# WORDLENGTH DETERMINATION ALGORITHMS FOR HARDWARE IMPLEMENTATION OF LINEAR TIME INVARIANT SYSTEMS WITH PRESCRIBED OUTPUT ACCURACY

*S. C. Chan and K. M. Tsui*

Department of Electrical and Electronic Engineering,
The University of Hong Kong, Pokfulam Road, Hong Kong.

***Abstract*****—**This paper proposes two novel algorithms for optimizing the hardware resources in finite wordlength implementation of linear time invariant systems. The hardware complexity is measured by the exact internal wordlength used for each intermediate data. The first algorithm formulates the design problem as a constrained optimization, from which an analytic closed-form solution of the internal wordlengths subject to a prescribed output accuracy can be determined by the Lagrange multiplier method. The second algorithm is based on a discrete optimization method called the Marginal Analysis method, and it yields the desired wordlengths in integer values. Both approaches are found to be very effective and they are well-suited to large scale systems such as software radio receivers. Design examples show that the proposed algorithms offer better results and a lower design complexity than conventional methods.

## I. INTRODUCTION

When implementing digital filters, it is well known that there are two sources of error, namely coefficient round-off error and signal round-off error [1]. The former happens when the real-valued coefficients of the digital filter, obtained say by the Parks-McClellan algorithm, are rounded to their fixed-point representations to simplify the hardware implementation. On the other hand, signal round-off error occurs when overflow occurs due to insufficient internal wordlength and improper scaling; and when rounding is performed for long intermediate data after multiplications with the filter coefficients. It is usually more difficult to handle in hardware implementation because complicated hardware for detecting overflows, etc., would significantly slow down the throughput of the system.

A considerable amount of researches has been done on the analysis of these two problems [1]. The efficient realization of digital filters however is still a very active area of research. One pioneer work in addressing the coefficient roundoff problem and efficient realization of digital filters is due to Lim *et al.* [2], where the filter coefficients are represented as sum-of-power-of-two (SOPOT) or canonical signed digits (CSD) representations. Since multiplications with SOPOT coefficients require limited shifts and additions, it gives rise to very efficient multiplier-less realization. Since then, various algorithms for determining these SOPOT coefficients were proposed.

On the other hand, to satisfy a given output accuracy, one usually employs a fixed and long wordlength for all intermediate data, which increases the hardware complexity. In [3], a flexible approach using a random search algorithm was proposed to minimize the hardware complexity of a target FIR system while satisfying the given output accuracy. The SOPOT coefficients, rounding options, and internal wordlengths are determined to minimize the number of adder cells to meet a given specifications on frequency response and output accuracy. The searching algorithm is similar to the mutation of genetic algorithm (GA) and the random walk in stimulated annealing. The main difference is that the search space is limited to a small neighborhood of the real-valued solution, greatly reducing the search time. However, its searching time will increase considerably when large number of variables is involved since the searching method is random in nature.

In this paper, we propose two novel algorithms to determine the minimum hardware complexity of linear time invariant systems subject to a prescribe output accuracy. The hardware complexity is measured by the exact internal wordlength used for each intermediate data. The output accuracy of the digital filters is specified statistically by its output noise power due to the rounding operations performed, using the commonly used uncorrelated white noise model. It is shown that if the wordlengths are treated as real-valued quantities, then this problem can be solved using the Lagrange multiplier method [4] and an analytic close-form solution for the wordlength of all intermediate signals can be obtained. A similar work can be found in [5], which was concerned with the real-time wordlength adaptation in adaptive FIR filters. One common limitation of these approaches is that the solution so obtained is most probably not integer-valued and hence it has to be rounded to the next largest integer. On the other hand, one may use it as an initial guess for further optimization, say using the random search algorithm proposed in [3]. This greatly reduces the searching time for a nearly optimal solution.

By recognizing that the close similarity between the wordlength determination problem and the bit allocation algorithm for data compression [6] – [8], we further propose an algorithm based on the marginal analysis method [7], [8]. More precisely, the basic idea of the proposed algorithm is to increase the wordlength of one of the intermediate output points successively in order to lower the output round-off noise power as much as possible, until the given bit accuracy or bit budget (total wordlength) is met. To illustrate the effectiveness of the proposed approaches, the hardware implementation of the software radio receiver proposed in [3] is considered. Unlike the random search algorithm, design result shows that the proposed algorithm works well with large number of variables. Furthermore, when coupling the optimal solution obtained from the previous method with the bit-allocation algorithm, a near optimal solution can be obtained within several seconds in a Pentium 4 personal computer.

