Fixed-point Configurable Hardware Components for Adaptive Filters

Romuald Rocher, Nicolas Herve, Daniel Menard, Oliver Sentieys
IRISA/ENSSAT
6 rue de Kerampont, 22300 Lannion
Email: name@irisa.fr

***Abstract*— To reduce the gap between the VLSI technology capability and the designer productivity, design reuse based on IP (Intellectual properties) is commonly used. In terms of arithmetic accuracy, the generated architecture can generally only be configured through the input and output word-lengths. In this paper, a new kind of fixed-point arithmetic IP is presented through the LMS and Delayed-LMS examples. The operator and memory word-lengths are optimized under an accuracy constraint defined by the user. To significantly reduce the optimization and design times, the architecture parameter determination is based on analytical approach.**

I. Introduction

The advance in VLSI technology offers the opportunity to integrate hardware accelerators and heterogenous processors in a single chip (System on Chip) or to obtain FPGA with several millions of gate-equivalent. Thus, complex signal processing applications can be now implemented in embedded systems. The time-to-market requires to reduce the system development time and thus, high-level design tools are needed. To reduce the gap between the hardware complexity and the designer productivity, design reuse [1] based on IP (Intellectual properties) has to be used.

To reduce the cost and the power consumption, the fixed-point arithmetic is required. For efficient hardware implementation, the chip size and power consumption have to be minimized. Thus, the goal of this hardware implementation is to minimize the operator word-length as long as the desired accuracy constraint is respected.

In an arithmetic point of view, the available IP are limited. The IP user can only configure the input and output word-length and sometimes the word-length of some specific operators. The link between the application performances and the data word-length is not immediate. Moreover, the fixed-point design search space can not be explored easily with this approach. Thus, the IP user must convert the application into fixed-point. But, the manual fixed-point conversion is a tedious, time-consuming and error prone task.

In this paper, a new kind of IP is presented through the LMS (Least Mean Square) and Delayed-LMS (DLMS) examples. These IP are configurable according to an accuracy constraint influencing the algorithm quality. The IP user specifies the accuracy constraint and the operator word-lengths are automatically optimized. The optimal operator word-lengths which minimize the architecture cost and respect the accuracy constraint must be researched. The accuracy constraint can be determined from the application performances through the technique presented in [2]. In our approach, analytical methods are used to evaluate the computation accuracy. Thus, the evaluation time is dramatically reduced compared to the simulation based methods and the fixed-point search space can be explored.

The paper is organized as follows. The LMS/DLMS algorithm and the generic architecture for the LMS/DLMS IP are presented in Section II. The fixed-point optimization process is detailed in Section III. The interest of our approach is underlined with several experiments in Section IV.

II. LMS/DLMS algorithm and architecture

A. LMS and DLMS algorithms

The aim of adaptive filters is to estimate a sequence of scalars from an observation sequence filtered by a system in which coefficients vary. These coefficients converge towards the optimum coefficients which minimize the mean square error (MSE) between the filtered observation signal and the desired sequence. This type of filters is used in different fields such as noise cancellation, equalization, linear prediction and channel estimation. The LMS based algorithms are the most common used because their implementation in embedded systems is simpler than the RLS algorithm. The LMS adaptive algorithm, presented in Figure 1.a, estimates a sequence of scalars y_n from a sequence of N-length vectors x_n [3]. The linear estimate of y_n is $w_n^t x_n$ where w_n is a N-length weight vector which converges to the optimal vector w_{opt}. The vector w_n is updated according to the following equation

$$w_{n+1} = w_n + \mu x_n e_{n-D} \quad \text{with} \quad e_n = y_n - w_n^t x_n \qquad (1)$$

where μ is a positive constant representing the adaptation step. The delay D is null for the LMS algorithm and different of zero for the Delayed-LMS.

