

SOFTWARE IMPLEMENTATION OF SHDSL TRANSCEIVERS ON A NOVEL DSP ARCHITECTURE

Manfred Riener, Andreas Bolzer and Gerald Krottendorfer

On Demand Microelectronics, Design Center Techgate
Donau City Str. 1, A-1220 Vienna, Austria (Europe)
phone: +43 1 2697985-67, fax: +43 1 297985-20, email: manfred.riener@ondemand.co.at
web: <http://www.ondemand.co.at>

ABSTRACT

The continuous development in communication technology demands high flexibility and short product cycles. The analysis of the communication standard SHDSL provides computational requirements, which can be used to specify a new type of signal processor. This paper examines the architectural issues for a vector processor, which shall handle communication algorithms as applied to SHDSL. It will be shown that enhancements to the conventional approach of vector computing significantly increases the utilization of scalar algorithms. This makes vector processing a suitable solution for communication applications.

1. INTRODUCTION

The steady progress in semiconductor technology enables the implementation of a state-of-art voice-band modem on a single signal processor core. The increasing importance of broadband access requires digital subscriber line (xDSL) technologies, where the computational load is a multiple of the voice-band technology. Since the processing power of available low-power DSP cores does not provide the required execution speed, high performing xDSL transceivers are implemented as dedicated hardware (HW) solutions.

Although dedicated HW saves chip size and power dissipation, a software (SW) based solution has significant advantages. It offers more flexibility to react faster on upcoming standards and feature enhancements. Bugs in the SW are easily removed, which saves expensive re-designs[1]. This fact increases in importance, because the probability of design errors grows rapidly for complex systems. Using SW solutions, the chip development is simplified because well tested intellectual property (IP) cores are integrated.

The SW development is achieved simultaneously to the HW design, so that both processes are integrated when first prototypes become available. This shortens the time-to-market as well as the turn-around time. Computational resources are re-used, which is especially important for multi-standard modems.

Current general purpose DSPs promise high processing power, but only few signal processors are available as IP core [2, 3]. The available cores suffer from high power dissipation or large chip area in relation to their processing power. The existing solutions do not allow an implementation of xDSL on a single-processor system. Even in multi-processor systems, the latency of the communication link is a hard restriction for the system design and is not competitive to dedicated HW.

Section 2 gives an overview of an SHDSL communication transceiver, following a classification of the used algo-

gorithms in section 3. Based on these considerations, section 4 shows the demands on a vector processing architecture to handle communication algorithms.

2. OVERVIEW SHDSL

SHDSL is a communication standard [4, 5], which allows a symmetric transmission up to 2312 kbit/s payload data rate (*PDR*) and uses a trellis-coded modulation (TCM).

Figure 1 depicts an overview of an SHDSL transceiver. Scrambled transmit data is passed to the TCM encoder, where 3 data bits are encoded into a 4-bit symbol. The symbol rate in kHz is given by $f_{sym} = (PDR + 8)/3$. Thus, the maximum symbol rate is $f_{sym} = (2312 + 8)/3 \approx 776$ kHz. The Tomlinson Precoder will be activated after the startup has been completed and the decision-feedback equalizer (DFE) will be disabled. Note that the Tomlinson Precoder is equivalent to the DFE in the receiver. The transmit (TX) filter is an up-sampling FIR structure, which forms the pulse shape to meet the standardized requirements.

The receive (RX) path consists of an FIR-type RX-filter an IIR-type pulse shorting filter. The echo cancellation, feed-forward equalizer (FFE) and decision-feedback equalizer (DFE) are LMS-adaptive FIR filters. The timing recovery consists of a timing detector and a loop filter. It is used to adjust the sampling phase in the analog front-end (AFE).

The necessary processing power for an SHDSL transceiver is summarized in Table 1. The values in the column *MIPS* are most comparable to a scalar processor, assuming that it can execute 1 multiply-add instructions (i.e. 2 operations) per cycle.

3. CLASSIFICATION OF THE ALGORITHMS

3.1 Vector-Scalar Algorithms

A typical operation which has vectorial operands and a scalar result is the matrix dot product. Many algorithms based on an FIR filter use this operation. The output of an FIR filter is given by

$$y(n) = \mathbf{x}^T \mathbf{h},$$

where the coefficients are

$$\mathbf{h} = [h(0), \dots, h(N-1)]^T,$$

and the input vector

$$\mathbf{x} = [x(n), x(n-1), \dots, x(n-N+1)]^T$$

is a snapshot of the input signal x at the time n .

