A STUDY ON DISCRETE WAVELET TRANSFORM IMPLEMENTATION FOR A HIGH LEVEL SYNTHESIS TOOL

Jean-Marc Tourreilles, Christophe Nouët, Eric Martin
LESTER, University of South Brittany,
2, rue Le Coat Saint-Haouen,
Lorient 56100, FRANCE
Tel: 33/02/97/88/05/45; fax: 33/02/97/88/05/51
e-mail: Jean-Marc.Tourreilles@univ-ubs.fr

ABSTRACT

Actually, the error from the fixed-point implementation is not taken in account in the high level synthesis tools. The processing signal algorithms are implemented without quality evaluation. This paper presents a fixed-point implementation methodology applied to the Discrete Wavelet Transform.

1 INTRODUCTION

Nowadays, signal processing applications implementation can be done using floating-point or fixed-point data format. The first choice leads to better numerical results than the second one because of the big dynamic range. This difference is also due to the overflow problem in fixed-point format. However the fixed-point data format is cheaper (area cost) than floating-point data format.
In this paper, we present a methodology to control overflow problems and truncation error in finite-precision implementation [8]. Firstly, we show a way to avoid overflow problems. Secondly, we present a method to evaluate computation noise power. Finally, we determine the data bit number to respect some quality criteria constraint. This methodology is integrated in High Level Synthesis Tool *GAUT* [7]. A Discrete Wavelet Transform example illustrates this methodology [4]. We consider that the image compression is done in a embedded system, then the data format used is the fixed-point one because of the area and comsumption low cost. For the best reception, we consider the reconstruction on ground, is done using floating-point format.

2 IMPLEMENTATION METHODOLOGY

The main steps of the methodology are detailed above.

2.1 Dynamic And Shifting

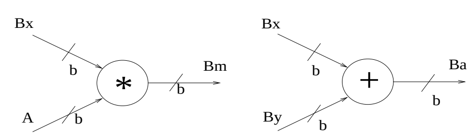
We propose two methods to avoid overflow. The first method, called external shifting, consists in a shifting of the input algorithm data.
Therefore, we evaluate the biggest computated values in the algorithm process. For such a purpose, we calculate the algorithm output value only considering the unity as input and modulus of intermediate results. Then we determine the smallest power of two ($EchL$) which is immediately greater than the biggest computation value. We divide all the input datas by $EchL$ which a shifting (linear shift register). The second method, called internal shifting, consists in executing a shifting as soon as overflow occurs.
From this, a new algorithm structure appears and must be analysed in terms of computation noise.

2.2 SNR Operation Modelisation

When one the shifting methods is executed, we must evaluate computation-noise power of the algorithm. The goal of this method is to establish an analytic formulation, function of the data bits number. For that, we must modelise operation. In operation output, we consider two noise origins:

- **Local Error**: From the fixed-point data truncation. As we use left-positioned fixed point arithmetic, this error only occures for multiplication and shifting.
- **Propagated Error**: from the accumulation of local error during the algorithm.

The choosen models are given above [2].



The local error and the propagated error are given by the following equations:

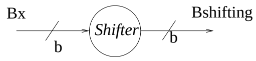
$$B_m^2 = A^2.E[B_x^2] + \frac{q^2}{12} \tag{1}$$

$$B_a^2 = E[B_x^2] + E[B_y^2] + \frac{q^2}{12} \tag{2}$$

- B_x and B_y are the noise power of x and y.
- $E[x]$ is the expectation of x.

- b is the data format.
- q is the quantum ($= 2^{-b}$).
- A is a constant.

