)$. Hence:

$$|F(j\omega)|^2 = (Real_{out}^2 + Imag_{out}^2) / (Real_{in}^2 + Imag_{in}^2) \quad (2)$$

Because the FFT module is 256-point one, we have to compare only the half frequencies. Similarly with the previous methodology, configurations that produce extensive saturation are penalized. Therefore, the fitness-score is derived by the following formula:

$$Fitness = \sum_{i=0}^{127} (|F(j\omega)|_{ideal_i}^2 - |F(j\omega)|_{evolved_i}^2) - penalty_{sat_i} \quad (3)$$

4 Simulation results

The simulation results that were obtained, concern the realization of two FIR filters (lowpass and highpass) by using both evolution strategies introduced in section 3. The simulation process consists of two components: the reconfigurable architecture, which is synthesizable and the GA unit (including the FFT module) that runs in a Verilog testbench environment that is not fully synthesizable at the moment. All the simulations in this paper refer to off-line reconfiguration. However, work is ongoing regarding on-line reconfiguration, whenever filters need to adapt to sensor dynamics or occurrence of faults. The characteristics of the filters, which were designed by the evolution of coefficients, are shown in table 4-1 and 4-2.

Table 4-1: Lowpass filter specification

Characteristics	Fixed point Equiripple Lowpass filter (13-taps)
Sampling frequency	1
Passband edge	0.1
Passband Attenuation	3
Stopband Edge	0.15
Stopband Attenuation	20

Table 4-2: Highpass filter specification

Characteristics	Fixed point Equiripple Highpass filter (9-taps)
Sampling frequency	1
Stopband edge frequency	0.01
Stopband attenuation	20
Passband edge frequency	0.1
Passband attenuation	3

4.1 Evolution of coefficients

In order to obtain the transfer function of the evolved filter, the impulse response is applied on the architecture at the beginning of each evaluation followed by 39 (maximum path delay) samples which are zero. The word length of the achieved coefficients can be up to 20-bit. Moreover, because a number of negative coefficients are used, a 2's complement encoding is required. Figure 4 depicts the transfer function of the ideal and the evolved lowpass filter. It is apparent that the evolved filter has exactly the same characteristics with the ideal one. However, the formed filter is a symmetrical 9-tap, instead of a symmetrical 13-tap one (ideal). The GA retrieves the best solution after 3500 generations.

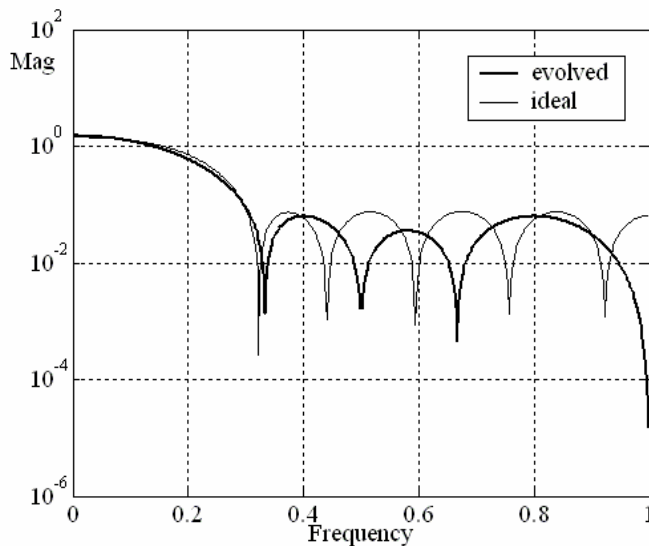


Figure 4: Transfer function of lowpass 13-tap filter

The process for the realization of the highpass filter is exactly the same as before. Figure 5 illustrates the comparison in the transfer function between the evolved filter and the specification. The GA achieves the best solution after 6300 generations and corresponds exactly to a 9-tap filter. However, as it can be seen in figure 5, there is a small deviation in the stopband and passband edge frequency.

