Degrading Precision Arithmetics for Low-power FIR Implementation

(Invited Paper)

Pietro Albicocco, Gian Carlo Cardarilli, Alberto Nannarelli[1], Massimo Petricca and Marco Re

Department of Electronics, University of Rome Tor Vergata, Rome, I taly
[1]DTU Informatics, Technical University of Denmark, Kongens Lyngby, Denmark

***Abstract*— In this paper a review of different techniques used to implement highly optimized DSP systems is presented. The case of study is the implementation of parallel FIR filters aimed to applications characterized by high speed and high selectivity in frequency where at the same time low power dissipation is mandatory. After a review of the possible "standard" optimization techniques, the paper addresses aggressive methodologies where power and area savings are obtained by introducing the concept of "Degrading Precision Arithmetic" (DPA). Three different approaches are discussed: DPA-I, based on selective bit freezing, DPA-II, based on VDD voltage scaling, and DPA-III, based on power gating. Some theoretical/simulative analysis of the introduced arithmetic errors and some implementation results are shown. A discussion on the suitability of these methodologies on standard cell technologies and FPGAs is also addressed. In our experience, these techniques are well known in the scientific community, but they are not extensively known in the design communiy, and, consequently, they are scarcely utilized.**

I. Introduction

As it is well known, the power consumption in a CMOS circuit composed by N gates is

$$P_{TOT} = \underbrace{\sum_{i=1}^{N} \left(V_{DD}^2 C_{Li} + {E_i}^{int}\right) a_i f_{clk}}_{\text{dynamic}} + \underbrace{\sum_{i=1}^{N} V_{DD} I_i^{leak}}_{\text{static}} \quad (1)$$

where V_{DD} is the power supply voltage, C_{Li} represent the capacitive load, i.e. the sum of fan-out and interconnect capacitance, ${E_i}^{int}$ (internal energy) accounts for the transistor short-circuit currents and the power dissipated in the switching of the internal nodes, a_i is the node's switching activity and f_{clk} is the clock frequency. Expression (1) suggests a number of ways to reduce the power dissipation such as

- lowering the supply voltage V_{DD};
- reducing the switching activity a_i in some nodes of the circuit, or better, reducing the product $C_{Li} \cdot a_i$

Activities a_i can be modified by acting at different levels, for example by changing the number representation from the two's complement system (TCS) to sign-and-magnitude (for example in the case of small value anticorrelated data[1]), or by minimizing the first term of summation (1) by minimizing the products $C_{Li} \cdot a_i$ (in RNS the analysis of the switching power shows that higher values of actity are concentrated on small capacitance nodes). Another approach to save power is the reduction in precision of the number representation (this reduces the switching nodes), as shown in [3], [4], [2].

Power can be also saved by lowering V_{DD} below the absolute maximum ratings, accepting errors introduced by longer propagation delays in the digital logic. For example, by carefully designing the carry propagation logic in adders, it is possible to obtain results with an acceptable accuracy at lower supply voltages [6], [5].

In addition to the dynamic part, static power dissipation due to device leakage (I^{leak} is device's leakage currents) must be accounted in deep sub-micron CMOS technologies, especially if the device must operate in very low power mode, for example in wearable electronics.

There are three main sources of leakage current in a CMOS transistor:

1) Reverse-biased junction leakage current
2) Gate leakage current
3) Subthreshold leakage current

All these currents are dependent on the device geometry and the temperature. The gate and subthreshold leakages are the most important contributions to the static power dissipation. In the paper, after a short review of the standard techniques, we present the innovative techniques for power reduction in FIR digital filters.

II. FIR filters Standard Architectural Optimizations

In parallel implementations of FIR filters, the standard architectural optimization techniques aimed at a reduction in complexity are mostly oriented to a reduction in complexity of the multipliers.

A. Fixed Point Arithmetic Optimization

Fixed Point (FXP) simulation permits to obtain a sub-optimal solution for the fixed point arithmetics. The fixed point analysis of the filter permits to choose the wordlength for the coefficients (that affects the frequency mask), and for the representation of the intermediate results. Usually, the introduction of truncation post multiplication or at the end of the accumulation is accepted because the filtering operation improves the input signal-to-noise ratio (SNR). At the end of this process, the FXP formats in the filter signal flow diagram (SDF) are defined.

