A NOVEL EQUALISER ARCHITECTURE WITH DYNAMIC LENGTH OPTIMISATION

Mark P. Tennant*†, A.T. Erdogan†, T. Arslan† and J. Thompson†

†School of Engineering and Electronics, The University of Edinburgh
The King's Buildings, Edinburgh EH9 3JL
United Kingdom

*Email: M.P.Tennant@sms.ed.ac.uk

Abstract— **This paper presents a novel architecture for tap-length optimisation of the linear LMS equaliser. No analysis has previously been carried out to determine any tradeoff that exists in circuit area against power saving achieved. A low-complexity length update algorithm is employed to dynamically adjust and optimise the number of taps in the linear equaliser according to channel conditions. The results show that the chosen algorithm presents minimal overhead and reduces power consumed due to optimisation of the equaliser length. This paper presents the first complete architectural VLSI implementation of the length optimised equaliser and includes a performance study in terms of area and power.**

I. Introduction

The popularity of portable, battery-powered consumer devices is continuing the high demand for low-power, high performance electronic components. This is especially so in the marketplace for wireless communication devices such as 2.5G/3G mobile phones, wireless enabled laptops and other data enabled devices which require powerful computation and real-time signal processing capabilities. The implementation of complex systems which facilitate wireless communication requires the use of efficient and flexible cores in their design. Attention here is concentrated on the design of adaptive filtering cores that must have heavily constrained power and area requirements for these applications. Linear equalisers are typically implemented using adaptive finite impulse response (FIR) filters [1] with filter coefficients being recursively updated using either the recursive least squares (RLS) or more commonly the least mean squares (LMS) algorithm used in this study. The number of taps in the FIR structure has a critical influence on the performance and computational complexity of the equaliser. An equaliser with too many taps will be computationally inefficient and may introduce a degradation in mean squared error (MSE) performance due to limitations of the LMS algorithm whereas, an equaliser with too few taps will be unlikely to reach its true potential level of distortion mitigation. Coupled with this is the time variant nature of wireless channels which ideally necessitates the ability of the equaliser to alter its number of taps with time. The technique of varying the length of the LMS filter was first presented by the authors of [2] with an algorithm which proved that a filter with fewer taps will have a faster convergence than that of a filter with a higher number of taps. This variable length stochastic gradient (VLSG) algorithm demonstrates an LMS adaptive filter which can accomplish a change in its length from being initially low, therefore aiding fast convergence, gradually increasing over time to achieve the low steady state MSE performance characteristic of higher order filters. In [3], Won et al. went on to propose another variable length LMS (VL-LMS) algorithm using a time-constant concept whereby several filter lengths are predetermined and filter length is increased to the next predetermined value when conditions are satisfied.

Both the algorithms referred to previously however offer only the ability to *increase* the filter length over time to satisfy the contradictory goals of fast convergence and good steady state performance. As a progression from the previous methods, the authors Riera-Palou et al. of [4] specifically present a linear equaliser using an algorithm that can dynamically and automatically increase or decrease the length of the filter. Using a segmented FIR filter structure and a weight update algorithm the optimum, and in this case minimum required, number of taps are operated. Further to this, [5] presents a method whereby the optimum length of the adaptive filter is determined. In this case the number of filter coefficients of an unknown system are found using the LMS algorithm in a system identification setup. This method uses the MSE output from a number of individual LMS adaptive filters in parallel to determine the number of unknown system coefficients and their values. This does not lend itself well to a circuit implementation which is constrained in terms of area and power. Finally in [6], the authors present the most recent proposal for the variable length LMS algorithm. This algorithm uses a method whereby the filter length is varied according to the negative gradient direction of the estimation error. The familiar gradient decent method is used to track the optimum filter length with constraints included to avoid unexpected behaviour and guarantee convergence.

0-7803-9390-2/06/$20.00 ©2006 IEEE

However, none of the above methods have been evaluated in terms of the power or area overhead required in comparison to a standard fixed length adaptive filter. Simulation based results are presented to show how varying filter lengths effects convergence times. In [4] results are presented to show equaliser length changing in accordance to input E/N_0. To realistically analyse power and area requirements, in this work a variable length LMS equaliser has been implemented in VLSI, targeting 0.18 micron standard CMOS technology.

II. Implementation

An N-tap FIR filter performs the following convolution:

$$y(n) = \sum_{k=0}^{N-1} b_k x(n-k) \tag{1}$$

where b_k's are the coefficients of the filter, $x(n)$ and $y(n)$ are the nth terms of the input and output sequences, respectively. The weight update equation for a least-mean squares (LMS) filter is

$$b_i(n+1) = b_i(n) + \mu e(n) x(n-i) \tag{2}$$

where μ is the step size and $e(n)$ is the adaptation error given by

$$e(n) = d(n) - y(n) \tag{3}$$

$d(n)$ is the desired output of the filter, i.e. the transmitted signal.

