

# The RISC Challenge in DSP Processing

*A new wave of microprocessors is taking the lead in advanced embedded computing offering computational power for control as well as dedicated DSP tasks. This new generation of processors is based on the RISC design philosophy and challenge traditionally separated controller and DSP processor solutions.*

## INTRODUCTION

These days, Digital Signal Processing (DSP) is working its way into common use. Software such as audio or speech processing make its way into consumer and even into deeply embedded applications. But algorithms like for example audio processing software demand DSP capabilities for a resource-efficient implementation. A conventional microcontroller alone is in most cases not able to execute these algorithms efficiently. Thus, dedicated DSP functionality has to be implemented in such advanced embedded systems (for an introduction to DSP see e.g. [1]). In principle, this could be done either by adding DSP functionality to the microcontroller core or by adding a separate DSP processor.

In parallel, there is an increasing demand for control functionality in DSP applications. But conventional DSP processors [2] offer only very limited control functionality because of their internal architecture. The lack of general purpose registers, bit-manipulation instructions, fast external memory and peripheral access are limiting the use of DSP processors in embedded applications. From a viewpoint of embedded applications, DSP processors are pure calculation engines comparable with co-processors or specialized ASIC's used for dedicated tasks.

## EMBEDDED MARKETPLACE

In the past, embedded systems were a synonym for embedded control applications requiring a cost-efficient microcontroller focusing on system control and integrated peripherals and memory. In the 32-bit embedded controller marketplace, integration of up to 256 or 512 Kbytes of flash memory are currently state-of-the-art. Peripherals such as parallel I/O ports, A/D converters, Timers, Watch Dog Timers, serial ports, or bus interface controllers are standard for an embedded microcontroller.

As embedded applications emerge which include also displays, Graphical User Interfaces (GUI), a Man Machine Interface (MMI), image processing or remote control capabilities via standard phone lines additional performance for DSP and multimedia functionality has to be offered. By using traditional approaches system design becomes more complicated and the overall costs are increased. These trends change the embedded marketplace in several ways and initiate new classes of processors, see [3].

## COMBINED CPU/DSP APPROACH

Another and more optimized solution is to integrate conventional general purpose control and dedicated DSP functionality into a single, unified architecture, see e.g. [4], [5]. This results in a combined CPU/DSP processor offering with a single instruction stream conventional CPU as well as dedicated DSP functionality.

An example of the latter approach is the subsequently described Hitachi SH3-DSP architecture featuring dedicated DSP support for next-generation embedded applications. Such an approach keeps the costs down while offering standard embedded operating system support and a simplified system approach.

## RISC VERSUS DSP

A typical embedded microcontroller has been optimized to perform the overall system control which includes driving and controlling of external devices and the execution of the main program. A typical control program consists out of a large sequence of instructions and only a small amount of data. In order to offer a higher performance and to reduce the overall costs, state-of-the-art microcontrollers are based on the RISC design philosophy [6]. The RISC design philosophy is based on several principles such as a simple instruction set, a powerful pipeline concept for a very fast instruction execution, register orientation, and an uniform instruction decode process.

Typically, a DSP processor executes completely different tasks. A large amount of data is manipulated by a more or less small program code. The algorithm used for manipulating the data is normally based on a small loop. This leads initially to a completely different implementation and design approach. The RISC approach used for embedded microcontrollers has to be modified and further extended to meet the requirements of typical DSP programs. A DSP processor is, for example, strictly memory oriented, i.e. instructions are operating directly on operands stored in memory locations and not in registers, instruction execution is normally not pipelined, and the instruction set has been selected to execute typical DSP programs such as Digital Filters or Fast Fourier Transformation programs. Substantial requirements for DSP applications include also a deterministic program execution, a very short interrupt response time and a fast data transfer capability for a typically continuous data stream processing. The integration of DSP functionality into a basic 32-bit RISC

architecture means to transform these basic DSP characteristics to the RISC methodology and at the same time to extend the RISC philosophy.