The paper is organized as follows: the signal round-off error model and the issue of overflow handling are briefly reviewed in Section II. Section III describes the proposed algorithms for determining the internal wordlength using the Lagrange multiplier and bit allocation algorithm. This is then followed by a design example and comparison in Section IV. Finally, conclusion is drawn in Section V.

## II. SIGNAL ROUND-OFF AND OVERFLOW ANALYSIS

### A. — Signal Round-off Analysis

Signal round-off errors occur due to rounding of the intermediate signal after multiplications. Since the exact round-off errors are difficult to analyze, they are usually treated as

0-7803-8834-8/05/$20.00 ©2005 IEEE.

uncorrelated white noises. For rounding operations, the quantization noise will have a zero mean with a variance $\sigma$ equal to $\varDelta^2/12$, where $\varDelta$ is the quantization step-size. In other words, the variance is determined by the number of fractional bits that is retained after multiplication**.** In fixed-point arithmetic, each intermediate signal can be represented in the form of $< I/F >$, where $I$ is the number of integer bits including the sign bit and $F$ is the number of fractional bits. In general, if $F$ bits are rounded to $B$ bits, where $B < F$, then the noise variance $P_e$ is given by:

$$P_e = \frac{\varDelta^2}{12} = \frac{2^{-2(B-1)}}{12} = \frac{2^{-2B}}{3}, \qquad (1)$$

where $\varDelta = 2^{-(B-1)}$. Without loss of generality, consider the round-off noise model of the LTI system in figure 1, where the signals to be quantized are $s_i[n]$ for $i = 1,...,M$ ; $M$ is the total number of rounding sources. From (1), if $s_i[n]$ is rounded to $b_i$ bits, then the variance of the quantization error, $e_i[n]$, is given by $2^{-2b_i}/3$**.** Let the transfer function from $s_i[n]$ to the output $y[n]$ be $H_i(\omega)$, $i = 1,...,M$. Furthermore, we assume that the noise sources are uncorrelated. Hence the variance of the output noise at $y[n]$ can be expressed as follows:

$$\sigma_e^2 = \sum_{i=1}^{M} c_i \sigma_i^2 = \frac{1}{3}\sum_{i=1}^{M} c_i 2^{-2b_i}, \qquad (2)$$

where $c_i = \frac{1}{2\pi}\int_{-\pi}^{\pi} |H_i(\omega)|^2 \, d\omega = \sum_k h_i^2[k]$ ; $H_i(\omega)$ is the transfer function from $s_i(n)$ to $y(n)$; and $h_i(k)$ is the impulse response corresponding to $H_i(\omega)$. The output accuracy $A$, in terms of the number of fractional bits, is then approximately given by:

$$A \approx \left|10 \cdot [\log_{10}(\sigma_e^2)]\right| / 6 \text{ bits}. \qquad (3)$$

It should be noted that the larger the number of noise sources, the lower will be the accuracy. The noise power can however be reduced by increasing the internal wordlengths for the fractional bits, at the expense of increased hardware complexity.

B. — Overflow Handling

Signal overflows occur when the allocated wordlength of the integer bits is insufficient to accommodate the growth in integer wordlength of the signal after additions. In order to avoid overflow, more bits must be allocated to the integer part of the adder output and the register holding it. There is, however, an option to retain or decrease the number of bits in the fractional part, depending on the required output accuracy. In FIR filters, it is possible to determine whether signal overflow will occur at a particular adder using the L1 scaling measure. More precisely, the input signal $x[n]$ is assumed to take on its maximum value denoted by $x_{\max}$. Then, the maximum value after implementing the *k-th* impulse response coefficient of the target system is bounded by:

$$y_{\max,k} = x_{\max}\sum_{n}^{k}\left|h[n]\right|. \qquad (4)$$

Using (4), it is possible to determine the worst-case integer wordlength of each adder and hence the size of its output register to avoid signal overflow. It should be noted that there are other methods such as L2 scaling to handle signal flows. However, there is still a small probability that overflows will occur. To determine this option, we can imagine that a noise is generated by the rounding option and the minimum acceptable wordlength is then determined as if it was a rounding source due to multiplication. If the minimum wordlength obtained is larger than the existing wordlength, then the wordlength has to be increased. Otherwise, rounding can be performed if the additional noise generated does not violate the prescribed accuracy. In IIR filters, scaling is usually performed at certain stages of the system to avoid overflow. Since scaling is a multiplication operation, it can be treated similarly by our model.