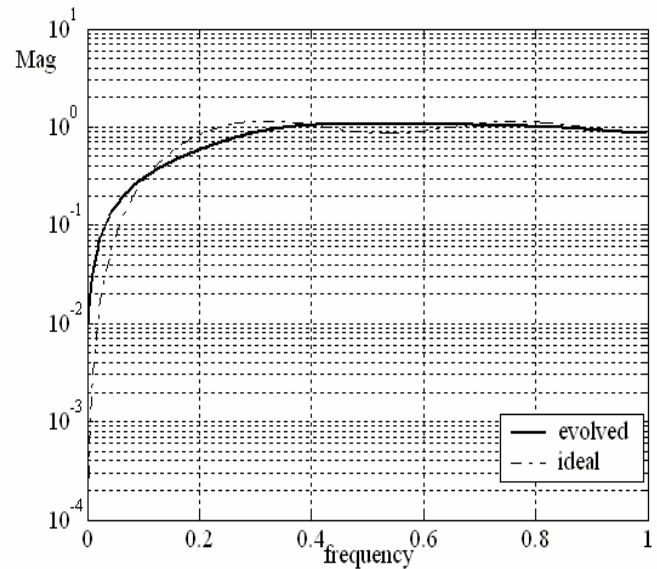


Figure 5: Transfer function of highpass 9-tap filter

4.2 Evolution of frequency response

This methodology presents superiority over the previous one because it calculates the magnitude from the frequency response of the input and output when real data is fed in the RA. Therefore, a more accurate estimation can be done in comparison with previous work done in [6], concerning overflow phenomena that generate noise in digital filters. Moreover, results prove that the GA needs much less generations to converge with respect to the previous methodology. However, it has the drawback that a single generation needs more computational time due to the FFT calculations.

It has been mentioned in section 2 that the maximum number of taps that can be achieved by this architecture is 36. Therefore, considering the 256-point FFT, 220 randomly selected data samples are fed in the RA. The rest samples until 256 are padded with zeros. The frequency response of the input is calculated only once, while the respective of the output changes in each generation. Figure 6 illustrates the comparison between the evolved lowpass filter and the applied specification. The actual number of generations needed is only 72. This reduction in the number of generations in comparison with the previous lowpass filter realization is quite sensible, considering the fact that the GA is not concerned to precisely match the value of each coefficient, but instead it tries to evolve a filter that has the same spectral density with the one applied on the specification. After the design of the filter, we obtained the coefficients of the filter by applying the impulse response in the RA and verified the achieved magnitude in matlab. Table 4-3 summarizes the results.

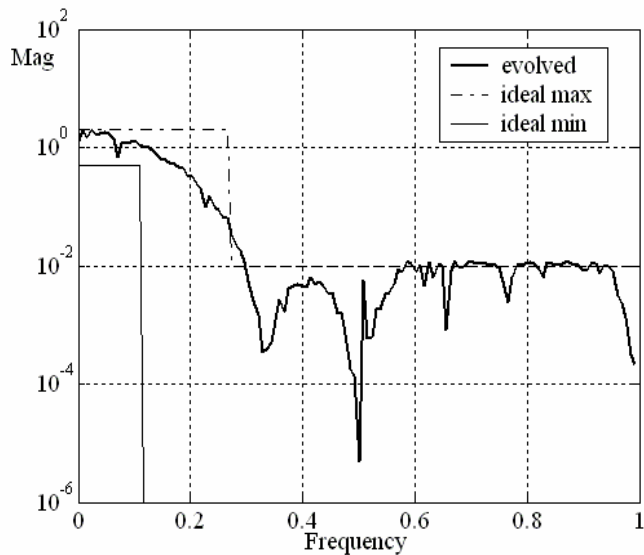


Figure 6: Magnitude of the evolved lowpass filter

Table 4-3: Coefficients of the 8-tap lowpass filter

C0	-0.125
C1	-0.129
C2	-0.203
C3	-0.254
C4	-0.207
C5	-0.203
C6	0.129
C7	-0.078

For the design of the highpass filter the GA needs 204 generations in order to find the optimal solution. In figure 7 is depicted the magnitude of the evolved filter in comparison with the ideal filter. Similarly with the previous example, table 4-4 shows the coefficients of the evolved filter.