978-1-61284-857-0/11/$26.00

B. Multipliers

Reduction in complexity in the implementation of the filter can be obtained by an accurate analysis of the application and selection of the right multiplier architecture

- **General Purpose Multipliers:** required if the filter frequency response must be changed or if adaptive filtering is required.
- **Constant Coefficient Multipliers:** useful when the filter frequency response is fixed. In this case, a constant coefficient multiplier is implemented and optimization techniques based on Multiple Constant Multiplication (MCM) can be used. ([7], [8])
- **Truncated multipliers:** if truncation post multiplication can be tolerated, a most aggressive earning in complexity is obtained by propagating the truncation back in the multiply unit by eliminating not only logic gates required to obtain the truncated bits (this optimization is normally automatically implemented by the synthesizer), but also by eliminating the logic required in the generation of the carry bits. In this way a multiplier characterized by half the complexity (in the case of half output bits truncated) is obtained. Some overhead is introduced by the logic for the truncation error compensation: truncated multipliers type 1 constant correction factor [9], and type 2 variable correction factor [10].

III. Non Traditional Number Representations

A different number system representation affects the performance of the architecture of the arithmetic operators involved in the computation. Different possibilities have been explored in the literature such as the use of special radix number systems (for example the quater-imaginary number systems [19]), and by using not weighted representations such as the Residue Number System (RNS) [13] or the Logarithmic Number Systems (LNS) [12]. For example, the use of RNS permits a reduction in complexity of the multiplication by using the isomorphism technique that transforms multiplication in the sum of isomorphic indexes. On the other hand, the use of a different number system requires input and output conversions from traditional weighted number systems that must be carefully considered in the evaluation of the advantages offered by the special number representation. Another disadvantage of the RNS implementation of FIR filters resides in the integer representation that cannot benefit from post multiplication truncation, and the introduction of a coding overhead [14]. However, when high speed and high dynamic range DSP is required, the number of multiply-accumulate (MACC) operations for each output sample is high. In these cases the advantages obtained with the multiplier implementation give very interesting gains in power consumption. Similar advantages are obtained when complex filters must be implemented. In this case, the Quadratic Residue Number Systems (QRNS) can be used, This method strongly reduces the architectural complexity of complex product giving advantages also when the Golub rule is used [17]. Different papers have been presented in the literature showing interesting gains both in standard cell and FPGA implementations [16], [16], [15].

IV. Special Optimizations: Degrading Precision Arithmetic

A digital system is generally developed to work in the most critical operating conditions. But it operates under more relaxed conditions for almost all the time. The degrading precision arithmetic provide a way to dynamically reduce power consumption when the system works under better conditions than the worst case for which it has been designed. Different approaches can be exploited at different levels. For example power gating could limit dynamic but also the leakage power consumption.

A. DPA-I

This approach reduces the switching activity by blocking the least significant bits to fixed values[18]. Two approaches are feasible: forcing of a subword in a fixed state (Forcing); preserving the least significant portion of the word at the value assumed at the time in which the clock is disabled (Freezing). Forcing can be implemented in two different ways: Forcing to 0 (0-Forcing) and Forcing to 0.5 (0.5-Forcing). From the error characterization point of view, 0-Forcing of the n least significant bits in a word has the same error statistics of the truncation scheme. The mean value and variance of the 0-Forcing error are

$$m_\epsilon = q\frac{2^n - 1}{2^{n+1}} \tag{2}$$

$$\sigma_\epsilon^2 = \frac{q^2}{12}(1 - 2^{-2n}) \tag{3}$$

where q represents the quantization step.
0.5-Forcing on the n least significant bits of a word has the error characteristic of the mid-raiser quantizer (Fig. 1). The resultant error Probability Density Function (PDF) are identical to the rounding error PDF. The mean value and variance of the 0.5-Forcing error are

$$m_\epsilon = -q\frac{1}{2^{n+1}} \tag{4}$$

$$\sigma_\epsilon^2 = \frac{((q/2) - (-q/2))^2}{12} = \frac{q^2}{12}. \tag{5}$$

The Freezing approach consists in disabling dynamically the clock in the least significant bits, so that they will not change. This procedure is equivalent to store a uniformly distributed random number in the least significant bits of tthe data. Consequently, the freezing error can be modeled by the following procedure: a 0-Forcing action that discards the n least significant bits followed by an addition of a value extracted by a uniformly distributed process. The error is consequently modeled as the sum of the two uniformly distributed random variables, i.e. their PDFs convolution

$$f_{E_{Froz}}(\epsilon_{Froz}) = f_{E_{0Forc}}(\epsilon_{0Forc}) * f_{X_{Frozen}}(x_{Frozen}) \tag{6}$$