## COST REDUCTION

In case of the combined CPU/DSP processor, several resources are used by control and DSP tasks. This approach avoids the doubling of already available

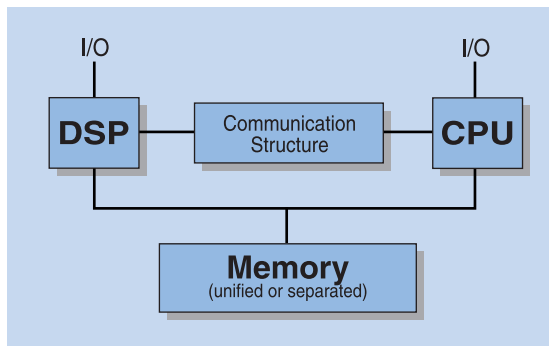


Figure 1. Separated CPU and DSP approach

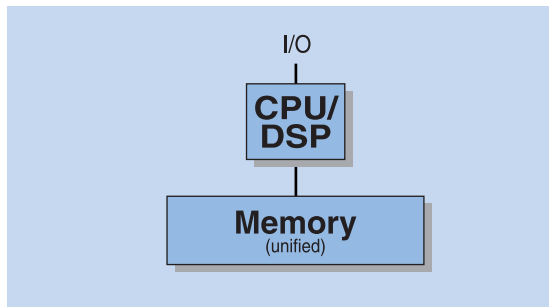


Figure 2. Combined CPU/DSP system solution approach

hardware and this results in an immediate cost reduction. Besides that, the combined CPU/DSP approach simplifies the overall system design. The separated CPU and DSP approach leads to a multiprocessor system with all the well-known problems. Even when integrated into a single ASIC such a multiprocessor solution makes debugging complex and difficult and

doubles resources such as multipliers and registers without using them permanently. Figure 1 and 2 illustrate the two different system solutions.

The combined CPU/DSP approach is based on a single instruction stream, it is only the programmer who decides if a task is a control or a DSP task. Every task of the entire program can use all resources of the device. This approach avoids synchronisation issues which occur in multiprocessor systems and leads to a

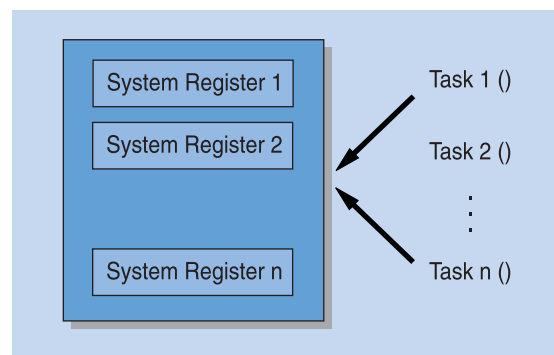


Figure 3. Single instruction stream of combined CPU/DSP architectures

higher reliability of the code. Figure 3 shows the single instruction stream approach of combined CPU/DSP architectures.

Actually, there is no separation anymore into a CPU and a DSP core, the separation is only done by separation of the instruction set into a control or CPU and a DSP instruction set.

This approach gives a system designer the flexibility to add to his embedded system additional multimedia capabilities. Additional hardware is not anymore necessary. By additional software modules, the embedded system can include modem support or image processing support. Other examples are voice processing or video decoding. Additional hardware or DSP processors could thus be obsoleted in the embedded system design.

