A DISTRIBUTED ARITHMETICS ARCHITECTURE FOR QUANTIZATION SNR IMPROVEMENT BASED ON LUT RECODING

José Rubio Fernández and Josep Sala Alvarez

Dept. of Signal Theory and Communications
Polytechnic University of Catalonia.
Campus Nord UPC, D5 Building, C/ Jordi Girona, 1-3.
08034 Barcelona (Spain)
Tel: +34 93 4015894 Fax: +34 93 4016447 E-mail: {jrubio, alvarez}@gps.tsc.upc.es

ABSTRACT

The architectural design of Application Specific Integrated Circuits (ASIC) for Digital Communications demands efficient algorithms for carrying out the signal processing involved in the modulation and demodulation procedures. The search for compact, well-performing architectures is therefore crutial in the implementation of communication systems on silicon.

In this paper we describe efficient Distributed Arithmetics (DA) architectures for Digital Filtering in Communication Systems. In particular, the specific design of FIR filters at bit-level is considered. We present a memory recoding method for DA look-up tables (LUT's) that leads to substancial improvement in quantization SNR without the necessity to increase the LUT wordlength, ASIC synthesis results will also be provided.

1. INTRODUCTION

1.1. DISTRIBUTED ARITHMETICS

In this section we will describe only its canonical form. Let $\mathbf{h}^{\mathrm{T}}=[h_0,h_1,\ldots,h_{L-1}]^{\mathrm{T}}$ be the vector containing the filter coefficients and let $\mathbf{x}(n)^{\mathrm{T}} = [x(n),x(n-1),\ldots,x(n-(L-1))]^{\mathrm{T}}$ be the vector containing the input (not quantized samples). Let $\mathbf{x}_q(n)^{\mathrm{T}} = [\mathrm{q}(x(n)),\mathrm{q}(x(n-1)),\ldots,\mathrm{q}(x(n-(L-1)))]^{\mathrm{T}}$ be the vector containing the input quantized samples. DA carries out the following operation with minimum quantization noise,

$$y(n)=\mathbf{h}^{\mathrm{T}}\mathbf{x}_q(n)=\sum_{i=0}^{L-1}h_i\mathrm{q}(x(n-i)) \qquad (1)$$

Note that the coefficients in $\mathbf{h}$ are not quantized. Let us consider an incoming stream of data quantized to b_x bits,

$$\begin{aligned}\mathrm{q}(x(n)) &= V2^{b_x-1}\sum_{m=0}^{b_x-1}x_{m,n}w_m2^{-m} \qquad (2)\\ w_m &= \begin{array}{ll}+1 & ,\ 1\le m\le b_x-1\\ -1 & ,\ m=0\end{array}\end{aligned}$$

where V is a constant defining the dynamic range of quantization and $x_{m,n}$ constitute the two's complement encoding of the data stream $x(n)$. Using the expression for $\mathrm{q}(x(n))$ in (1) we get

$$\begin{aligned}y(n) &= \sum_{i=0}^{L-1}h_i\overline{V}\sum_{m=0}^{b_x-1}x_{m,n-i}w_m2^{-m} \qquad (3)\\ &= \sum_{m=0}^{b_x-1}w_m2^{-m}\overline{V}\sum_{i=0}^{L-1}h_ix_{m,n-i}\\ &= \sum_{m=0}^{b_x-1}w_m2^{-m}y_m(n)\end{aligned}$$

with $\overline{V}=V2^{b_x-1}$. Therefore, the output $y(n)$ is constructed with the shift and addition of the b_x auxiliary terms $y_m(n)$. These are just filtering operations themselves, taking the streams $x_{m,n-i}$ as inputs to filter $\mathbf{h}$. These auxiliary filtering operations are pre-calculated in floating point[1] precision and stored in a table $\mathcal{M}$. This table is looked up at the addresses $\mathbf{x}_m^{@}(n)=\left[x_{m,n},x_{m,n-1},\ldots,x_{m,n-(L-1)}\right]$ so that $y_m(n)=\mathcal{M}\left[\mathbf{x}_m^{@}(n)\right]$. Distributed Arithmetics builds on this decomposition using the quantized version of the table,

$$\mathcal{M}\left[\mathbf{x}_m^{@}(n)\right]\overset{def}{=}\mathrm{q}\left(y_m(n)\right)$$