The choosen model of the computation noise from a shifting is given by:



Where

$$B_{shifting} = \frac{q^2}{2^{3d}} \sum_{i=1}^{2^d - 1} i^2 + B_x \tag{3}$$

Where d is the number of shift made on the data and q is the quantum.
The flow graph provides the operation on the processing data. With this flow graph and the operations models choosen, we can establish analytic formulation of computation noise power. Then noise power is given by:

$$B = Const \times \frac{q^2}{12} \tag{4}$$

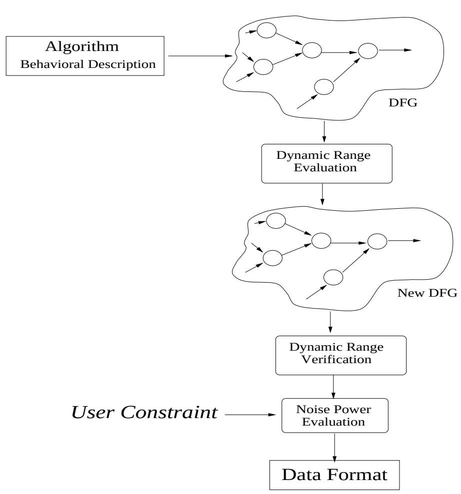
Where $Const$ is a constant value, q is the quantum ($q = 2^{-b}$) and b the number of bits used to data format codage.

2.3 Application SNR Constraint And Data Format

At first, signal processing algorithm can specified by its quality criteria. Secondly, the analytic formulation of the algorithm noise power provides the computation-noise power as a function of the bits number. From this we can obtain the number of bit needed for good implementation. The High Level Synthesis Tool user have to choose some criteria matched to the kind of signal processing.

2.4 Integration in GAUT

This methodology aims to help for circuit designer. Its integration in $GAUT$ tool [7] is done from data flow graph. It is detailed above.



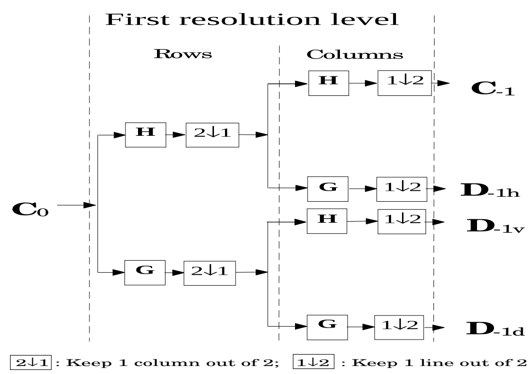
A recursive computation method allows us to obtain the $EchL$ factor and to establish a new algorithm data flow graph. This new data flow graph is transformed in noise flow graph and gives us the noise power. The $GAUT$ tool user gives a constraint and the number of bits needed is obtained. So the synthesis is done using appropriated operators librairy.

3 AN APPLICATION : THE DISCRETE WAVELET TRANSFORM 2-D, 3 STAGES

In this section we apply the method to a Discrete Wavelet Transform algorithm. The algorithm choosen is the $DWT\ 2-D$ on three stages. The used format is the left-positioned fixed-point format. First, we detail the main characteristics of the DWT algorithm.

3.1 DWT Implemetation Methodology

The characteristics of the DWT studied are given in [1], [3], [6]. The multiresolution analysis step is decomposed in three stages. The l^{th} DWT stage gives us , D_{-lh}, D_{-lv}, D_{-ld} coefficients and low resolution image C_{-l}. For the analysis step, we consider both propagated and local error, but only propagated errors occure during the reconstruction step because of the floating-point computation format.



The choosen filters are given in the following table [1]:

	filter coefficients				
	h(0)	h(±1)	h(±2)	h(±3)	h(±4)
H	0,602949	0,266864	-0.078223	-0.016864	0.026749
G	0,557543	-0,295636	-0,028772	0,045636	
H̃/2	0,557543	-0,295636	-0,028772	0,045636	
G̃/2	0,602949	0,266864	-0.078223	-0.016864	0.026749

The filters used to compute the wavelet coefficients are given in the following table:

DWT coef	D_{-1h}	D_{-1v}	D_{-1d}	$\cdots$	D_{-3h}	D_{-3v}	D_{-3d}
filters	H,G	G,H	G^2	$\cdots$	H^4,G	H^4,G,H	H^4,G^2

The filters used to obtain the low resolution image are the **H** filters (six times used).

