

Efficient Implementation of Trace-back Unit in a Reconfigurable Viterbi Decoder Fabric

Cheng Zhan¹, Sami Khawam¹, Tughrul Arslan^{1,2}, Iain Lindsay¹

¹ School of Electronics and Engineering
University of Edinburgh, King's Buildings, Mayfield
Road, Edinburgh, EH9 3JL, UK

² Institute for System Level Integration
The Alba Centre, Alba Campus
Livingston, EH54 7EG, UK

{c.zhan, Sami.Khawam, Tughrul.Arslan, Iain.Lindsay}@ed.ac.uk

Abstract—This paper presents a reconfigurable viterbi fabric with efficient track-back unit in a system on chip device. The proposed reconfigurable fabric can support Viterbi implementations with constraint lengths ranging from 3 to 9, and code rates in the range 1/2-1/3. Our results illustrate that the proposed architecture has superior power consumption and throughput characteristics compared with a generic field programmable gate array (FPGA) and a digital signal processor (DSP), respectively.

I. INTRODUCTION

In order to support the various communication applications and services, future portable communication systems need to be more flexible, as well as providing low power consumption and high throughput. All of these requirements increase the need for a flexible, low-power and high-speed architecture for future mobile systems.

The Viterbi algorithm [1] is commonly used for decoding the convolutional code, a widely used channel coding techniques in today's digital communication systems. The configuration of the convolutional codes is disparate in each communication standard which imparts different requirements on the Viterbi decoder. Thus, a reconfigurable Viterbi decoder with low power consumption and high throughput is a key challenge for future portable devices. In this paper, we present a novel domain specific Viterbi decoder architecture with an efficient trace-back unit targeting reconfigurable SoC platforms for future portable devices. Such a domain specific reconfigurable fabric exhibits a good compromise of flexibility, power-consumption and throughput when compared to application specific integrated circuits (ASIC), FPGAs and DSPs.

The paper is organized as follows: In section II describes the reconfigurable system-on-chip system. The domain specific reconfigurable fabric for Viterbi decoder is addressed in section III and its performance is assessed in section IV.

II. SYSTEM OVERVIEW

Since a DSP incurs significant overhead of fetching, decoding and executing a stream of instructions which means most of the clock cycles and power are wasted in fetching the instruction and setting the appropriate control signals. So some communication digital signal processing algorithms, such as Viterbi algorithm, are best handled in a domain specific reconfigurable array to achieve low power consumption and high throughput rate. The data to be processed is sent to the domain specific reconfigurable fabric by DSP and read back to the DSP via a system-on-chip bus. The programming of the reconfigurable fabric is done by the micro-processor. By writing the different configuration bits to reconfigurable fabrics, these fabrics can be configured to different implementations during run-time.

III. RECONFIGURABLE FABRIC FOR VITERBI DECODER

A. Viterbi Algorithm

The Viterbi algorithm is known to be an efficient method to perform maximum likelihood sequence detection of the convolutional codes. It can be viewed as a technique to find the shortest path metric in a trellis diagram. The computation of Viterbi algorithm is an iterative operation. The most computationally intensive steps in each iteration are as follows:

- Branch metric (BM) computation
- *Add-compare-select* operation to update the path metric (PM) of each trellis state and generate decision bits
- Decision bits store and output decoding sequence

The detailed processing units of this proposed domain specific reconfigurable fabric will be talked about below.

B. Branch Metric Unit

The *BM* computation [2] is to calculate the distance between the received signals and the ideally transmitted signals at each trellis stage. In a specific application, the *BM* calculation diagram is varied with convolutional code rate R (R bits are transmitted for each bit in the message, $R-1$ of them being parity bits). Since the *BM* for code rate $1/2$ will be used in calculating the *BM* for code rate $1/3$, we do not need to design two different hardware circuits for each code rate. We can reuse the code rate $1/2$ *BM* calculation circuit to decrease the power and area overhead.

C. Add-Compare-Select Unit

The path metric update for each new trellis state according to the following equation:

$$PM(i)_{t+1} = \min_{\text{all possible } j} (PM(j)_t + BM(j,i))$$

Where the $PM(i)_{t+1}$ and $PM(j)_t$ are corresponding to the path metric of state i at time stage $t+1$ and state j at time stage t , respectively. The $BM(j,i)$ is associated with the branch metric from state j to state i which is generated by the *BM* unit. So we call, this kind of data-dependent feedback loop, add, compare and select (*ACS*) process. In order to increase the modularity and decrease the complexity of the reconfigurable design, we combine two *ACS* operations together, named *butterfly operation*. In every butterfly processing unit, we exploit multiplexers and LUTs to fit the different constraint lengths.

D. Trace-back Unit

Each processing cycle, in addition to update two *PMs* for next iteration, every butterfly processing unit offers two decision bits d_t^i and d_t^{i+1} , where d_t^i represents the decision bit of trellis state i at time cycle t , and the decision vector, $d_t^{2^k-1} d_t^{2^k-2} \dots d_t^0$, which is a set of decision bits, one per trellis state, at time cycle t , will be stored in the same decision memory location. The final objective of the Viterbi decoder is to exploit the decision vector to reconstruct the trellis path. The previous trellis path state S_{t-1}^k given by the current path state S_t^k according to the following update [3]:

$$S_{t-1}^k = d_t (S_t^k \gg 1)$$

which corresponds to a right shift of the current state introducing the value of surviving bit d_t in the vacant position. Since the length of S_t^k is equal to $k-1$, the size of the right shift register need to vary with constraint length k , we use seven 2-to-1 multiplexers to implement this varying sized right shift register, shown in Figure 1.

