efficiency embedded

White Paper
RISC & DSP Architecture

Version 0a | August 16, 2007

Hyperstone GmbH
Line-Eid-Strasse 3
78467 Konstanz
Germany
Phone: +49 7531 98030
Fax: +49 7531 51725
Email: info@hyperstone.de

Hyperstone Inc. - USA
4800 Bethania Station Road
Winston-Salem, NC 27105
USA
Phone: +1 336 744 0724
Fax: +1 336 744 5054
Email: us.sales@hyperstone.com

Hyperstone Asia Pacific - Taiwan
3F., No. 501, Sec.2, Tiding Blvd.
Neihu District, Taipei City 114
Taiwan, R.O.C.
Phone: +886 2 8751 0203
Fax: +886 2 8797 2321
Email: taiwan@hyperstone.com

www.hyperstone.com

hyperstone®

Unifying RISC and DSP

The Hyperstone E1-32XSR core represents a unique microprocessor architecture: The combination of a high performance RISC processor with an additional powerful DSP instruction set and on-chip micro-controller functions. The high throughput is not achieved by raw clock speed, but is due to a sophisticated architecture, which combines the advantages of RISC and DSP technology. It offers a powerful set of variable length instructions. Programs for the Hyperstone E1- 32XSR require less than half the memory size of most RISC µPs. Most instructions execute within one clock cycle. The fast multiply unit at high clock frequency makes it one of the fastest CPUs on the market with regard to DSP functionality. For many applications, the Hyperstone E1 makes the use of additional DSP chips obsolete.

Load-Store Architecture

The Hyperstone RISC technology is based on a load store architecture. It is register oriented and built around a 32-bit wide register stack that holds general purpose local registers and 26 global registers. Load and store instructions are pipelined to a depth of 2 stages at the memory bus.

Global Registers

The global registers include a Program Counter, Status Register, Stack Pointer, Upper Stack Bound, Bus Control registers, Timer registers and 14 general-purpose global registers.

Local Registers

The local registers are organized into a 64-word, circular register stack to hold function/subroutine stack frames. The stack crosses the register-memory boundary. Organized into stack frames of up to 16 words, the current frames are kept on-chip and are automatically pushed down to off-chip memory as the register stack fills up. Likewise, as the frames are popped off the stack, stack frames from memory are automatically passed to the on-chip stack.

Register Stack with Overlapping Frames

The current stack frame can overlap with the previous stack frame at a variable range to allow fast parameter passing. The overflow and underflow of the register stack is managed automatically, relieving the programmer of this task.

Variable-length instructions

Variable-length instructions make program codes more compact. The basic size of a Hyperstone instruction is a 16-bit halfword, however, the variable-length instructions can have up to three 16-bit halfwords. As a result, 32-bit constants and 32-bit native addresses are provided, thus making pre-instructions for generating longer addresses or constants obsolete. These variable-length instructions provide a program code that is more compact compared to other RISC and CISC architectures.

Integrated Timers

The Hyperstone E1-32XSR has two hardware timers integrated with a common time base and a resolution of 1 µs. The system timer is a general-purpose timer. In combination with Real-time operating systems, several virtual timers in stack-level tasks and virtual timers in interrupt-level tasks can be provided. Depending on the work load of the CPU, the latency of these virtual timers can be under 1 µs. Programming of these timers is very easy because only the delay has to be defined. Very important is that none of these timers generates any overhead CPU cycles for pending time events. A processing overhead of approximately 1 µs is required only when a timer event occurs. The other timer can be directly controlled by the user. The signals of this timer are directly accessible at one of the chip's I/O pins without any latency. It is synchronized to the clock.

Among others, this timer is ideally suited for measuring pulse widths or generation of pulse sequences.

Interrupts

Interrupts can be caused by external interrupt signals, by the general-purpose timer interrupt, or by an I/O Control Mode. Interrupts do not require a task switch. An interrupt causes an interrupt-level task to be entered. This interrupt-level task runs on the stack of the current task executing, just a new stack frame is created. Therefore, a full context switch is avoided. The interrupt latency time can be below 0.1 µs when no other interrupt is presently being served. Up to 7 priority-controlled external interrupt signals can be connected directly.

