

Session: IP Design

**A Flexible, Low Power, High Performance DSP IP Core
for Programmable Systems-on-Chip**

Paul M. Heysters, Gerard K. Rauwerda and Lodewijk T. Smit

Recore Systems

P.O. Box 217, 7500 AE, Enschede, The Netherlands
paul.heysters@recoresystems.com

Abstract – Recore Systems provides semiconductor IP solutions for programmable systems-on-chip. The coarse-grained reconfigurable technology of Recore promises to solve the flexibility, performance, power consumption and cost requirements of semiconductor businesses. In this paper, the Montium™ DSP core is introduced and an example is given of how this core can be used in programmable systems-on-chip.

Keywords – DSP core, low power, system-on-chip, coarse-grained reconfigurable, Montium™

I. INTRODUCTION

Recore Systems develops processor cores, programming tools and applications. The processor cores of Recore offer low power, high performance and flexibility. This is realized by using coarse-grained reconfigurable technology. Complex programmable systems-on-chip can be developed seamlessly by using the processor cores as Intellectual Property (IP) building blocks.

The increasing complexity of integrated circuits causes custom designs to become too expensive. Upcoming systems-on-chip require a large number of interlocking pieces of IP to form a complete platform: processing cores, peripherals, specialized I/O modules, operating systems and development tools. More complex systems can be built in a shorter timeframe by purchasing off-the-shelf IP blocks. In particular, the design productivity will increase with 200% [3].

Recore's first core is called Montium. This core is based on coarse-grained reconfigurable technology developed at the University of Twente, The Netherlands [5]. This technology has established itself in numerous academic research projects. Recore adopted the concepts of the proven reconfigurable technology and redeveloped and reengineered the Montium technology into a product for use by chip developers. Besides a processing core, the Montium technology also

constitutes other hardware modules, program development tools and application designs. The program development tools for the Montium core comprise the Synsation compiler, the Simsation simulator and the Insation editor.

II. POWER, PERFORMANCE, PROGRAMMABILITY

Conventional processing architectures, such as general purpose processors, DSPs, FPGAs and ASICs, cannot satisfy the combination of extremely low power consumption, high performance, flexibility and low costs. Yet, the upcoming need for processors that meet all these requirements is undeniable. The need for low costs is universal, but why are the requirements for low power, high performance and flexibility imminent?

A. Why low power?

Most notably (for consumers), low power processors improve the operating time of battery powered devices. Lower power also means a lower operating temperature. Due to cost, noise and packaging constraints it is not always possible to use cooling fans. In today's world, environmental and economical incentives are important considerations as well.

B. Why high performance?

People use ever more digital processing. For example, consider cell phones, personal multimedia players, photo and video cameras and navigation systems. This trend will definitely continue and will drive the demand for ambient processing power. Especially upcoming multimedia and digital communications applications require massive digital processing. New standards provide better quality by using advanced algorithms, which are extremely computational demanding. For instance, the computational requirements for wireless communications will soon exceed the performance of the most powerful microprocessor.

C. Why flexibility?

Flexibility is the key to a short return-on-investment. New technical standards emerge ever faster and become more complex. Product development cannot be postponed until the new standards are stable due to a short time-to-market requirement. The reuse of flexible processor cores reduces time and money consuming hardware design. In addition, flexible processors are programmable, which leads to a short design cycle for new applications and gives rise to the reuse of existing software components. Products using flexible hardware can be repaired or upgraded with new features when they are already in the field.

III. MONTIUM PROCESSOR CORE

The Montium Tile Processor (TP) is a programmable architecture that obtains significant lower energy consumption than DSPs for fixed-point digital signal processing algorithms. The Montium TP targets computational intensive algorithm kernels that are dominant in both power consumption and execution time.

The Montium TP is typically used as an accelerator core in combination with a lightweight general purpose processor. In contrast to a conventional DSP, the Montium TP does not have a fixed instruction set, but is configured with the functionality required by the algorithm at hand. In particular, the Montium TP does not have to fetch instructions and, hence, does not suffer from the Von Neumann bottleneck. Once configured, the Montium TP resembles more an ASIC than a DSP.

The Montium TP can be reconfigured almost instantly, as the size of the configuration binaries is very small. The size of a typical configuration is less than 1 KB and reconfiguration typically takes less than 5 μ s.

The Montium TP has a low silicon cost, as the core is very small. For instance, the silicon area of a single Montium TP with 10 KB of embedded SRAM is 2 mm² in 0.13 μ m technology. The power consumption in this technology is typically less than 600 μ W/MHz (including memory access).