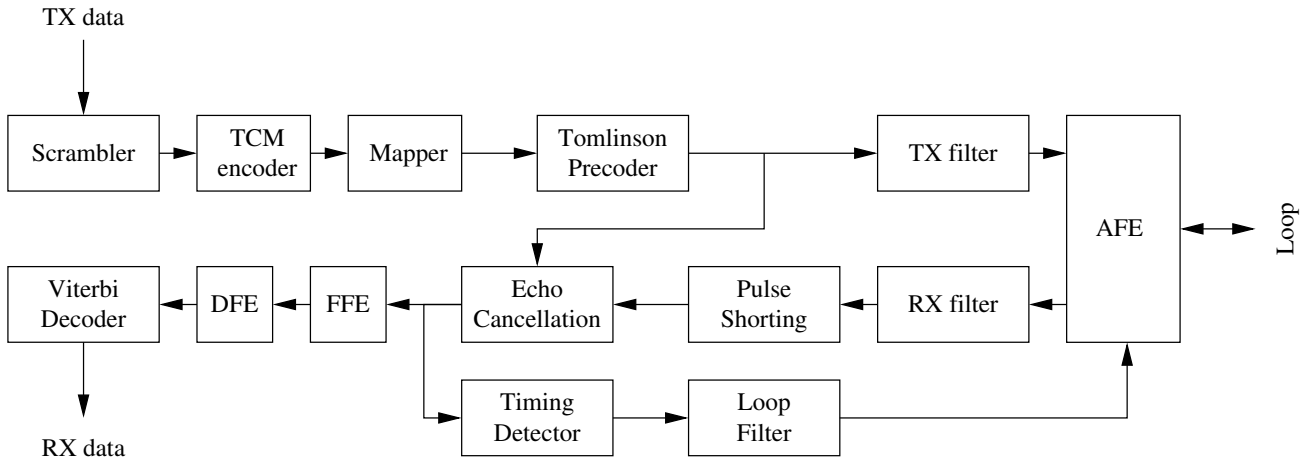


Figure 1: Block diagram of an SHDSL transceiver.

The Tomlinson Precoder, the TX filter, the RX filter, the non-adaptive parts of the Echo Cancellation, the FFE, the DFE and the minimum search of the Viterbi algorithm will use this operation.

3.2 Vector-Vector Algorithms

Many algorithms process vectors and return a vectorial result. Block processing operations (e.g. FFT) are representative applications. A simpler example is the LMS update of FIR filters. The coefficients will be changed according to

$$\mathbf{h} = \mathbf{h}_{old} + e\beta\mathbf{x},$$

where e is the error signal for this iteration and β is a measure for the adaption speed.

For the SHDSL transceiver, the LMS update will be applied to the EC, the FFE and the DFE. The calculation of the Viterbi metric can be also expressed as vector-vector algorithm.

3.3 Scalar Algorithms

Usually, scalar algorithms do not take advantage of a vector architecture. Fortunately, in many cases the algorithm can be split into multiple smaller parts [6]. Hence, the scalar algorithm can be computed in multiple parallel parts.

The mapper, the pulse shorting, the timing detector and the loop filter are examples of scalar algorithms. Each of these algorithms can be executed, simultaneously.

3.4 Bit-oriented Algorithms

The majority of bit-oriented algorithms are processed by the framer. The conversion of the bit-stream into symbols can be seen as a part of the signal processing algorithm. Thus, it can be implemented on a signal processor, although it is implemented more efficiently on framer architectures.

4. MAPPING OF ALGORITHMS TO ARCHITECTURE

Most computational power for SHDSL will be needed for vector arithmetic. Thus, the optimal architecture should be based on a vector processor [7] with enhancements from [6].

Function	Type	MIPS/slice	MIPS
Scrambler	bit	7.0	7.0
TCM encoder	bit	8.0	8.0
Mapper	bit/v.s.	0.5	0.5
Tomlinson Precoder	v.s.	23.3	186.4
TX filter	v.s.	65.2	521.6
RX filter	v.s.	28.0	224.0
Pulse Shorting	s.	15.4	15.4
Timing Detector	s.	15.4	15.4
Loop Filter	s.	15.4	15.4
Echo Cancellation	v.s./v.v.	81.7	653.6
FFE	v.s./v.v.	18.6	148.8
DFE ¹	v.s./v.v.	58.3	466.4
Viterbi Decoder	v.s./v.v.	92.3	738.4
Total performance		429.1	3000.9
Peak performance		370.8	2534.5
Required number of slices			26

Table 1: Processing power of SHDSL

Figure 2 shows an architecture, which is capable to compute vector operations as introduced in section 3.2. The main data path for this kind of computation is emphasized in this figure.

The realization for the second class of vector algorithms as described in section 3.1 is shown in figure 3. The results of all slices are joined into the global arithmetic unit. For matrix dot-products, the global arithmetic adds the partial sums and passes the result to the output.

For the efficient implementation of the scalar algorithms as described in section 3.3, the arithmetic entities (slices) are connected like a chain. Each slice passes its result to the next stage, and afterwards all slices compute their partial algorithms in parallel. Figure 4 shows the configuration of a vector processor for this task.

Bit-oriented algorithms are a special case of scalar algorithms. The architecture may not be used efficiently due to the fact that a single-bit operation does not take advantage from wide data paths. Thus, a general purpose DSP

¹SHDSL uses the DFE only during the startup, while other modules are not operational.