As is outlined in [4], splitting an N tap FIR into K concatenated subfilters of P taps each, such that $N = KP$, produces an estimate $y_s(n)$ (with $0 \leq s < K$) of the transmitted data ($d(n)$). The various equaliser outputs $y_s(n)$, can be used to compute a corresponding error signal $e_s(n)$ according to (3) such that

$$e_s(n) = \left| d(n) - y_s(n) \right| \tag{4}$$

The distinct error signals can be squared and averaged to obtain an output MSE measure for each subfilter. The performance criteria used to evaluate the different subfilters is the accumulated squared error (ASE) and is defined as

$$ASE_s(n) = \sum_{i=1}^{n} \left| d(i) - y_s(i) \right|^2 = \sum_{i=1}^{n} e_s(i)^2 \tag{5}$$

The advantage of using the ASE is that the repetitive computation of division, used in the calculation of MSE, is removed.

The aim of the length update algorithm is to detect the subfilter at which the ASE becomes insignificantly smaller or even larger than the previous subfilter. The algorithm proposed in [4] to control the number of active subfilters involved in equalisation, assuming the equaliser has L active segments, is outlined as

$$ASE_{L-1}(n) = \sum_{i=1}^{n} \beta^{n-1} \left| d(i) - y_{L-1}(i) \right|^2 \tag{6}$$

$$ASE_L(n) = \sum_{i=1}^{n} \beta^{n-1} \left| d(i) - y_L(i) \right|^2 \tag{7}$$

$$\begin{aligned} &\text{If } ASE_L(n) \leq \alpha_{up} ASE_{L-1}(n) \\ &\quad \Rightarrow +1 \text{ subfilter (P extra taps)} \end{aligned} \tag{8}$$

$$\begin{aligned} &\text{If } ASE_L(n) \geq \alpha_{dw} ASE_{L-1}(n) \\ &\quad \Rightarrow -1 \text{ subfilter (P fewer taps)} \end{aligned} \tag{9}$$

where $0 < \alpha_{up} \leq \alpha_{dw} \leq 1$ and determine the amount of worsening or improvement necessary to force the equaliser to expand or contract. β is a forgetting factor and is ≤ 1. A more detailed derivation of this algorithm can be found in [4].

A. Conventional Adaptive Filter Core

In order to evaluate the performance of the variable length equaliser a conventional adaptive filter core based on equation (1) and using the LMS algorithm with it's equations (2) and (3) was implemented. This conventional adaptive filter has two functional blocks, as seen in Fig. 1(a).

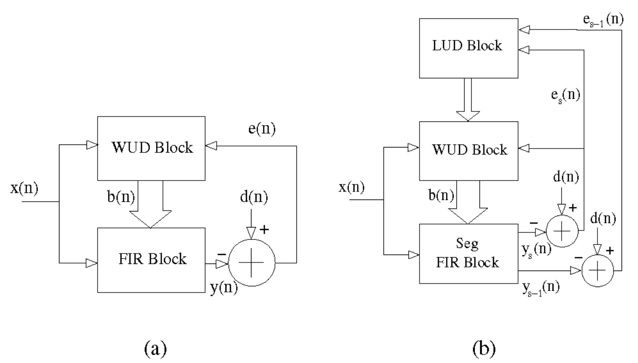

Fig. 1. a) Conventional Adaptive Filter b) Variable Length Adaptive Filter.

A weight update block (WUD), which uses the data input samples and an error signal to calculate new coefficients, and a filter block, which is generally an FIR filter and employs the coefficients calculated by the WUD block. Here a conventional direct form FIR filter was used.

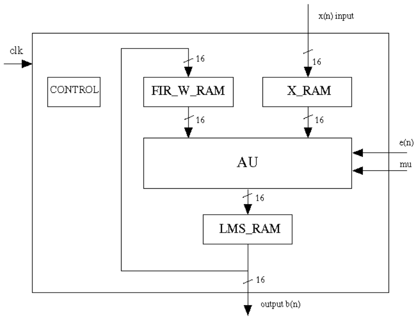

Fig. 2. Block Diagram of Conventional WUD Block.