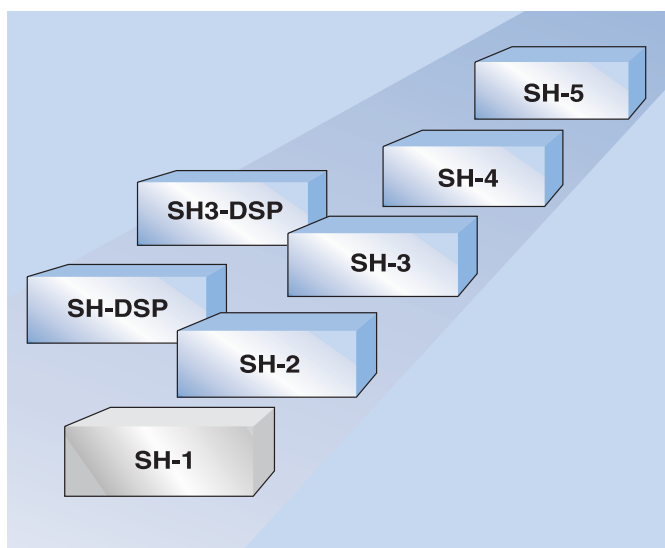


Figure 4. SuperH™ Architecture Overview

## SUPERH™ 32-BIT EMBEDDED PROCESSOR FAMILY

A common 32-bit RISC architecture has been used to create an upward compatible family of 32-bit embedded microcontrollers and microprocessors. Starting with the SH-1 embedded core, further advanced SH-2, SH-3 and SH-4 cores have been introduced which are the basis for several SH series and lines offering a wide variety of individual devices offering an application oriented peripheral mix. A second upward compatible series has been developed integrating additional DSP functionality and extensions, the so-called SH-DSP and the new SH3-DSP architectures. Figure 4 illustrates the upward compatibility of the various series. The SH-DSP series includes a SH-2 core with integrated DSP

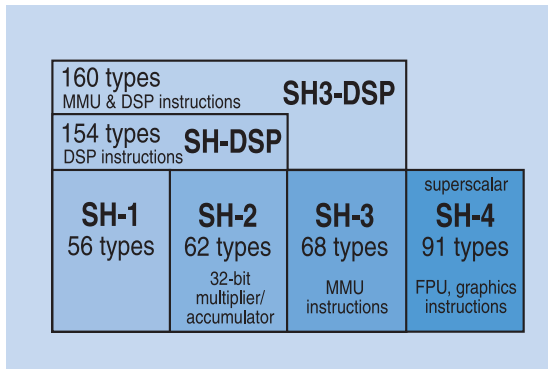


Figure 5. Instruction Set Upward Compatibility

functionality, the SH3-DSP series features a SH-3 core plus an extended DSP instruction set. The DSP instructions of the SH-DSP and the SH3-DSP series are also upward compatible to enable a simple upgrade path. Figure 5 shows the basic instruction set upward compatibility of the SH series. That means that object code running on a SH-1 device is also running on a SH3-DSP device. Every evolution step represents additional architectural features, for example, all SH-3 and higher devices feature an integrated Memory Management Unit (MMU) allowing virtual memory management. The MMU could be disabled without any performance loss. Figure 6 introduces the current SH devices available.

## SH3-DSP ARCHITECTURE

The SH3-DSP architecture is an object-code upward compatible line-up of the SH-DSP and the SH-3 architecture. The instruction set of the SH3-DSP is a super-set of the two other architectures resulting in a powerful, high-performance embedded processor upgrade solution.

The SH3-DSP architecture has the following main features:

- Basic 16-bit fixed-length RISC instruction set
- 16/32-bit length DSP instruction set extension
- 16/32/40-bit internal data paths
- Large general purpose register file
- Additional DSP register file
- Basic 5-stage pipeline
- 4 Gbytes address space
- Extended Harvard architecture for DSP datapath
- 2 Addressing units
- Multiplier, ALU, Barrel Shifter
- Maximum of 4 parallel operations (ALU, Multiply, 2 Load/Stores)
- Looping hardware
- Fixed-point support
- X/Y on-chip memory structure/datapath