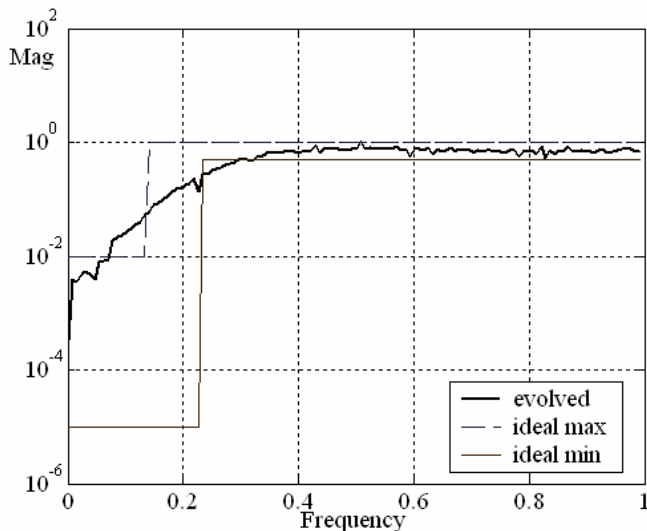


Figure 7: Magnitude of the evolved highpass filter

Table 4-4: Coefficients of the 12-tap highpass filter

C0	0.0098
C1	-0.0117
C2	-0.0215
C3	-0.0019
C4	-0.0058
C5	-0.1152
C6	0.6328
C7	-0.2520
C8	-0.2340
C9	-0.1289
C10	0.0313
C11	0.0313

5 Synthesis and power analysis

For the purpose of this paper a reconfigurable 4x12 array has been synthesized using Synopsys synthesis tool and UMC 0.13 technology cell library. The post-layout netlist was then employed in order to calculate the power consumption of the RA by running a simulation with 1000 data samples both for the lowpass and highpass filter. Table 5-1 summarizes the area results of the RA.

Table 5-1: Area results

Area (sq. mm)	A/S - CALU	L/R - CALU	RA(4x12)
Combinational	0.00473	0.0036	0.2613
Sequential	0.00208	0.0020	0.1087
Interconnections	0.00120	0.0010	0.1028
Total chip	0.00801	0.0066	0.4728

Figure 8 depicts the power consumed by the evolved lowpass and highpass filters, which have 8 and 12 taps, respectively. The power is measured in mW and corresponds to clock frequency equal to 1MHz. It is important to mention that the operational voltage of the design is 1.08 Volts, which implies that the RA pulls current equal to 0.32 mA and 0.34 mA for the lowpass and highpass filter, respectively.

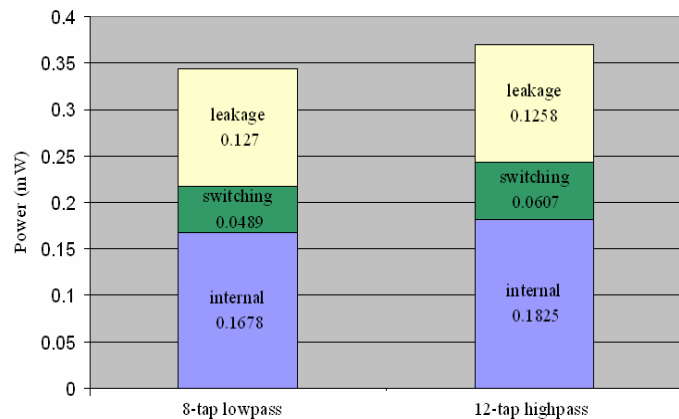


Figure 8: Power analysis of the evolved filters

It is also interesting to see the power that is consumed by each CALU, because they are clocked through AND gates and therefore some CALUs can be switched off, when they do not participate in a certain configuration and further reduce the total power consumption. In the example of the highpass filter realization, the utilization of the CALUs is 81%. This implies that there are 9 CALUs (5 adders/subtractors and 4 left/right shifters), which can safely be switched off. Figure 9 depicts the power results concerning the two kinds of CALUs and the 4:1-multiplexer, respectively. Table 5-2 summarizes the power consumed by our RA compared to the AT6000 series FPGAs [9], specifically for the realization of FIR filters. Based on the power that the evolved 12-tap filter consumes and on the utilization of the CALUs, the power consumption for 16, 24, and 32-tap filter has been estimated.

