

# Architecture, performance modeling and VLSI implementation methodologies for ASIC vector processors: A case study in telephony workloads



Vassilios A. Chouliaras<sup>a,\*</sup>, Konstantia Koutsomyti<sup>b</sup>, Simon Parr<sup>c</sup>, David Mulvaney<sup>a</sup>, Mark Milward<sup>a</sup>

<sup>a</sup> School of Electronic, Electrical and Systems Engineering, Loughborough University, Loughborough LE11 3TU, United Kingdom

<sup>b</sup> Processor Division, ARM Ltd., 110 Fulbourn Road, Cambridge, GB-CB1 9NJ, United Kingdom

<sup>c</sup> M-Stack, Albert House, Quay Place, 92-93 Edward Street, Birmingham, B1 2RA, United Kingdom

## ARTICLE INFO

### Article history:

Available online 16 October 2013

### Keywords:

VLSI  
Microprocessors  
Vector processors  
System-on-Chip (SoC)  
Electronic System Level (ESL) design

## ABSTRACT

This research discusses hardware architectures, script-based automation and software and hardware methodologies for developing customized System-on-Chip scalar/vector processors within the example application domain of telephony codes. The approaches researched include Register-Transfer-Level methodologies resulting in an SIMD-enhanced processor known as the ITU-VE1, and Electronic System Level methodologies resulting in a multi-parallel vector processor known as the SS\_SPARC. The example applications were the ITU-T G.729A and G.723.1 speech codecs chosen for their abundant data-level parallelism and availability for research purposes. Results indicate the proposed scalar/vector accelerators achieve a maximum speed-up of 4.27 and 4.62 for the G729.A and G723.1 encoders respectively for 512-bit wide SIMD configurations. Both vector processors resulting from the proposed methodologies were implemented as VLSI macros and compared at the silicon level. Compared to the Register-Transfer-Level flow, the Electronic System Level flow implementing the same datapath results in increased power consumption of 3–15% however delivers an area reduction of 2–18% and substantially shortens design and verification time making it a viable alternative to established RTL methodologies.

© 2013 Elsevier B.V. All rights reserved.

## 1. Introduction and motivation

Embedded processor cores with a fixed instruction set architecture (ISA) are widely used in the design of System-on-Chip (SoC) embedded systems. Such architectures present a good compromise for the execution of general-purpose code, such as user interfaces, protocol processing and embedded operating systems (OSs). However, they lack the necessary processing power for satisfying the digital signal processing (DSP) requirements of many of the core algorithms prevalent in modern consumer electronics or telecommunication applications. To address this shortfall in performance capability, architects have implemented embedded DSP engines operating in parallel with the main scalar processor to accelerate performance-critical components of the application [1]. The drawback of this approach is that additional silicon area is required and the more complex programming model that is needed and which involves multiple address spaces, additional ISA support, scatter-gather direct-memory access dataflows and ‘mailbox-type’ communications [2]. A potential solution to these issues is the hardwired implementation of the core DSP code of the target

application, but this involves the development and validation of parallel code at the register transfer level (RTL) and results in solutions that are specific to the task at hand and offer little or no flexibility. This lack of adaptability represents a serious deficiency in the consumer/telecommunications marketplace, where short design cycles are needed to adapt to ever-evolving standards and fast augmentation with the latest features is vital to remain competitive.

In the last few years, a new type of embedded central processing unit (CPU) has emerged, namely the configurable, extensible processors which allow the system architect to extend their architecture (programmers model and ISA) and microarchitecture (execution units, streaming engines, coprocessors, local memories) [3]. In telecommunications applications, these CPUs provide the specific benefit of allowing custom ISA and execution/storage resources to be defined, thus giving the advantage of post-fabrication adaptability to evolving standards. The ability to synthesize logic to execute instructions customized to a particular application domain results not only in a faster execution time, but the consequent reduction in dynamic instruction count for the target application (combined with the use of streaming local memories rather than data caches) which reduces power consumption. Such CPUs have been used in various diverse applications including hybrid shared memory/message-passing multiprocessors

\* Corresponding author. Tel.: +44 1509227113.

E-mail address: [v.a.chouliaras@lboro.ac.uk](mailto:v.a.chouliaras@lboro.ac.uk) (V.A. Chouliaras).