Shallow pipeline to accelerate branches plus an innovative instruction cache

The Hyperstone's shallow, two-stage pipeline accelerates standard and delayed branches. The innovative instruction cache provides automatic pre-fetch of instructions. This mechanism already loads the next instructions from memory into the cache, thus achieving the same high hit-rate as larger caches of other architectures.

Memory and I/O Address Spaces

The Hyperstone architecture provides separate memory and I/O address spaces. The memory address space of 4 GByte in total is divided into four memory areas with separate bus timing and bus width. A DRAM controller is integrated for the first memory area. It uses the fast page mode of the DRAMs, thereby producing burst cycles automatically. Hence, no external logic is required to connect SDRAM. All memory areas can also be assigned to SRAM, (Flash-)EPROM or other memory devices, each with its own bus timing - and all without external logic. Consequently, all memory devices can be directly connected pin-by-pin to Hyperstone microprocessors. A portion of the memory address space is also used by a single-cycle 16 kByte on-chip RAM. Hyperstone single-core RISC/DSP: ALU, DSP unit and load/store unit can work in parallel I/O devices are assigned to a separate I/O address space. Each I/O address has its own bus timing and virtually all peripheral components available on the market can be connected without external timing control logic.

Comprehensive on-chip bus interface

The comprehensive on-chip bus interface includes memory control (refresh, RAS-CAS multiplexer, parity) as well as chip-select and R/W-signals. This makes system design with Hyperstone microprocessors very simple because no interface logic is required to connect memory or I/O.

On-chip DSP-features

Up to now, separate DSPs and CPUs have been necessary for a number of applications, in particular for multimedia or telecom designs. Such applications can finally be realized through just one Hyperstone microprocessor because a DSP unit is already integrated into the architecture. The DSP unit operates on the register set of the architecture in parallel to the ALU and load/store unit. It is executing a dedicated DSP instruction set. Like the other instructions, the DSP instructions are strictly following RISC-principles. During the latency cycles of DSP instructions the ALU and load/store unit can execute other instructions.
Thus, a much higher flexibility is achieved compared to conventional DSP implementations. Additionally, up to three operations per clock cycle can be executed. Therefore, peak performance is tripled e.g. when running at 200 MHz, roughly 200 MIPS or up to 300 MOPS can be achieved. The DSP unit supports 16-bit and 32-bit data types. In order to achieve highest data throughput the DSP unit provides dedicated result registers and a 32-bit hardware accumulator as well as a 64-bit hardware accumulator. Among the dedicated DSP-type instructions are:

16-bit data format:

- multiply (single-cycle, pipelined)
- multiply-accumulate (single-cycle, pipelined)

- complex multiply
- complex multiply-accumulate
- addsub
- fixed-point shift

32-bit data format:
- multiply
- multiply-accumulate
- multiply-subtract

DSP Software Library

A collection of powerful DSP subprograms with ANSI C interfaces is optimally supported by the C Compiler. The DSP Software Library offers programmers fast prototyping of DSP software and algorithms for the Hyperstone RISC/DSP processors. It provides subroutines for the most important algorithms needed by DSP applications including:
- Digital FIR Filtering
- Digital Adaptive FIR Filtering
- Digital IIR Filtering
- Fast Fourier Transformation
- Discrete Cosine Transformation
- Multidimensional Arithmetic
- Standard Functions
- Utility Functions