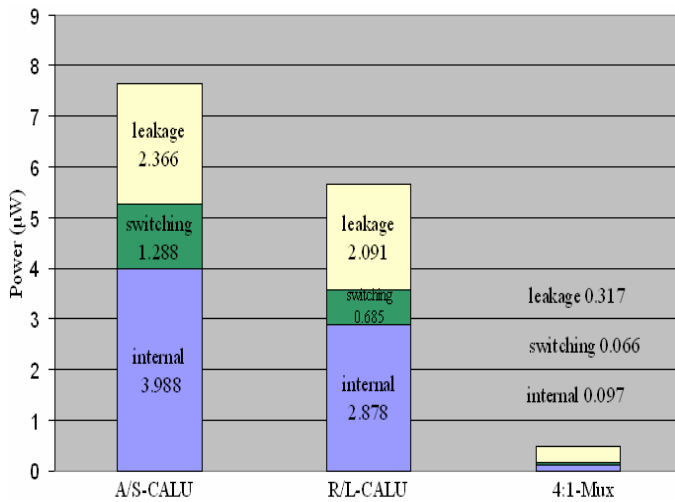


Figure 9: Power analysis of the CALUs

Moreover, we have implemented an algorithmically power-optimized, reprogrammable 128-tap FIR filter. It comprises an ASIC implementation that exploits efficient algorithms in order to reduce power [10]. More specifically, the first algorithm decomposes individual coefficients into two sub-components, such that a part can be produced using a single shift operation leaving another one with a reduced word-length to be applied on the multiplier. As a result, the effective switched capacitance decreases on the input of the multiplier and coefficient buses. The second algorithm reduces the switching activity not only at the multiplier inputs but also on the address and data buses, by processing multiple data samples at the same time rather than one at a time. The filter has been synthesized using the same technology cell library (UMC 0.13), with the RA. The power results of this design are also depicted in table 5-2. Finally, analysis of the work in [1] reveals that the ASIC core that has been designed to

implement the filtering tasks of the sensor consumes 4.5mW for a 128-tap filter (ASIC operating frequency is equal to 37MHz), by using 0.25µm technology cell library and 2.5 core supply voltage.

Table 5-2: Power Comparison

FIR filter	AT6000 (mA/MHz)	RA (mA/MHz)	ASIC (mA/MHz)
8-tap	1.048	0.32	3.30x10 ⁻³
12-tap	-	0.34	5.00x10 ⁻³
16-tap	1.29	0.49	6.60x10 ⁻³
24-tap	1.94	0.66	10.1x10 ⁻³
32-tap	2.62	0.83	13.3x10 ⁻³

6 Conclusions

This research work presents a power-optimized, EHW architecture that targets the accomplishment of the control and filtering operations of the JPL-Boeing micro-machined gyroscope. The obtained simulation results emphasize the capability of this platform to realize FIR filters by using different GAs. Results show that the evolution of the amplitude spectral density of the filter is more efficient than evolving the coefficients themselves directly, considering the number of generations that the GA needs to converge. This improvement is translated to a reduction of approximately 97% in the number of generations for both the realization of the lowpass and highpass filter. Finally, the obtained power results classify our design in an intermediate category between FPGA and ASIC technologies. More specifically, it presents a reduction in power that is about 68% over the AT6000 series FPGAs and an increase that comes up to 98% over a programmable FIR filter ASIC.

References

- [1]. M'Closkey, R., Challoner, A. D., "Modeling, identification and control of micro-sensor prototypes". American Control Conference, 2004. Proceedings of the 2004, vol.1, pp. 9-24, 2004.
- [2]. Bull, D. R. and Horrocks, D. H., "Primitive operator digital filters", *IEE Proc. Circuits, Devices and Systems*, pp. 401-412.
- [3]. Samuelli, H., "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients". *IEEE Trans. Circuits Syst*, vol.36, pp. 1044-1047.
- [4]. Redmill, D. W., Bull, D. R., and Dagless, E., "Genetic synthesis of reduced complexity filters and filter banks using primitive operator directed graphs". *IEE Proc. Circuits Devices Syst*, vol.147, pp. 303-310.
- [5]. Vinger, K. A. and Torresen, J., "Implementing evolution of FIR filters efficiently in an FPGA", *EH, 2003. Proceedings, NASA/DoD Conference on*, pp. 26-29, 2003.

[6]. Tufte, G. and Haddow, P. C., "Evolving an adaptive digital filter", EH, 2000. *Proceedings. The 2nd NASA/DoD Workshop on*, pp. 143-150, 2000.

[7]. Hounsell, B. and Arslan, T., "A programmable multiplierless digital filter array for embedded SoC applications", *IEE Electronics Letters*, vol. 37, no. 12, pp. 735-737, 2001.

[8]. Erba, M., Rossi, R., Liberali, V. and Tettamanzi, A. G., "An evolutionary approach to automatic generation of VHDL code for low-power digital filter". In *European Conference on Genetic Programming*, pp. 36-50, 2001.

[9]. www.atmel.com/dyn/products/app_notes.asp?family_id=623&part_id=2073

[10]. Erdogan, A. T., Hasan, M. and Arslan, T., "Algorithmic low power FIR cores", *Circuits, Devices and Systems, IEE Proceedings*, vol.150, Issue.3, pp. 155-160, 2003.