FIR Filter for Adaptive Equalization in PRML Read Channels

Nikola Nedovic and Vojin G. Oklobdzija
Advanced Computer Systems Engineering Laboratory
Department of Electrical and Computer Engineering
University of California
Davis, CA 95616

and

Michael Leung
Bay Area Design Center
Silicon Systems Inc.
2460 N. First St.
San Jose, CA 95131

ABSTRACT

A novel design of FIR filter for adaptive equalization in PRML read channels is presented. Two aspects of a design are considered: architecture level and implementation. The realization of this FIR structure introduces new architectural solutions. Those solutions are exploiting beneficial properties of both conventional FIR architectures, which allows for low computation latency at high operating frequencies, and ease of trading between parallel and pipelined structure. The implementation of this FIR structure uses the channel noise property knowledge to determine acceptable level of truncation/rounding and applies optimal automated special-purpose multiplication method to further minimize overall latency of the filter.

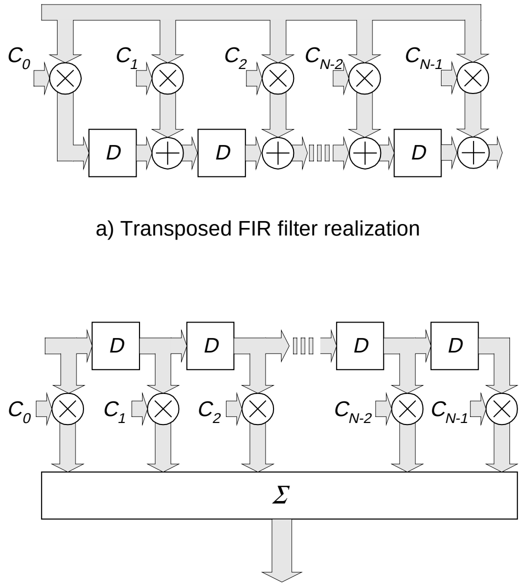
Keywords: Adaptive equalization, LMS, PRML, FIR filter, filter architecture, pipeline

1. INTRODUCTION

Partial Response signaling with Maximum Likelihood detection (PRML) is widely used in read channel applications. Achieving superior performance at higher levels of Intersymbol Interference (ISI) over previously dominant Peak Detection, it is prevailing technique used today. Operating mechanism of PRML system is based on equalization of the received input stream from the read head in a manner essentially equivalent to communication channel equalization and decoding of the intermediate signal using Maximum Likelihood algorithm.

The equalization can be performed in analog (continuous-time) domain, digital domain, or some combination of the two. The most common practice is use of analog equalization for rough shaping of the signal followed by sophisticated and costlier digital adaptive equalizer. The latter is usually performed by utilization of some form of Least Mean Square (LMS) algorithm [1] that provides coefficients to Finite Impulse Response (FIR) filter.

The main design objective of adaptive equalizer is to achieve as low computational latency as possible while maintaining acceptable distortion level. This is important since the equalizer is one of most time-consuming components in Clock Recovery Loop whose effectiveness is highly dependent on total loop latency. Recently, with rapid increase of operating frequency of read channels,



a) Transposed FIR filter realization

b) Direct FIR filter realization

Figure 1. FIR Filter Realizations

the speed of computation becomes important factor as well.

Optimal FIR filter configuration and architecture solution therefore plays important role in performance of PRML system. Today, FIR filters are realized in one of two popular forms: direct and transposed (Fig. 1). Though both of them have some advantageous properties, neither is suitable for this application. Direct realization is potentially better for high-frequency operation, but suffers from high latency compared to the transposed one. In the transposed realization, the input signal has to be multiplied and added to the accumulated value in a single pipeline stage, limiting the clock frequency. Also, this realization has a disadvantage of imposing the additional pressure to the implementation by high fan-out requirement on the input value.

