Domain Specific Reconfigurable Fabric Targeting Viterbi Algorithm

Cheng Zhan[1], Sami Khawam[1], Tughrul Arslan[1, 2]

[1] *School of Electronics and Engineering*
University of Edinburgh, King's Buildings,
Mayfield Road, Edinburgh, EH9 3JL, UK
c.zhan@ed.ac.uk
Sami.Khawam@ee.ed.ac.uk

[2] *Institute for System Level Integration*
The Alba Centre, Alba Campus
Livingston, EH54 7EG, UK
Tughrul.Arslan@ee.ed.ac.uk

Abstract

This paper presents a novel embedded reconfigurable fabric targeting efficient implementation of the Viterbi decoder within a System-on-Chip device. The proposed reconfigurable fabric can support constraint lengths ranging from 3 to 9, and code rates in the range 1/2-1/3.Our results demonstrate that this novel architecture has superior throughput and power consumption characteristics when compared to generic DSPs and FPGAs respectively.

1. Introduction

In order to cope with large amount of applications and services, future portable communication systems will to be more flexible, have low power consumption and high throughput. All of these requirements put tremendous pressure on current hardware implementation technologies. A flexible, low-power and high-speed architecture for future mobile systems is crucial. The continuing progress of semiconductor technologies and the increasing complexity of the silicon devices have enabled the emergence of the "Reconfigurable-System-on-Chip" (RSoC) concept. Reconfigurable SoC solution is seen by many researchers as the implementation methodology for future mobile systems [1] [2]. A typical reconfigurable SoC platform contains a general purpose micro-processor, digital signal processor (DSP), a number of domain specific reconfigurable fabrics, as well as some glue logics and ASIC parts.

Convolution code is widely used channel coding techniques in today's digital transmission systems. Because of its high performance, the Viterbi algorithm [3] is commonly used to decode the convolution code in different communication standards and communication environments. For instance, the GSM standard adopts a Viterbi decoder with the constraint length of 5, a rate of 1/2. Whereas the WLAN standard specifies a constraint length of 7, rate of 1/2 for the Viterbi decoder. The Viterbi decoder for 3G standards, on the other hand, has a specification of constraint length of 9 and a rate of 1/2 or 1/3. Thus, a flexible, low-power and high-throughput Viterbi decoder design is a key challenge for future portable devices. The low throughput characteristic of general DSPs and high power consumption of the general FPGAs make these less attractive than domain specific reconfigurable fabrics which could be directly integrated into an overall reconfigurable SoC platform.

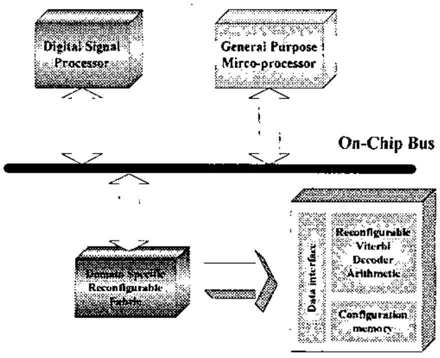

Figure 1 Reconfigurable System-on-Chip

In this paper, we present a novel domain specific Viterbi decoder architecture targeting an overall reconfigurable SoC platform for future portable devices, as shown in Figure 1. The dedicated reconfigurable Viterbi decoder fabric can be reconfigured for constraint lengths from 3 to 9 and code rates 1/2 and 1/3. The paper is organized as follows: In Section 2, the structure of the reconfigurable module for Viterbi decoder is overviewed. Comparisons with non-reconfigurable fixed Viterbi decoder and generic FPGA and DSP are presented in Section 3.

0-7803-8652-3/04/$20.00 © 2004 IEEE

2. Structure of the Reconfigurable Module for Viterbi Decoder

There are five major components in this fabric, the branch metric unit (BMU), the add-compare-select unit (ACSU), survivor memory unit (SMU), power control unit and configuration memory unit. Due to the demands of proposed architecture in terms of flexibility, low power consumption and high throughput, the following technologic features are exploited in this architecture.

2.1 Branch Metric Unit

The BMU is to compute the distance between the received quantization signal and the ideally transmitted signals. The BM calculation equations for code rate 1/2 and 1/3 are shown in the Table 1, the x, y, z denote the quantized inputs.

We can find that if we want to calculate the branch metric for these two rates, we do not need to design two different specific circuits. Since the bm00, bm01, bm10, bm11 are recalculated at rate of 1/3. Thus, we can reuse the code rate 1/2 circuit to decrease the power and area overhead.