The Montium TP is programmed using the proprietary Montium LLL language.

A. Architecture

A diagram of the Montium TP is shown in Figure 1. The hardware organization is very regular. Five identical processing parts in a tile exploit spatial concurrency to enhance performance. This parallelism demands a very high memory bandwidth, which is obtained by having 10 local memories in parallel. The local memories are also motivated by the locality of reference principle, which is a guiding principle to obtain energy-efficiency

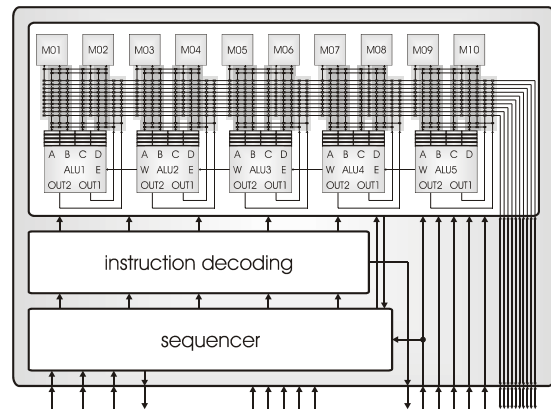


Figure 1: Montium Tile Processor

The datapath width of a Montium core and the memory capacity are customizable at design time and depend on the computational requirements. The arithmetic and logic units (ALUs) support both signed integer and signed fixed-point arithmetic. Input registers provide the most local level of storage.

The five processing parts together are called the Processing Part Array (PPA). A relatively simple sequencer controls the entire PPA by selecting configurable PPA instructions. This control paradigm is very different from a VLIW architecture, despite a deceptive resemblance.

IV. MULTIPROCESSOR SYSTEM

The Montium is typically used in a heterogeneous multiprocessor system. For instance, one or more Montium cores can be used to perform digital signal processing tasks for a general purpose processor. Figure 2 shows an example of a simple reconfigurable subsystem that can be used for this purpose.

In Figure 2, four Montium processing tiles are connected by a network-on-chip. A processing tile consists of a Tile Processor (TP) and a Communication and Configuration Unit (CCU). Each processing tile is connected to a router of the network-on-chip. There are two routers in the reconfigurable subsystem of Figure 2. Routers can either be packet or circuit switched. For this tiny network, a circuit switched router is used. Both routers are connected to the AHB bridge, which connects the reconfigurable subsystem to the general purpose processor and the rest of the system-on-chip.

A. Hydra Communication and Configuration Unit

The Communication and Configuration Unit (CCU), called Hydra, implements the network interface controller between the network-on-chip and the Montium TP. The Hydra CCU provides configuration and communications services to the Montium TP. These services include:

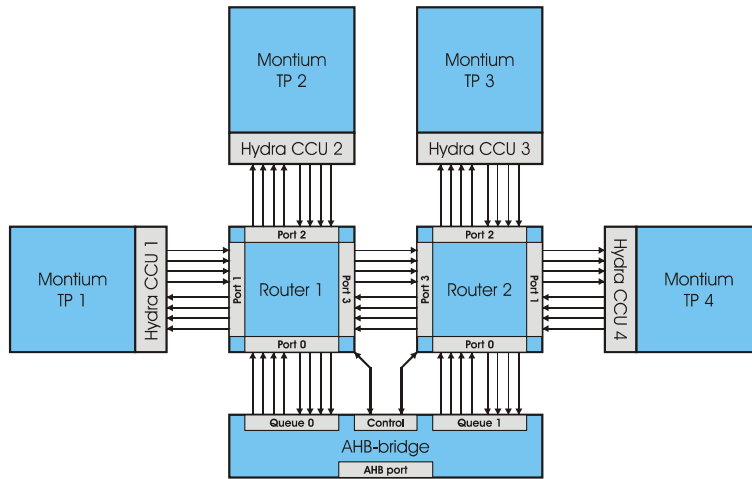


Figure 2: Example of a reconfigurable subsystem

- *Configuration* of the Montium TP and parts of the Hydra CCU itself
- *Frame based communication* to move data into or from the Montium TP memories and registers (using direct memory access)
- *Streaming communication* to stream data into and/or out of the Montium TP while computing

In streaming communication mode, data words (e.g. samples) are processed as they become available from the network. A streaming implementation of an algorithm has a higher throughput than its frame based alter ego.

B. Circuit Switched Router

A network-on-chip router is used to build a Montium TP reconfigurable subsystem (i.e. a multiprocessor). Routers are physically connected to CCUs and to other routers. Routers can also be used to create a heterogeneous system-on-chip comprising a variety of hardware modules.