The estimate of the quantized filter output, $y_{DA}(n)$, is thus calculated as,

$$y_{DA}(n)=\sum_{m=0}^{b_x-1}w_m2^{-m}\beta\mathcal{M}\left[\mathbf{x}_m^{@}(n)\right]$$

The $\beta=\frac{1}{2}$ factor has been introduced in the definition because the accumulations performed in (3) are subject to overflowing: therefore, words from the look-up table must be extended by one bit before accumulation.

Also, care must be taken at this stage with the dynamic range associated with the previous quantization operation in table $\mathcal{M}$. If we want to minimize the quantization noise introduced by the table we should make sure that the maximum absolute value of all possible memory positions coincides with $V_{\max}$ as defined in (12). Hence, the more advantageous definition of DA must be adopted,

[1] C's 'double' type or MATLAB's 8-byte precision.

$$y_{DA}(n) \stackrel{def}{=} \sum_{m=0}^{b_x-1} w_m 2^{-m} \beta \mathcal{M}\left[\mathbf{x}_m^{@}(n)\right] \qquad (4)$$

$$\mathcal{M}\left[\mathbf{x}_m^{@}(n)\right] \stackrel{def}{=} \mathrm{q}\left(A \sum_{i=0}^{L-1} h_i x_{m,n-i}\right)$$

with the re-scaling factor A suitably chosen to fulfil,

$$V_{\max} = A \max_{\{x_{m,n-i}=0,1\}} \left| \sum_{i=0}^{L-1} h_i x_{m,n-i} \right| \qquad (5)$$

1.2. ARCHITECTURES

The DA scheme represented in equation (4) is expressed in terms of the summation of several values. Each of these values may be generated sequentially with the same hardware or in parallel and added to a summation network [1]. In the first case, the filter output will not be available until after as many clock cycles as input quantization bits, while in the second case, one output sample will be generated every clock cycle. We will distinguish therefore between Sequential and Parallel Distributed Arithmetics: SDA (Figure 1) and PDA (Figure 2).

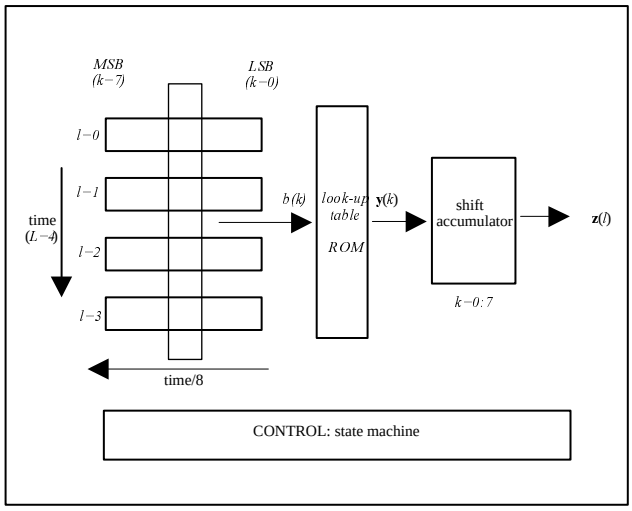

Figure 1: Description of Sequential Distributed Arithmetics. Some overhead is necessary for the control state machine and data routing.

1.3. HISTOGRAMS

A useful tool in the evaluation of complex fixed point arithmetic structures is the histogram. The distribution profile of data input values can be used to improve performance in DA implementations [3].In this paper we will use the histogram of LUT's (ROM's) as a tool to characterize some details in Distributed Arithmetics implementation of digital filters. It serves the purpose of monitoring the word-length occupancy in concatenated arithmetic operations when large signal records are used to probe fixed-point implementations of signal processing algorithms.