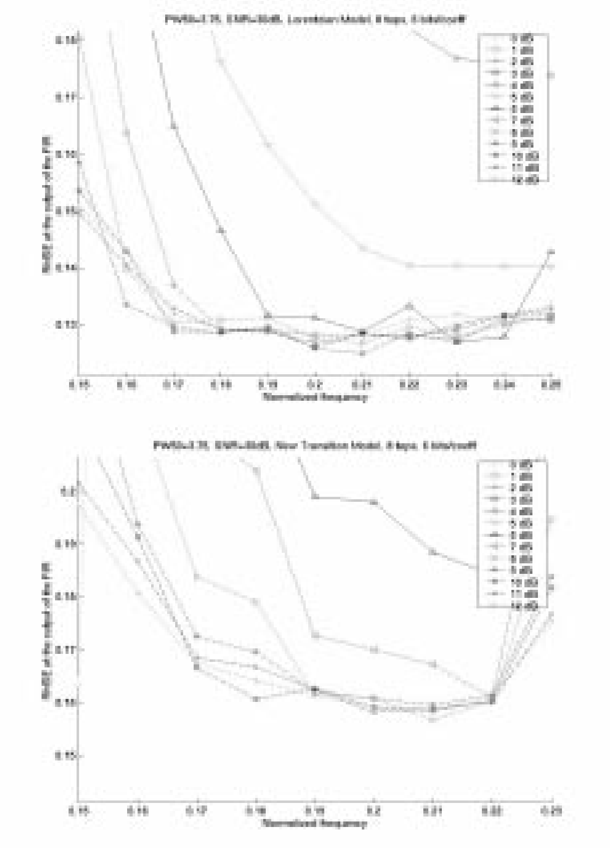


Figure 2. Comparison of CT filters for PW50=3.75

The paper is organized as follows: Section 2 describes the environment used for simulation, Section 3 covers the topic of optimal FIR filter configuration, Section 4 describes architecture of the FIR filter, in Section 5 discussion on arithmetic issues is given and Section 6 concludes the paper.

2. SIMULATION

In order to determine the characteristics of the equalizer and to obtain optimal configuration, a number of simulation conditions have been defined:

1. The models of input signal (signal from the read head) that are dealt with are standard Lorentzian and an Texas Instruments proprietary model, with PW50 parameter (PW50 determines the relative width of the step response with respect to the sampling period) up to 3.75
2. All other sources of signal distortion are approximated by the additive Gaussian noise with SNR up to 30dB
3. Continuous-Time filter (CT) is represented by the Impulse Invariant Transform (IIT) of the CT filter used in the real system
4. Oversampling of 10 is introduced in order to follow the realistic method of restoring the phase shift and to overcome limitation of IIT related to filters with non-negligent characteristics in higher frequencies region
5. Saturation of the coefficients and rounding of each signal are applied. The aim is to obtain results that are as accurate representation of the real system as possible

Since only EPR4 signaling is used in the simulations and in this particular configuration, is not guaranteed that the chosen configuration is optimal if other signaling is used. However, this method is not restricted to this particular signaling type and can be easily applied to any other system.

Table 1. *CT filter Set Chosen for Simulations*

	Lorentzian Model			New Transition Model		
	Boost	$f_{C,min}$	$f_{C,max}$	Boost	$f_{C,min}$	$f_{C,max}$
PW50=2.5	2	0.21	0.23	2	0.21	0.23
	3	0.19	0.225	3	0.19	0.23
	4	0.17	0.22	4	0.19	0.22
	5	0.175	0.22	5	0.18	0.21
PW50=3.75	6	0.22	0.24	8	0.2	0.22
	7	0.215	0.23	9	0.19	0.22
	8	0.185	0.24	10	0.18	0.225
	9	0.185	0.225	11	0.18	0.225
	10	0.185	0.21	12	0.19	0.22
	11	0.19	0.205			

Since the CT filter that serves as pre-equalizer is the component with the great influence on output RMSE, it is important to determine its parameters. This is done with the initial assumption that FIR filter used for evaluation of CTF (8 tap, 6 bits/coefficient) will result in output

RMSE similar to the one obtained with supposed optimal FIR, which is verified in later phase of the design. Results are shown in Figure 2. Although some fluctuations that are consequence of phase error change due to relatively small over-sampling rate are present, general characteristics can be observed. The choice of the phase (which down-sampled waveform to choose) is made upon ‘static’ (zero-forcing) performance.