From this analysis, appears that 0-Forcing represents the worst case, due to the non zero mean value of the truncation

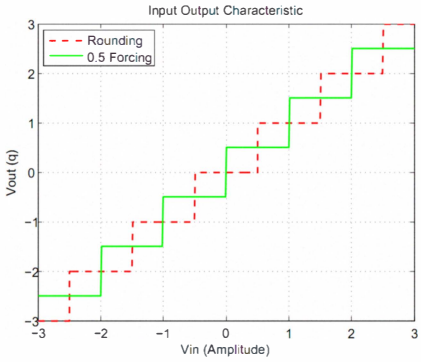

Fig. 1. 0-Forcing, 0.5-Forcing and Freezing error characteristics.

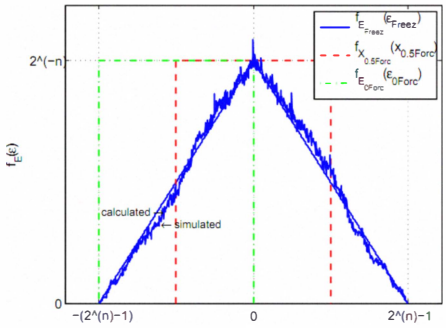

Fig. 2. 0-Forcing, 0.5-Forcing and Freezing error PDF comparison.

operation. The 0.5 Forcing and Freezing are characterized by a quasi zero mean value but by a different shape in the error PDF. Figure 2 shows the PDFs of the error in the case of 0-Forcing, 0.5-Forcing and Freezing (theoretical and estimated by simulation).

Freezing and/or a mix of freezing and 0/0.5 Forcing can be used in the implementation a FIR filter (in this work we refer to direct form FIR filters) to save power at the expenses of precision. It is interesting to note that none of these techniques require additional hardware differently from the case of the rounding operator implementation. Just clock select logic is required to implement a variable precision system.

To lower the power consumption in a filter, we operate by limiting the precision at the input and/or after multiplication. Obviously, the best results in terms of saved power are obtained by acting on the filter input, in fact, this affects the size of all the subsequent blocks (in a parallel FIR filter all the multipliers). The type of limiting precision operator on the input signal depends on its statistical properties. If the input is uniformly distributed, the best technique is 0.5-Forcing. The application of these techniques to the internal nodes of the filter SDF requires some considerations. For example, taking into account that, in a FIR filter the coefficients amplitude decreases as $|sin(x)/x|$, the outer taps correspond to very small values a_k, so the dynamic range at the output of the multipliers is small. In this situation the application of a 0.5-Forcing introduces errors larger than a 0-Forcing. For this reason, the use of Freezing in the implementation of a FIR

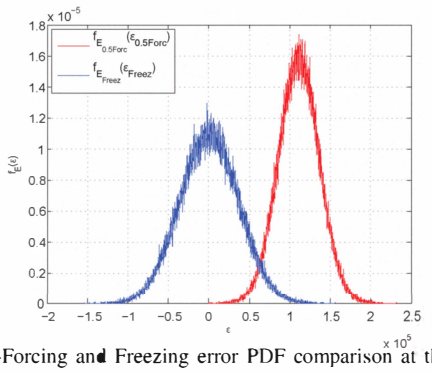

Fig. 3. 0.5-Forcing and Freezing error PDF comparison at the output of a 512 order FIR filter.

gives a sort of adaptive behavior that shows an interesting effect on the pdf of the error at the filter output. To evaluate this "adaptive behavior" the following experiment has been realized To validate this behaviour N=512 FIR filter with the following characteristics have been simulated

