

DSP Cores for Mobile Communications: Where are we going?

Gerhard Fettweis

Mobile Communications Systems

Dresden University of Technology, D-01062 Dresden

email: fettweis@ifn.et.tu-dresden.de

ABSTRACT

Digital signal processors (DSPs) have become a key component for the design of communications ICs. Application customization leads to key market advantages but also to enormous problems of having too many different DSPs and their software development tools. First, by analysis of the problem open issues are pointed out. Then, a possible solution named CATS is presented, which allows for customization without the generation of too much heterogeneity in hardware and tools.

1. INTRODUCTION

The importance of digital signal processors (DSPs) for communications, and in particular mobile communications, has been ever increasing. Today DSPs present a key technology for executing baseband modem and lower layer protocol functions.

Historically DSPs were designed around one multiplier as stand-alone integrated circuits (ICs). In the light of VLSI technology the processing power and complexity of DSPs has been increasing to today's levels. Very importantly, implementation of DSPs as embedded cores including other logic functions on the same die has become feasible. This has led to a new situation for semiconductor manufacturers. As DSPs were designed into systems as stand-alone ICs, a semiconductor manufacturer was able to design many products, leaving the DSP IC to be supplied by another source. Due to the embedded integration, non-DSP manufacturers see themselves faced with either loosing their market or having to own a DSP as well. This situation led to new DSP-core design startups providing DSP cores to the market. Hence, today embedded DSPs have been widely adopted and are becoming mainstream.

In the future, however, the market can evolve even further. Large customers of embedded DSP ICs, as "tier one" mobile terminal equipment manufacturers, today need to have ASIC design expertise to define the custom logic around the embedded DSP. This way they can ensure a proprietary solution with a competitive advantage. In future a need to have a direct access to a propri-

etary DSP architecture within the ASIC design environment could evolve. This will take place if and only if owning proprietary DSP hardware architecture advantages has a direct impact on market advantages of the end product. Otherwise, owning software and ASIC functions will suffice.

Hence,

- foundries need access to DSP hardware technology today,
- equipment manufacturers may need to own DSP hardware technology tomorrow.

This paper shall first provide a brief understanding of DSP technology. Following, the impact of DSP architecture technology on the competitive standing is highlighted, leading to many open questions. Finally, our research project at Dresden University named CATS (Concept for Application Tailored Signal Processors) will be presented as a possible solution.

2. MOTIVATION

2.1 Achieving a Competitive Advantage

The communications market is very dynamic and has a high growth rate. Hence DSPs for communications must evolve to continue being a platform for achieving and sustaining a competitive standing. How can this be achieved?

The performance of DSPs is evolving further by advances in semiconductor technology. This leads e.g. to higher clock frequencies as well as a reduced power consumption per MIPS. Additional performance improvements can be gained by the development of new DSP architectures, where performance is measurable by a reduced MIPS requirement per algorithm (improved efficiency), reduced power consumption, or allowable higher clock frequency.

Riding on advances in semiconductor technology alone for achieving a competitive advantage can be extremely dangerous. Therefore, architecture technology is a key.

2.2 How to Get Ahold of DSP Technology?

Typical money maker ICs have gained a competitive advantage by sustaining a technical and/or marketing

advantage. A technical advantage as

- power consumption
- die size/cost
- performance
- package, I/O, chip-set integration

is achieved by combined *architecture-application* optimization. How can this be achieved for DSPs?

As mentioned in section 1, a non-DSP semiconductor manufacturer can licence a core from a core provider. However, it is difficult to gain a competitive edge based on a publicly available licensed core, possibly even running licensed software. In this case the manufacturer becomes a pure wafer fab.

Hence, this leads to today's licensing dilemma:

- Licensing fixed DSP cores is no long-term solution. It can be a short-term solution. In the long-term the manufacturer is at the mercy of the core provider's architecture technology advancement. And, no proprietary competitive advantage can be gained.
- However, easy and fast access to software modules is desirable, for which standard accessible cores are superior platforms.

2.3 What is Driving DSP Technology?

To analyze what is driving DSP architecture technology, one must briefly recall the key factors that come into play for any application to become a technology driver:

- It needs to have a high growth market,
- it needs to have a substantial market volume today,
- it requires to push technology limits (clock, die size, power consumption, packaging)

Clearly, communications is the DSP technology driver today. Mobile communications DSPs in particular are becoming a semiconductor technology driver due to high performance as well as stringent power consumption and packaging constraints. To understand the DSP market, it therefore is very important to analyze DSPs for (mobile) communications in further detail.