Set of CT filters for PW50 parameters of 2.5 and 3.75 that results in best performance is given in Table 1. $f_{C,min}$ and $f_{C,max}$ represent cut-off frequency normalized to the sampling frequency.

3. FIR FILTER CONFIGURATION

Next step is to determine FIR filter configuration based on the performance of the system that uses set of ‘optimal’ CT filters. Some of representative results are shown in Figure 3. It can be noticed that the minimum can be identified at generally 8 taps and no less than 6 bits per coefficient. Some simulations performed prior to this work resulted in the same optimal FIR filter configuration, but with the flat tail of the characteristics, which claims that the same performance would be obtained with any higher number of taps. It is shown that this is not true in the real system where finite precision is inevitable and noise is present. The slope can be explained by the adaptation fluctuation on the side taps dominated by the noise, which is not correlated with the input signal. The more side taps (taps with insignificant value), the bigger the noise introduced by their wondering around optimal value (which is close to zero), and therefore the worse the output RMSE. Increasing the number of taps beyond the optimal point is not only useless, but leads to performance degradation. This behavior is noticed for all realistic input conditions and can be considered general.

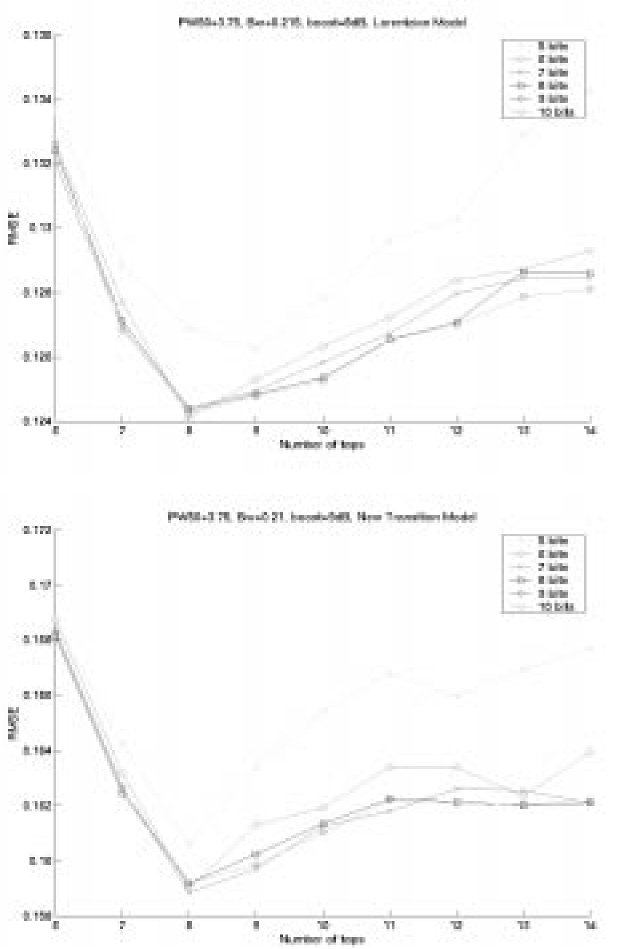


Figure 3. FIR filter performances for PW50=3.75

4. FIR ARCHITECTURE

As mentioned earlier, FIR filters today utilize either direct [2] or transposed realization [3,4]. It is also stated that neither of them is suitable for high-performance adaptive equalizer application: direct realization because of the large computation latency and transposed one because of its inherent limit on clock period and high fanout imposed on input signal. Most of PRML adaptive equalizer architectures today choose to use transposed realization since it results in lower total computation latency, which is most critical requirement of the application, leaving the problems it imposes to be dealt with by circuit design.