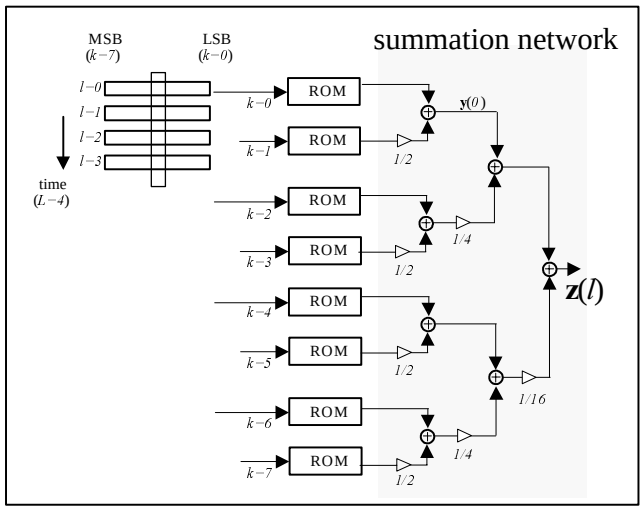

Figure 2: Description of Parallel Distributed Arithmetics. The overhead in SDA is not necessary but ROM look-ups and additions are replicated, not shared.

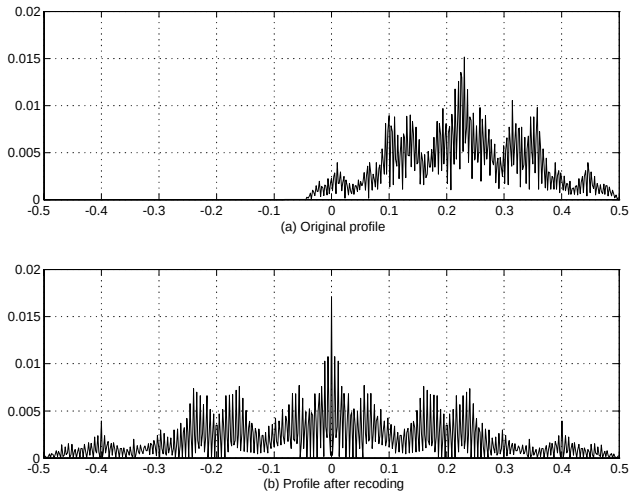

Figure 3: Profiles of a look-up table: before (upper) and after recoding (lower).

2. MEMORY RECODING (I)

In a previous section we defined the way in which the DA look-up tables are generated in the classical scheme. In this chapter we will show that recoding the look-up table leads to substantial improvements in quantization SNR without the necessity to increase the look-up table wordlength. Let us now examine the profile of several typical DA look-up

tables (see Figure 3(a)). We can appreciate that their histogram is biased towards the positive range. In terms of quantization this is suboptimum as the available memory wordlength is not fully used. We will seek to improve quantization SNR by several dB by removing any offset present in the look-up table and re-scaling to use up the full dynamic range. We will show that this offset can be added back outside the look-up table.

Let us define the mean of the look-up table,

$$\mu = 2^{-L} \sum_{k=0}^{2^L-1} [\mathcal{M}_k]_{\text{not quantized}} \tag{6}$$

We will generate a new look-up table as the quantization of,

$$\begin{aligned} y'_m(n) &= A' \left(y_m(n) - \mu \right) \\ \mathcal{M}'_k &= \mathrm{q}\left(y'_m(n) \right) \end{aligned} \tag{7}$$

with A' a new scaling factor to use up the whole dynamic range of the b_m-bit memory wordlength. Let us now proceed with $\mathcal{M}'_k$ as,

$$\begin{aligned} \mathcal{M}'_k &= \mathrm{q}\left(y'_m(n) \right) \\ &= \mathrm{q}\left(A' y_m(n) - A'\mu \right) \\ &= \mathrm{q}\left(A' y_m(n) \right) + \mathrm{q}\left(-A'\mu \right) + \\ &\quad \left\langle A' y_m(n) \right\rangle + \left\langle -A'\mu \right\rangle - \left\langle \left\langle A' y_m(n) \right\rangle + \left\langle -A'\mu \right\rangle \right\rangle \\ &= \mathrm{q}\left(A' y_m(n) \right) + \mathrm{q}\left(-A'\mu \right) + \varepsilon_m(n) \end{aligned}$$

