A STRUCTURAL APPROACH FOR DESIGNING PERFORMANCE ENHANCED SIGNAL PROCESSORS: A 1-MIPS GSM FULLRATE VOCODER CASE STUDY

Matthias H. Weiss, Ulrich Walther, and Gerhard P. Fettweis

Mobile Communications Systems
Dresden University of Technology, 01062 Dresden, Germany
{weissm,walther,fettweis}@ifn.et.tu-dresden.de

ABSTRACT

Recent performance enhanced DSP (Digital Signal Processor) architectures incorporate either datapath add-ons such as dual-MAC architectures or tailored datapaths such as Viterbi accelerators. Both strategies strongly influence the instruction set architecture (ISA). Since common ISAs are not designed for architectural enhancements, either a complete redesign is required or architectural enhancements cannot be fully exploited by the ISA.
Taking the GSM Fullrate Vocoder in this paper a structural approach is presented to how datapath add-ons or tailorizations can be applied to increase DSP's performance. To efficiently utilize architectural enhancements we propose a modified VLIW (very long instruction word) ISA, called TVLIW (tagged VLIW). TVLIW combines both VLIW performance and DSP codewidth requirements. To demonstrate the applicability, we applied the TVLIW ISA to a highly pipelined quadruple-MAC architecture, incorporating only one dualport RAM and a 26-bit wide instruction word.

1. INTRODUCTION

The ongoing advances in semiconductor technology are the enabler for more powerful processors at smaller power consumption. However, shrinking transistor dimensions strongly influence a processor's architecture. While increased number of transistors on a chip can be exploited by parallel processing, pipelined architectures allow faster clock rates. Along with this an increasing gap between memory access time and gate latency can be observed.
In the development of microprocessor architectures this is already taken into account, e.g. by implementing superscalar architectures and complex memory hierarchies.
DSP architectures on the other hand support inherent parallelism, e.g. by allowing concurrent memory accesses and arithmetic calculations. However, to efficiently support new DSP algorithms more processing power must be accomplished by incorporating both parallelism and pipelining into architecture design while avoiding an increase in memory bandwidth for supporting low-power and cost effective designs [1].
Thus, a common strategy is to replicate arithmetic units or implement tailored datapaths. Both methods strongly effect the instruction set which ensures DSP's programmability and flexibility. Furthermore, support of both pipelining and parallelism is mainly determined by the instruction set. The organization of this instruction set is referred to as instruction set architecture (ISA).
To illustrate current strategies for implementing new datapaths and how they affect the ISA two recent DSP architectures are chosen: the C54x Lead-DSP by TI [2] and the DSP1618 by Lucent [3], former AT&T, which both incorporate Viterbi accelerators.
A tailored add compare select (ACS) datapath was added to the former C5x architecture by TI. Furthermore, MAC was separated from ALU. These architectural changes lead to a complete new instruction set, requiring a new instruction decoder, and thus a complete redesign of the processor, yielding the C54x.
Lucent on the other hand improved the DSP1600's architecture in two different ways by not changing the ISA and thus allowing code compatibility. For Viterbi acceleration they choose a coprocessor approach which limits the DSP's flexibility. Furthermore, a Bit Manipulation Unit (BMU) was added to the DSP1600 core. Due to limitations in the ISA the BMU cannot be employed concurrently with other parts of the processor, which limits performance. Thus, current performance enhanced DSPs either require a complete processor redesign as in the TI case or architectural improvements cannot be fully exploited by the ISA.
This paper describes, how datapath add-ons can be implemented into architecture by avoiding ISA redesign

This work was sponsored in part by the Deutsche Forschungsgemeinschaft (DFG) within the Sonderforschungsbereich (SFB) 358.

at no performance loss. To show the applicability, in Section 2 we analyze the RPE-LTP speech coding algorithm used in the GSM vocoder for choosing the appropriate tailored or replicated datapaths. In Section 3, a structural approach for designing the suitable ISA is given. In Section 4, results for the selected algorithm are presented.

2. ALGORITHM-ARCHITECTURE INTERACTION

To demonstrate how algorithm structures can be exploited by architectural enhancements we chose the RPE-LTP speech coding algorithm used in the GSM vocoder [4] for two reasons. First, it contains typical signal processing algorithms such as filters of different lengths, auto- and cross-correlations, lattice IIR and FIR, etc. Furthermore, it is a widely accepted benchmark algorithm.