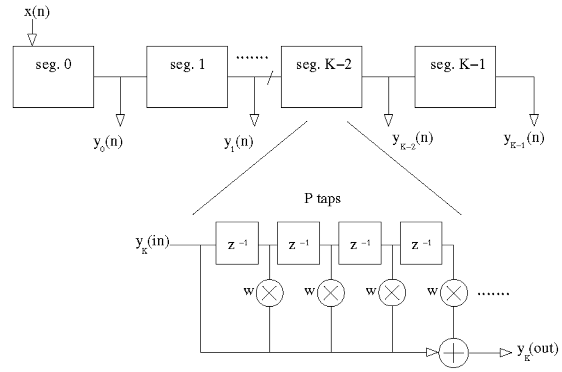

Fig. 3. Block Diagram of Segmanted FIR.

Within the WUD block there are five functional blocks as shown in Fig. 2. It consists of three memory blocks for storing the input data (X RAM), the calculated filter coefficients (LMS RAM) and the filter weights from the previous $(n-1)$ sample (FIR W RAM), an arithmetic unit (AU) and a control block (CONTROL). A brief description of these blocks is given below:

- CONTROL: The controller is based on a counter which provides addresses for the RAM blocks and is responsible for the synchronisation of activity for every block in the WUD unit.
- X RAM: This is a RAM used for the storage of the input data $x(n)$. The input data sequence is clocked into this RAM for the use of the AU when calculating filter weights.
- LMS RAM: Upon calculation of filter weights the values are stored in this RAM for the use of the FIR filter.
- FIR W RAM: When clocking filter weights out of the LMS RAM (by the FIR filter addresses) the values are immediately copied into this FIR W RAM for subsequent calculation of further filter weights for the next $(n+1)$ sample.
- AU: This arithmetic unit consists of two multipliers and an adder. This arithmetic unit performs the direct calculation of filter weights according to the weight update equation (2). The calculated values are latched into LMS RAM according to the signals created by CONTROL.

B. Variable Length Adaptive Filter Core

Basing implementation of the design on the algorithm described in [4], the variable length adaptive filter consists of three functional blocks, seen in Fig. 1(b). A WUD block, a length update (LUD) block, used to calculate the ASE values and provide tap-length data to the WUD block, and a segmented FIR filter block. The segmented structure of the FIR filter allows the calculation of subfilter outputs $y_s(n)$ and thus the error signals used to evaluate the error performance of the N tap filter and $(N-P)$ tap filter (P being the number of taps in one subfilter or segment). In this design the segmented FIR filter will calculate an output dependent on the number of coefficients presented to it by the WUD block. It is the output of subfilter results that are critical to the operation of the variable length adaptive filter design. The structure of the segmented FIR filter can be seen in Fig. 3. The control element in the conventional WUD block is replaced by a finite state machine (FSM) design which accepts the tap-length data from the LUD block, controls all timing signals and correctly addresses the various RAM elements in the WUD block. This enhanced control element is the main alteration made to the WUD and enables the length of the adaptive filter to increase or decrease based on the output value of the LUD block.

The LUD block takes the error signals from the last two FIR subfilters, $e_s(n)$ and $e_{s-1}(n)$ and calculates the respective ASE of equations (6) and (7) using accumulator units. The ASE values are latched and presented to a comparator unit in accordance with the length update algorithm of equations (8) and (9). From the comparator result a state machine controller in the LUD block then increments or decrements an up/down counter to provide a tap-length value. This LUD controller initialises the up/down counter and controls timing of the arithmetic units present. The number of taps initialised at reset is selected by the user and the maximum number of taps is set by the maximum size of the memory blocks and counters in hardware. The algorithm can vary the tap length of the filter to integer multiples of subfilter size P with the minimum size being P taps.

III. Results

Two different adaptive filter cores have therefore been implemented, the first being the conventional (CON) and secondly the variable length (VAR-LMS). Both have been analysed in terms of area usage and power consumption and in channel equaliser configuration. The cores were designed using Verilog HDL and then synthesised using Design Compiler^{TM} targeting the UMC 0.18μ standard cell CMOS library. The requirements of the synthesis were identical for all cores. This was vital in order to allow for consistent power consumption and area usage comparisons. A netlist was

TABLE I

AREA ANALYSIS FOR DIFFERENT ADAPTIVE FILTER CORES

Core	Area (μm^2)	% Difference
CON	451255.0	–
VAR LMS	479232.81	+6.2

TABLE II

POWER CONSUMPTION ANALYSIS FOR DIFFERENT EQUALISER TAP–LENGTHS

Core (No of Taps)	Dynamic Power (mW)	% Difference
CON (64)	21.01	–
VAR LMS (64)	21.89	+4.19
VAR LMS (56)	20.77	–1.14
VAR LMS (48)	19.63	–6.57
VAR LMS (40)	18.48	–12.04
VAR LMS (32)	17.37	–17.33
VAR LMS (24)	16.20	–22.89
VAR LMS (20)	15.62	–25.65
VAR LMS (16)	15.05	–28.36

created for each core and back–annotated netlist simulations for a uniformly distributed random input bipolar binary data of 10000 samples using Verilog–XL™ simulator were performed and verified against Matlab™ simulation results. The resulting data, including switching activity of the circuit nets was then used by Synopsys DesignPower™ to determine power consumption for the different adaptive filter cores. In all of the above stages a clock frequency of 100 MHz and a supply voltage of 1.8 Volts were used.