with $\varepsilon_m(n)$ including all the residue terms. Note that in the third equality of the previous equation, the argument of $\mathrm{q}(A' y_m(n))$ is exceeding the dynamic range associated with quantization. Nevertheless, this does not constitute a problem for two reasons: (a) the function $\mathrm{q}(x)$, by definition, does not clip values with $|x| > V_{\max}$ and (b) the variable $A' y_m(n)$ is only an intermediate arithmetic value, inexistent in the hardware implementation of this scheme. In fact, we are only *virtually* exceeding the dynamic range. Therefore, if we define the new DA output as,

$$y'_{DA}(n) \stackrel{def}{=} \sum_{m=0}^{b_x-1} w_m 2^{-m} \beta \mathcal{M}' \left[\mathbf{x}_m^{@}(n) \right]$$

we get,

$$\begin{aligned} y'_{DA}(n) &= \sum_{m=0}^{b_x-1} w_m 2^{-m} \beta \mathrm{q}\left(A' y_m(n) \right) \\ &\quad + \sum_{m=0}^{b_x-1} w_m 2^{-m} \beta \mathrm{q}\left(-A'\mu \right) \\ &\quad + \sum_{m=0}^{b_x-1} w_m 2^{-m} \beta \varepsilon_m(n) \end{aligned} \tag{8}$$

The second term in 8 can be expressed as,

$$\begin{aligned} y'_{DA}(n)_2 &= \beta \mathrm{q}\left(-A'\mu \right) \sum_{m=0}^{b_x-1} w_m 2^{-m} \\ &= \beta \mathrm{q}\left(-A'\mu \right) \left(-1 + \sum_{m=1}^{b_x-1} 2^{-m} \right) \\ &= \beta \mathrm{q}\left(-A'\mu \right) \left(-1 + \frac{1}{2} \cdot \frac{1 - 2^{-b_x+1}}{1 - 2^{-1}} \right) \\ &= -\beta \mathrm{q}\left(-A'\mu \right) 2^{-b_x+1} \end{aligned}$$

Hence, we have that we can define the output $z_{DA}(n)$ of a new DA architecture as the accumulation of values of look-up table $\mathcal{M}'$, $y'_{DA}(n)$, plus a constant term,

$$\begin{aligned} z_{DA}(n) &= y'_{DA}(n) + \beta \mathrm{q}\left(-A'\mu \right) 2^{-b_x+1} \\ &= \sum_{m=0}^{b_x-1} w_m 2^{-m} \beta \mathrm{q}\left(A' y_m(n) \right) + \\ &\quad + \sum_{m=0}^{b_x-1} w_m 2^{-m} \beta \varepsilon_m(n) \end{aligned} \tag{9}$$

A new noise term, $\sum_{m=0}^{b_x-1} w_m 2^{-m} \beta \varepsilon_m(n)$, appears, but it is far exceeded by the amplification A'. This new amplification factor was not possible in last chapter's scheme because it would have saturated the memory wordlength.

2.1. OFFSET POST-CORRECTION

The correction term $-y'_{DA}(n)_2$ provides an additional advantage: in the last accumulation performed by DA, the weight $w_0 = -1$ is enforced. In two's complement, this is equivalent to inverting all bits and adding 1. The addition of the correction term is roughly equivalent to substracting 1. Hence, during the final clock cycle in distributed arithmetics, it is only necessary to invert all bits of the bit-extended looked-up word, add it up to the shift-accumulator register and correct the overall result with the addition of the constant correction term,

$$\begin{aligned} K &= \overline{V} \beta 2^{-b_x+1} + \beta \mathrm{q}\left(-A'\mu \right) 2^{-b_x+1} \\ &= \beta \left(\overline{V} + \mathrm{q}\left(-A'\mu \right) \right) 2^{-b_x+1} \\ &= \beta \overline{\mathrm{q}}\left(A'\mu \right) 2^{-b_x+1} \qquad , \ \mu > 0 \end{aligned}$$