2.4 What Kind of DSPs are Needed?

There has been discussion on DSPs versus microprocessors. This was mainly based on general purpose floating-point DSPs. Actually, DSPs cover a very wide range of architectural customizing for applications. We can divide DSPs into three general classes, i.e.

- application specific DSP (AS-DSP)
- domain specific DSP (DS-DSP)
- general purpose DSP (GP-DSP).

Following, we refer to a circuit being a DSP only if it is software programmable by an assembly language. DSPs as defined e.g. in [1] we call datapath processors.

AS-DSPs are typically customized to an application to serve high-end application performance requirements, or to minimize die size/cost. Generally the market volume must allow for a custom solution to be developed, and customizing is carried out to gain market advantages. However, time-to-market constraints must allow for a long design cycle. Examples of AS-DSPs can be found

e.g. for speech coding [2,3]. Application customizing can be found in the datapath, address generation, bus architecture, memory, and I/O.

DS-DSPs are targeted to a wider application domain, as cellular modems (TI C540, TCSI Lode). They can be applied to a variety of applications, however they were designed "with a target application in mind". Due to special instructions and additional hardware they can run domain specific algorithms efficiently (e.g. Viterbi algorithm and equalizers). A DS-DSP is designed for a market with a volume high enough to allow specialized solutions. Its main advantage over an AS-DSP is its fast availability, and access to a small software library base. *GP-DSPs* have evolved from the classic FFT/filtering multiply-accumulate design paradigm. Examples are TI C50, Lucent 16xx, Motorola 563xx, ADI 21xxx, and DSP-Semi's Oak/Pine. GP-DSPs are readily available, are widely applicable, and have a large software base. However, they lack in performance when compared to more customized solutions for specific applications.

2.5 Architecture Development Technology Flow

To show a path of architecture technology evolution the new multi-MAC DSP trend shall be analyzed.

In the early 90s the Japanese digital cellular (PDC) half-rate speech coder was standardized [4]. It was too complex to run on available DSPs. Hence, NTT designed a SIMD-DSP (single instruction multiple data) as an implementation test-bench based on a classical GP-DSP architecture with 2 parallel MAC datapaths. This resulted in a very large DSP of 200 mm² size in 0.8μ CMOS [3]. Targeting the same application, at TCSI together with AKM we designed a classical path of how the first DSP with an integrated dual-MAC. It is an AS-DSP and extremely small (70 mm² in 0.7μ CMOS, including the A/D audio codec). This clearly shows the power of application specific customizing.

The basic dual-MAC datapath idea was generalized by the author, which led to TCSI's Lode DS-DSP core [5], targeted for cellular phone modem and speech coding. Currently TI and Lucent have picked-up the trend and are designing multi-MAC GP-DSPs [6,7].

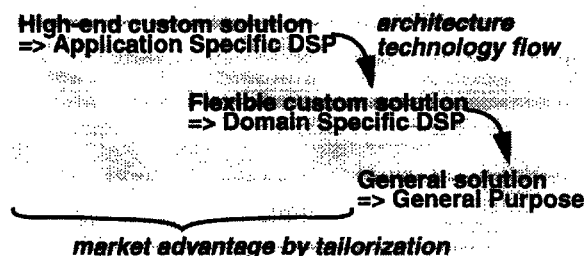


Fig. 1 Architecture technology flow

High-end applications require innovation and application specific customizing to enable the design of solutions. As semiconductor technology evolves, these architecture ideas can be applied to DS-DSPs, and then to GP-DSP architectures, as shown in Fig. 1.

This case study clearly allows drawing two conclusions

- tailorization of DSPs enables key market advantages
- DSP technology is *not one core* only.

3. The Software-Hardware Nightmare

Today's mobile communications equipment, as cellular phones, comprise several functional units which require signal processing tasks

- baseband modem (a typical DSP task)
- speech codec (a typical DSP task)
- protocol and control unit (on a microcontroller)

In the previous section we learned that tailorization leads to key advantages. This, however leads to a heterogeneous design with 3 different processor platforms, one for each area of customizing. The resulting problem for the communications equipment designer is to maintain and develop software on 3 different incompatible platforms. Adding future functional units can worsen this problem, see Fig. 2. Semiconductor manufacturers on their side need to maintain and update multiple processor platforms.

