A TRELLIS-BASED ALGORITHM FOR THE REALIZATION OF FIR FILTERS FOR MULTI-MODE RECEIVERS

Sang Hyun Lee, Kwang Soon Kim, and Kyung Hi Chang

Electronics and Telecommunications Research Institute (ETRI)
161 Gajeong Dong, Yuseong Gu, Daejeon 305-350, Korea

ABSTRACT

A trellis-based algorithm is proposed for the realization of finite impulse response digital filters for multi-mode receivers. With this algorithm, the resource-sharing configuration of filter coefficients minimizes overall costs of the multi-mode receiver. In an example design, this algorithm can provide simpler realization with reduced costs.

1 INTRODUCTION

Software-defined radio (SDR) concept has come to stand for a software-reconfigurable radio that derives its flexibility from programmable RF, IF and baseband signal processing stages [1]. Digital signal processing (DSP) affords greater flexibility and higher performance than traditional analog technology. This makes DSP extend from baseband to IF processing stages. The digital IF processing stage mainly processes signal up/down conversion. A digital up/down converter typically consists of a digital local oscillator, a digital mixer and a digital filter. A digital filter in digital IF processing stage is used for decimation, reduction of the out-of-band noise and also baseband signal shaping. As for implementation of digital filter, due to its stability and phase linearity, the FIR filter is preferred [2]. The phase linearity makes easy the compensation for signal distortion experienced through the communication channel. With this property, the FIR filter has been considered attractive for communication applications. Multi-mode receiver for a SDR should accommodate all FIR filters necessary for different communication modes. However, straightforward integration of all hardware resources is not a preferred choice with respect to the cost and complexity of hardware. Therefore, we present the realization structure of FIR filters with the reduced cost and complexity of hardware and the algorithm transforming all filter structure to the proposed realization structure.

2 THE PROPOSED STRUCTURE OF THE MULTI-MODE FILTER

2.1 Analysis Model

The coefficients of FIR filter for each mode are designed to replace multipliers as sums of signed-powers-of-two terms using MILP [3]. The designed filters have different lengths depending on their desired frequency responses. Also, they cost different amounts of hardware resource and it is therefore found that the hardware resource for the longest-length filter is the essential cost for implementation of multi-mode filter. The practical realization strategy for reducing total cost of multi-mode filter is to extract any repeating parts of hardware resource among the shorter-length filters' coefficients and to merge them into the longest-length filter's coefficients. The subexpression sharing technique reported in [4] was able to achieve hardware reduction under this rule. We apply this technique for filter coefficients of all filters along with an additional constraint, for layout regularity and routability, that the order of calculation for each tap of shorter-length filters should not be changed. Figure 1 illustrates an example of filter with two modes realized based on this strategy.

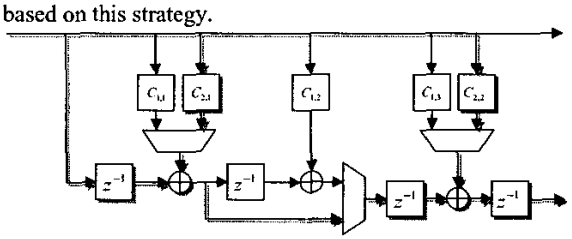

Figure 1. An example of realization for the proposed structure.

The shorter-length filter uses only shaded parts of the hardware resource in the Figure 1. Taps of the shorter-length filter is allocated to those of the longer-length filter so as to maximize the shared parts of resource. The price to be paid is the multiplexers for selection of the taps for each mode and routing those taps. However, since, in general, an adder costs about 2 to 4 times more hardware resource than a multiplexer, the resource sharing achieves reduction of hardware resource even if the tap shares only one addition. The proposed strategy for realization of the filters for multi-modes is summarized as follows.

- All filters are designed such that they have their coefficients as sums of signed-power-of-two terms.
- All filters are implemented only with adders and shifters. Therefore, by sharing the repeated patterns of filter coefficients, the total cost of filters is reduced.
- Taps of the shorter-length filter are allocated to those of the longer-length one, with its processing order

0-7803-7946-2/03/$17.00 ©2003 IEEE

fixed, so as to maximize the shared hardware resources of both filters.

- The allocated taps are routed and connected with multiplexers for filters of different modes.