where the first term is accounting for the addition of a digital one after bit-extension by one bit and the second term is the correction by $-y'_{DA}(n)_2$. Using the convention $\overline{V} = 1/2$, we have,

$$K = \left(\frac{1}{2} + \mathrm{q}\left(-A'\mu \right) \right) 2^{-b_x}$$

An alternative procedure is to initialize the accumulator in distributed arithmetics to the value,

$$y_{acc,0} = \beta \left(\overline{V} + \mathrm{q}\left(-A'\mu \right) \right) \tag{10}$$

so that after the b_x iterations (equivalent to $b_x - 1$ shifts) of the shift-accumulator it will become $\beta\left(\overline{V} + \mathrm{q}\left(-A'\mu\right)\right) 2^{-b_x+1}$ (intermediate rounding operations are not taken into account!) which is precisely the term that must be added.

3. MEMORY RECODING (II)

We will show here that the memory recoding procedure previously presented can be posed in terms of a recoding of the $x_{m,n}$'s. We will consider the following mapping[2],

$$\begin{array}{rcl} \bar{x}_{m,n} & = & 2x_{m,n} - 1 \\ \{x_{m,n} : 0, 1\} & \longmapsto & \left\{\bar{x}_{m,n} \colon -1, +1\right\} \end{array}$$

Also, we will find a new expression for the mean μ introduced in 6. Let $\mathbf{b}_k = [x_{m,n}, x_{m,n-1}, \ldots, x_{m,n-L+1}]^{\mathrm{T}}$ denote the k-th binary address(in the $\{0,1\}$ encoding) among all 2^L possible addresses and let $\mathbf{c}_k = \left[\bar{x}_{m,n}, \bar{x}_{m,n-1}, \ldots, \bar{x}_{m,n-L+1}\right]^{\mathrm{T}}$ denote the k-th binary address (in the $\{-1,+1\}$ encoding). Let $\mathbf{1}$ be the all-ones vector, then, the mean μ can be expressed as,

$$\begin{aligned} \mu &= 2^{-L} \sum_{k=0}^{2^L-1} A\mathbf{h}^{\mathrm{T}}\mathbf{b}_k \\ &= 2^{-L} A \sum_{k=0}^{2^L-1} \mathbf{h}^{\mathrm{T}} \frac{1}{2}\left(\mathbf{c}_k + \mathbf{1}\right) \\ &= 2^{-L-1} A\mathbf{h}^{\mathrm{T}} \sum_{k=0}^{2^L-1} \mathbf{c}_k + 2^{-L-1} A \sum_{k=0}^{2^L-1} \mathbf{h}^{\mathrm{T}}\mathbf{1} \end{aligned}$$

but $\sum_{k=0}^{2^L-1} \mathbf{c}_k = 0$ as each possible $\mathbf{c}_{k'}$ has its $-\mathbf{c}_{k'}$ counterpart. Therefore,

$$\mu = \frac{1}{2} A\mathbf{h}^{\mathrm{T}}\mathbf{1} = \frac{1}{2} \sum_{i=0}^{L-1} Ah_i$$

using this equality in 7 we get,

$$\begin{aligned} y'_m(n) &= A'\left(y_m(n) - \mu\right) \qquad (11) \\ &= A'\left(A \sum_{i=0}^{L-1} h_i x_{m,n-i} - \frac{1}{2} \sum_{i=0}^{L-1} Ah_i\right) \\ &= \frac{1}{2} A' \sum_{i=0}^{L-1} Ah_i \left(2x_{m,n-i} - 1\right) \\ &= \frac{1}{2} A' \sum_{i=0}^{L-1} Ah_i \, \bar{x}_{m,n-i} \end{aligned}$$

It is shown that the memory recoding procedure outlined in the previous section is equivalent to evaluating the outputs of the filter when the input data is $\{-1,+1\}$ encoded.