## III. Wordlength Determination

### A. — Analytic Solution

The problem of determining the wordlengths for a given output noise power $\sigma_e^2 = P_{spec}$ can be formulated as the following constrained optimization problem:

$$\min_{\boldsymbol{b}} \sum_{i=1}^{M} w_i b_i = \boldsymbol{w}^T \boldsymbol{b}$$

$$\text{subject to } \frac{1}{3}\sum_{i=1}^{M} c_i 2^{-2b_i} = P_{spec}. \qquad (5)$$

where $\boldsymbol{w}$ is a constant weight vector, $\boldsymbol{b}$ is the variable vector representing the fractional part of the internal wordlengths to be determined. In most cases, $w_i$ are chosen as one for all *i*. If we allow $\boldsymbol{b}$ to be taken on real values, instead of integer values, then the minimization problem in (5) can be solved analytically using the method of Lagrange multiplier [4]. Define the following Lagrangean function:

$$L(\boldsymbol{b},\lambda) = \boldsymbol{w}^T \boldsymbol{b} + \lambda\left(\frac{1}{3}\sum_{i=1}^{M} c_i 2^{-2b_i} - P_{spec}\right), \qquad (6)$$

where $\lambda$ is the Lagrange multiplier associated with $\boldsymbol{b}$. Taking the partial derivatives and setting them to zero yields:

$$\frac{\partial L(\boldsymbol{b},\lambda)}{\partial b_i} = w_i - \frac{2\lambda c_i}{3} \cdot 2^{-2b_i} \cdot \ln 2 = 0.$$

From which, we obtain $\lambda$ as follows:

$$\lambda = \frac{3w_i}{2c_i \ln 2} 2^{2b_i}, \ i = 1,...,M. \qquad (7)$$

Equating the left hand side of (7) for *i* = 1, one gets:

$$\frac{3w_1}{2c_1 \ln 2} 2^{2b_1} = \frac{3w_i}{2c_i \ln 2} 2^{2b_i} \Leftrightarrow \left(\frac{c_i}{c_1}\right)\left(\frac{w_1}{w_i}\right) = 2^{2(b_i - b_1)},$$

and after slight manipulation, it gives:

$$b_i = b_1 + \frac{1}{2}\log\left(\frac{w_1 c_i}{w_i c_1}\right), \ i = 1,...,M. \qquad (8)$$

Substituting (8) into the identity $\frac{1}{3}\sum_{i=1}^{M} c_i 2^{-2b_i} = P_{spec}$ enables us to solve for the optimal value of $b_1$ as follow:

$$P_{spec} = \frac{1}{3}\sum_{i=1}^{M} c_i 2^{-2b_1} \frac{w_i c_1}{w_1 c_i} = \frac{c_1 \cdot 2^{-2b_1}}{3w_1}\sum_{i=1}^{M} w_i,$$

$$\text{and } b_1^{opt} = \frac{1}{2}\log_2\left(\frac{c_1}{3w_1 P_{spec}}\sum_{k=1}^{M} w_k\right). \qquad (9)$$

Finally, we have:

$$b_i^{opt} = \frac{1}{2}\log\left(\frac{c_{1,M}}{3w_1 P_{spec}}\sum_{k=1}^{M} w_k\right) + \frac{1}{2}\log_2\left(\frac{w_1 c_i}{w_i c_1}\right), \ i = 1,...,M,$$

and after slight manipulation the desire result:

$$b_i^{opt} = \frac{1}{2}\log_2\left(\frac{c_i}{3w_i P_{spec}}\sum_{k=1}^{M} w_k\right), \ i = 1,...,M. \qquad (10)$$

Alternatively, we can minimize $\sigma_e^2$ subject to a given bit budget: $\sum_{i=1}^{M} w_i b_i = B_{spec}$. The design problem becomes:

$$\min_{\boldsymbol{b}} \frac{1}{3}\sum_{i=1}^{M} c_i 2^{-2b_i} \quad \text{subject to } \sum_{i=1}^{M} w_i b_i = B_{spec} . \qquad (11)$$

Using again the Lagrange multiplier method, the optimal solution of $b_i$ is found to be:

$$b_i^{opt} = \frac{1}{2}\log_2\left(\frac{c_i}{w_i}\right) + \frac{B_{spec} - \frac{1}{2}\sum_{k=1}^{M} w_k \log_2\left(\frac{c_k}{w_k}\right)}{\sum_{k=1}^{M} w_k} , \qquad (12)$$

$i = 1,...,M$ .