The set of new architectures is proposed. It is based on combination between direct and transposed realization. The care is taken that fanout of the inputs is less than some reasonable value. On the other side, significant reduction in computation latency is achieved comparing to direct realization. This is accomplished by treating the computation (multiply-add-based FIR equation) as a regular multiplication with common partial product generation and inserting pipeline registers in multiplier tree in the convenient paths so that they also serve as functional delay elements (taps of the filter). Some amount of functional registered delay is still performed by traditional direct realization-like taps in order to

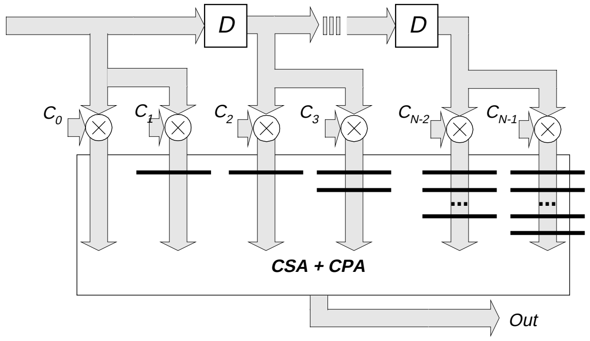


Figure 4. Proposed FIR Filter Architecture

decrease fan-out imposed on input. This way similar fan-out property as in direct realization and similar computation complexity in the last stage (which determines latency) as in transposed realization are obtained.

The principal scheme of proposed full-rate FIR filter is shown in Figure 4. Block marked with X can represent either Booth multiplexors or regular partial product generators, depending on the particular multiplier realization. Black lines over signal paths represent the pipeline registers for each particular path in CSA. The computation is done between two pipeline registers such that only those signals in the same stage are processed together. Initial assignments of the signals to the pipeline stages are done so that overall registered delay from the partial product generator c_i to the output equals the number of sample periods required by the FIR filter function (i). The combination of number of pipeline stages and amount of input delay achieves the balance between computational complexity and fan-out on input signals.

Each sample of the output is sum of N sets of partial products generated by N previous input samples and corresponding coefficients. The partial product generation on input sample in_{n-i} and coefficient c_i is performed in cycle *n-ceil(i/k)*, where k is order of parallelism for each tap output (number of multipliers driven by each tap; k=2 for the example on Figure 5) and *ceil* stands for ceiling function. In order to obtain correct output, the signals generated in that path have to be delayed for *ceil(i/k)* more cycles, which is accomplished by inserting *ceil(i/k)* pipeline registers in the multiply-add structure. These registers are used to split the total path delay to several equal parts (more detailed explained in section 5.2) and to achieve less constraint on the clock period. The latency of the filter is determined by that of the first path (in_n*c_0) and the level of compression of remaining paths in previous cycles.

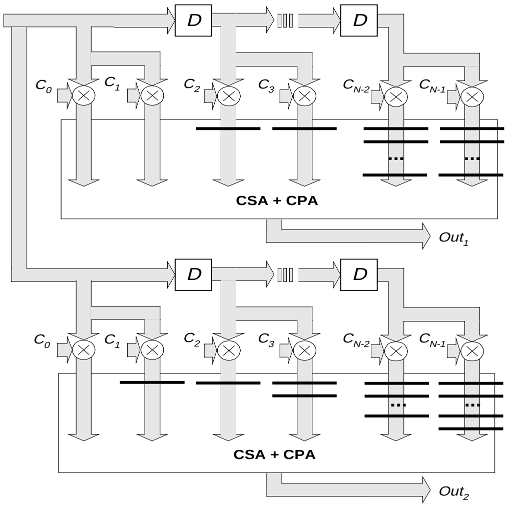


Figure 5. Half-Rate Filter Architecture

This approach can be further generalized by using different or non-uniform number of partial product generators (non-uniform fanout) to change particular path's pipeline depth and thus affect the total latency, fanouts in the structure and minimal clock period. The only constraint from functional point is that total of registers in path that multiplies c_i – tap delay line plus pipeline registers inside the multiply-add circuitry – must be equal to i.