In summary, the histogram of the look-up table will now be symmetric. The profile of this table will display reflection symmetry. Hence, it will be necessary to store only one half of the memory.

[2] In [1], Withe use the same mapping to explain a method to reduce the memory size by half to a 2^k word ROM

filter ID	roll-off	n^o coef.	ΔSNR (dB)	$\overline{\mu}$ ratio
f1	0.75	14	5.29	0.91
f2	0.6	14	5.12	0.89
f3	0.5	16	4.99	0.87
f4	0.35	16	4.78	0.84
f5	0.25	24	4.40	0.79
f6	0.2	32	4.17	0.76

Table 1: Simulation results.

4. SNR IMPROVEMENT

The theoretical evaluation of the expected improvement in quantization SNR associated with recoding is formulated in terms of the amplification factor A'. This factor can be obtained as,

$$V_{\max} = A' \max_{\{x_{m,n-i}=0,1\}} \left| A\left(\sum_{i=0}^{L-1} h_i x_{m,n-i} - \mu\right)\right| \qquad (12)$$

Hence, as the relationship between the median of the look-up table and the filter coefficients is

$$\mu = \frac{1}{2} \sum_{i=0}^{L-1} h_i$$

and take into account the definition of the amplification factor A from (12), we have that the amplification factor A' will be,

$$A' = 2 \cdot \frac{\max\left(\sum_{h_i>0} h_i \, , \, -\sum_{h_i<0} h_i\right)}{\sum_{i=0}^{L-1} |h_i|} \leq 2 \qquad (13)$$

which shows that, at most, we can get a one bit improvement (approximately 6 dB) in the quantization SNR,

$$\Delta SNR = 20 \log_{10} A' \leq 20 \log_{10} 2 \approx 6dB \qquad (14)$$

The resulting histogram for the recoded version of the LUT's corresponding to Figure 3a, is showed in Figure 3b.

4.1. SIMULATION RESULTS

We have applied the explained recoding to some benchmark filters to be used in matched filtering in a IF sampling scheme. The modulation is uncoded QPSK on a square root raised cosine pulse (SQRRC). Some parameters of the benchmark filters and the associated SNR improvement are show in Table 1.

b_m MSB ... LSB	E_b/N_o dB	$(E_b/N_o)_Q$	SNR_Q
8 8 8 8 8 8 8 8	43.03	45.94	41.93
8 8 8 8 8 8 8 3	42.89	45.94	41.77
8 8 8 8 8 8 4 3	42.70	45.94	41.61
8 8 8 8 8 5 4 3	42.62	45.94	41.55
8 8 8 8 6 5 4 3	42.42	45.94	41.30
8 8 8 7 6 5 4 3	41.92	45.94	40.77
8 8 7 6 5 4 3 2	41.44	45.94	39.60
8 8 6 5 4 3 2 1	37.95	45.94	35.99
8 8 7 6 5 4 3 1	41.14	45.94	39.07
8 8 7 6 5 4 2 2	41.39	45.94	39.34
8 8 7 6 5 3 2 2	41.32	45.94	38.94

Table 2: Filter f3. Results for PDA architecture with variable quantization.

b_m MSB ... LSB	E_b/N_o dB	$(E_b/N_o)_Q$	SNR_Q
9 9 9 9 9 9 9 9 9	40.97	45.10	42.22
9 9 9 9 9 9 9 9 4	40.95	45.10	42.20
9 9 9 9 9 9 9 9 3	40.90	45.10	42.10
9 9 9 9 9 9 9 9 2	40.80	45.10	41.96
9 9 9 9 9 9 9 9 1	40.29	45.10	41.09
9 9 9 9 9 9 9 4 3	40.89	45.10	41.99
9 9 9 9 9 9 5 4 3	40.76	45.10	41.93
9 9 9 9 9 6 5 4 3	40.69	45.10	41.71
9 9 9 9 7 6 5 4 3	40.63	45.10	41.60
9 9 9 8 7 6 5 4 3	40.14	45.10	41.20
9 9 9 9 7 7 4 4 3	40.28	45.10	41.41

Table 3: Filter f4. Results for PDA architecture with variable quantization.