By the constraint imposed on the realization structure that the order of taps for the shorter-length filter should not be changed, $\binom{N_{longer}}{N_{shorter}}$-tap combinations are possible where N_{longer} and $N_{shorter}$ denote the lengths of the longer-length and shorter-length filter, respectively. However, if the difference between the lengths of both filters is more than 10, the number of the possible combinations becomes tremendously many. Thus, it is practically not possible to find the optimal combinations of taps by the exhaustive search. Fortunately, we find that this problem is not in NP, that is, there exists an algorithm with polynomial complexity, and the practical but optimal allocation algorithm is presented in the next section.

3 ALGORITHM FOR THE COEFFICIENTS ALLOCATION

The process of allocating filter taps under the order constraint can be viewed as a linear resource allocation problem with longer-length filter's taps as its resources and savings in hardware by sharing as its objective function. Dynamic programming (DP) has been used effectively to solve this resource allocation problem [5]. In our algorithm, we exploit the "bottom-up" feature of DP to successively allocate the shorter-length filter's taps to the longer-length one's until all taps are allocated.

To facilitate the discussion on the algorithm, we present a trellis by which search is performed, as depicted in Figure 2. By N and a_i we denote the length and i-th filter tap of the shorter-length filter, while by L and b_i we denote the length and i-th filter tap of the longer-length filter, respectively. Each column of the trellis means each tap index of the shorter-length filter. The states in the i-th column of the trellis mean longer-length filter's taps to which the i-th tap of shorter-length filter can be allocated.

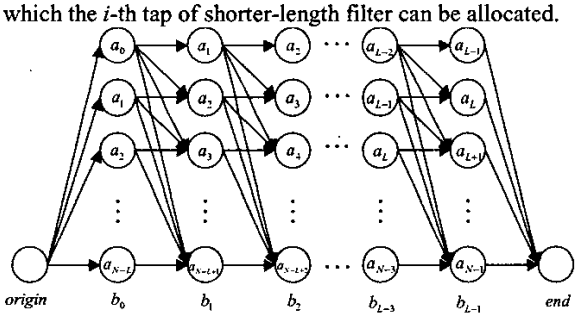

Figure 2. A trellis for DP-like algorithm.

For example, assume that we need to allocate taps of filter with length 3 to those of length 5. The first tap can be allocated only to the first to third taps, otherwise one of the remaining taps cannot be allocated. Thus, the possible range of tap indices to which the first taps is allocated is from 0 to N-L+i and, at the i-th column of the trellis, the range of tap indices is from i to N-L+i.

Now that the states in each column are defined, we define the branches emanating from each state. Unlike usual trellis for DP, due to the order constraint the number of branches emanating from each state varies with the row location of that state. The i-th state from the top in each column has exactly i branches. For example, if the second tap of shorter-length filter is allocated to the third tap of longer-length filter, the first tap of shorter-length filter can be allocated only to the first to second taps due to the order constraint.

The last to be defined is the cost to be paid when taking branches in the trellis. The cost functions are so associated to all branches as depicted in Figure 3.

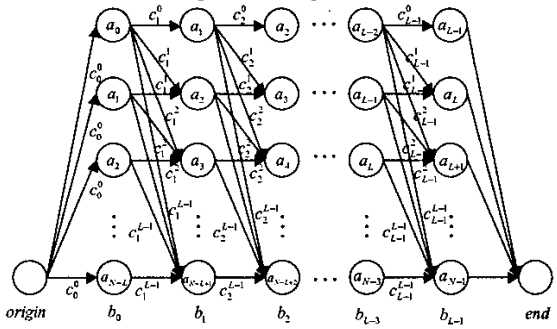

Figure 3. A trellis structure defined cost functions for search algorithm.

Let c_i^j denote the cost function of all branches merging into the j-th state from the top at the i-th column. Since the cost functions of branches merging into a given state are identical, we easily determine the best branch among the merging branches by observing the overall path cost up to the current step. So the complexity of this algorithm is slightly less than that of the conventional DP. The cost associated to each branch is defined as the amount of shared hardware resource. A savings in hardware resource by sharing is the difference between the number of total hardware building blocks necessary for straightforward implementation and that of the hardware building blocks used for merged tap coefficients.

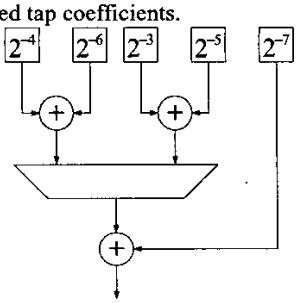

Figure 4. An example of filter coefficients realized by using resource sharing.