A. Analyzing Algorithms

At the first stage we analyzed how different datapath add-ons can be employed by the overall GSM full-rate algorithm. Starting with a basic architecture we computed a cycle-count (measured in Million Instructions Per Second (MIPS)). This cycle-count approximately corresponds to cycle-counts found in current high-end DSP solutions such as TCSI's Lode and TI's Lead C54x. Based on this cycle-count in table 1 the MIPS requirement of the major algorithms are shown.

Algorithm	appr. MIPS Requirement	
	1-Mac-Arch.	4-MAC-Arch.
Lattice-FIR	0.27	0,054
Lattice-IIR	0.16	0,081
Cross-Correlation	0.72	0,222
FIR	0.13	0,046
Remainder	0.52	0,44

Table 1: Major Algorithms in GSM-Fullrate

To increase the performance we added datapath extensions to the architecture as shown in Fig. 1. Starting with the cycle-count of the basic architecture we estimated the impact of adding specialized units for input-scaling, output-scaling, and division, a tailored MAC unit computing a 31 by 16-bit multiplication, and one or three additional MAC units. We found specialized scaling-units not being able to improve the overall performance. The division-operation, though very expensive in terms of cycle-counts, is only moderately used in the algorithm [5]. Implementing a 31 by 16 MAC functionality can save approximately 6% of the MIPS and can be folded into a existing MAC unit to save cost. Implementing a dual-MAC reduces the MIPS count significantly by 30%. Since dual-MAC solutions already exist, e.g. by [7] or by [6], we took a step beyond by implementing a quadruple-MAC which can save 47% of the MIPS count.

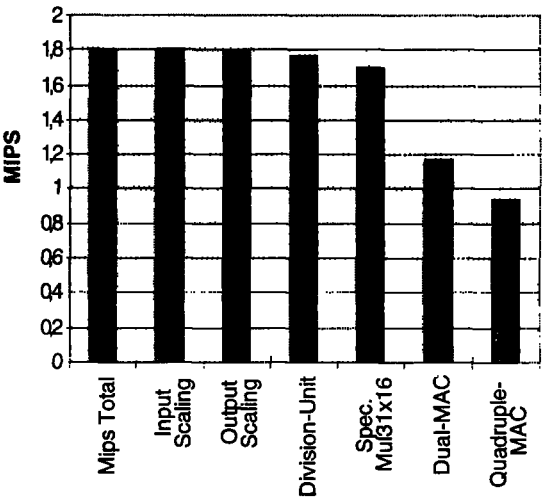

Fig. 1. MIPS reduction by different datapath add-ons

B. Architecture

To be comparable with current DSP solutions, we implemented an architecture with a dual-ported RAM. Fig. 2 shows the blockdiagram of the final datapath. Analyzing particularly the major algorithms (table 1) of the GSM-Fullrate we found two memory-accesses per cycle to be sufficient to feed up to five arithmetic units with data. In order to exploit special properties found in digital signal processing algorithms we implemented a 9-entry ring-buffer, which both reduces memory I/O operations and provides the necessary input delay required in DSP algorithms [7].
For independent usage we separated ALU from MAC units, e.g. to perform scaling and auto-correlation in a pipelined fashion - scaling input-values before performing multiplications - and thus employing both units in parallel. Furthermore, the 31 by 16 multiplication functionality was added to one MAC unit. Finally, our architecture incorporates a 16 entry program cache, a pipelined data memory and a pipelined instruction decoder, resulting in a instruction pipeline of variable length.

3. STRUCTURAL APPROACH FOR ISA DESIGN

Controlling the pipeline mainly incorporates resolving hazards. In a data-stationary CISC ISA pipeline hazards must be resolved by hardware [6]. Thus, if the instruction set is modified, both the instruction decoder and the hazard resolving hardware must be changed. To allow the programer or compiler to control the pipeline, time-stationary ISAs are applied as in Lucent's DSP1618.

