

A NOVEL 32 BIT RISC ARCHITECTURE UNIFYING RISC AND DSP

Christoph Baumhof

Frank Müller

Otto Müller

Manfred Schlett

hyperstone electronics GmbH
Am Seerhein 8, D-78467 Konstanz, Germany
cbaumhof@hyperstone.de

ABSTRACT

A novel 32 bit RISC architecture is presented which is the basis of a powerful general purpose microprocessor and in parallel a 16/32 bit fixed point DSP processor. This unifying of RISC and DSP was not achieved by simply using a microprocessor and DSP core, but a new concept for the implementation of DSP processors has been developed. With the architecture presented it has been proven that a DSP processor can be implemented using strictly the RISC design philosophy. Besides providing basic 16 bit fixed point functionality, the architecture implements a set of DSP instructions that support an efficient mapping of common DSP algorithms to the processor.

1. MOTIVATION

The emerging telecommunication and multimedia markets are creating new demands for embedded control performance. More and more applications require both: a microcontroller and a programmable DSP processor. The first step in implementing such systems was the simple and straightforward utilization of two separate processors: one for the controller tasks and one for the DSP algorithms.

The concept presented here narrows the gap between the microcontroller and DSP worlds by integrating a 32 bit microprocessor and a 16 bit fixed point DSP in a single architecture. Besides the advantage of combining the two worlds, this approach offers a simplified system design and reduces the overall costs. In addition, the RISC philosophy [1, 2] offers a clear roadmap to a higher DSP performance with a minimum of gates required for implementing the DSP functionality [3].

2. RISC-BASED DSP

The new concept that we call “RISC-based DSP” aims to rival classic DSP architectures by enhancing a RISC microprocessor by a DSP unit executing the basic expressions of DSP algorithms [2, 4]. This unit has been optimized for 16 bit fixed point arithmetic but also provides support for 32 bit numbers.

For a successful DSP design, fast execution of the basic DSP expressions alone is not sufficient. Issues like fast loop processing, high data bandwidth and deterministic program flow have to be addressed. Furthermore, the 16 bit data format has to be integrated into the 32 bit architecture in a way that ensures efficient data handling.

2.1. The Basic RISC Architecture

The basic 32 bit RISC architecture provides the following characteristics supporting fast DSP processing:

- simple two stage decode/execute pipeline with single cycle delayed branches
- almost all ALU instructions are single cycle providing fast loop control, address calculation and index updates
- simple 128 byte instruction cache organized as a circular buffer to ensure a deterministic program flow
- pipelined memory access for a high bandwidth and to avoid wait cycles during loading or storing data from external memory to registers and vice versa
- 96 general purpose registers (32 bit) for a high programming flexibility
- register windowing technique for fast interrupt processing and fast parameter passing to subroutines.

A block diagram of the E1-32 processor implementing these characteristics is shown in figure 1.

For the integration of the 16 bit data format into the 32 bit RISC architecture, the 32 bit registers can be split into an upper and a lower 16 bit part. This introduces a new register data format: a 32 bit register can hold two 16 bit numbers. The DSP unit operates directly on the two 16 bit words given in a 32 bit source register. This technique is known as subword processing, see for example [5, 6].

2.2. The DSP unit

The DSP unit enhances the instruction set of the basic RISC architecture by a set of DSP instructions executing the arithmetic part of a DSP algorithm. Supported

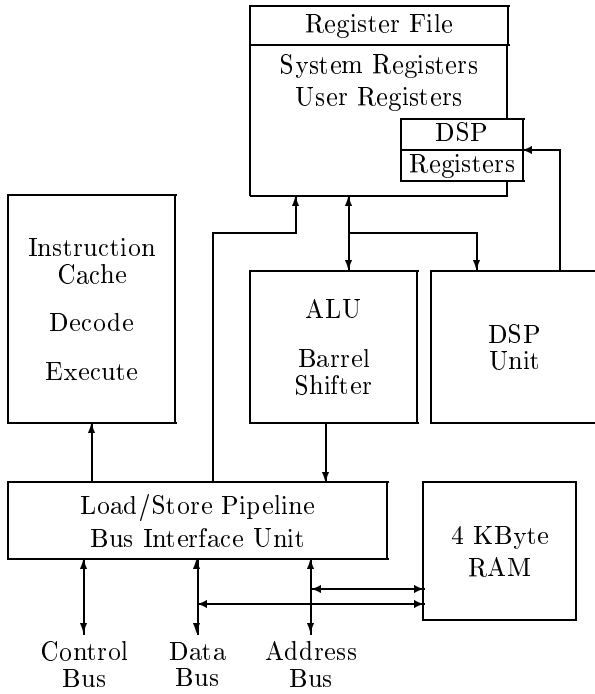


Figure 1. E1-32 block diagram

data types include 16 bit integer, 16 bit fixed point and 16 bit complex fixed point as well as 32 bit integer. The following set of DSP instructions has been implemented:

- multiply (16 and 32 bit)
- multiply-accumulate (16 and 32 bit)
- complex multiply
- complex multiply-accumulate
- add-subtract
- add-subtract with fixed point adjustment.