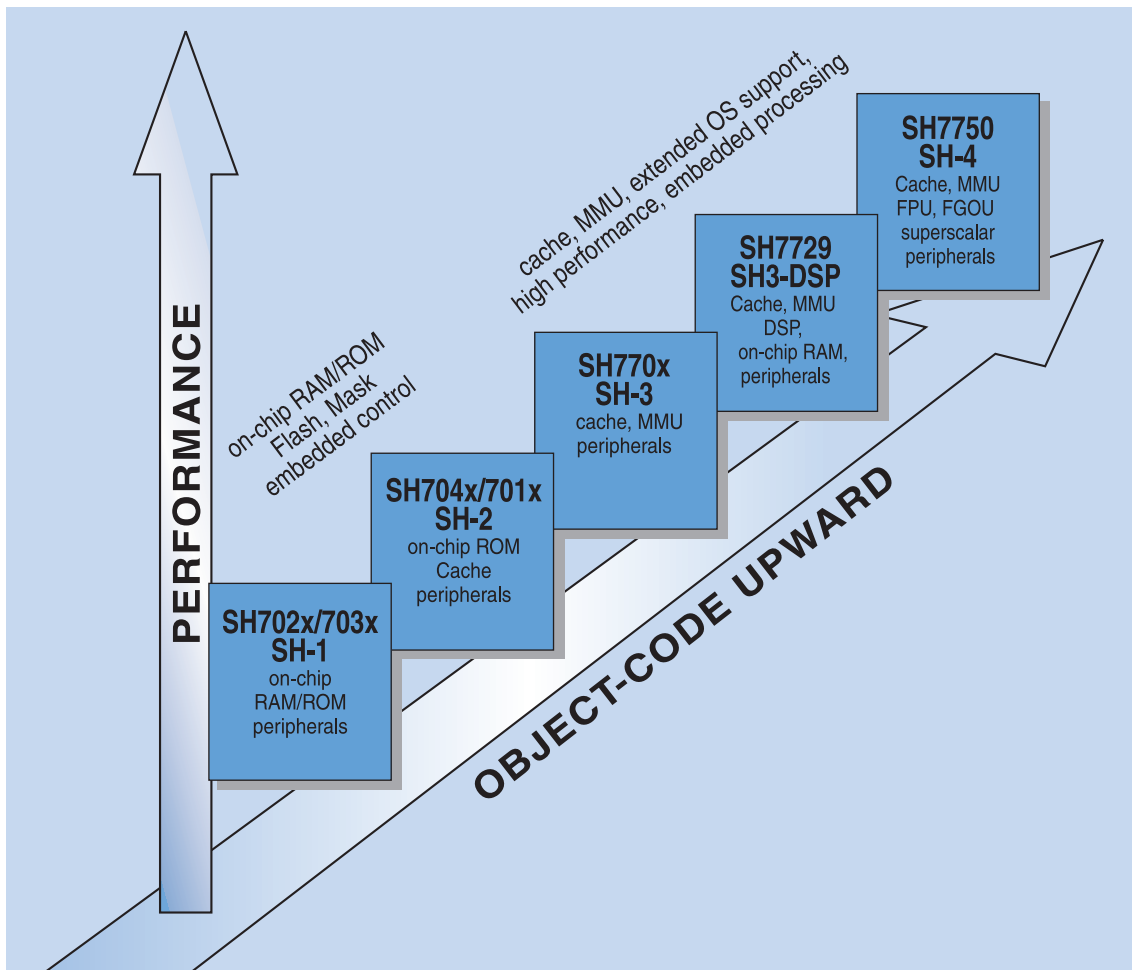


Figure 6. Hitachi SH 32-bit CPU Series and Derivatives

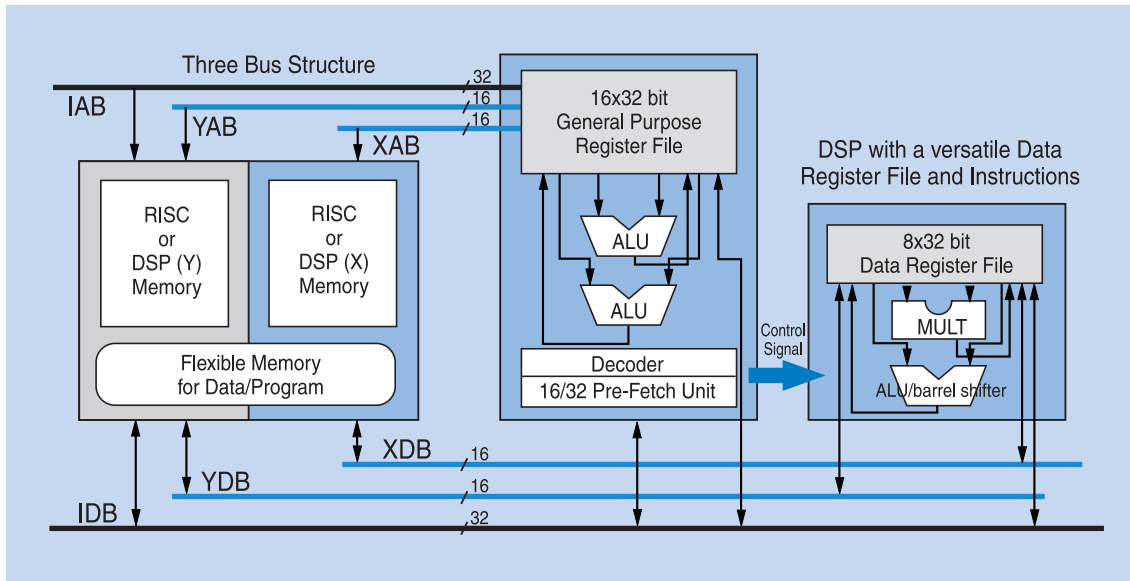


Figure 7. Integrated DSP unit

Especially useful for DSP programs are the extended bus structure, the looping hardware, and the fixed-point arithmetic support. Figure 7 shows the integration of the DSP unit into the RISC architecture.

One of the most important features of DSP functionality is the efficient manipulation of fixed-point data. This kind of arithmetic requires the availability of a set of fixed-point data formats directly processed by the instruction set.

Normal CPU's only support integer or floating-point numbers. The DSP instructions of the SH3-DSP also support fixed-point data in the range of  $-2^8$  to  $(2^8-2^{-31})$ . Programmers can make use of an accuracy of up to 31 mantissa bits. Additionally, the SH3-DSP also supports DSP type integer data. Important is the availability of guard bits to handle accumulation overflow, the SH3-DSP's DSP type data format supports up to 9 guard bits.

During the execution of a DSP algorithm an intermediate result (for example, the result of an accumulation) is very often a 32-bit number, but after finishing the operation sequence the result has to be rounded to 16 bits. The SH3-DSP instruction set also supports the rounding feature and the programmer can choose the rounding mode.

Zero-overhead loops are directly supported by the SH3-DSP via hardware. The programmer can determine with register values the start and stop addresses of loops which results in a straight-forward implementation of digital filter or other loop-intensive DSP algorithms.

Finally, by providing additional addressing modes, such as the modulo addressing mode, programmers have full access to instructions necessary for the implementation of efficient and high-performance DSP algorithms.