This problem we call the *software-hardware nightmare*.

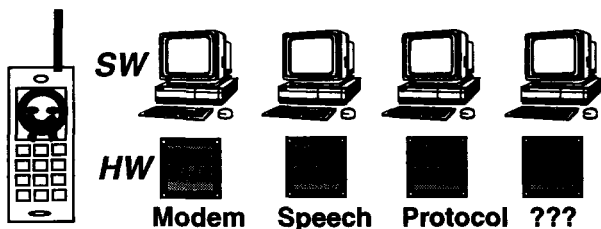


Fig. 2 Software-hardware nightmare

Semiconductor as well as equipment manufacturers need one integrated library based DSP family design approach, as sketched out in Fig. 3. Using the same software design tools as well as library based reuse of hardware units in an integrated DSP design system is key to solving the heterogeneous software-hardware nightmare. Lucent addresses this problem by allowing customized acceleration units to be added to the DSP 16xx core [8] (and has announced customizing flexibility for its new Sabre DSP [7]). This approach, however, only allows for customized add-ons that have no direct communication with the memory. Large performance gains by customizing require direct memory access, as a galois-field datapath for error correction coding [9].

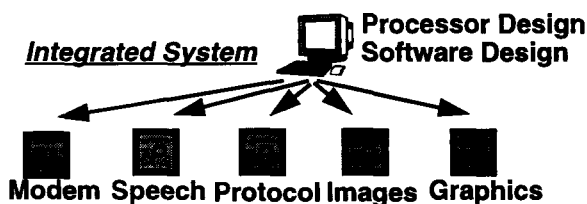


Fig. 3 Integrated hardware/software DSP design

The baseline of the software-hardware nightmare is: We need a solution as sketched in Fig. 3: the tailorization of DSPs within an integrated DSP architecture family, programmable with *one* set of software design tools.

4. CATS: Concept for Application Tailored Signal Processors

The Concept for Application Tailored Signal processors (CATS) is a research project under design at the Mobile Communications Chair in Dresden. The goals are

- One integrated computer assisted processor development platform for achieving application tailored processors with customized execution acceleration as well as general purpose signal processors based on one processor architecture family.
- One generic software development platform (assembler, debugger, compiler) which is independent of the application specific tailorization of the processor.
- A real-time development/debugging environment which can be delivered to the customer with a new application tailored processor within weeks after receiving the processing specification.
- An extendable library-based hardware and software design to allow easy maintenance and technology migration as well as debugging and testing.

4.1 Instruction Set Architecture Goals

To achieve the CATS design goals, the interaction and co-design of hardware/architecture and software/algorithms must be carried out. The operational functionality during task execution on a DSP can be orthogonalized/divided into data transfer and data manipulation. Data transfer includes move/load operations and transfers to/from the PC (program counter), i.e. call, return, branch, etc. It is clear that data transfer is very similar for any tailored DSP, and therefore can be standardized. Arithmetic/logic operations need further analysis. Following, three different equations are given as examples for this class of operations.

$$y_k = \sum_i |a_i - x_{k-i}|$$

$$y_k = \sum_i a_i x_{k-i}$$

$$y_k = \max_i (a_i + x_{k-i})$$

All three equations perform convolution operations on data vectors, where the access and order of use of data is identical, i.e. the data transfer. However, the executed arithmetic, i.e. data manipulation, is not. Hence, also the arithmetic/logic operations can be classified according to data transfer and data manipulation independently. Assuming that algorithm specific differentiation lies more in data manipulation than in data transfer, data transfer hardware can be kept identical for different application tailored DSPs. This leads to the following:

- Algorithm specialization must only be carried out in the arithmetic datapath where data manipulation is carried out, and in memory bandwidth.
- All units of a DSP concerning data transfer can be designed independently of the specific algorithm (memory management, program control, memory, buses). Different requirements of memory bandwidth

will require a classification of data transfer and yield a family of bus architectures.

For the instruction set architecture (ISA) to support this classification efficiently, the instruction word is divided into two separate fields or words, one for data transfer and one for data manipulation, respectively.

An example for this classification is RISC, as shown below in Fig. 4.