A block diagram of the DSP unit implemented in the E1-32 processor is shown in figure 2.

The DSP instructions offer up to four latency cycles, but with four operands this means a throughput of one operation per cycle. The latency cycles can be efficiently used for memory accesses and index update and loop control instructions. This three-fold parallelism between ALU, DSP unit and memory access pipeline based on using latency cycles efficiently removes the need to implement superscalar features.

In order to avoid register conflicts when loading new data into the source registers of a preceding DSP instruction, the DSP instruction results are always stored in two dedicated 32 bit DSP result registers. Thus, new operands can be loaded immediately after the issue of a DSP instruction. The two result registers can be addressed by other instructions just like conventional registers. They can be organized as four 16 bit registers, two 32 bit registers or as one 64 bit register. The programmer can use them as a 32 bit or a 64 bit

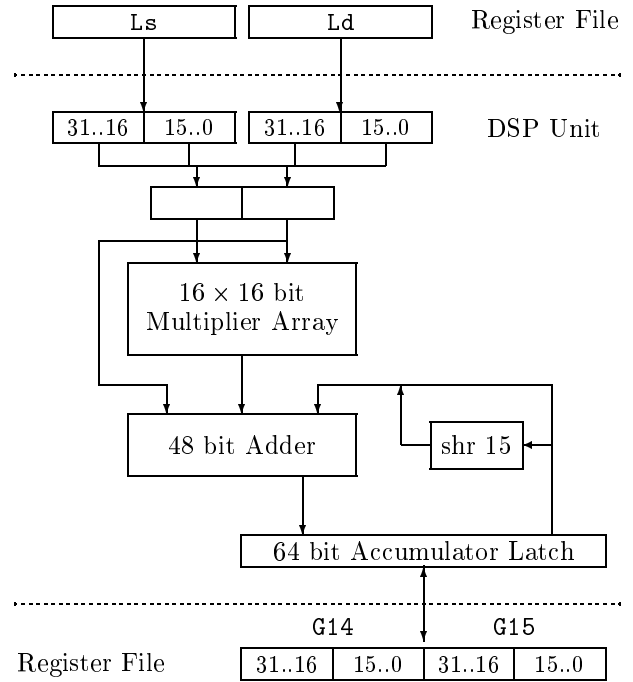


Figure 2. DSP unit block diagram

accumulator in the multiply-accumulate instructions. Longer accumulators can easily be realized in software using general purpose registers.

As the memory access pipeline is capable of loading or storing 32 bits of data per clock cycle, data bottlenecks are avoided. Two 16 bit data words can be loaded in each clock cycle even from external memory. The E1-32 load and store instructions support 8, 16, 32 and 64 bit operation.

Based on the philosophy of keeping the architecture as simple as possible, we implemented overflow handling with a user trap. Programmers can enable the overflow exception handling that traps to a user-specified software routine when an overflow occurs. All types of saturation and exception handling can be programmed.

3. ALGORITHMIC MAPPING

The DSP unit design is based strictly on RISC principles. Thus, in order to achieve a high processing throughput, it is necessary to keep this RISC philosophy in mind when an algorithm is implemented. It is especially important to use the large register file and the memory load/store pipelines efficiently and to make sure that the ALU and DSP unit execute in parallel wherever possible.

As an example, consider the inner loop of a FIR filter computation where the dot product $s_n = \sum_{i=0}^{N-1} c_i x_{n-i}$ of the filter coefficients c_i and the last N filter input values x_{n-i} (filter history) is computed. Figure 3 shows

```

LDW.P  L15, L11 ; L15 coeff ptr, load two coeff.
LDW.P  L7, L12  ; L7 history ptr, load two hist.
MOVD   G14, 0   ; clear G14//G15 accumulator

```

Loop:

```

LDW.P  L15, L13 ; L13 next two coefficients
LDW.P  L7, L14  ; L14 next two history val.
ADDI   L8, -4   ; decrement loop counter
EHMACD L11, L12 ; halfword mult/acc
LDW.P  L15, L11 ; L11 next two coefficients
LDW.P  L7, L12  ; L12 next two history val.
DBGT   Loop    ; delayed branch if L8 > 0
EHMACD L13, L14 ; halfword mult/acc