## DSP INSTRUCTION SET EXTENSION

The RISC instruction set of the SH-3 architecture has been extended by the following classes of DSP instructions:

- System control instructions providing support for loop control and data transfer between registers and/or memory
- Data transfer operations between memory and registers inside the DSP unit to save program code size
- Parallel data processing operations inside the DSP unit offering up to four parallel operations (ALU, multiply, load, store)

An example of a parallel DSP operation sequence is given below:

```
PADD A0,M0,A0  PMULS X0,Y0,M0  MOVX.W
(R4)+,X0      MOVY.W (R6)+,Y0
```

In this example, R4 and R6 belong to the general purpose register file, A0, M0, X0, Y0 belong to the dedicated DSP register file. The SH3-DSP organises the above operation sequence as double data transfer field 'A' and double data processing operation field 'B'. These operations are then executed in parallel. There are several kinds of operation groups for each field comprising a set of instructions which can be combined for parallel instruction execution.

## X/Y-MEMORY

The SH3-DSP architecture has been designed to support on-chip X- and Y-memory. It can be used by the CPU and the DSP unit to store instructions and data. This memory has a user selectable mapping mechanism and 3 independent read/write ports supporting:

- 16/32-bit access from the CPU
- maximum two 16-bit accesses from the DSP unit
- 8/16/32 bit accesses from a DMA controller

The DSP unit can access this memory via the XBUS and YBUS. Each access is 16 bit wide. This bus structure does not exist on the SH-3 architecture and is especially useful for Digital Filters.