A possible problem with the analytical formula above for the wordlength is that $b_i$ are real-valued. To obtain an integer solution, they need to be rounded to the next largest integers. Moreover, for extremely low bit budget or large target variance, $b_i$ can even become negative. On the other hand, for high bit budget or small target variance, the problem is less serious and the solution so obtained is more accurate. It is interesting to note that the above analysis is similar to a classical problem in signal compression, known as bit allocation problem. There are in general two different approaches to solve this problem, namely the discrete Lagrange multiplier method [6] and the Marginal Analysis method [7], [8]. Next, we shall extend the latter to solve the wordlength determination problem.

### B. — Bit Allocation Algorithm

The first problem we address below is to minimize $\boldsymbol{w}^T\boldsymbol{b}$ subject to a given noise power $P_{spec}$. The variable $b_i$ is first initialized to zero. Then the algorithm allocates one bit to one of $b_i$'s until the target noise power is met. In each step, the one with the largest reduction in output noise power is selected and its wordlength will be increased by one bit. More precisely, the pseudo code of the algorithm can be summarized as follows:

```
$b_i = 0$ ; (or $b_i = floor(b_i^{opt})$ ; // from (10))
while ( $\frac{1}{3}\sum_{i=1}^{M} c_i 2^{-2b_i} > P_{spec}$ ) {
    compute $k = \arg\max_i \xi_i$
    where $\xi_i = \left.\frac{\Delta D}{\Delta B}\right|_i = \left|\frac{c_i(2^{-2(b_i+1)} - 2^{-2b_i})}{w_i}\right|$ ;
    $b_k \leftarrow b_k + 1$ ;}
```

Table 1: Minimization of $\boldsymbol{w}^T\boldsymbol{b}$ subject to prescribed noise power $P_{spec}$ .

Note that $b_i$ 's are both non-negative and integer-valued. A similar algorithm for minimizing $\sigma_e^2$ subject to a given bit budget $B_{spec}$ can be derived as follows:

```
$b_i = 0$ ; (or $b_i = floor(b_i^{opt})$ ; // from (12))
while ( $\sum_{i=1}^{M} w_i b_i < B_{spec}$ ) {
    compute $k = \arg\max_i \xi_i$
    where $\xi_i = \left.\frac{\Delta D}{\Delta B}\right|_i = \left|\frac{c_i(2^{-2(b_i+1)} - 2^{-2b_i})}{w_i}\right|$ ;
  $b_k \leftarrow b_k + 1$ ;}
```

Table 2: Minimization of $\sigma_e^2$ subject to prescribed bit budget $B_{spec}$ .

Again, $b_i$ are both non-negative and integer-valued. For multiplier-less realization using SOPOT coefficients, one can easily compute the wordlength required to achieve a given output error variance. Once it is determined, the exact rounding operation at each node can be determined and hence the complexity of the adders and registers can be determined exactly. The overflow prevention can also be determined according to section II-B if the maximum input format is known. Finally, the algorithms described in section III-A and III-B can be combined to shorten the search time, as we shall illustrate in next section.

## IV. Design Example

In this example, all wordlength determination algorithms, including the random search algorithm proposed in [3], are implemented using Matlab Ver. 6.0 in a Pentium 4 personal computer. For comparison purposes, the hardware implementation issue of a digital intermediate frequency (IF) receiver for software radios having the same specification of the example in [13] was considered. Figure 2 shows the corresponding IF architecture which consists of a compensated cascaded integrator and comb (CIC) filter, a multistage decimator, a Farrow-based fractional delay digital filter (FDDF) and a halfband filter (HBF). In the compensated CIC filter, a second-order CIC compensator is used to compensate the passband droop of the conventional CIC filter. The Farrow-based FDDF is used as an arbitrary sample rate changer (SRC) and it converts the sampling rate of the incoming signal to that required by the base-band processor. The programmable FIR filter in the conventional IF architecture can thus be replaced by the HBF with fixed coefficients, which is placed immediately after the FDDF. The overall downsampling ratio $M^*$ supported by this structure is:

$$2 \le M^* \le M_{CIC} \cdot 2^k \cdot M_I ,$$

where $M_{CIC}$ is the downsampling ratio of the compensated CIC filter, $M_I$ is the rational downsampling ratio of the FDDF, and *k* is the number of the remaining decimators to be selected. Since the coefficients of all these filters are fixed, they can be implemented without any multiplications using SOPOT representations. Furthermore, by implementing the filters using multiplier-block (MB) [9], significant savings in hardware resources can be achieved. Once the SOPOT coefficients are determined, the transfer functions of these filters are known. The internal wordlength are then minimized subject to the prescribed 16-bit accuracy using the random search algorithm. The results so obtained are summarized as follows: $\boldsymbol{w}^T\boldsymbol{b} = 4196$ ; $\sigma_e^2 = 2.021\times10^{-10}$ (i.e. 16.157 bit accuracy) and the computational time is about 20 minutes. Interested reader can refer to [3] and [10] for more details regarding the design aspects of the digital IF, such as the determination of SOPOT coefficients and the optimization of the internal wordlengths subject to the prescribed output accuracy using the random search algorithm. Next we shall employ our proposed wordlength determination algorithms to solve the same problem.

With the same specification in [3], there are totally *M* = 236 rounding sources in the receiver in figure 2. Using (10), the optimal wordlength format for each intermediate signal is obtained. The optimal value of $\boldsymbol{w}^T\boldsymbol{b}$ is found to be 4145.2. As mentioned earlier, the entries of the vector $\boldsymbol{b}$ are not integer-valued. Therefore, for practical implementation, they are rounded to the closest integer just larger than them such that the 16-bit accuracy is still met. The corresponding value of $\boldsymbol{w}^T\boldsymbol{b}$ becomes 4276 and the total noise power $\sigma_e^2$ is decreased to

$1.212\times10^{-10}$ (or 16.527 bit accuracy). For the bit allocation method, we obtain the following results in table 3: $\boldsymbol{w}^T\boldsymbol{b} = 4171$; $\sigma_e^2 = 2.5034\times10^{-10}$ (16.002 bit accuracy) and the computation time is within one minute. This method gives the best solution among the three algorithms studied with a much lower computational time than the random search algorithm. This suggests that the proposed bit allocation algorithm is well-suited even for a large scale system. It should be noted that the computational time of the proposed algorithm can be further reduced to a few seconds if the solution obtained from (10) is used as an initial guess. In order to avoid overflow, the worst-case integer bit format of each intermediate signal can then be calculated as described in section II-B, assuming that the input signal $x[n]$ to the compensated CIC filter has a format of <1/13>, i.e. 14-bits with $x_{\max} = 0.99988$. The final output is found to have a wordlength format of <9/19>. The wordlength formats of each filter output format are also shown in figure 2. Table 3 summarizes the design result in this example.

## V. Conclusion

Two novel algorithms for the wordlength determination of linear time invariant systems subject to a prescribed output accuracy are presented. The first one is able to determine a closed-form analytic solution of this problem using Lagrange multiplier method, assuming that the wordlength is a real-valued quantity. The second one is based on the Marginal Analysis method and gives integer-valued solution. Using the software radio receiver as an example, design results show that proposed approaches offer better results and a lower design complexity than conventional methods.

## References

[1] A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing*, Englewood Cliffs, NJ: Prentice-Hall, 1989.

[2] Y. C. Lim and S. R. Parker, “FIR filter design over a discrete power-of-two coefficient space,” *IEEE Trans. ASSP-31*, pp. 583-591, April 1983.

[3] S. C. Chan and K. S. Yeung, “On the design and multiplier-less realization of digital IF for software radio receivers with prescribed output accuracy,” in *Proc. DSP2002*, vol. 1, pp. 277-280, July 2002.

[4] R. Fletcher, “Practical Methods of Optimization,”*, 2nd Edition, Chichester: John Wiley & Sons*, 1987.

[5] P. D. Fiore and Li Lee, “Closed-form and real-time wordlength adaptation,” in *Proc. ICASSP’1999*, vol. 5, pp. 1897-1990, 1999.

[6] A. Segal, “Bit allocation and encoding of vector resource,” *IEEE Tran. Info. Theory*, vol. 22, pp. 162-169, Mar. 1976.