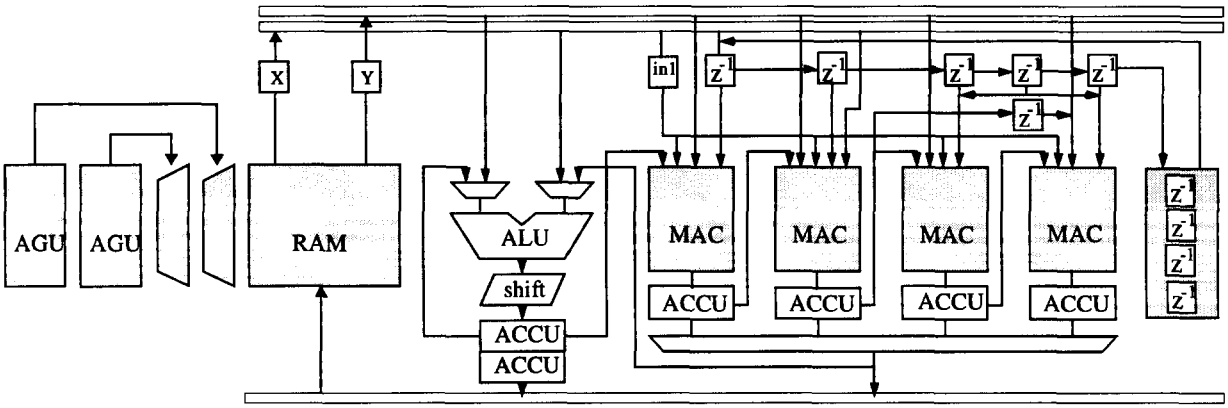

Fig. 2. Processor architecture

A. ISA-Architecture Interaction

As mentioned above, architectural enhancement can be achieved either by adding a new datapath, such as the ACS unit, or by adding new functionality to an existing datapath, e.g. by folding the 31 by 16 multiplication or Galois Field Arithmetic into an existing MAC [9]. If the datapath should be controlled independently, the time-stationary instruction word must be expanded in order to orthogonalize the instruction set. This leads to the very long instruction word (VLIW) ISA.

The VLIW ISA is already recognized as a good candidate for high performance processors, e.g. used by the Phillips TriMedia processor [10]. However, VLIW ISA comprises two main disadvantages. First, modifying the hardware architecture requires a change in the instruction word. Thus, the original VLIW ISA is not code compatible, which is a typical disadvantage against a CISC ISA. However, code compatibility achieved by the development from TI's C1x and C2x to C5x could not be attained by developing the C54x.

The second disadvantage of VLIW is code size explosion. Several methods for limiting code size have been suggested [10], [11]. For usage in DSP we proposed a Tagged VLIW (TVLIW) scheme [6].

B. TVLIW-ISA

By classifying DSP code into in-line and in-loop code different behavior can be observed. While in-loop code requires the full VLIW's functionality, in-line code, although requiring the biggest part of the program memory, cannot exploit parallelism. Thus, TVLIW supports different requirements of in-line and in-loop instructions by assembling the actual instruction word dynamically. A very long instruction word (VLIW) consists of a number of functional unit instruction words (FIW). Each FIW controls the associated function unit (FU) independently from the remaining FIWs. The idea of the TVLIW scheme is to assemble the actual VLIW out of limited number of FIWs. If the full VLIW's functionality is required, this assembling may require several cycles. However, these instructions mainly occur within loops. With the help of a loop cache, this overhead is only necessary during the first iteration. With the help of this TVLIW scheme, an existing architecture can be expanded by datapath add-ons without requiring an ISA redesign.

4. RESULTS

Implementing the GSM full-rate algorithm we focused on both the effect of implementing the TVLIW scheme in comparison to a conventional VLIW approach and performance results applying a quadruple-MAC architecture.

A. Instruction Set Architecture

For constructing the ISA by the TVLIW scheme we chose 13 classes of function units (table 2), each requiring an 8 bit wide instruction word, resulting in a 26 bit TVLIW instruction word.In contrast, the DSP design given in [5] requires a 42 bit instruction word controlling a less parallel hardware. For coding the complete

ISA Groups	ISA FUs	#
Program Control Instruction Words	• Program Control, • Loop, • Short Immediate, • Long Immediate.	4
Arithmetic/Logic Instruction Words	• ALU, • MAC (4 x).	1 + 4
Load/Store Instruction Words	• Load/Store (2 x), • Address Generation (2 x).	2 + 2
Total number of ISA FUs		13.