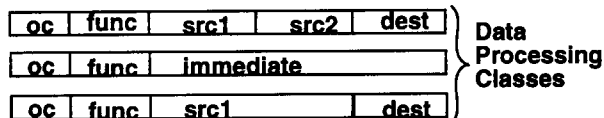


Fig. 4 RISC instruction set example

In a RISC ISA the function field defines the data manipulation, and data transfers are explicitly defined as sources, destinations, and immediate values.

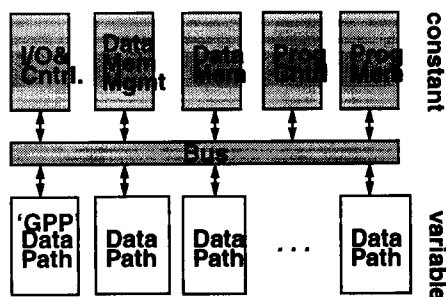


Fig. 5 CATS DSP hardware approach

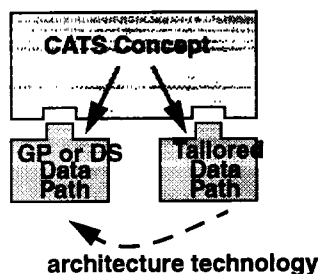


Fig. 6 Architecture technology flow

4.2 Results used in CATS

Now we only need to recognize that a DSP architecture is defined by enabling data transfers (its program control unit, address generation, buses, I/O) except for the datapath. Hence, basing a tailored DSP design (AS-DSP or DS-DSP) on one predefined ISA which partitions data manipulation and data transfer (see e.g. [matt]), allows for designing an integrated concept for the design of application tailored DSPs (CATS). The features can be summarized as follows:

1. Orthogonalization of instructions into data transfer and data manipulation.
2. Data transfer operations include all information necessary for the design of simulators and compilers. Hence, leaving the data transfer classes constant within CATS allows for the design of generic software tools. Data manipulation operations are linked into the tools as separate libraries.

3. Tailorization is mainly given by custom datapaths, and the memory bandwidth, see Fig. 5. Note that a datapath design is very simple compared to the other control intensive units of a DSP.
4. GP-DSPs can be created by adding a GP datapath within the CATS concept as well. Hence, architecture technology developed for custom datapaths can easily be transferred to GP-DSPs within the CATS framework, Fig. 6.
5. All data transfer units of a DSP are kept constant within CATS, and therefore can be implemented on an ASIC. An attached FPGA that supports custom datapath implementations then allows for very fast evaluation board designs.

6. CONCLUSIONS

Access to DSP architecture technology is key for semiconductor manufacturers. Hardware tailorization allows to gain competitive advantages. CATS is a concept which is a step in this direction. It features

- one integrated software development platform,
- extendable library based hardware design,
- fast turn-arounds (datapath design only),
- fast evaluation board turn-around (FPGA based),
- DSP architecture updates instead of new designs.

Hence, it is a possible solution for the licensing dilemma as well as the software-hardware nightmare.

REFERENCES

- [1] IMEC Report 1996.
- [2] G. Fettweis, S. Wang, et al., "Strategies in a cost-effective implementation of the PDC half-rate codec for wireless communications," IEEE 46th Veh. Techn. Conf., Atlanta, USA, pp. 203-7, 1996.
- [3] Y. Okumura, T. Ohya, Y. Miki, T. Miki, "A study of DSP circuits applied to speech codec for digital mobile communications," Proc. of the Fall Meeting of the IEICE, B-294, p.2-294, 1993.
- [4] T. Ohya, H. Suda, T. Miki, "Pitch Synchronous Innovation CELP (PSI-CELP) PDC Half-Rate Speech Codec," Technical Rep. IEICE, RCS93-78, pp. 63-70, Nov. 1993.
- [5] I. Verbauwhede et al, "A low-power DSP engine for wireless communications," VLSI Signal Processing IX, IEEE, eds. W. Burleson et al, pp. 469-78, 1996.
- [6] "DSP battle brews as TI ups Mflops", Electronic Engineering Times, 10-21-96, p. 01.
- [7] "Lucent looks ready for '97", Electronic Engineering Times, 12-09-96, p. 90.
- [8] "Lucent 16xx programmers guide," Lucent.
- [9] W. Drescher et al., "VLSI Architecture for Datapath Integration of Arithmetic over $GF(2^m)$ on Digital Signal Processors," Proc. ICASSP'97.
- [10] M. Weiss, U. Walther, G. Fettweis, "A structural approach for designing performance enhanced signal processors: ...," Proc. ICASSP'97.