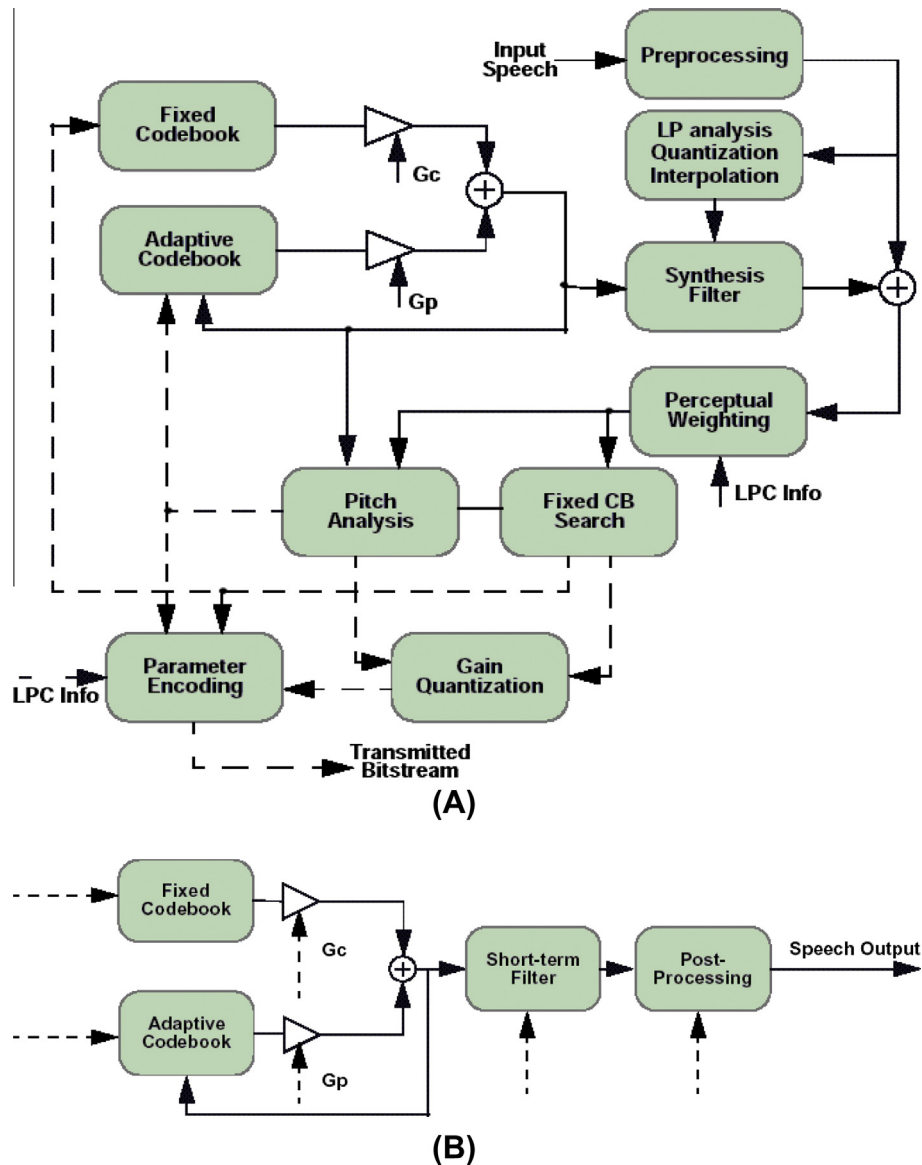


Fig. 1. G729.A signal flow diagram for (A) encoder and (B) decoder.

[4], Software Defined Radio (SDR) [5], video coding [6,7], telephony [8], audio processing [9] and real-time operating system acceleration [10].

The research described in this paper investigates in detail methodologies for vector processor architectures tightly integrated to configurable and extensible System-on-Chip (SoC) processors and targeting data-parallel workloads. Two such methodologies were studied, the first based on established RTL descriptions known resulting in the ITU-VE1 processor and the second, making use of a novel ESL design flow to implement the same data-parallel accelerators at a fraction of the design time while making use of a more potent, superscalar processor known as the SS\_SPARC. In this study, we selected as example workloads the ITU-T G.729A and G.723.1 speech codecs which exhibit substantial data-level parallelism however, our findings are directly relevant to all applications exhibiting similar levels of data-independent computations. Although current desktop CPUs can sustain the real-time execution of these example workloads, the target platform of the current research is mobile as both the ITU-VE1 and SS\_SPARC are SoC processors. Also, state-of-the-art, highly parallel, very long instruction word (VLIW) DSPs can sustain the real-time execution of these codecs, but such engines are both silicon and power expensive [11]. Further, the workloads executing on these VLIW DSPs are highly

optimized, commercially protected making it practically impossible for researchers to evaluate them on any other architecture or to perform microarchitecture research.

The main motivating factors behind our research were (a) to develop the vector accelerators (datapaths and load/store infrastructure) that improve substantially the real-time performance of the reference workloads and (b) To study two methodologies for the development of such accelerators namely, RTL and ESL. In the first case, this research resulted in the RTL-designed ITU-VE1 processor which efficiently utilizes the main scalar CPU of a typical telecoms application-specific integrated circuit (ASIC) as the driving engine for a highly customizable and parameterizable vector engine. For the ITU-VE1, cycle-accurate simulation results demonstrate speed-ups of the order of 4.5 (in terms of cycles) compared to the same code executing on an un-accelerated baseline embedded processor architecture thus making the RTL-based architecture a valid implementation alternative to embedded VLIW processors [11]. Following the development of ITU-VE1, we identified automation methodologies and microarchitectural improvements which led to the development and customization of a potent superscalar simultaneous-multithreaded processor known as the SS\_SPARC. In this case, much of the infrastructure developed around the

Download English Version:

<https://daneshyari.com/en/article/463025>

Download Persian Version:

<https://daneshyari.com/article/463025>

[Daneshyari.com](https://daneshyari.com)