Calculation of savings in the hardware resource consists of two stages of sharing process. In the first stage, we find the common pattern of hardware resource, extracting that part and transforming the remaining part. The remaining

parts are connected to the common part through multiplexer. Multiplexer is used to select the partial coefficients used by only one side of taps according to operation modes. If the coefficients of both filters are identical, multiplexer is not necessary. It is easily found that sharing n shifts saves n additions.

Suppose that we need to merge the coefficients of $2^{-7}+2^{-4}+2^{-3}$ and $2^{-7}+2^{-6}+2^{-5}$. Figure 4 shows an example of merged taps with a shift 2^{-7} shared. One shared shift saves one addition. In the second stage of the calculation, the subexpression sharing is applied to filter taps obtained by the preceded common part sharing. In the subexpression sharing, we consider two shifts and one addition as a group and transform the tap in order that this group is shared by both filters.

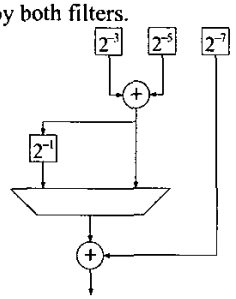

Figure 5. An example of filter coefficients realized by proposed structure using subexpression sharing.

Figure 5 illustrates the result of the subexpression sharing applied for the merged tap in the Figure 4. With two stage's sharing process, two additions and two shifts are saved. The more precision of filter coefficients is used, the more savings is obtained by resource sharing. Using these techniques, we calculate the shared resource associated to each branch in the trellis. The final costs applied for branches are defined as the weighted sums of the number of additions and shifts. The weights highly depend on implementation methods of hardware building blocks.

Based on the trellis defined with such cost functions, the dynamic-programming-like optimization is done, as depicted in Figure 6. Figure 6 shows a specific example of filter tap allocation. The result of tap allocation is illustrated in Figure 6a and searching process such that the result in Figure 6a is achieved is shown in Figure 6b. For each state in a given column, we determine the best branch among all possible branches from itself to the other states in the most recent previously considered column, that is, the column to the left.

Once searching process arrives at the end of the trellis, we get the "survivor" path associated to the optimal combination of filter taps achieving maximal resource sharing. The branch in the survivor path corresponds to a longer-length filter's tap to which a shorter-length filter's tap is allocated, with the state set to "survived" and the remaining states being determined by the path to which the branch connects that state. Therefore, the sequence of states obtained by the search directly gives the sequence of longer-length filter's taps to which shorter-length filter's are allocated.

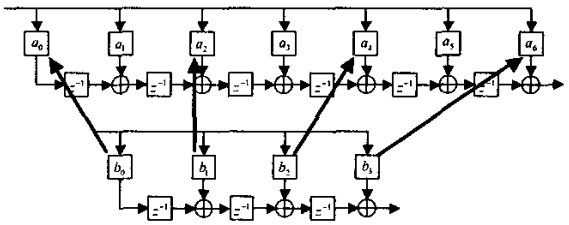

Figure 6a. An example of filter coefficient allocation.

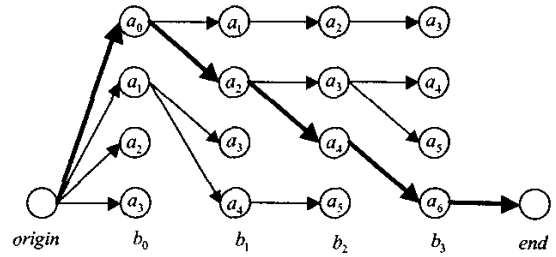

Figure 6b. Searching process representing the above filter coefficient allocation.

The algorithm can be readily extended to filter with an arbitrary many number of modes. The algorithm starts by applying trellis search for the longest-length filter and the second longest-length filter. By considering this solution as "new" longest-length filer, it repeatedly applies this new filter for the next longest-length filter of decreasing size until it arrives at the filter with taps into which all filters' tap is merged. In this case, the solution is a suboptimal solution while the solution of algorithm is guaranteed to be the optimal solution in the case with two modes.

- Construct a trellis with the number of columns being the same as the length of sorter-length filter, as illustrated in Figure 4.
- For each state, determine the cost function by calculating the difference between the resource of straightforward implementation and the resource of merged implementation realized by the resource sharing.
- Perform the trellis search.
- Obtain a sequence of filter taps from the survivor path and configure the implementation structure corresponding to the sequence.