```

D=decode, E=ALU execute, W=memory access wait
L=load, S=DSP unit execute

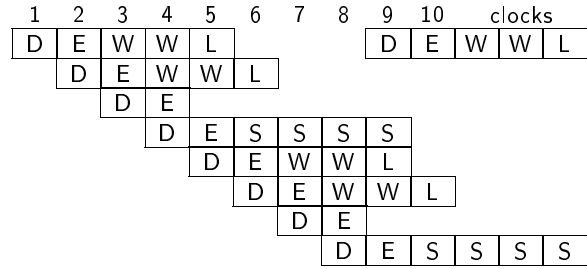


Figure 3. FIR filter inner loop code and pipelined execution

the program code for a sample implementation of this inner dot product loop and a diagram displaying the activity of the various units during the execution of the loop.

The load instructions each fetch two 16 bit numbers in one 32 bit word. These are processed by the EHMACD instruction that performs two 16 bit multiplications and adds the two products into the 64 bit accumulator G14//G15.

The ADDI and DBGT instructions are used for the loop control. In each iteration of the loop, four products are computed and accumulated. Thus, the loop counter is decremented by four. The flags for the branch instruction are set by the decrement instruction. Furthermore, the load/store instructions and the DSP instructions do not affect the condition flags so that a separate check of the loop counter before the branch instruction is not necessary.

The pipeline diagram shows how the DSP unit latency cycles (S) are used by the ALU for the loop control and the data load instructions. Using two sets of registers (L11, L12 and L13, L14) for the data, the two load pipelines effectively hide the two access cycle memory latency from the program. In this way, the DSP unit can be kept busy during the FIR computation. Using 8 clock cycles per loop iteration, the resulting throughput of the FIR filter is two clock cycles per filter tap.

The subword processing technique used in the FIR filter example is illustrated in figure 4. Four 16 bit operands located in two general purpose registers serve as input to the DSP unit. The EHMACD instruction performs two 16 bit multiplications and accumulates the products using the 64 bit accumulator G14//G15. Alternatively, the EHMAC instruction uses the 32 bit accumulator G15 for the product summation.

This subword processing technique effectively enables the parallelism based on latency cycles. While the DSP unit performs the operation on four subwords, the ALU and load pipelines perform the instructions neces-

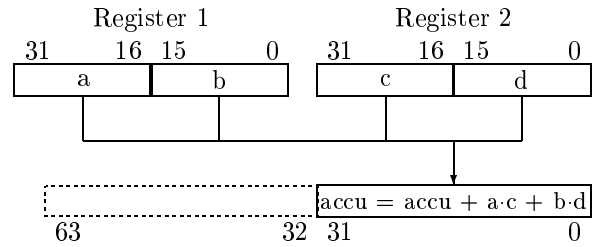


Figure 4. Subword processing

sary to get the next operands and for the loop control. Even though the basic architecture is still a single-issue architecture, different numbers of latency cycles permit the parallel operation of different units.

For fixed point computations, the DSP unit does not specify a particular fixed point format. The format is specified by the programmer. For example, for a filter computation in 1.15 fixed point format (one sign bit and 15 bits of fraction giving a numeric range of $-1 \leq x \leq 1 - 2^{-15}$), the accumulation result is shifted right by 15 places after the accumulation is finished. This is done using normal ALU instructions. For a different fixed point format, the programmer specifies a different shift width. Thus, the RISC approach to DSP gives the programmer additional flexibility manipulating data.

4. IMPLEMENTATION DETAILS AND PERFORMANCE

The first realization of the RISC-based DSP architecture is the E1-32 microprocessor [7]. Using a 0.8 μm CMOS technology a performance of 60 MHz has been achieved. The typical power dissipation is less than 0.8 W at a power supply voltage of 5 V. The die size is 56 mm² including pads. The E1-32 requires 220,000 transistors in total including 4 KByte on-chip RAM, 96 general purpose 32 bit registers and the 128 byte instruction cache. Thus, the E1-32 is ideally suited as a core technology for an ASIC design. Figure 5 shows

