RC64,
A RAD-HARD MANY-CORE HIGH-PERFORMANCE DSP FOR SPACE APPLICATIONS

Ran Ginosar, Peleg Aviely, Hagay Gellis, Tuvia Liran, Tsvika Israeli, Roy Nesher,
Fredy Lange, Reuven Dobkin, Henri Meirov, Dror Reznik
Ramon-Chips Ltd., 5 HaCarmel Street, Yokneam 2069201, Israel
Email: [ran, peleg, tuvia,,tsvika, roy, fredy, reuven, miki, henri, dror]@ramon-chips.com, Hagay.Gellis@ceva-dsp.com

Abstract

RC64, a novel rad-hard 64-core signal processing chip targets DSP performance of 75 GMACs (16bit), 150 GOPS and 38 single precision GFLOPS while dissipating less than 10 Watts. RC64 integrates advanced DSP cores with a multi-bank shared memory and a hardware scheduler, also supporting DDR2/3 memory and twelve 3.125 Gbps full duplex high speed serial links using SpaceFibre and other protocols. The programming model employs sequential fine-grain tasks and a separate task map to define task dependencies. RC64 is implemented as a 300 MHz integrated circuit on a 65nm CMOS technology, assembled in hermetically sealed ceramic CCGA624 package and qualified to the highest space standards.

RC64 Progress

In 2012, 2013 and 2014 we presented in DASIA early descriptions and progress of the RC64 project [3][4][5]. The two earlier reports assumed 130nm technology and 64 simple RISC cores, which imposed many limitations. Those papers described the development of the RC64 architecture, starting with 64 simple RISC cores and discussing preliminary performance of the ESA NGDSP benchmark #5 (modem) on RC64 [1][2].

Meanwhile, the RC64 project has made significant progress, first described in [5]. An advanced technology node of 65nm was selected together with a 624-pin ceramic CCGA package. Sixty-four advanced DSP cores from CEVA [6] were selected and integrated in a simulator and as a FPGA implementation. Both platforms demonstrate several applications. The DSP core tools were used for providing a state-of-the-art integrated development environment for software development and performance analysis. Physical design has started, including preparation of standard cell libraries, radiation effects mitigation methodology and interface IP selection. These were tested for radiation effects in a dedicated test chip [7]. Further progress has been made in building a test-bed and benchmark suite for validation and performance demonstration within the MacSpace FP7 project [8]. The project contributed a set of requirements for high end payload computers in multiple application domains. Further progress has been made within the QI2S FP7 project [9], where a hyperspectral image processing application has been developed, demonstrating advanced algorithmic design using preliminary RC64 tools and implementation.

RC64 Architecture

The new package and new process technology enable the inclusion of twelve integrated full duplex high speed serial links (HSSL) using CML SERDES interfaces on chip at a 3.125Gbps rate each, with an aggregated 60Gbps throughput. Several protocols such as SpaceFibre and SRIO are considered for HSSL, aiming at efficient connectivity among multiple RC64 chips and other FPGAs, ASICs. HSSLs minimize PCB complexity. They eliminate the need for multiple external SERDES components on board, save power and reduce the number of pins required. The HSSLs in RC64 support off-board and long range communication, as well as multi-lane channels when higher data rates are required.

The advanced technology and package also enable supporting faster and denser DDR3 SDRAM interface while keeping DDR2 as an option. Reed-Solomon ECC is employed to protect from DDR2/3 SEFI and SEE. The 32-bit wide DDR2/3 interface supports up to 25Gbps throughput.

Other I/O interfaces in RC64 include two SpaceWire interfaces for control and twelve for instrument data, parallel LVDS interfaces for advanced ADC and DAC device connectivity and interface to flash memory.

The on-chip shared memory system of RC64 is based on each DSP core having its own write-through data cache, an instruction cache, and a private store, supporting the unique task-oriented programming (TOP) model. All DSP cores access the single shared memory with 256 ports and a 64-to-256 ports multistage interconnection network,

enabling simultaneous access of all processors to shared memory with a low number of conflicts. Thanks to the DSP core's data cache, access to shared memory happens either for fetching a complete cache line (the interfaces and the interconnection network are optimized for transferring complete cache lines rather than individual words) or for writing a single word, due to the write-through mechanism. While write-through may result in a higher traffic rate to memory than write-back, it eliminates the need for complex inter-core cache coordination mechanisms such as snooping, locking and directories. Instead, the programming model minimizes memory conflicts and prevents software from relying on shared memory synchronization.

The on-chip 4MByte shared memory acts as a local-store memory. Access to off-chip DDR2/3 memory is facilitated by software-controlled DMA. This approach simplifies software development and it is found to be very useful for DSP applications, which favor streaming over cache-based access to memory.

The many-core architecture is depicted in Figure 1. A central scheduler assigns tasks to processors. Each processor executes its task from its cache storage, accessing shared memory only when needed. When task execution is done, the processor notifies the scheduler, which can subsequently assign a new task to that processor. Access to off-chip streaming channels, DDR2/3 memory, and other interfaces happens only via programmable DMA channels.