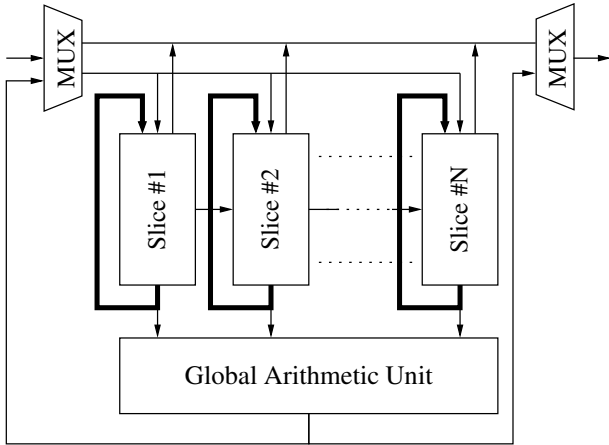


Figure 2: Computing architecture for vector-vector operations.

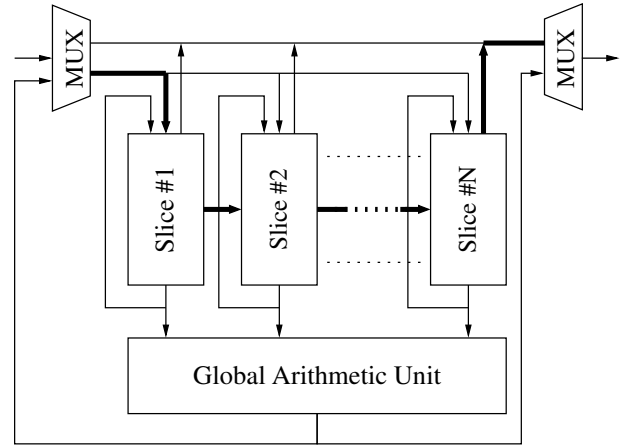


Figure 4: Computing architecture for scalar operations.

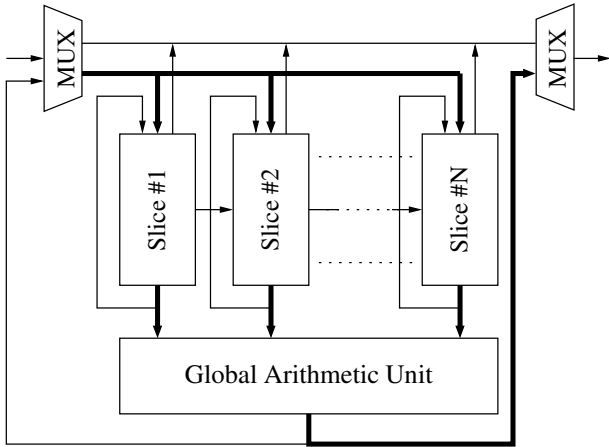


Figure 3: Computing architecture for vector-scalar operations.

would not be the right choice for bit-oriented algorithms and should therefore be implemented in specialized framing processors. In certain cases (e.g. scrambler, TCM encoder), the algorithm can be mapped to scalar bit-field operations, which can then be implemented on the same architecture (Figure 4) used for scalar operations.

Table 1 summarizes the classification of SHDSL algorithms and shows the necessary processing power on a scalar DSP architecture. The MIPS measurement assumes an architecture that can execute 1 multiply-add instruction per cycle. The vector algorithm within this work can be shared on multiple slices and therefore, it reduces the MIPS per slice. Assuming a vector architecture with a capability of 100 MIPS per slice, the number of slices N_{slices} is given by

$$N_{slices} = \frac{\text{Total MIPS}}{\text{Available MIPS per slice}} \quad (1)$$

The appli-ance of Equation 1 to the required processing power of SHDSL (Table 1) reveals that SHDSL needs an architecture with 26 slices.

5. CONCLUSION

The discussion throughout this paper demonstrates that high performance algorithms as used in advanced communication technologies are very suitable to vector processors because of the computing performance and power dissipation. Enhancements to such an architecture lead to a new approach, which increases the utilization while processing scalar algorithms on a vector engine. The scalability of this processor core enables tailored solutions, which are not only limited to communication applications. Suitable applications in the area of high performance computing and real-time systems may be image and video processing.

REFERENCES

- [1] M. Holzer, P. Belanovic, B. Knerr, and M. Rupp, "Design methodology for signal processing in wireless systems," *Informationstagung Mikroelektronik 2003*, Vienna, Oct. 2003.
- [2] B. R. Wiese and J. S. Chow, "Programmable implementations of xDSL transceiver systems," *IEEE Comm. Mag.*, pp. 114–119, May 2000.
- [3] Berkeley Design Technology, *Buyer's Guide to DSP Processors*. 2003.
- [4] ITU, *G.991.2: Single-pair high-speed digital subscriber line (SHDSL) transceivers*. 2001.
- [5] W. Y. Chen, *DSL: Simulation Techniques and Standards Development for Digital Subscriber Line Systems*. Macmillan Technical Publishing, 1998.
- [6] A. Bolzer, G. Krottendorfer, and M. Riener, "A new vector processor architecture for high performance signal processing," in *Proc. European Signal Processing Conf.*, 2004.
- [7] K. Asanovic, *Vector Microprocessors*. PhD thesis, University of California, Berkeley, 1998.