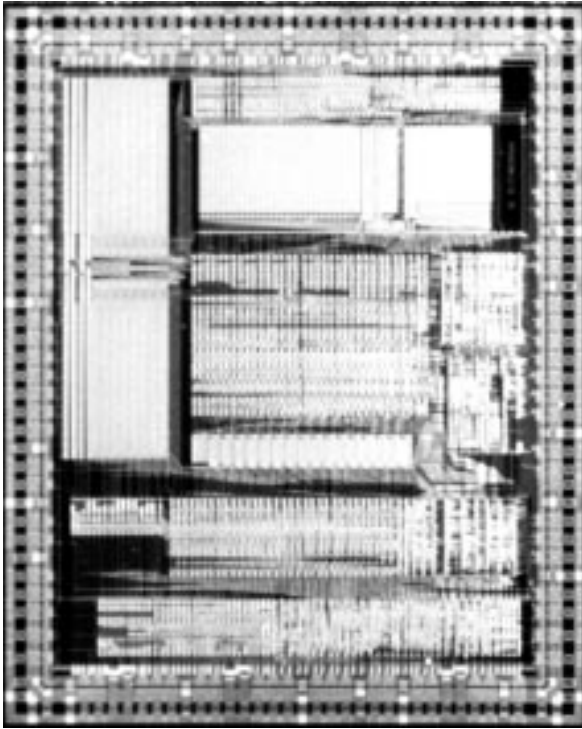


Figure 5. E1-32 die photo

a die photo of the E1-32.

The current version of the E1-32 is manufactured using a widely available $0.8\ \mu\text{m}$ CMOS process. This version is currently ported to a $0.6\ \mu\text{m}$ process where clock rates of 80 MHz are achieved. With a port to $0.5\ \mu\text{m}$ and $0.35\ \mu\text{m}$, clock rates of 120 MHz and above will be possible [8].

The general purpose microprocessor performance of the E1-32 has been measured with the Dhrystone 2.1 benchmark suite. Running at 60 MHz a performance of 97.6k Dhrystones has been achieved [8].

For performance measurements of the E1-32 as a DSP processor, a number of common DSP benchmarks have been executed [9]. The following figures illustrate the DSP performance of selected 16 bit algorithms running on a 50 MHz E1-32 using external static memory for program and data:

1024 point complex FFT:	0.87 ms
100 tap FIR filter:	$4.5\ \mu\text{s}$
100 tap LMS adaptive FIR filter:	$12.3\ \mu\text{s}$
8×8 point 2-dim DCT:	$25.2\ \mu\text{s}$

The FIR benchmarks include the filter history update as well as all loop control and branching, the adaptive filter includes the FIR filter computation and the LMS coefficient update after each input sample. The FIR benchmarks use the 64 bit accumulator for the computation.

As an example illustrating the performance of the memory load/store pipelines, the FFT benchmark has been executed with the data in external dynamic RAM. The 1024 point complex FFT benchmark executes in 0.92 ms in this case. This behaviour illustrates the advantage of the RISC-based DSP approach over conventional approaches. Even with slow external memory a very high DSP performance is achieved.

5. CONCLUSION

The RISC concept in combination with an enhanced functionality for Digital Signal Processing is capable of surpassing conventional DSP approaches. Especially in the growing field of multimedia and telecommunication applications, the unique combination of RISC controller and DSP features offers a simplified system design with high performance at low cost. Due to the strict RISC philosophy in the DSP unit design, this concept illustrates a clear roadmap to clock rates of 100 MHz and above.

REFERENCES

- [1] K. Nadehara et al., "Low-power Multimedia RISC," IEEE Micro, vol. 15, no. 6, pp. 20–29, 1995
- [2] M. Schlett, "A New Way of Integrating RISC and DSP," in: Proceedings of the 7th International Conference on Signal Processing Applications & Technology, pp. 653–657, Boston, 1996
- [3] M. Dolle, M. Schlett, "A Cost-effective RISC/DSP Microprocessor for Embedded Systems," IEEE Micro, vol. 15, no. 5, pp. 32–40, 1995
- [4] M. Dolle, M. Schlett, "A 32-bit RISC Microprocessor with integrated DSP Unit," in: Proceedings of the 22nd European Solid-State Circuits Conference, pp. 268–271, Neuchâtel, 1996
- [5] A. Peleg, U. Weiser, "MMX Technology Extension to the Intel Architecture," IEEE Micro, vol. 16, no. 4, pp. 42–50, 1996
- [6] R. Lee, "Subword Parallelism with MAX-2," IEEE Micro, vol. 16, no. 4, pp. 51–59, 1996
- [7] "hyperstone E1-32/E1-16 User's Manual," hyperstone electronics GmbH, 1996
- [8] M. Dolle, "EURICO: A 32-bit RISC/DSP Microprocessor for Embedded Systems," in: C. Müller-Schloer et al., "Embedded Microprocessor Systems," IOS Press, pp. 474–482, Amsterdam, 1996
- [9] "hyperstone Digital Signal Processing Library Reference Manual," hyperstone electronics GmbH, 1996
- [10] M. Bellanger, "Digital Processing of Signals," John Wiley and Sons, New York, 1989