## SH7729

The first implementation of the SH3-DSP architecture is the SH7729 device. The SH7729 integrates a SH3-DSP core and a mix of peripherals:

- SH3-DSP Core
- Memory Management Unit
- 16 kbyte cache memory, mixed instruction/data
- 16 kbyte of X/Y memory
- Interrupt controller
- User break controller
- Bus state controller
- Serial debug interface
- Timer
- Real-time clock
- Serial communication interfaces
- DMA controller
- I/O port
- A/D and D/A converter
- Clock pulse generator / PLL

The SH7729 has the following features:

- Up to 133 MHz internal clock frequency, 133 MIPS / 266 MOPS
- Up to 66 MHz external bus frequency
- Up to 33 MHz peripheral bus frequency
- 3.3 V I/O
- 1.8 V Core power supply
- QFP-208 Package

The general purpose register file and the organisation of the DSP unit enable a fast interrupt behaviour of the SH7729. Shadow registers serve as fast content-saving mechanism, the integrated interrupt controller includes a priority identifier and handles the interrupt requests of external and internal devices and units. The interrupt response time is the sum of the following consecutive operations:

- Time for priority decision and Status Register mask bit comparison
- Wait time until end of pending instruction execution
- Time from interrupt exception handling until fetch of first instruction of exception handler is started

The basic interrupt response time depends on the interrupt source and the memory access wait time. For this purpose, the SH7729 integrates a bus state controller offering very fast external bus accesses and a SDRAM interface. An interrupt response time of 0.3 – 0.6  $\mu$ s can be achieved in case of operand cache hit or external memory access is performed with wait=0.

The cache memory of the SH7729 has a capacity of 16 kbytes for instructions and data and is 4-way set associative. The cache can also be disabled if a hard real-time behaviour and a deterministic program flow are required. If the cache is enabled, the cache will be searched whenever instructions or data in memory (not X/Y-memory) are accessed. The cache is a physical cache and holds physical data in its address section. The cache also supports memory mapping onto a virtual address space.

## APPLICATION EXAMPLES

The SH7729 has been designed for embedded systems requiring a high-performance 32-bit CPU with an additional need for DSP performance. The DSP unit is especially useful to run programs such as:

- V.34 modem code
- JPEG (de) compression algorithms
- Voice / audio (de) compression algorithms such as G.723.1
- Speech recognition and synthesis
- H.263 image (de) compression software

This means that even an already existing embedded system can be upgraded by additional software modules to support additional multimedia features.

## SUMMARY

The SH3-DSP architecture presented is a further development of the SH-DSP and SH-3 architecture featuring object-code upward compatibility. With a performance up to 133 MIPS (Dhrystone 1.1) and 266 MOPS, the SH7729 implementation is a powerful platform for next generation embedded applications requiring additional multimedia and/or DSP capabilities. The SH3-DSP is based on a single instruction stream approach combining general purpose CPU and dedicated DSP functionality in a single architecture. This makes for a wide variety of applications additional hardware accelerators or DSP processors redundant and provides in parallel a simplified system approach. ■

## REFERENCES

- [1] M. Bellanger, Digital Signal Processing, John Wiley and Sons, New York, 1998
- [2] P. Lapsley et al., DSP Processor Fundamentals, Berkely Design Technology, Inc., Fremont, 1996
- [3] M. Schlett, Trends in Embedded Processor Design, IEEE Computer, August 1998, Vol. 31, No. 8, pp. 44-50 (reprinted in Real-Time Magazine, 98-4, pp. 14-20)
- [4] M. Dolle, M. Schlett, A Cost-Effective RISC/DSP Microprocessor for Embedded Systems, IEEE Micro, October 1995, Vol. 15, No. 5, pp. 32-40
- [5] M. Schlett, A New Way of Integrating RISC and DSP, Proc.7th Int. Conf. Signal Processing Appl. And Techn. ICSPA '96, Boston MA, pp. 653-657
- [6] D. Tabak, RISC Architecture, John Wiley and Sons, New York, 1987

*Manfred Schlett is a marketing engineer at Hitachi Europe GmbH. His research interests include micro-processor architectures, advanced VLSI design, signal processing, and multimedia. Previously, he worked for Hyperstone Electronics as a design engineer and later on as a project manager of the European OMI project EURICO and as a marketing manager. Schlett received his Diploma degree in technical mathematics and his PhD in mathematics from the University of Karlsruhe, Germany. He is a member of the IEEE Computer Society. Contact the author at manfred.schlett@hitachi-eu.com.*