- variable cut-off frequency $f_c \in (0.6, 0.7)$ step 0.01
- Number of bits for the FXP implementation: $x(n)$=12 bits, a_k=12 bits
- DPA-I post multiplication on the 12 lsbs, 0-Forcing and 0.5-Forcing

100 simulation slots have bee simulated, each one representing a different frozening state then the statistics of the output error (as the difference between the actual simulation and the fixed point one) has been estimated. The simulation results are shown in figure 3 where we can observe how the mean value of the 0.5-Forcing PDF is greater than that of the Freezing. This result shows that when the designer evaluate the different available strategies to adaptively degrade the arithmetic precision in a filter, better SNR can be obtained by using freezing at no expense of the hardware resources. Consequently the Freezing technique can be very useful in different situations.

B. DPA-II

Differently from DPA-I, this technique operates on the reduction of the supply voltage V_{DD}. In this case, any V_{DD} reduction will imply a quadratic reduction of the power dissipation. As explained in [18], because of the increased delay, the units in the datapath have to be redesigned to make sure the error is introduced only in the least significant portion of the datapath to keep the degradation from being too large. DPA-II paradigm is interesting but requires an ad hoc redesign of the arithmetic units in order to limit the error introduced by timing violations. The adaptation of the arithmetic blocks depends on the final application. In fact, specific constraints are imposed by different applications. For example the case of image processing and audio, (where the final evaluatation is based on psycometric evaluations) is really different from communication systems where the quantitative measurement of BER is mandatory.

C. DPA-III

The idea of the DPA-III is similar to that of the DPA-I, but in this case, in place of disabling the clock, the power supply

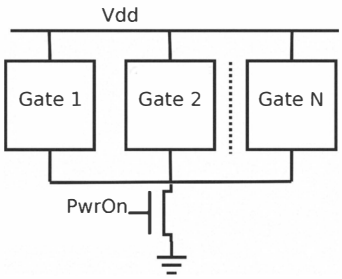

Fig. 4. Power gating structure.

is disconnected from the cells. This can be done with power gating. The advantage over DPA-I is that the static power is reduced as well, and that portions of the combinational units can be disabled independently of clock gating. The basic scheme used in power gating structure is given in Fig. 4. In standby mode the supply voltage is turned off by using a NMOS transistor in series with the transistors on each logic block as shown in Fig. 4. In the active state, the $PwrOn$ transistor is on. To guarantee the proper functionality of the circuit, the $PwrOn$ transistor must be carefully sized to limit the voltage drop. In order to reduce the area and power overhead, a single transistor can be used for each group of gates (N gates in Fig. 4). In the case of standard cell design, a single transistor can be used for each standard cell. In DPA-III the n least significant bits of the data, wiil be disabled by power gating. The errors introduced on the final result are equivalent with those described in DPA-I, but static power dissipation should be greatly reduced. Since the switching of the transistor $PwrOn$ requires additional dynamic power, the method is more effective when active and standby states are maintained for long periods. The disadvantage of power gating is that it is a post-layout technique and that the sleep/wake-up times are longer than disabling/enabling the clock. Clearly, at present, power gating cannot be applied to systems implemented on FPGAs.

V. Conclusion

In this paper, a review of the most important standard techniques that can be used to improve area, speed and power in the implementation of FIR filters is presented. Beyond the more standard techniques a review on methods based on non traditional number representations is given. Moreover, some new trends that are based on the concept of degraded precision arithmetic are discussed. This new approach allows to save power at the expense of precision. The case of DPA-I has been investigated more in depth showing that, when a variable precision system is implemented, the freezing technique permits to lower the error without overhead in hardware. Some brief consideration on more innovative technologies such as DPA-II and DPA-III are presented to open the discussion. Some of these techniques should be suitable to FPGA implementation due to the increased infractuctures for clock gating offered in the new families. More details will be given in the presentation.

References

[1] A. P. Chandrakasan, R. W. Brodersen, *Minimizing Power Consumption in Digital CMOS Circuits*, Proc. of IEEE, vol. 83, no. 4, pp. 498-523, April 1995.

[2] A. Bonanno, A. Bocca, A. Macii, E. Macii, M. Poncino, *Data-Driven Clock Gating for Digital Filters*, Proc. of 19th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS 2009), pp. 96-105, 2009.