[7] B. Fox, “Discrete optimization via marginal analysis,” *Management Science*, vol. 13, pp. 210-216, Nov. 1966.

[8] S. W. Wu and A. Gersho, “Rate-constrained picture-adaptive quantization for JPEG baseline coders,” in *Proc. ICASSP’1993*, vol. 5, pp. 390-392, 1993.

[9] A. G. Dempster and M. D. Macleod, “Use of minimum-adder multiplier blocks in FIR digital filters,” *IEEE Trans. CAS. II*, vol. 42, pp. 569-577, Sept. 1995.

[10] K. S. Yeung and S. C. Chan, “On the design and multiplier-less realization of digital IF for software radio receivers,” in *Proc. EUSIPCO’2002*, vol.1, pp. 695-698, Sept. 2002.

|  | $\boldsymbol{w}^T\boldsymbol{b}$ | **Output noise power** | **Output bit accuracy** |
|---|---|---|---|
| Random search algorithm [3] | 4196 | $2.021\times10^{-10}$ | 16.157 |
| Analytic solution | 4145.2 | $2.512\times10^{-10}$ | 16 |
| Rounded analytic solution | 4276 | $1.212\times10^{-10}$ | 16.527 |
| Bit allocation algorithm | 4171 | $2.503\times10^{-10}$ | 16.002 |

Table 3. Summary of design results.

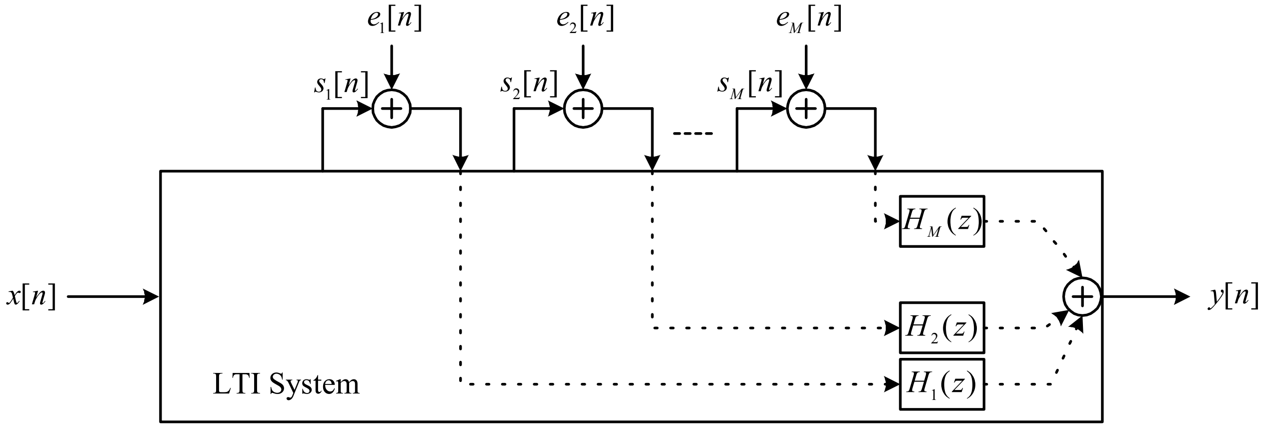

Figure 1: Round-off noise model of linear time invariant system.

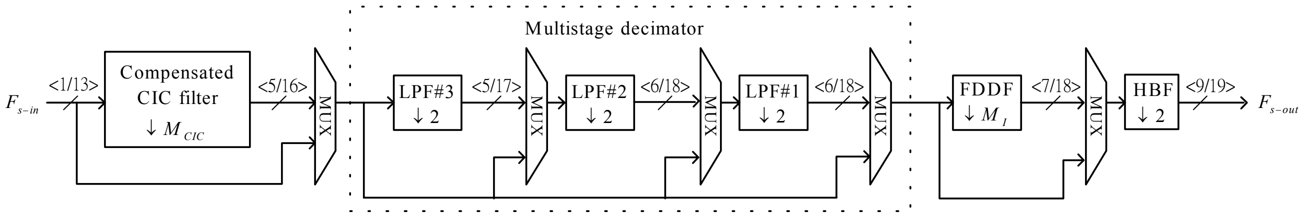

Figure 2: Digital IF architecture for software radio receiver in [3] and [10].