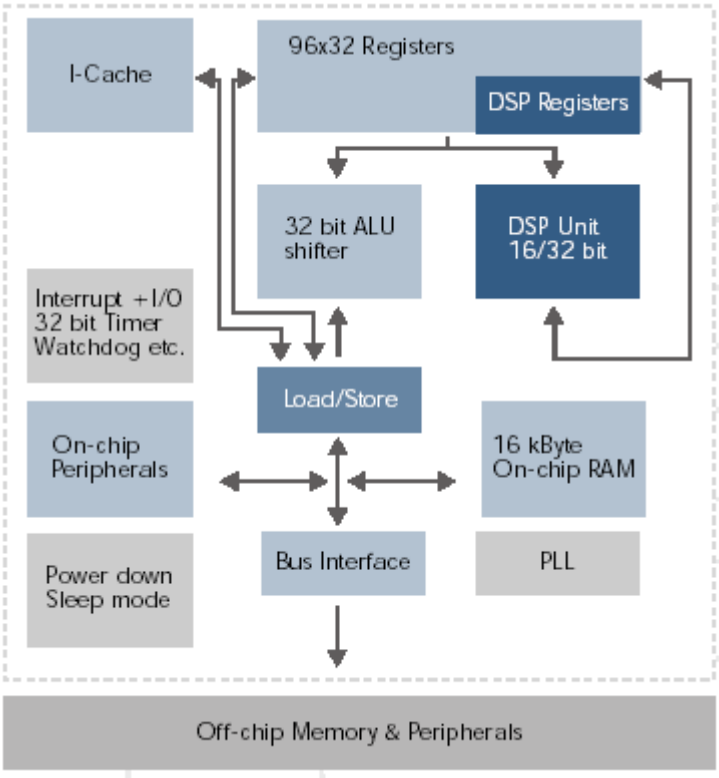

Figure 1: Hyperstone RISC & DSP CPU Architecture

Low power consumption

The Hyperstone's minimum transistor count results in a low power consumption. Depending on process technology and maximum frequency e.g. at 200 MHz about 50 mW worst case are consumed by the processor. An automatic power down reduces power consumption even further in

many applications. Due to the on-chip bus interface, total power consumption depends on the external load connected to the chip. The low power consumption makes very small packages possible.

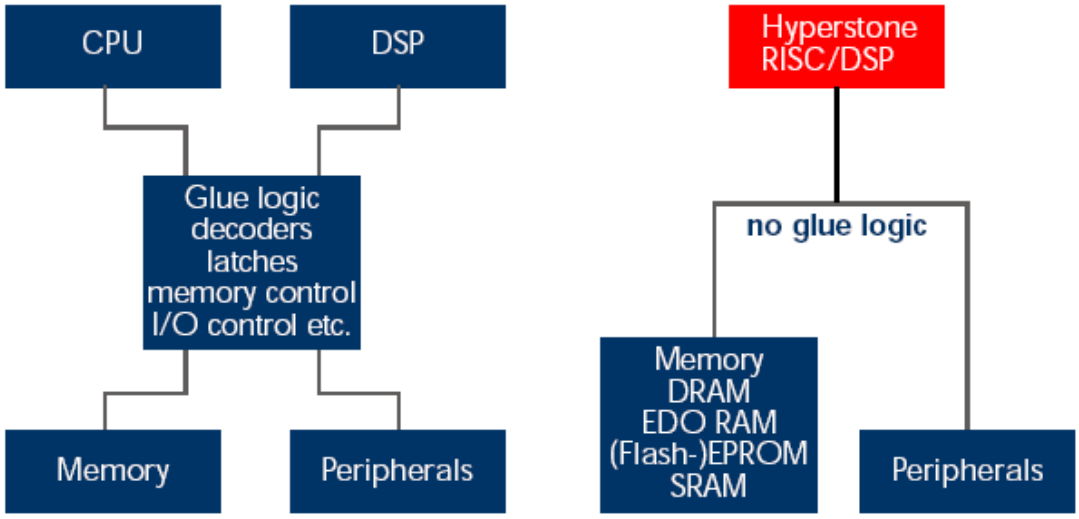

Figure 2: Architectural integration of RISC & DSP requires no glue logic

Conclusion

All Hyperstone processors are based on the E1 CPU architecture and offer the RISC&DSP advantage. The efficient integration results in an ultra low power CPU core. The processor is available in several different versions including the E1-32XSR, the E2 and other Application Specific Standard Products (ASSP) such as all hyNet derivates.