3.1.1 Overflow

Two methods are proposed to avoid overflow.

External Shifting: The DWT basic structure is the FIR. Treated datas are computed by six FIR during the analysis step and by six other FIR during the reconstruction step. The maximum of dynamic is the output data of the third step analysis and is given by:

$$\max(y_n) = \left[\sum_{j=0}^{8} |h_j|\right]^6 = 6,917249327 < 2^3 \tag{5}$$

Then we can divide the compression input datas by 8 (i.e. three shiftings). To get the output original pixels value we have to mutiply the reconstruction output data by 8.

Internal Shifting: In this case, we have to find where overflow occurs. Knowing that the output filter are bounded by the following expressions,

$$\max\left(y_{n,h}\right) = \sum_{j=0}^{8} |h_j| = 1,38 \tag{6}$$

$$\max(y_{n,g}) = \sum_{j=0}^{6} |g_j| = 1,30 \tag{7}$$

where $y_{n,h}$, $y_{n,g}$ are respectivelly the output of H and G, we can divide the input of each H, G filter by 2 (one shifting). Then we have to multiply by 4 the data output of each reconstruction step.

3.1.2 Evaluation of the Noise Power

External Shifting: If P_i is from the propagated error and L_i from the generated error (local error), the output noise power for the i^{th} FIR of analysis is given by:

$$\begin{aligned} B_{output,i} &= B_{input,i} \times P_i + L_i \\ B_{input,i+1} &= B_{output,i} \end{aligned} \tag{8}$$

Where $B_{input,i}$ is the input data noise power for the i^{th} FIR of the process. The low resolution image and DWT-coefficients noise power of the l^{th} step analysis is given by:

$$B_{D_{-l}} \simeq B_{C_{-l}} = B_{C_0} \prod_{i=0}^{2l-1} P_i + \left[\sum_{i=0}^{2l-2} \left(L_i \prod_{i=k+1}^{2l-1} P_i\right) + L_{2l-1}\right] \tag{9}$$

Where B_0 is given by:

$$B_{C_0} = \left[\frac{3}{2^{3d-2}} \sum_{i=1}^{2^d-1} i^2\right] \frac{q^2}{12} = \left[\frac{3}{2^7} \sum_{i=1}^{7} i^2\right] \frac{q^2}{12} \tag{10}$$

During the reconstruction, only propagated error occurs. If $\tilde{P}_i$ is is from the propagated error for $\tilde{h}$ and $\tilde{g}$ filters, then:

$$B_{output,i} = B_{input,i} \times \tilde{P}_i \tag{11}$$

The three steps give us:

$$B_R = 8^2 \left\{ B_{R_2} \left[\sum_{j=0}^{n_{\tilde{h}}} \tilde{h}_j^2\right]^2 + B_{C_1} \left[2 \sum_{j=0}^{n_{\tilde{g}}} \tilde{g}_j^2 \sum_{j=0}^{n_{\tilde{h}}} \tilde{h}_j^2 + \left(\sum_{j=0}^{n_{\tilde{g}}} \tilde{g}_j^2\right)^2 \right] \right\} \tag{12}$$

The numerical application gives us:

$$B_R = 1392 \frac{q^2}{12} \tag{13}$$

Internal Shifting: In this case, the ouput noise of a filter with a unity shifting on input is given by :

$$B_{input,i+1} = \frac{B_{output,i}}{4} + B_d \tag{14}$$

The numerical application gives us:

$$B_R = 32709 \frac{q^2}{12} \tag{15}$$

3.1.3 DWT Noise Criteria

To deduce the number of bits from the noise power formulation, we have to define some DWT quality criteria [5]. We will consider three criteria.