This architecture does not impose any additional constraint to parallelization. The care should be taken, however, that number of pipeline registers per path is different than in full-rate structure. Example of half-rate realization is given in Figure 5.

5. ARITHMETIC ISSUES

Truncation Effects

In order to reduce computational complexity, a study of errors involved by removing certain number of summand bits is undertaken. The study is conducted for the architecture proposed in section 4 and for several different methods of error generation:

1. Truncation without coefficient pre-multiplying (direct partial product adder)
2. Rounding without coefficient pre-multiplying (direct partial product adder)
3. Truncation with coefficient pre-multiplying (Booth encoding)
4. Rounding with coefficient pre-multiplying (Booth encoding)
5. DC involving method [5]
6. Novel statistics-based method [6]

DC-based method replaces truncated part of the partial products with their average value, while statistics-based method compensates truncated part with the number estimated with the sum of the input bits.

These six methods are compared in terms of RMSE involved as well as maximum error obtained for some particular set of inputs. Probability of inputs to the filter is taken to be uniform (which is not quite the accurate approximation for one tap, but in average, this approximation is believed to be good). Another assumption is that numbers are represented by magnitude, which is believed to be very similar to two's complement since the truncation principle is the same and [6] shows good matching between these two multiplier types performances.

The results are shown in Figure 6. It can be noticed that although [6] claims that statistics-based method is the best one, for particular case (eight multiplications of two six-bit numbers and comparison criterion chosen - RMSE), simple truncation is the most advantageous method. Note that truncation should be compared to other methods' performance on one bit higher position since their computational complexities are similar in that case. Particularly truncation of Booth-encoded number is beneficial since some amount of lower bits' influence is already transferred to higher bits in the process of pre-multiplying. Advantage of truncation over rounding could also be argued. Indeed, results of truncations and rounding are similar in the area of interest, with the slight advantage of truncation. Computational complexity and delay are shown to be less for the truncation.

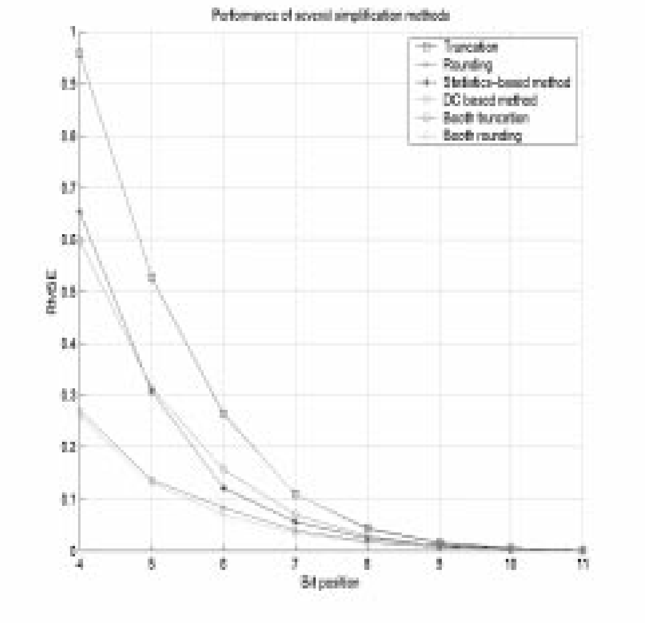


Figure 6. Introduced RMSE vs. Elimination Bit Position for Considered Simplification Methods

The bit position of truncation should now be determined. The strategy is adopted that RMSE of the truncation should not significantly affect noise and ISI RMSE value. Since the FIR filter is dealing with RMSE of order of 0.14, first bit position that meets the condition is position 8 in Figure 6, which corresponds to truncation of 4 bits of Booth output. In this case, RMSE is 0.0286, which boosts total RMSE for approximately 2%.