[3] P. Larsson, C. J. Nicol, *Self-Adjusting Bit Precision for Low-Power Digital Filters*, Proc. of IEEE Symposium on VLSI Circuits, pp. 123-124, June 1997.

[4] S. Yoshizawa, Y. Miyanaga, *Use of a Variable Wordlength Technique in an OFDM Receiver to Reduce Energy Dissipation*, IEEE Transactions on Circuits and Systems-I, vol. 55, no. 9, pp. 2848-2859, October 2008.

[5] D. R. Kelly, B. J. Phillips, S. Al-Sarawi, *Increasing Throughput of a RISC Architecture Using Arithmetic Data Value Speculation*, Proc. of 43rd Asilomar Conference on Signals, Systems, and Computers, pp. 915-920, Nov. 2009.

[6] S.-L. Lu, *Speeding Up Processing With Approximation Circuits*, IEEE Computer Magazine, vol. 37, no. 3, pp. 67-73, March 2004.

[7] O. Gustafsson, A. G. Dempster, L. Wanhammar, *Extended Results for Minimum-Adder Constant Integer Multipliers*, IEEE International Symposium on Circuits and Systems, vol. 1, pp. I-73, 2002.

[8] Y. Voronenko, M. Püschel, *Multiplierless Multiple Constant Multiplication*, ACM Transactions on Algorithms, Vol. 3, No. 2, May 2007.

[9] M. J. Schulte, E. E. Swartzlander Jr., *Truncated Multiplication with Correction Constant*, in VLSI Signal Processing VI, pp. 388-396, IEEE Press, (Eindhoven, Netherlands), October 1993.

[10] E. J. King and E. E. Swartzlander, Jr., *Data-Dependent Truncation Scheme for Parallel Multipliers*, in Proceedings of the 31st Asilomar Conference on Signals, Systems, and Computers, vol. 2, pp. 1178-1182, (Pacific Grove, CA), November 1997.

[11] J. M. Jou, S. R. Kuang, R. D. Chen, *Design of Low-Error Fixed-Width Multipliers for DSP Applications*, IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 46, pp. 836-842, June 1999.

[12] Y. Sun, M. S. Kim, *A High-Performance 8-Tap FIR Filter Using Logarithmic Number System*,Proceedings of IEEE International Conference on Communications, June 2011.

[13] G. C. Cardarilli, A. Del Re, A. Nannarelli, M. Re *Low Power and Low Leakage Implementation of RNS FIR Filters*, IEEE Thirthy-Ninth Asilomar Conference on Signals, Systems and Computers, pp. 1620-1624, October 28, November 2005.

[14] G. C. Cardarilli, A. Del Re, A. Nannarelli, M. Re, *Impact of RNS Coding Overhead on FIR Filters Performance*, IEEE Asilomar Conference on Signal Systems and Computers, pp. 1426-1429, November 2007.

[15] G. C. Cardarilli, A. Nannarelli, M. Re, A. Del Re, *Power Characterization of Digital Filters Implemented on FPGA*, ISCAS 2002, IEEE International Symposium on Circuits and Systems, vol. 5, pp. 801-804, May 2002.

[16] Haohuan Fu, O. Mencer, W. Luk, *Optimizing Residue Arithmetic on FPGAs*, Best Paper Award Proceeding of International Conference on Field-Programmable Technology 2008 (ICFPT'08), December 2008.

[17] G. C. Cardarilli, A. D. Re, A. Nannarelli, and M. Re, *Low-Power Implementation of Polyphase Filters in Quadratic Residue Number System*, Proc. of IEEE International Symposium on Circuits and Systems, vol. 2, pp. 725-728, September 2004.

[18] M. Petricca, G. C. Cardarilli, A. Nannarelli, M. Re and P. Albicocco, *Degrading Precision Arithmetic for Low Power Signal Processing*, Proc. of IEEE Asilomar Conference on Signal Systems and Computers, pp. 1163, Pacific Groove (CA), November 2010.

[19] G. C. Cardarilli, A. Nannarelli, M. Re, *On the Comparison of Different Number Systems in the Implementation of Complex FIR Filters*, IEEE VLSI SOC 2008 Post Conference Book, Springer Publishers, Book Editors: D. Soudris, C. Piguet, R. Reis