4 EXAMPLE

As an example, we applied the proposed trellis-search algorithm for the realization structure of digital transmit-and-receive filters of 3GPP and 3GPP2. The desired frequency responses of the filters are specified in [6] and [7], respectively. Digital filter for 3GPP is the truncated root-raised-cosine filter with roll-off factor of 0.22 and the desired frequency response of digital filter for 3GPP2 is identical to that of IS-95 CDMA digital cellular standard. Both filters were designed by the MILP algorithm with the normalized peak ripple of –50 dB [3]. The coefficient wordlength was 10 bits. Designed filters for 3GPP and

3GPP2 are the 63 and 81-tap linear phase FIR filters, respectively.

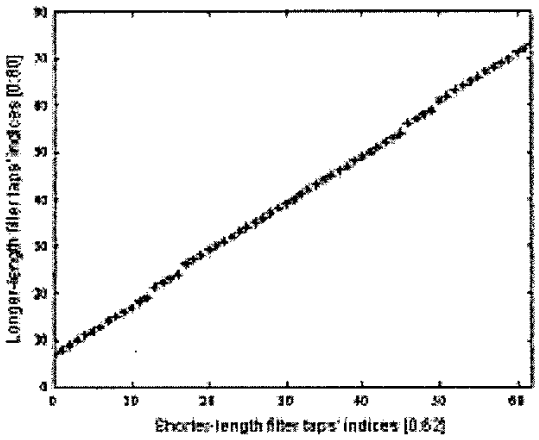

Figure 7. Result of filter coefficient allocation.

In this case, we have 4.567×10^{17} possible combinations of filter taps allocation. Each tap of shorter-length filter has 19 possible combinations and the number of columns in the trellis is 63. Therefore, we constructed the trellis with 63 columns and 19 states for each column. By the trellis search, we obtained the result shown in Figure 8. Figure 7 shows the mapping of shorter-length filter's tap indices onto longer-length one's tap indices. Note that, in the case of linear phase filter, the mapping is symmetric about the center tap of shorter-length filter with high probability. Table I compares the complexity of the filter structure designed by the trellis search method to that of the straightforward implementation. The trellis search algorithm realizes a savings in additions of 40% and delays of 43%. The gate counts are estimated based on the design rule of STD90/STDM90 (0.35um, 3.3V) ASIC fabrication process. The proposed method achieves a savings in gates of 40%.

Table 1. Comparison of total costs.

	Additions	Shifts	MUX's	Delays	Estim. gates
3GPP2 (L=63)	130	131	-	62	20691
3GPP (L=81)	143	138		80	23698
Sum	273	269	-	142	44389
Saved	108	66	-	62	18076
Final	165	203	60	80	28203

5 CONCLUSION

We have presented an efficient trellis search algorithm for minimal cost realization of FIR filters with coefficient implemented as sums of signed-powers-of-two terms for the multi-mode receiver. The proposed algorithm allocates one filter's taps to the other's taps such that the savings in the hardware resource is maximized by exploiting redundancy within the coefficients. An example of realization of FIR transmit-and-receive filters with sums of signed-power-of-two coefficients has shown that the proposed algorithm reduces the required hardware resources.

6 REFERENCES

[1] J. Mitola, "Software radio," *IEEE Communications Magazine*, vol. 33, May 1995.

[2] D. B. Chester, "Digital IF filter technology for 3G systems: An introduction," *IEEE Communications Magazine*, vol. 37, Feb. 1999.

[3] Y. C. Lim and S. R. Parker, "Design of discrete-coefficient-value linear phase FIR filters with optimum normalized peak ripple magnitude," *IEEE Transactions on Circuits and Systems*, vol. 37, Dec. 1990.

[4] R. I. Hartley, "Subexpression sharing in filters using canonical signed digit multipliers," *IEEE Transactions on Circuits and Systems – II : Analog and digital signal processing*, vol. 43, Oct. 1996.

[5] D. P. Bertsekas, Dynamic programming : deterministic and stochastic models, Prentice Hall, Sept. 1987.

[6] TIA/EIA Interim standard, Physical layer standard for cdma2000 spread spectrum systems, TIA/EIA/IS-2000-2, Telecommunications Industry Association, Aug. 1999.

[7] 3G TS 25.104: "UTRA (BS) FDD; Radio transmission and Reception," 3GPP Organizational Partners, Mar. 2000.