Multiplier realization

The multiplier realization has huge impact on total system performance. The Three Dimensional Method (TDM) [7] is used for partial products compression since it results in low computational latency and it can deal with arbitrary input arrival profile, which is used for inserting different number of pipeline registers for different taps. This approach keeps the record of delays for all intermediate signals for each multiplier column and in each step uses full adder or half adder on the earliest signals in the group of bits that should be summed. The process ends when number of bits in the column reaches two. The last step is addition of two remaining numbers to obtain final result, which can be done using some fast addition method.

The automated method for generation TDM CSA tree with inserted pipeline/tap registers is developed. It modifies TDM algorithm by intentional initial delay for each path, which is complement of the number of pipeline registers for that path, given in section 4. This step corresponds to the assigning the taps to the pipeline stages. In the process of building the CSA tree, the pipeline register is placed prior to the detection of cycle time excess for each particular pipeline stage in the path.

Original TDM algorithm is also improved by choosing the inputs that are summed in each step based on the relation between the delays from each input to the outputs of single basic elements (full adders and half adders) rather than using two or three fastest inputs ([7]). Experiments show further delay reduction of several percents compared to original TDM algorithm.

Considering the method for partial product generation, Booth algorithm of higher radices is particularly suitable in this application for several reasons. First, pre-multiplication of the FIR filter coefficients is done out of the critical path, which means that some of the computation is already done prior to the CSA tree. Second, as already mentioned, truncation effects in the multiplier are less harmful since some of the lower bits' information is transferred to higher bit positions prior to the truncation.

6. CONCLUSION

Novel design of FIR filter for adaptive equalization in PRML read channel is presented. The configuration of the filter is obtained based on extensive simulations of its environment under what is believed to be realistic assumptions on both the system and the component itself. New architectural solution that overcomes inherent disadvantages of both conventional FIR architectures in the particular application is proposed. It is also shown that further performance increase can be achieved at the expense of proper truncation in the multiplier and by use of more sophisticated and adjustable multiplication algorithm. Combination of proposed architecture and multiplication algorithm has beneficial property of high level of controllability and ease of reconfiguration, imposing no unnecessary constraints on clock frequency and achieving low computation latency. Several different architectures can be derived from this design, including easy trading between parallelism and pipelining, trade-off between depth of pipeline and maximal fan-out in the design and partial product generation method. Preliminary comparative study on performance of half-rate realization of this filter and some of representative contemporary filters speaks in favor of the new design.

7. ACKNOWLEDGEMENTS

The authors would like to thank Borivoje Nikolic of UC Berkeley and Leo Fu of Silicon Systems Inc. for their useful suggestions.

8. REFERENCES

[1] S. U. H. Qureshi, “Adaptive equalization”, Proceedings of the IEEE, Sept. 1985.

[2] C. S. H. Wong, J. C. Rudell, G. T. Uehara, P. R. Gray, “A 50 MHz eight-tap adaptive equalizer for partial-response channels”, IEEE Journal of Solid-State Circuits, March 1995.

[3] L. E. Thon, P. Sutardja, Fang-Shi Lai, G. Coleman, (Edited by Wuorinen, J.H.), “A 240 MHz 8-trap programmable FIR filter for disk-drive read channels”, 1995 IEEE International Solid-State Circuits Conference, Digest of Technical Papers, IEEE 1995.

[4] Robert Staszewski, Khurram Muhammad, Poras Balsara, “A 550Msample/s 8-tap FIR Digital Filter for Magnetic Recording Read Channels”, 2000 IEEE International Solid-State Circuits Conference, Digest of Technical Papers, IEEE 2000.

[5] S. S. Kidambi et al. “Area-efficient multipliers for digital signal processing applications”, IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, IEEE, Feb. 1996.

[6] Jer Min Jou et al. “Design of Low-Error Fixed-Width Multiliers for DSP Applications IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, IEEE, June 1999.

[7] V. G. Oklobdzija, D. Villeger, S. S. Liu, “A method for speed optimized partial product reduction and generation of fast parallel multipliers using an algorithmic approach”, IEEE Transactions on Computers, IEEE, March 1996.