The network router interface consists of two identical unidirectional physical channels. Each physical channel contains four lanes. Lanes contain data and network control signals.

In order to send data over the network, a processing tile has to send a network packet onto one of its output lanes. The packets sent by a source tile are received on an input lane of the destination tile. Up to four outgoing packets can be sent in parallel by a single tile, by using all four outgoing lanes.

C. AHB Bridge

The Advanced High performance Bus (AHB) bridge connects the reconfigurable subsystem to embedded processors, high performance peripherals, DMA controllers, on-chip memory and I/O interfaces. The AHB protocol is part of the

AMBA on-chip bus specification. AMBA is an open de facto standard for the interconnection and management of functional blocks that make up a system-on-chip [1].

V. APPLICATION EXAMPLES

The Montium TP is a flexible core that can be used for digital signal processing applications. In this section, three sample applications are highlighted: baseband processing, error correction and digital down conversion.

A. HiperLAN/2 baseband processing

HiperLAN/2 [4] is a wireless local area network access technology similar to IEEE 802.11a. In [6] a Montium TP implementation of a HiperLAN/2 receiver and simulation results are presented. The implementation uses three Montium TPs. The simulation results show that the implementation can realize the minimum required bit error rate of $2.4 \cdot 10^{-3}$ after error correction. The partitioning of the HiperLAN/2 receiver on the tile processors is shown in Figure 3.

These tile processors can meet the real-time performance requirements at fairly low clock frequencies: typically ranging from 25 to 72 MHz, depending on the function. Also, the one-time configuration overhead is low: ranging from 274 to 946 bytes. This is small enough to enable dynamic reconfiguration. Implementation details are given in Table 1.

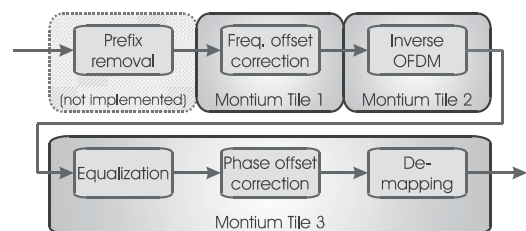


Figure 3: Partitioning of the HiperLAN/2 receiver

Table 1: HiperLAN/2 implementation details

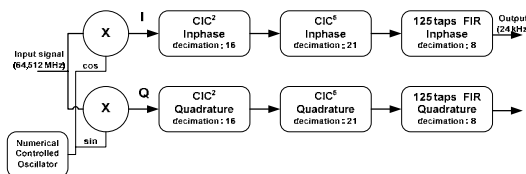
	Frequency offset correction	Inverse OFDM	Equalizer, Phase offset, de-map
Execution time in clock cycles	67	204	110
Communication time in clock cycles (32-bits/clock frame based)	128	116	100
Minimal clock frequency in MHz (32-bits/clock frame based)	49	80	53
Minimal clock frequency in MHz (32-bits/clock@100MHz frame based)	25	72	37
Minimal clock frequency in MHz (64-bits/clock streaming)	17	51	28
Configuration size in bytes	274	946	576
Configuration time in clock cycles	137	473	288

B. Viterbi forward error correction

Viterbi forward error correction [8] is used in many wireless standards, including DRM, DAB, IEEE 802.11 and DVB. In [7] an adaptive Viterbi decoder is implemented on a single Montium TP. The implemented Viterbi decoder is adaptive in many ways. Parameters such as constraint length and rate can be reconfigured dynamically, depending on the communication system required by a specific application. The decision depth of the Viterbi decoder is also configurable and can be dynamically reconfigured while the Viterbi decoder is operating.

The total configuration size of the Montium TP Viterbi implementation is 1,356 bytes. The configuration time is 6.78 μ s when configuring at 100 MHz. Once the Montium TP is configured as a Viterbi decoder, partial reconfiguration can be used to instantly adjust the constraint length, decision depth or rate. For example, the required decision depth depends on the quality of the wireless channel. Therefore, the decision depth is typically adjusted at run-time, using partial reconfiguration.

The worst-case energy consumption of a Montium TP based Viterbi decoder for Digital Audio Broadcasting (DAB) is conservatively estimated to be 24 nJ/bit. This is up to 24 times the energy consumption of an ASIC implementation of the DAB Viterbi decoder. An implementation on an ARM9 general purpose processor is estimated to dissipate about 5000 times more energy than the ASIC implementation. However, the ARM9 does not have enough processing power to deliver an output rate of 1.8 Mbit/s, which is required for DAB.