CEVA-X1643 DSP

The CEVA-X1643, instantiated 64 times, is a DSP based on a 8-way VLIW (Very Long Instruction Word) model combined with SIMD (Single Instruction Multiple Data) concepts.

The CEVA-X1643 architecture is based on load/store computer architecture, utilizing in-order RISC operations and instructions. The architecture has dedicated Load/Store units responsible for loading/storing data from/to the data memory directly to/from the registers and accumulators. All other computation instructions always utilize those registers and accumulators as sources and destinations. The CEVA-X1643 has a rich set of DSP and arithmetic instructions as well as unique control-code related instructions.

The Instruction Set of the CEVA-X1643 is built for code compactness. Instructions are either 16-bit or 32-bit wide, and up to eight such instructions can be grouped to form an instruction-packet, executed in a single cycle. The four computation units (M0, M1, MS, ML) can perform several integer operations each cycle with data types of 8bit, 16bit, 32bit or 40bit integers.

Excellent FFT/FIR performance can be achieved due to the 4 MAC operations per cycle and a single complex MAC operation in a cycle.

The CEVA-X1643 includes an innovative Power Scaling Unit (PSU), which provides advanced power management for both dynamic and leakage power. The PSU supports multiple operational modes ranging from full operation, debug bypass and memory retention to complete power shut-off (PSO). This is highly important for the RC64 being in a space environment with heat dissipation challenges.

More details in http://www.ceva-dsp.com/CEVA-X1643

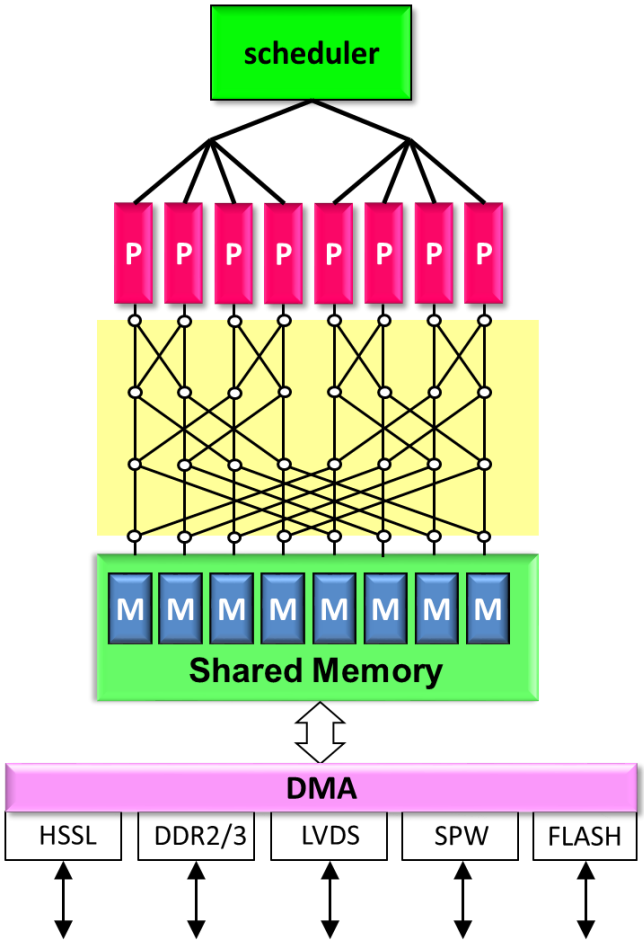

Figure 1: RC64 many-core architecture

Figure 2: CEVA-X1643 Block Diagram

Software development tools include a state-of-the-art GUI environment enabling C-language application development. Performance is achieved through instruction level optimization using a detailed profiler, employing libraries for optimized code, validated on a cycle accurate simulator.

RC64 Extensions to CEVA-X1643 DSP

Being a general purpose DSP, the CEVA-X1643 does not include **floating point** support. Several applications and use-cases identified during MacSpace [8] project require an intensive use of floating point computation. These applications are hard to implement using the optimized fixed point arithmetic and calls for adding a floating point co-processor. Such a co-processor extends the instruction set of the DSP to include single precision floating point multiply-accumulate, multiply, add/sub, convert and compare. These instructions will be used by the compiler to implement 'float' data type related computations. The co-processor will be instantiated 64 times and will be connected to each DSP core, providing peak performance of 64 x 2 x 300M=38.4Gflops for MAC dominated algorithms.

Radiation tolerance for processors requires special care for SRAM modules used for local memory and cache. These modules suffer from a high probability of SEE due to the high density of memory cells. CEVA-X1643 DSP implementation within RC64 combines error detection and invalidation for program and data caches. Extra parity bits are combined into the tag, program and data cache memory arrays, indicating soft errors when accessed for reading. A cache miss sequence is initiated when parity error detected, reading the highly protected original value from the shared memory. The detected errors are monitored for later analysis.

