

An Efficient Reformulation Based Architecture For Adaptive Forward Error Correction Decoding In Wireless Applications

Yao Gang, Tughrul Arslan, Ahmet T Erdogan
 School of Electrical Engineering, Edinburgh University
 Edinburgh UK, EH93JL
 Email: Y.Gang@ed.ac.uk

Abstract— The authors present a reformulation based architecture for Threshold Selection in Adaptive Forward Error Correction (FEC) Decoding in wireless applications. The reformulation technique results in an efficient VLSI architecture with a significant reduction in hardware complexity. The paper describes the reformulation technique, its applications on the Architecture For Adaptive Forward Error Correction (FEC) Decoding Algorithm and its implementations. We demonstrate that in addition to significant reduction in data path complexity, there is also a 25% to 47% storage reduction in the Path Metric Unit (PMU).

I. INTRODUCTION

Wireless networks usually employ sophisticated Forward Error Correction (FEC) techniques such as Viterbi Algorithm [1] (VA) to improve reliability. The computation complexity and storage requirement of the conventional VA increased exponentially with the constraint length of the Viterbi Algorithm [1]. Considering the strict constraints on the area and the power in the mobile systems especially the battery based handsets, it is prohibitively complex to implement the conventional FEC scheme based on VA.

II. CONVENTIONAL ADAPTIVE VITERBI ALGORITHM

Chan [2] proposed an efficient FEC scheme referred to Adaptive Viterbi Algorithm VA (AVA), which computed the received channel symbol adaptively according to the channel environment. Up till now, Sriram's work [3] is the only one that described the VLSI architecture of AVA algorithm in literature. In [3], the minimum path metric is stored into the Path Metric Memory Unit (PMU) in the current iteration of the trellis recursion and are read out for calculating the path metric and determining the survivor state in the next iteration. The conventional Threshold Selection Unit architecture is shown in Fig 1.

III. PROPOSED REFORMULATED APPROACH

A. Algorithm Reformulation

Let $i, j \in S$ represent the states of the trellis graph and state i is the preceding state of state j associated with a state transition from i to j .

Let PM_n^i represents the accumulated path metric of state i at time step n . Specifically, PM_n^{min} represents the minimum path metric of the surviving states at time step n .

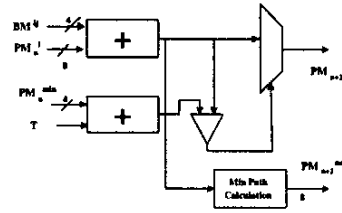


Fig. 1. Threshold Selection Unit architecture : conventional

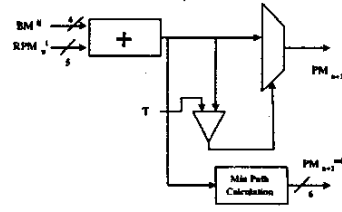


Fig. 2. Threshold Selection Unit architecture : reformulated

So rules of the AVA algorithm can be expressed as:

$$PM_{n+1}^j \leq PM_n^{min} + T \quad (1)$$

As the state i is the preceding state of j , and BM^{ij} denotes the branch metric that refers to state transition i to j :

$$PM_{n+1}^j = PM_n^i + BM^{ij} \quad (2)$$

While PM_n^{min} is the minimum path metric of time step $n-1$, we let RPM_n^i denote the rescaled version of path metric for the state i at time step n ,

$$RPM_n^i = PM_n^i - PM_n^{min} \geq 0 \quad (3)$$

By plugging (2) and (3) into (1), then AVA equation (1) can be reformulated as

$$RPM_n^i + BM^{ij} \leq T \quad (4)$$

As shown in Fig 2 we reformulated the algorithm from eq 1 to eq 4. In the reformulated scheme, the comparison is simplified from variable based to constant based. Moreover, the RPM_n^i is calculated on the fly and are not stored.