Area results are shown in Table I and power results are shown in Table II. A 64–tap adaptive filter was analysed (CON being fixed and VAR LMS being max possible length) in both cases with fixed step–size μ = 0.005. Power analysis was carried out using an input E/N_0 of 15 dB and fixed channel profile. The channel profile used is identical to that found in [4]. Various equaliser lengths were manually chosen to analyse the effects on power consumption for various tap–lengths. For example, with input E/N_0 of 15 dB, results in [4] show there is no improvement in steady state MSE performance achieved for tap lengths greater than 20. Table II therefore shows that for the VAR LMS core a decrease in tap–length results in an increase in the power saving achieved. These results also identify the overhead present in the VAR LMS architecture. For the VAR LMS core when 64 taps are in operation the power consumed is 0.88 mW higher than in CON when 64 taps are in operation. It can also be seen that the overhead in core area amounts to an increase of 6.2% in VAR LMS over CON.

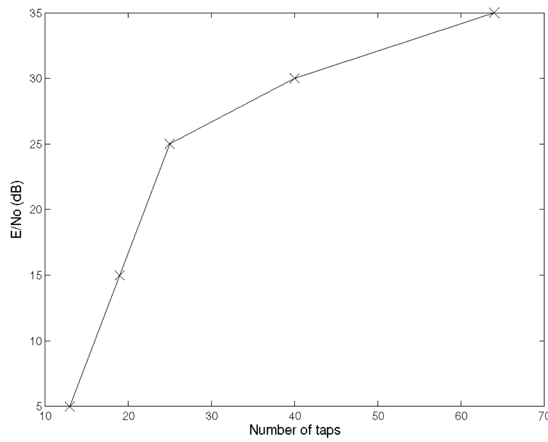

Fig. 4. Number of taps for varying E/N_0

When the length update algorithm is employed to determine the optimum number of taps, power is saved due to the fact that a filter of greater fixed length will not achieve a better steady state MSE. The graph of Fig.4 shows the optimised steady state filter length of the VAR LMS core for various input E/N_0 levels. In this case K = 24 and P = 3 taps/segment.

IV. CONCLUSION

This paper has presented a novel architectural VLSI implementation of a dynamically length optimised LMS adaptive filter for use in channel equalisation. The technique was implemented in a 64 tap adaptive filter core and demonstrates length optimisation with varying input E/N_0. The results demonstrate a power saving is achieved by optimising the number of taps in operation. Results have shown a power saving of 28% can be achieved for a variable length architecture optimised to 16 taps over a conventional 64 tap fixed length adaptive filter architecture. It has also been shown that the low–complexity of the additional circuitry needed for the variable length adaptive filter presents minimal overhead for this architecture.

REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*, Prentice–Hall, 2000. 4th edition.

[2] Z. Pritzker and A. Feuer, *Variable length stochastic gradient algorithm*, IEEE Trans. Signal Processing. Vol. 39, pp. 997–1001, Apr. 1991.

[3] Y.K. Won, R.H. Park, J.H. Park and B.–U. Lee, *Variable LMS Algorithms using the time constant concept*, IEEE Trans. Consumer Electron. Vol. 40, pp. 655–661, Aug. 1994.

[4] F. Riera–Palou, J.M. Noras and D.G.M. Cruickshank, *Linear equalizers with dynamic and automatic length selection*, Electron. Lett. Vol. 37, no. 25 pp. 1553–1554, Dec. 2001.

[5] R.C. Bilcu, P. Kuosmanen and K. Egiazarian, *A new variable length LMS algorithm: Theoretical analysis and implementations*, Proc. 9th Int. Conf. Electronics, Circuits and Systems. Vol. 3, pp. 1031–1034, 2002.

[6] Y. Gu, K. Tang, H. Cui and W. Du, *LMS algorithm with gradient descent filter length*, IEEE Signal Processing Lett. Vol. 11, no. 3, pp. 305–307, March 2004.