Notice that the improvement ΔSNR is directly related to the roll-off as equation (13) is a measure of the time oscillations of the impulse response $h(n)$ and hence, of the filter selectivity.

If we look at the mean μ, we obtain also the relationship,

$$\mu = \frac{V_{\max}}{2} \frac{\sum_{h_i>0} h_i + \sum_{h_i<0} h_i}{\max\left\{\sum_{h_i>0} h_i, -\sum_{h_i<0} h_i\right\}} = \frac{V_{\max}}{2}\overline{\mu} \qquad (15)$$

and hence,

$$-\frac{1}{2}V_{\max} \leq \mu \leq +\frac{1}{2}V_{\max} \qquad (16)$$

The ratio $\overline{\mu}$ is also tabulated for the benchmark filters in 1,which also displays dependence on the associated roll-off. Of course, $\overline{\mu}$ and A' are not independent as the larger $\overline{\mu}$, the larger A' (a better improvement of ΔSNR).

5. PDA ARCHITECTURE WITH VARIABLE QUANTIZATION

The PDA architecture, constituting an unfolding in the space domain of the iterations carried out by SDA in the time domain, has the inconvenient that the area cost of the original SDA scheme is multiplied approximately by the number of input quantization bits. Nonetheless, PDA is advantageous in the sense that as one memory is instantiated for each input bit of weight 2^{-k}, quantization of that memory can be made variable depending on the importance the corresponding weight. Hence, memory look-ups associated with the least significant input bit vectors can be quantized in a rougher way. Simulations have been run to test PDA with variable quantization.

5.1. SIMULATION RESULTS

Simulation has proceeded for the six benchmark filters used in section 4. The equivalent E_b/N_o measuring the contribution of ISI and quantization noise at the output of the digital filter provides only information on performance at the strobe. Alternative measures of signal to quantization noise ratio are also necessary for comparison. Therefore, the following parameters are defined,

1. b_m, number of memory quantization bits.
2. E_b/N_o evaluated at the output of distributed arithmetics due to the ISI plus quantization noise. The centroid with respect to which the equivalent ISI is computed is calculated for the ideal not-quantized filter when the input signal is one single pulse, not quantized.
3. $(E_b/N_o)_Q$, E_b/N_o defined as bit-energy with respect to output quantization noise associated with output rounding: $\sigma_q^2 = \frac{1}{12}\Delta^2 = \frac{1}{12}2^{-2b_y}$. For the previous E_b/N_o measure to be reliable it must hold that $(E_b/N_o)_Q >> E_b/N_o$. Otherwise, the mean absolute value of the ISI is comparable to or lower than the output quantization step.
4. SNR_Q, measured signal to quantization noise ratio at the output of the DA filter. The reference signal for determining the noise power term is the not-quantized input filtered by the not-quantized, full precision ideal filter. It includes the contribution of input quantization, filter quantization and output rounding.

The obtained results corresponding to filters with ID f3 and f4 in Table 1, are shown in Table 2 and Table 3, respectively. The used number of input quantization bits and number of quantization bits at the output of distributed arithmetics (rounding) are 8 in all the cases.

The results shown that the application of variable quantization to the PDA architecture could decrease further the associated complexity at a small loss in performance.

6. REFERENCES

[1] White, S. A., "Applications of Distributed Arithmetic to Digital Signal Processing: A Tutorial Review", *IEEE ASSP Magazine,* vol. 63, pp. 4-19, (July 1989).

[2] A. Peled, B. Liu, "A New Hardware Realization of Digital Filters", *IEEE Trans. on ASSP*, vol. ASSP-22, (December 1974).

[3] Mehendale, M., Sinha, A., Sherlekar, S.D., "Low Power Realization of FIR Filters Implemented Using Distributed Arithmetic", *Proceedings of Design Automation Conference*, Asia and South Pacific, pp. 151-156, (1998).