Table 1 Branch Metric Calculation Function

Code Rate	1/2	1/3
BM Calculation Function	bm00 = x + y	bm000 = x + y + z = bm00+z
		bm001 = x + y + (-z) = bm00+(-z)
	bm01 = x + (-y)	bm010 = x + (-y) + z = bm01+z
		bm011 = x + (-y) + (-z) = bm01+(-z)
	bm10 = (-x) + y	bm100 = (-x) + y + z = bm10+z
		bm101 = (-x) + y + (-z) = bm10+(-z)
	bm11 = (-x) + (-y)	bm110 = (-x) + (-y) + z = bm11+z
		bm111 = (-x) + (-y) + (-z) = bm11+(-z)

2.2 Butterfly Unit

The ACSU takes in the output branch metric from the BMU, calculates the path metric and the decision bits at each state node of the transmission trellis and stores the new path metric as the input for next iteration. Due to the path metric is saved in finite-length registers, metric normalization is required to prevent errors because of the overflow during the loop calculation of the path metric. According to [4], the module normalization is the best way to avoid the catastrophic overflow and the minimum bit width of the path metric register is 8 bits. In the VLSI architecture for the add-compare-select (ACS) calculation, 8-bits ripple adder and the modified ripple comparer to get more efficiency in area and speed.

In order to increase the modularity and decrease the complexity of the reconfigurable design, we reorder the trellis into groups of states and each group has the identical elements, called butterfly unit. Each butterfly unit is made of 2 ACS units and contains 2 source and 2 destination states.

Considering the fully parallel architecture in this application, the reconfigurable switch among the butterfly units becomes an important issue. After observing the reordered trellis structures of different constraint length k, we realize that one path metric input in butterfly unit varies with k, another is fixed. In this way, by adopting a 7-input multiplexer, the interconnection between the path metric inputs of each butterfly and the path metric register can be configured for different constraint length.

Meanwhile the four branch metric inputs of the butterfly unit repeat irregularly, which fully depend on the generator polynomials of the convolution encoder. Thanks to the symmetric characteristic of the generator polynomials, two branch metric inputs are equal the other two. So we can exploit two 8-input multiplexers to select the branch metric inputs for each butterfly unit. The whole structure of this reconfigurable butterfly unit is shown in Figure 2.

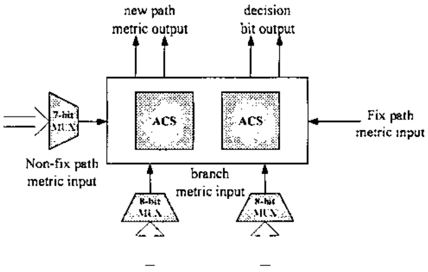

Figure 2 Structure of one butterfly unit

2.3 Survivor Management Unit

The survivor management unit is responsible for finding the decode data by using the decision bits from the butterfly unit. There are two basic algorithms for this unit, namely register exchange and trace-back. We choose trace-back because it can achieve higher implementation efficiency, lower power consumption, and easier to reconfigurable.

2.3.1 Trace-back strategy

Four independent same size memory blocks are used to implement this unit whose read/write pointer using the same frequency. This strategy allows four processes, Write, Trace-back, Decision and Idle, to parallel access four same size memories, and each of them processing a different memory block at the same time.

In the Trace-back and Decision blocks, the previous state can be got from $S_{n-1}^{k} = d_n(S_n^k >> 1)$ [5], where S_n^k is the state at time n, constraint length is k, and d_n is its decision bit. Since the length of S_n^k is equal to k-1, the size of the right shift register need to vary with constraint length k. We exploit 7 multiplexers to implement this varying size right shift register, shown in Figure 3.

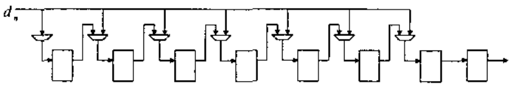

Figure 3 Reconfigurable right shift register

2.3.2 Reconfigurable RAM structure

In order to ensure the path can trace-back to the merged state, the depth of the memory needs to be more than 5 times of the constraint length k and the width of the memory is equal to 2^{k-1}. Since the constraint length k is from 3 to 9, the memory block need to reconfigure its depth and width.

In our reconfigurable RAM architecture, the small dual-port RAM (32x4bits) is as the basic element to buildup the big ram with the connect boxes. The input reconfiguration bits can be used to control the connect box to combine the neighbor basic rams together or not. The Figure 4 shows four basic ram elements with three neighboring connection boxes to implement the 64x8bits ram.

2.4 Power Control Unit

As mentioned earlier, we use fully parallel butterfly architecture in the ACSU. For different constraint length, not all the butterfly units are being used. We adopt tri-buffers to turn off the clock input of the unused butterfly units, thus neither the inputs, nor the outputs can toggle, and the non-used butterfly units can not consume power at all. Similarly, writing the different reconfiguration bits to the connection boxes, the relevant RAM blocks will be powered down. Thus, the power consumption overhead is litter when compared with non-reconfigurable fixed Viterbi decoder.