TABLE I
COMPARISON FUNCTION BASED ON STANDARD CELL TECHNOLOGY(
 $T = 20$)

	Area	Power	Delay
$a < T$	1	1	1
$a < b$	8.34	2.03	1.88
$a < b + T$	9.87	2.18	2.47

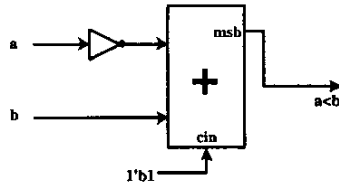


Fig. 3. Stand Cell Approach For The Variable Based Comparator

B. Standard Cell Technology

In standard cell CMOS technology, the constant comparison can be implemented much more efficiently than the variable comparison. For example, a 3 bits wide variable based comparator can be implemented by a full adder. The stand cell implementation for the function $a < b$ is described in Fig 3. above description. Three types of basic comparison units are mapped into UMC's .18u standard cell technology. TABLE I summarizes the normalized results on area, power and delay of the comparison functions ($a < T$, $a < b$ and $a < b + T$, where $T = 20$) is a typical constant used in AVA Decoding in wireless LAN applications and a and b are both 5 bits wide vectors.

C. Analysis

It can also be noted that in the reformulated approach the calculation of rescale metric RPM_n^i is introduced as an overhead. It shall be shown that this kind of computation contains no redundancy at all. The rescaled version of the path metric corresponds to the rescaled method which aims to overcome the problem of overflow for the path metric [4]. Although for Viterbi Decoders (VDs) reported in the literature, modulo logic is always adopted with a freedom of the calculation of the minium path metric, which is a heavy overhead in hardware, the reformulated scheme exploits the inherent parallism of the threshold selecting rule and rescale operation without modulo method's penalty of the increase in the width of the path metric. For example, in a $R = 1/2$, $K = 9$ VD's with 3 bits soft quantization, the width of a path metric employing modulo logic is 8 [5]. While according to our approach, only 7 bits wide path metric is needed in the first instance. Moreover, as the compare select operation is based on the constant T , which has an optimal value range of 20 to 30 according to [2], we only need 5 bits to store the path metric because all the paths with a metric larger than T will be discarded. In TABLE II, the width requirement for path metric memory in

TABLE II
WIDTH REQUIREMENT FOR MEMORY IN PMU

K(Constraint Length)	4	6	7	9
Conventional	4	8	8	9
Reformulated	3	4	5	5

TABLE III
AVA ARCHITECTURE TARGETING FOR STANDARD CELL TECHNOLOGY

	Area	Power	Delay
Conventional	1	1	1
Reformulated	0.53	0.43	0.93

PMU is presented. A storage efficiency of [25%, 47%] can be achieved under different constraint length specifications.

IV. IMPLEMENTATION RESULTS

The specification of the implementations are :

- 64 states, constraint length $K = 7$, code rates $R = 1/2$
- 3 bits, 8 level soft decision inputs
- ASIC Approach : UMC .18u stand cell library

The results of the stand cell approach are normalized in TABLE III. 47%, 57% and 7% improvement are achieved in area, power efficiency and speed respectively. Significant improvement can be observed. In addition, the power efficiency is enhanced compared to the basic comparison unit results presented in TABLE I. Further reduction in power can be contributed to the reduction of complexity in minium path metric determination unit and path metric adder.

V. CONCLUSION

The reformulated architecture exploits the inherent parallism between the Add Compare Select Operation and rescale operation in Adaptive FEC Decoding and leads to a significant reduction in area, power and delay to the whole VD. It should be noted that the proposed technique will also achieve a similar power, area and speed efficiency with different specifications e.g. $K = 9$.

REFERENCES

- [1] A.J.Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Transaction on Communications*, vol. COM-19, p. 751 to 771, October 1971.
- [2] F. Chan and D. Haccoun, "Adaptive viterbi decoding of convolution codes over memoryless channels," *IEEE Transaction on Communications*, vol. 45, p. 1389 to 1400, November 1997.
- [3] R. T. Sriram Swaminathan and D. Goeckel, "A dynamically reconfigurable adaptive viterbi decoder," in *FPGA'02, International Symposium on Field Programmable Gate Arrays Archive Proceedings of the 2002*, p. 227 to 236, February 2002.
- [4] A.P.Hekstra, "An alternative to metric rescaling in viterbi decoders," *IEEE Transaction on Communications*, vol. 37, p. 1220 to 1222, November 1989.
- [5] P. J.Black and T. H.Meng, "A 140-mb/s 32-state, radix-4 viterbi decoder," *IEEE Journal Of Solid-State Circuits*, vol. 27, p. 1877 to 1885, December 1992.