B. Generic LMS architecture

The generic architecture for the LMS/DLMS algorithm is presented in Figure 1.b. The architecture is made-up of a filter part and an adaptation part to compute the new coefficients values. To satisfy the throughput constraint the filter part and the adaptation part can be parallelized. For the filter part, K multiplications are used in parallel and for the adaptation part K MAD (Multiply-Add) patterns are used in parallel. The

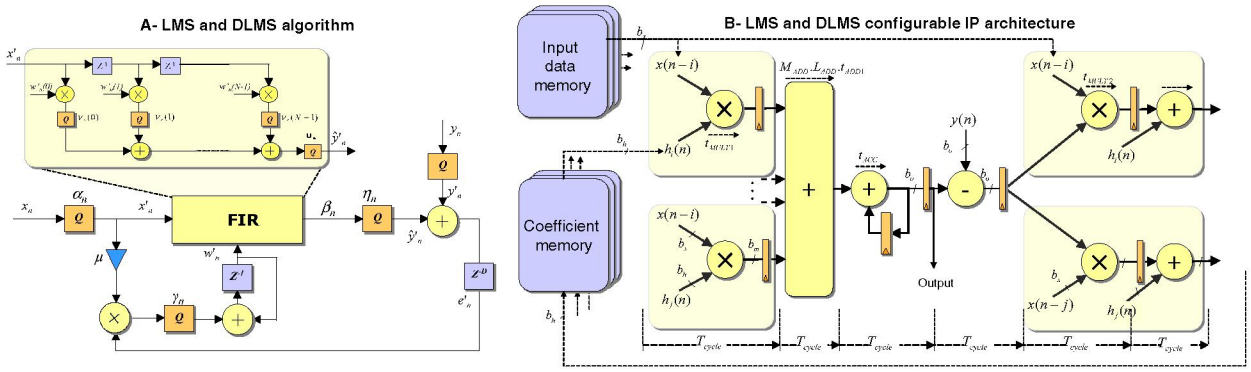

Fig. 1. LMS/DLMS algorithm and generic Architecture for the IP

different data word-lengths b in this architecture are b_x for the filter inputs, b_m for the filter multiplier output, b_h for the filter coefficients and b_e for the filter output.

To accelerate the computation, the processing is pipelined and the operators work in parallel. Let T_{cycle} be the cycle-time corresponding to the clock period. This cycle-time is equal to the maximum value between multiplier and adder latency. The filter part is divided into several pipeline stages. The first stage corresponds to the multiply operation. To add the different multiplication results, an adder based on a tree structure is used. This tree is made-up of $log_2(K)$ levels. This global addition execution is pipelined. Let L_{ADD} be the number of additions which can be executed in one cycle-time. Thus, the number of pipelined stages for the global addition is given by the following expression

$$M_{ADD} = \left\lceil \frac{log_2(K)}{L_{ADD}} \right\rceil \quad with \quad L_{ADD} = \left\lceil \frac{T_{cycle}}{t_{ADD1}} \right\rceil \qquad (2)$$

where t_{ADD1} is the adder latency. The last pipelined stage for the filter part corresponds to the final accumulation. The adaptive part is divided into three pipelined stages. The first one is for the subtraction. The second stage corresponds to the multiplication and the final addition composes the last stage. The timing constraint management is detailed in Section III-C.

III. Fixed-point optimization

The IP generation methodology is presented in Figure 2. The methodology first stage corresponds to the data dynamic range determination. In linear time-invariant systems, analytical approaches [5] can be used. But for the LMS algorithm, none of these methods is applicable. So a floating-point simulation is made to evaluate the dynamic range from the input data. Then, the binary-point position is deduced from the dynamic range to ensure that all data values can be coded to prevent overflow. The third stage is the data word-length optimization. The architecture cost C (area, energy consumption) is minimized under an accuracy constraint as expressed in the following expression

$$\min(C(\vec{b})) \; with \; SQNR(\vec{b}) \geq SQNR_{min} \qquad (3)$$

where $\vec{b}$ represents all data word-lengths. The optimization process requires to evaluate the architecture cost $C(\vec{b})$ and the computation accuracy $SQNR(\vec{b})$ defined through the Signal to Quantization Noise Ratio (SQNR) metric. These two processes are detailed in sections III-A and III-B. To determine the parallelism level K which allows to respect the throughput constraint, the architecture execution time is evaluated as explained in section III-C. Once the different operator word-lengths and the parallelism level are defined, the VHDL code representing the LMS or DLMS architecture at the RTL level is generated.