- Maximal Error:

$$e_{max} = \max \frac{\left(I_0(l,c) - \underline{I_0}(l,c)\right)^2}{LC} \qquad \forall \quad l,c \tag{16}$$

- Mean Squared Error:

$$MSE = \sum_{l,c} \frac{\left(I_0(l,c) - \underline{I_0}(l,c)\right)^2}{LC} \tag{17}$$

- Peak Signal To Noise Ratio:

$$PSNR_{dB} = 10 \log_{10} \frac{1}{\sum_{l,c} \frac{\left(I_0(l,c) - \underline{I_0}(l,c)\right)^2}{LC}} \tag{18}$$

The criterion e_{max} gives us a error peak that it isn't matched to ours statistical operation models. The criteria MSE and $PNSR_{dB}$ seem to be well-matched to our study. When DWT is implemented, the fixed-point noise has to be insignificant compared with the $PNSR_{dB}$. First we can define the $PSNR_{computation}$:

$$PSNR_{computation} = 10 \log_{10} \left(\frac{1}{B_R}\right) \tag{19}$$

A typical value of a $PSNR_{dB}$ is 30 dB, then considering the previous assumption, we can obtain the following inequality:

$$\begin{aligned} &10 \log_{10} \left(\frac{1}{A \frac{q^2}{12}}\right) > 50dB \\ \Longrightarrow \quad & b > \frac{1}{2} \log_2 \left(\frac{10^5 A}{12}\right) \end{aligned} \tag{20}$$

The two shifting methods results give us 12 and 15 bits for respectivelly external and internal shifting.

3.2 Results

The *DWT* is synthesises by a high level synthesis tool *GAUT*. The implemented image processing is a $512 \times 512 - image\ DWT$. The tool gives a structural description and an area evaluation of the *ASIC* architecture processing unit (UT). The synthesis is done under a 40 ms time constraint (25 images per second). The results are summarised in the following table.

	external shifting	internal shifting
number of bits	12	15
CPU area $mm/10^{-}3$	14586	20736

The high level synthesis gives us different architectures for the two methods. Then the area cost is different between internal-shifting *ASIC* and external-shifting *ASIC*.

4 CONCLUSION

This presented methodology allows us to implement signal and image processing in finite precision format under signal to noise ratio. This methodology is implemented in an high level synthesis tool. Overfows problems are treated before the evaluation of the computational noise power. From a user constraint, a data format is given and is taken into account during the synthesis. The interest of this methodology is to consider the parameter "computational noise" as a quality criterion. This leads to an optimization of the target architecture (ASIC) area, and allows us to decrease the consumption of the circuit (relative to floating point format).

References

[1] M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, Image Coding Using Wavelet Transform, *IEEE Transactions On Image Processing*, Vol. 1, No. 2, April 1992.

[2] M. Bellanger, Traitement numérique du signal, *Technical and Scientifical collection of Telecommunications CNET-ENST*, 1984.

[3] I. Daubechies, Orthonormal Bases of Wavelets with Finite Support - Connection with DiscreteFilters, AT&T Bell Laboratories, 600 Mountain Avenue, Murray Hikk, NJ07974, U.S.A.

[4] A. Grzeszczak, M. K. Mandal, S. Panchanathan, VLSI Implementation of Discrete Wavelet Transform, *IEEE Transactions On VLSI Systems*, Vol. IV No. 4, December 1996.

[5] C. Lecordier, *Internal report LESTER lab.*, 1997.

[6] S. G. Mallat, A Theory for Multiresolution Signal Decomposition: The Wavelet Transform, *IEEE Transactions On Pattren Analysis And Machine Intelligence*, Vol. II, No. 7, July 1989.

[7] E. Martin, O. Sentieys, H. Dubois, J.L. Philippe, GAUT, an architectural synthesis tool for dedicated signal processors, *Proceedings EURO-DAC 93.*

[8] J.M. Tourreilles, C. Nouët, E. Martin, Méthode d'implantation d'algorithmes de traitement du signal en précision finie, *GRETSI 97, Proceedings of the* 16^{th} *Symposium on Image and Signal processing*, Grenoble, September 1997.