Table 2: Functional Units of Instruction Word

algorithm, consisting of both the encoder and decoder

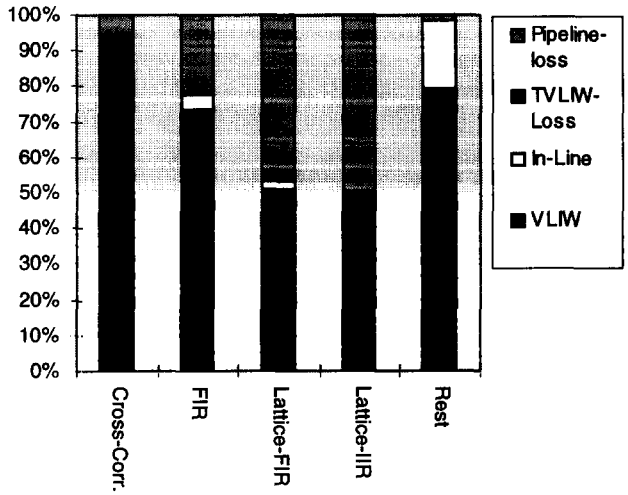

Fig. 3. Cycle distribution of major algorithms

part, less than a 1 K-word program memory was required such as in the DSP 56301 implementation, but coding an orthogonal instruction set. The algorithm was implemented on a C++ simulator allowing us to model the parallelism of a TVLIW architecture.

B. Performance Results

In spite of the small 16-entry program cache which often holds inner loops only, we measured a 90%-usage for the complete algorithm. In the remaining 10% performance-loss of TVLIW is neglegable, since either a limited number of functional units is required or the pipeline is filled or emptied. This is shown in Fig. 3 for the main algorithms from table 1 in detail. Due to a special mechanism of TVLIW cycles for filling or emptying the pipeline can be used for reading additional parts of the actual VLIW from the program memory indicated by the light grey area. Once the pipeline is filled the whole VLIW performance can be exploited indicated by the dark grey area. The actual cycles spent for assembling the VLIW word - indicated by the black bars - are almost neglegable. Applying a quadruple-MAC solution the MIPS requirement of the major algorithms can be cut almost by 4 as shown in table 1.

By the given architecture and ISA we could reduce the GSM 6.10 full-rate encoder and decoder to 0,85 MIPS which is 5 times faster than the MIPS requirements of a standard DSP such as Motorola's DSP65301 [12].

5. CONCLUSIONS AND FUTURE WORK

Taking the GSM full-rate algorithm we showed a structural approach for designing performance enhanced signal processors. By adding datapath add-ons the proposed architecture requires less than one MIPS, cutting the MIPS count of a basic architecture by 2 and of a standard DSP by 5. However, the main intention was not to outperform standard DSPs, but to show how hardware extensions such as a quadruple-MAC solution can be added to a standard architecture without requiring a complete redesign. Thus, by applying the proposed method, adding a datapath does not influence the overall architecture and ISA. This can be achieved by applying the tagged VLIW (TVLIW) scheme.

In future work we will further analyze the DSP architecture in order to get detailed insight into gate counts, cost of datapath extensions, possible clock rates etc. Furthermore, compiler methods which already exist will be applied to avoid assembly coding.

REFERENCES

[1] G. Fettweis, „DSPs for Mobile Communications: Where are we going?", *this conference*

[2] Texas Instruments, *TMS320C54x User's Manual, Preliminary Edition*, 1995

[3] Lucent Technologies, *DSP1611,DSP1617, DSP1618,DSP1627, Information Manual*, 1996.

[4] Recommendation GSM 06.10, July 5, 1989

[5] J. Nurmi et al., „A DSP core for speech coding applications", *Proc. ICASSP-94*, vol.2, pp. II/ 429-432

[6] M. Weiss and G. Fettweis, „Dynamic Code-width Reduction for VLIW Instruction Set Architectures in Digital Signal Processors", *3rd International Workshop on Image and Signal Processing, 1996,* pp. 517-520

[7] G. Fettweis et al., „Strategies in A Cost-Effective Implementation of The PDC Half-Rate Codec for Wireless Communications", *VTC '96*, vol. 1, pp. 203-207

[8] T. Shiraishi et el., „A 1.8V 36 mW DSP for the Half-Rate Speech CODEC", *IEEE 1996 Custom Integrated Circuits Conference,* pp. 371- 374

[9] W. Drescher and G.Fettweis, „VLSI Architecture for Multiplication in GF(2m) for Application Tailored Signal Processors", *1996 IEEE Workshop on VLSI Digital Signal Processing,*, pp. 55-64

[10] P. Clarke, „Compressed VLIW meets multimedia [microprocessors]", *Electronic Engineering*, Vol. 67, pp. 45-47, 1996

[11] H. Miyajima, K. Murakami, Y. Saitoh, and T. Hironaka, „Hyperscalar processor architecture and the preliminary performance evaluation", *Internal Technical Report, Kyushu University,* May 1995

[12] L. McLuckie and C. Cox, „GSM Full- and Half-Rate Speech Codec on The DSP56301", *DSP Deutschland 95*, pp. 75-84