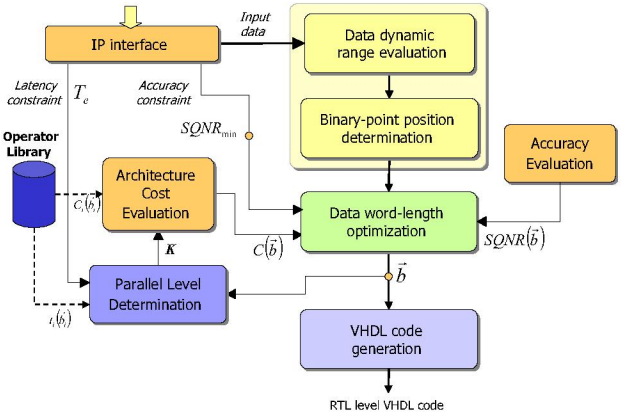

Fig. 2. Methodology for the Fixed-Point IP generation

A. Computation Accuracy Evaluation

To evaluate the accuracy, analytical methods are preferred to simulation based on methods which lead to too long simulation time. So the analytical expression of the SQNR in the LMS algorithm is computed as in [4].

a) The Fixed-point LMS Algorithm: In a fixed-point implementation, quantization noises are introduced as presented in Figure 1.a. The updated coefficient expression becomes

$$w'_{n+1} = w'_n + \mu e'_n x'_n + \gamma_n \qquad (4)$$

where γ_n is the noise associated with the term $\mu e'_n x'_n$ and depends on the way the filter is computed. The error in finite precision is given by

$$e'_n = y'_n - w'^t_n x'_n - \eta_n \tag{5}$$

with η_n the global noise in the inner product $w'^t_n x'_n$. This global noise is the sum of each multiplication output noise and output accumulation noise.

$$\eta_n = \sum_{i=0}^{N-1} v_n(i) + u_n \tag{6}$$

Moreover, a new term ρ_n is introduced

$$\rho_n = w'_n - w_n \tag{7}$$

ρ_n is the N-length error vector due to the quantization effects on coefficients. This noise can not be considered as the noise due to the quantization of a signal. The mean of each term is represented by m whereas σ^2 represents its variance and can be determined as explained in [7].

b) Noise power expression: The study is made at steady-state, once the filter coefficients have converged. The noise is measured at the filter output. The power of the error between filter output in finite precision and in infinite precision is determined. It is composed of three terms.

$$E(b_y)^2 = E(\alpha_n^t w_n)^2 + E(\rho_n^t x_n)^2 + E(\eta_n^2) \tag{8}$$

At the steady-state, the vector w_n can be approximated by the optimum vector w_{opt}. So the term $E(\alpha_n^t w_n)^2$ is equal to $|w_{opt}|^2(m_\alpha^2 + \sigma_\alpha^2)$ with $|w_{opt}|^2 = \sum w_{opt_i}^2$.

The second term is detailed in [6] and is equal to

$$E(\rho_n^t x_n)^2 = m_\gamma^2 \frac{\sum_{i=1}^{N}\sum_{k=1}^{N}(R_{ki}^{-1})}{\mu^2} + \frac{N(\sigma_\gamma^2 - m_\gamma^2)}{2\mu} \tag{9}$$

The last term $E(\eta_n^2)$ depends on the specific implementation chosen for the filter output computation (filtered data).

B. Architecture Cost Evaluation

The IP processing unit is based on a collection of operators extracted from a library. This library contains the arithmetic operators, the registers and the multiplexors for the different possible word-lengths. Each library element is automatically generated and characterized in terms of area and energy consumption from scripts for the Synopsys tools.

The IP architecture area and energy consumption are obtained from the sum of the different basic element area and energy consumption. The elements correspond to the memory (coefficients w_n and input data x_n), the operators (multiplier, adder, subtracter), the registers and the multiplexors used inside the datapath.