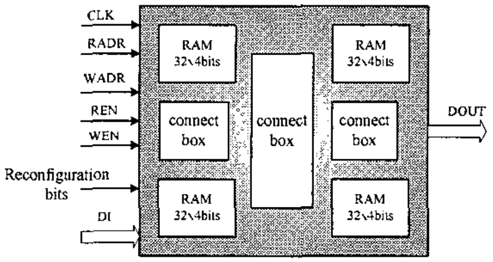

Figure 4 64x8 configurable RAM

2.5 Configuration Memory Unit

This unit stores the reconfiguration bits to program the switch of multiplexers and the status of the tri-state buffers. The commercial FPGAs use SRAM to store the reconfiguration bits, but SRAM cannot be synthesized. Hence, we use flip-flop composing the shift chain to save the reconfiguration bits. The total reconfigration bits of this fabric are 807 bits, shown in Table 2.

Table 2 Total configurable bits

Component	BMU	Trace-back	RAM	ACS	Power Contr-oller
Number of configuration bits	1	8	20	771	7

When the constraint length is less than 9, configuration does not occupy the full memory area. We divide the memory into 8 chunks, and each of them has separate address information. Using the address information, the micro-processor or DSP can choose the specific memory chunk to download the reconfiguration bits in. This kind of selective reconfiguration can avoid writing the whole reconfiguration data to the configurable memory unit when only partial architecture requires reprogramming.

3. Implementation Results

The domain specific reconfigurable Viterbi decoder module is targeted on UMC 0.18um CMOS technology library. For comparative results, we have also implemented the non-reconfigurable architecture using a standard UMC 0.18um ASIC technology and on Xilinx Virtex-E FPGA (XCV300e-8-PQ240). All of these devices use the 0.18um CMOS technology and run at 1.8V. For the Xilinx FPGA, the maximum frequency can go up to 70MHz whereas the maximum frequency of an ASIC and our reconfigurable architecture can reach up to 200 MHz. Table 3 shows the comparative results between an ASIC and our reconfigurable architecture. We have implemented constraint lengths of K=3, K=5, K=7, and K=9 on the ASIC and our reconfigurable architecture and ran at 20 MHz. We can clearly see that the power overhead for reconfiguration is less than 46% and area overhead is nearly 52%.

Table 3 Reconfigurable Vs non-reconfigurable

Constraint Length K		3	5	7	9
Recon	Power (mW)	1.6	4.4	16.9	57.1
	Area (um^2)	46530	103729	325521	1193841
Fixed	Power (mW)	1.1	3.1	11.6	39.4
	Area (um^2)	32539	68198	221443	785422
Power Overhead (%)		45.5	42.0	45.7	45.0
Area Overhead (%)		43.0	52.1	47.0	52.0
Frequency (MHz)		20			

Table 4 illustrates the power consumption (sans the quiescent power) of the Viterbi decoder running on the Xilinx Virtex-E. We can clearly deduce that the power consumption on the FPGA is much higher than that of our architecture. The power figure for the Xilinx FPGA makes it unsuitable for the low power requirement of future portable devices.

Table 4 Power consumption on Xilinx Viretex-E

Constrain Length K	3	5	7
Power (mW)	21.72	67.35	386.66
Max. Frequency (MHz)	70		

The Viterbi algorithm was also implemented on a TI TMS320C6416T DSP with a Viterbi Coprocessor customized for 3G wireless infrastructures. This processor can provide a maximum of 5 Mbps decoding throughput [6]. However, the maximum frequency of our architecture can reach up to 200MHz or a decode rate of 200Mbps.

4. Summary

We have proposed a novel domain specific reconfigurable Viterbi decoder architecture as a reconfigurable module which can be integrated into a reconfigurable SoC platform to provide high flexibility as well as good performance characteristics for future portable devices. By writing the different reconfiguration bits to the configuration memory, the status of the switch and connection components can be reconfigured and the different versions of the Viterbi decoder can be implemented on this architecture.

This flexible, low-power and high-speed domain specific reconfigurable architecture provides a compromise between the ASICs, generic FPGAs and DSPs. After comparison with non-reconfigurable fixed ASIC, Xilinx FPGA and TI DSP, the power and area overhead for reconfiguration were found to be less than 46% and 52%, respectively. However, the architecture provides much higher throughput than the TI DSP and much lower power consumption than Xilinx FPGAs. These results confirm that the proposed architecture is suitable for a high-performance communication system.

5. Reference

[1]Khawam S., Arslan T., Westall F., "Embedded reconfigurable array targeting motion estimation applications", *Proceedings of the 2003 IEEE International Symposium on Circuits and Systems (ISCAS'03)*, May 2003, Vol.2, Page(s): 760-763

[2]Abnous, A.; Rabaey, J., "Ultra-low-power domain-specific multimedia processors" *VLSI Signal Processing, IX, 1996*, 30 Oct.--1 Nov. 1996 Page(s):461 – 470

[3] 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

[4] 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

[5] J.A.Heller and I.M.Jacobs, "Viterbi decoding for satellite and space communication," *IEEE Transactions Communication Technology*, vol. COM-19, no.5, Page(s): 835-848, Oct, 1971

[6] TI TMS320C6416T DSP data sheet, http://www.ti.com