The local data memory, which has no redundant refresh source, requires error detection and correction. Each byte of the local data memory is protected for single error events. This enables on-the-fly correction of errors when reading the data. Storing data modifies the hamming code bits. Detected errors are monitored and are useful for deciding on scrubbing the local data memory to remove the error as these can accumulate over time and become un-correctable. The scrubbing sequence interrupts the application and therefore is scheduled by software when the core is idle or is performed periodically using driver code, both without programmer intervention.

RC64 Configuration for CEVA-X1643 DSP core

Configuring the DSP core for RC64 required a different approach due to the 64 instances and the shared memory architecture. The size of the shared memory (4Mbyte) enables holding the program, executed by all the DSPs, available with few penalty cycles using cache miss sequence. This requires only a minimal program cache size of 8Kbyte per DSP core, fitting a kernel code executing highly efficient DSP algorithms.

Data cache size must fit algorithm kernel memory capacity and support data streaming from shared memory to the DSP core. This requires only minimal memory size of 8Kbyte. The cache pre-fetch mechanism enables hiding some of the latency due to cache miss sequence while performing computation.

Local data memory is core private and useful for storing constants, stack and temporary working data set. This fast memory doesn't suffer any latency due to cache miss sequences and can be configured to hold memory structures defined during compilation time to be available for executed kernels of the application. The selected 8Kbyte size is found to be a good trade-off for this purpose and chip area efficient.

RC64 integration of CEVA-X1643 DSP

Connecting the 64 instances of the DSP core into the RC64 interconnect includes the following interfaces:

Program and Data memory interfaces are connected through a bridge to the shared memory network. The two AXI master interfaces perform write and read transactions during cache miss sequence or when non-cacheable memory is accessed. The shared memory keeps the store ordering requirement of the DSP core, enabling delayed write transactions, minimizing stall cycles.

Floating point extension interface is memory mapped using data AXI master interface. A strong order mechanism supported by the DSP core enables keeping load/store sequence for such dedicated address space, providing in-order transactions, holding both data and command information.

Scheduler interface uses data AXI slave, storing scheduler data into pre-defined address in the local data memory. A software driver enables communicating with the scheduler interface, responding to task allocations and messaging using GPIO interfaces of the DSP core. A power down sequence is initiated by the program when no task is allocated, saving dynamic power, waking up in just few cycles when required to perform a new task.

Debug is implemented using a JTAG chain of all DSP cores, enabling visibility into the current state of the program, local and shared data memory. It enables controlling each DSP core, stepping through the program code, setting break points etc. A break sequence for the system is supported by connecting debug state indications of each DSP core to the other cores break input. Entering debug mode in any of the cores leads to all other cores entering debug mode within a few cycles, freezing the program state for proper debug actions.

RC64 System software

Application development for RC64 requires adaptation software for abstraction of hardware details and for easy configuration and operation of the interfaces.

The following software libraries are provided for applications:

Boot sequence: the library is based on the CRT code of the CEVA-X1643 DSP core. It is modified to be performed by many cores in parallel where only one of the cores performs the shared memory content initialization. The code includes DSP core self-test, cache clearing, memory protection configuration and finally, execution status notification to the external controlling host.

Runtime loop: This library performs the scheduling function for the DSP core. It interacts with the hardware scheduler, receives task allocation details, launches the task code and responds with task termination when the task is finished. The code initiates the power down sequence when no task is received for execution.

Initialization task: The first task allocated by the scheduler is responsible for loading the application task map into the scheduler. This code is automatically generated during pre-compile stage according to the task map definition. Programmer's tasks will start to be allocated after this task is finished.

DSP Library: Task code optimization is possible using available DSP libraries, efficiently using the DSP core resources for instruction level efficiency.

Interfaces library: Configuring the interfaces requires special sequences such as link detection and activation, clock enabling, DMA configuration etc. Each interface has its own set of parameters according to the required connectivity, storage type, data rate etc.

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 313105 (www.QI2S.eu) and under grant agreement n° 607212 (www.MACSPACE.eu).

References

[1] R. Trautner, ESA's Roadmap for Next Generation Payload Data Processors, DASIA 2011.

[2] ESA/ESTEC, Next Generation Space Digital Signal Processor Software Benchmark, 2008. Available via http://www.esa.int/TEC/OBDP/

[3] Ginosar, Aviely et al., RC64: A Many-Core High-Performance Digital Signal Processor for Space Applications, DASIA 2012.

[4] Ginosar, Aviely et al., Architecture, design and implementation of RC64, a many-core high-performance DSP for space applications, DASIA 2013.

[5] Ginosar, Aviely et al., RC64, A rad-hard many-core high-performance DSP for space applications, DASIA 2014

[6] CEVA www.ceva-dsp.com

[7] Liran, Ginosar et al., 65nm RadSafe™ technology for RC64 and advanced SOCs

[8] MacSpace www.macspace.eu

[9] QI2S www.qi2s.eu