C. Throughput constraint

The system must verify a given constraint to ensure a real-time execution. The LMS Architecture presented in Figure 1 and detailed in section II-B is divided in two parts corresponding to the filter part and the adaptation part. The execution time of the filter part is obtained with the following expression

$$T_{FIR} = \frac{N}{K} T_{cycle} + M_{ADD} T_{cycle} + T_{cycle} \tag{10}$$

The execution time of the adaptation part is given by

$$T_{Adapt} = T_{cycle} + \frac{N}{K}(T_{cycle}) + T_{cycle} \tag{11}$$

The system throughput constraint depends on the chosen algorithm. For the LMS algorithm, the sampling period T_e must satisfy the following expression

$$T_{FIR} + T_{Adapt} < T_e \tag{12}$$

Even if the Delayed-LMS algorithm has a slower convergence speed compared to the LMS Algorithm, as the error is delayed, the filter part and the adaptation part can be computed in parallel which gives it a higher execution frequency. The constraints become

$$T_{FIR} < T_e \quad \text{and} \quad T_{Adapt} < T_e \tag{13}$$

The parallelism level is obtained by solving the expression 12 and 13. These expressions require the knowledge of the operator latency which depends on the operator word-lengths. Thus, firstly, the operator word-lengths are optimized with a K equal to 1. The obtained operator word-lengths allow to determine the operator latency. Secondly, the term K is computed from the throughput constraint and then, the operator word-lengths are optimized with the real value of K.

IV. Experiments and results

The LMS and DLMS IP blocks have been used for different experiments to underline the necessity to optimize the operator and memory word-lengths under an accuracy constraint. The IP users have to supply the reference and the input signal. For the architecture generation, the throughput constraint T_e and the accuracy constraint $SQNR_{min}$ must be defined.

The LMS and DLMS IP have been tested for different values of the throughput constraint T_e and the accuracy constraint $SQNR_{min}$. For each T_e and $SQNR_{min}$ value, the operator and memory word-lengths are optimized under the accuracy constraint. Then, the architecture is generated. The architecture area, the parallelism level and the energy consumption are measured and the results are presented respectively in Figure 3.a, 3.b and 3.c. The operator library has been generated from the $0.18\ \mu m$ technology from ST Microelectronics. The results are presented for an timing constraint between 60 ns and 170 ns and for an accuracy constraint between 30 dB and 90 dB.

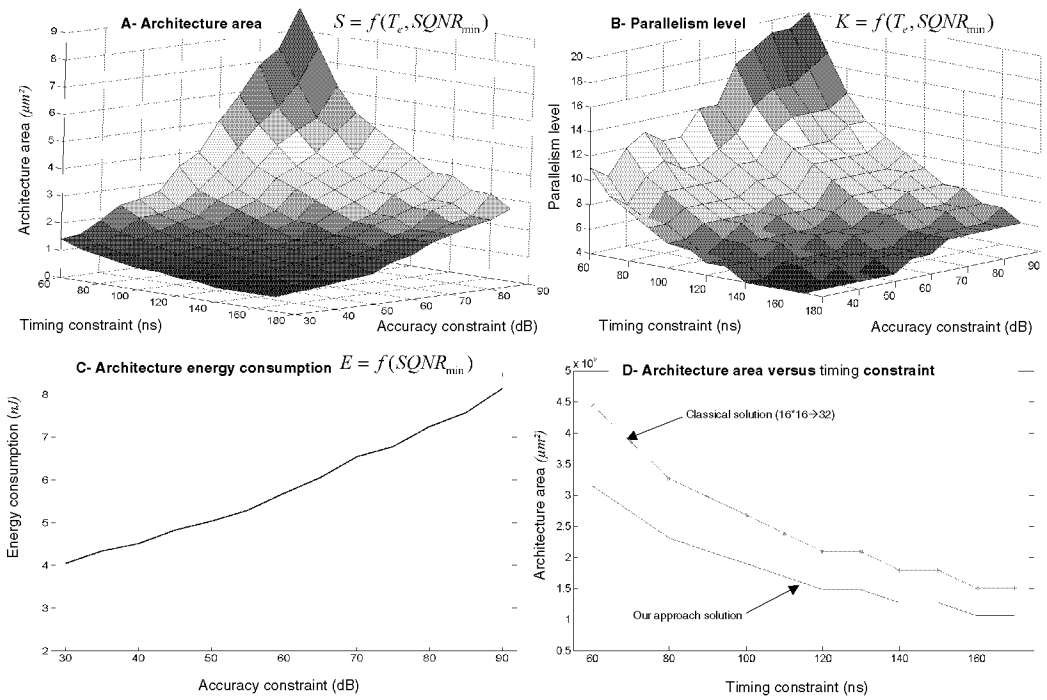

Fig. 3. Experiment results : architecture area, energy consumption and parallelism level for different value of accuracy and timing constraints