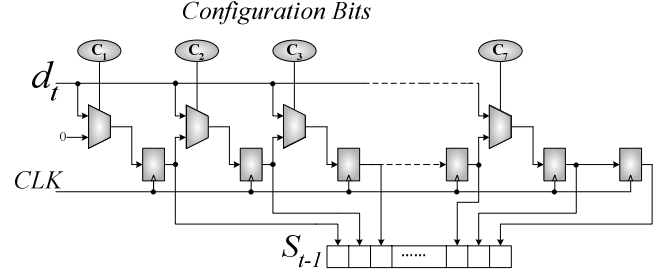


Figure 1. Reconfigurable Right Shift Register

Meanwhile, the size of decision memory block and the address generator also need to be reconfigurable for different constraint length. In addition, considering reducing the employ of the on-chip bus and saving the memory access time, we design the memory unit, shown in figure 2, which is inside the reconfigurable fabric rather than the on-chip memory to store and read these decision vectors.

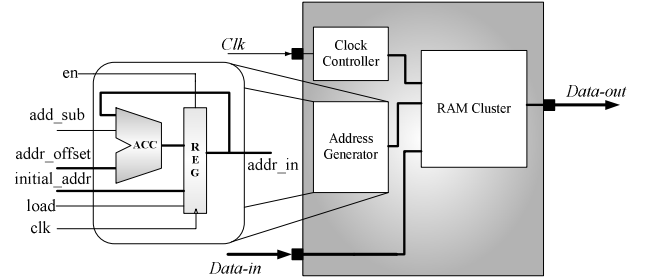


Figure 2. Data Memory Unit

This reconfigurable data memory unit is composed of three components, RAM cluster, clock controller, and address generator. Due to a tradeoff between the memory size and access time, we select 64x4 dual-port block RAM as the basic element and there are eight of these in the RAM cluster. So the width of this RAM cluster can be configured from 4bits to 32bits. In addition, changing the configuration of the clock controller, the clock input of the unused block RAM can be disabled and the relevant RAM blocks will be powered down. Thus, the power consumption overhead of the whole architecture can be obviously cut. The address generator inside the data memory unit is used to add the flexibility of write and read operations of the RAM cluster. Using this generator, the memory unit can be configured as *FIFO* (first in first out), *LIFO* (last in first out) and random access *RAM* which will be used in the domain specific application.

IV. IMPLEMENTATION RESULTS

This domain specific reconfigurable Viterbi decoder fabric is targeted on a UMC 0.18um CMOS technology library. For comparative results, Viterbi decoders are also realized using standard ASIC technology and Xilinx Virtex-E FPGA (XC300e-8-PQ240). The comparison results of the different architectures are detailed in the Table I, II and

III, respectively. All of these architectures use the 0.18μm CMOS technology and run at 1.8V and 20MHz. Table I and II show the power consumption and area of an ASIC and our reconfigurable architecture, respectively. We can see that the power and the area of our architecture are around 5.4 times and 1.5 times bigger than that of ASIC as the tradeoff for the flexibility.

Table III illustrates the results of the Viterbi implementation on the Xilinx FPGA, and the power consumption in this table do not include the quiescent power (Quiescent power = 540 mW). We can clearly deduce that the power consumption is 5 times more than that of our architecture. Due to this power figure, the reconfigurable SoC platform based on embedded FPGAs is unsuitable for the low power requirement of future portable devices. 80% power reduction is achieved in our architecture in comparison to Xilinx FPGA.

TABLE I. RESULTS OF THE RECONFIGURABLE IMPLEMENTATION

Constrain Length K	Power (mW)		Size (mm ²)
	Logic	Memory	
3	1.09	1.33	0.05
5	1.55	10.64	0.1
7	5.27	42.94	0.34
9	20.13	169.67	1.19

TABLE II. RESULTS OF THE ASIC IMPLEMENTATION

Constrain Length K	Power (mW)		Size (mm ²)
	Logic	Memory	
3	0.49	1.33	0.03
5	0.94	1.70	0.07
7	3.39	5.51	0.22
9	12.52	34.17	0.78

The Viterbi algorithm can also be implemented on a TI TMS320C6416T DSP with a Viterbi Coprocessor customized for 3G wireless infrastructures [4]. This

processor can provide a maximum of 5Mbps decoding throughput. However, the maximum frequency of our architecture can reach up to 200MHz or a decode rate of 200Mbps, which is 40 times faster than DSP.

TABLE III. RESULTS OF THE IMPLEMENTATION ON XILINX VIRTEX-E

Constrain Length K	Power (mW)	Max Freq. (MHz)	# LUT	# FF
3	22.97	105.2	434 (7%)	193 (3%)
5	61.02	100.1	1045 (17%)	381 (6%)
7	213.17	83.7	4351 (70%)	1174 (19%)

V. SUMMARY

We have addressed a novel domain specific reconfigurable Viterbi decoder architecture with an efficient Trace-back unit as a reconfigurable core for integration into a reconfigurable SoC platform in order to provide high flexibility as well as good power and timing performance characteristics for future portable devices. After comparison with Xilinx FPGA and TI DSP, we found that power consumption is reduced considerably by around 80% over FPGA and the throughput improved 40 times over DSP. These results confirm that the proposed reconfigurable SoC platform which is based on domain specific reconfigurable fabrics is highly suitable for a high-performance communication system.

REFERENCES

- [1] A.J Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. IT-13, pp.260-269, April 1967
- [2] J.Proakis, *Digital Communications*, 3rd ed. McGraw Hill, 2001
- [3] P.J.Black and T.H.Meng, "A 140-Mb/s, 32-state, radix-4 Viterbi decoder," *IEEE Journal of Solid-State Circuit*, vol. 27, no.12, pp.1877-1885, December 1992
- [4] TI TMS320C6416T DSP data sheet, <http://www.ti.com>