**Figure 4: DDC algorithm****Table 2: Comparison of DDC implementations**

Device	Process	Frequency	V_{dd}	Power	Die size
TI GC4016	0.25 μ m	80 MHz	2.5	115.0 mW	n.a.
Low Power ASIC	0.18 μ m	64.512 MHz	1.8	27.0 mW	1.7 mm ²
ARM922T	0.13 μ m	6.697 GHz	1.08	1.674 W	3.2 mm ²
Altera Cyclone I	0.13 μ m	64.512 MHz	1.5	141.4 mW	n.a.
Montium TP	0.13 μ m	64.512 MHz	1.2	38.7 mW	2.2 mm ²

C. Digital Down Conversion

Digital Down Conversion (DDC) is a method to reduce the sample rate by selecting a limited frequency band out of a stream of samples. By attenuating the unwanted frequencies, the signal can be resampled at a lower rate. In [2] a DDC implementation is compared for five different architectures: two ASICs, an ARM922T, an Altera Cyclone FPGA and a Montium TP. The DDC algorithm used is depicted in Figure 4 and consists of a 2-stage Cascading Integrating Comb (CIC) filter, a 5-stage CIC filter and a 125-tap Finite Input Response (FIR) filter. The input sample rate is 64.512 MHz and the output is 24 KHz. This DDC configuration is suitable for DRM.

An overview of the comparison is presented in Table 2. The ARM9 cannot perform the DDC algorithm in real-time. It needs to run at 6.697 GHz, whereas the maximum clock frequency is 250 MHz. The ARM9 power estimation considers only the power consumption of the core and excludes any memory access.

As can be expected, the customized low power DDC ASIC is the most energy-efficient. It consumes 27 mW. When scaling the technology to 0.13 μ m, this is roughly equivalent to 9 mW. The power consumption of the Montium TP DCC implementation is 4 times higher than an ASIC solution and 4 times better than an FPGA solution

VI. CONCLUSION

Shortening time-to-market constraints will drive semiconductor companies to acquire complete and flexible IP solutions that can be integrated instantly. The market for DSP cores that combine low power, high performance, flexibility and low costs is growing rapidly.

The Montium TP is a DSP core that satisfies these requirements. The Montium TP can be used as a single DSP accelerator core or clustered in a (large) reconfigurable subsystem. The core is in particular suitable for battery operated devices and embedded systems that may not generate a lot of heat, such as (portable) consumer electronics, digital communications and top-of-the-hood automotive systems.

REFERENCES

- [1] ARM, "AMBA Specification (Rev 2.0)", *ARM IHI 0011A*, ARM Limited, 1999.
- [2] Bijlsma T, Wolkotte P.T and G.J.M. Smit, "An Optimal Architecture for a DDC", *submitted to Proceedings of the 20th International Parallel & Distributed Processing Symposium Reconfigurable Architectures Workshop (RAW 2006)*, Rhodes, Greece, April 2006.
- [3] D. Edenfeld, A.B. Kahng, M. Rodgers, Y. Zorian, "2003 Technology Roadmap for Semiconductors", *IEEE Computer*, vol. 37 issue 1, pp. 47-56, January 2004.
- [4] ETSI, "Broadband Radio Access Networks (BRAN); HiperLAN type 2; Physical (PHY) Layer", *Technical Specification ETSI TS 101 475 V1.2.2 (2001-02)*, February 2001.
- [5] P.M. Heysters, "Coarse-Grained Reconfigurable Processors – Flexibility meets Efficiency", *Ph.D. Dissertation*, University of Twente, Enschede, The Netherlands, September 2004, ISBN 90-365-2076-2.
- [6] P.M. Heysters, G.K. Rauwerda and G.J.M. Smit, "Implementation of a HiperLAN/2 Receiver on the Reconfigurable Montium Architecture", *Proceedings of the 18th International Parallel & Distributed Processing Symposium Reconfigurable Architectures Workshop (RAW 2004)*, Santa Fé, New Mexico, U.S.A., April 2004, ISBN 0-7695-2132-0.
- [7] G.K. Rauwerda, G.J.M. Smit, W. Brugger, "Implementing an Adaptive Viterbi Algorithm in Coarse-Grained Reconfigurable Hardware", *Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA'05)*, pp. 62-68, Las Vegas, Nevada, U.S.A., June 27-30, 2005, ISBN 1-932415-74-2.
- [8] A.J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", *IEEE Transactions on Information Theory*, 13(2):260-269, April 1967.