The architecture area increases when the timing constraint decreases. Indeed, to respect this constraint, the parallelism level K must be more important. More operators are needed and thus the processing unit area is increased. The architecture costs (area, energy consumption) increase with the accuracy constraint. High values of accuracy constraint require to use operators and data with a greater word-length. This operator word-length rising, increases the energy consumption and the area of the processing and memory units. Moreover, this operator word-length rising, increases the operator latency. Thus to respect the timing constraint, the parallelism level K must be more important and the processing unit area is increased.

Our results have been compared to a classical solution based on $16 \times 16 \rightarrow$ 32-bit multiplications and 32-bit additions. This solution leads to a $SQNR$ of 52 dB. The cost has been evaluated for the classical and our optimized approach for an accuracy constraint of 52 dB and with different timing constraints. The results are presented in Figure 3.d. For the same computation accuracy our approach reduces the architecture area by 30% and the power consumption by 23%.

V. Conclusion

In this paper, a new kind of fixed-point arithmetic IP has been proposed. The LMS/DLMS IP blocks have been detailed. A generic architecture has been proposed to adapt the parallelism level according to the timing and the computation accuracy constraints. To reduce the operator word-length optimization, the cost, the accuracy and the throughput constraints are evaluated analytically. The results underline the need to optimize the operator and memory word-lengths. Compared to a classical approach, for a same computation accuracy, the architecture area and the energy consumption are reduced respectively by 30 % and 23 %. With our approach, the user can optimize the trade-off between the architecture cost, the accuracy and the execution time. Accuracy models have been defined for other specific applications like NLMS, APA [10]. Moreover, an automatic and generic floating-to-fixed-point conversion methodology is under development [9].

References

[1] M. Keating and P. Bricaud. *Reuse Methodology Manual.* Kluwer Academic Publishers, 3rd edition, 2000.

[2] D. Menard, M. Guitton, S. Pillement, and O. Sentieys. Design and Implementation of WCDMA Platforms : Challenges and Trade-offs. In Proceedings of the *International Signal Processing Conference (ISPC 03)*, Dallas, April 2003.

[3] S. Haykin, "Adaptive Filter Theory", *Englewood Cliffs, NJ:Prentice-Hall, 2^{nd} edition, 1991.*

[4] C. Caraiscos, B. Liu, "A Roundoff Error Analysis of the LMS Adaptive Algorithm", *IEEE Transactions Acoustic, Speech, Signal Processing, vol ASSP-32, no.1, february 1984.*

[5] R. Kearfott. Interval Computations: Introduction, Uses, and Resources. *Euromath Bulletin*, 2(1):95–112, 1996.

[6] R. Rocher, D. Menard, O. Sentieys and P. Scalart, "Accuracy Evaluation of Fixed-Point LMS algorithm", Proceedings of the *IEEE ICASSP 2004, Montreal, Pages: 57-60, Vol. 5, may 2004.*

[7] G. Constantinides, P. Cheung and W. Luk "Truncation Noise in Fixed-Point SFGs", *IEE Electronic Letters, 35(23) : 2012-2014, november 1999.*

[8] D. Katsushige, N. Kiyoshi and K. Hitoshi, "Pipelined LMD Adaptive Filter Using a New Look-Ahead Transformation", *IEEE Transactions on Circuits and Systems, vol 46, january 1999*

[9] N. Herve, D. Menard, and O. Sentieys. Data wordlength optimization for fpga synthesis. In Proceedings of the *IEEE International Workshop on Signal Processing Systems, SIPS'05*, Athens, Grece, nov 2005.

[10] R. Rocher, D. Menard, O. Sentieys and P. Scalart, "Accuracy Evaluation of Fixed-Point APA algorithm", Proceedings of the *IEEE ICASSP 2005, Philadelphia, march 2005.*