Reed Electronics Group

MICROPROCESSOR REPORT

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

Motorola Attacks ASICs

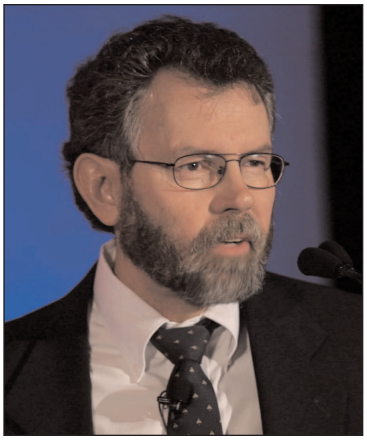
Programmable Communications Processor Offers Design Flexibility

By Tom R. Halfhill {7/14/03-01}

More frightening than any Halloween mask is the over–$1 million price tag on a deep-submicron mask set. No wonder everyone is looking for ways to exorcise the demon. Motorola's latest weapon is the MRC6011, a new chip that has a programmable RISC controller, internal peripherals, and six DSP cores, each with 16 function units. Designed primarily for wireless infrastructures, the MRC6011 is an off-the-shelf alternative to a costly ASIC project or a conventional DSP.

Motorola disclosed architectural details of the MRC6011 last month at **Embedded Processor Forum 2003**. The chip is suitable for many compute-intensive applications, but the instruction set, microarchitecture, and DSP cores make it particularly useful for baseband processing in 3G-cellular and wireless-LAN base stations.

In that role, the MRC6011 can replace a fixed-function ASIC or programmable DSP while maintaining high performance. Because it's fully programmable, field upgrades are easier than with systems based on custom ASICs. The ability to deploy soft upgrades—perhaps remotely over the network—is a valuable feature when communications protocols and industry standards are rapidly evolving.



ROSS MEHAN

Roman Robles, manager of Motorola's Digital Technologies Operation, describes the MRC6011 at EPF2003.

Reconfigurable Means Reprogrammable

By far the most interesting feature of the MRC6011 is what Motorola calls a reconfigurable compute fabric (RCF), which will also appear in future chips in this series. However, Motorola's use of the term "reconfigurable" doesn't mean the chip has reprogrammable gates, as with an FPGA. Instead, the reconfigurable elements are small DSP cores that have their own local registers, memories, and arrays of function units. Once programmed, these self-contained cores can independently execute all or part of an algorithm locally, without fetching instructions from off-core or off-chip memory, so all their I/O bandwidth is available for data throughput.

Although local execution within arrays of function units is a valid alternative to the conventional execution model of a programmable processor, *MPR* doesn't consider the MRC6011 to be a true reconfigurable architecture. We reserve the term "reconfigurable" for chips whose logic can be modified after manufacture, especially at run time. (See the sidebar, "Defining Reconfigurable Processing.")

Although the DSP cores in the MRC6011 enjoy local autonomy, they are controlled by an on-chip RISC processor and share data over an internal I/O fabric. The DSP cores can carry out a task independently or jointly, depending on the computational requirements. To simplify the programmer's model, Motorola provides a function library—focused on communications—callable from a C program. The functions are written in assembly language and run exclusively on the

DSP cores. Customers write their high-level code in C for the RISC controller, which parcels out the work to the cores as required. At that level, the DSP cores are completely transparent; C programmers don't have to know how they work. However, Motorola provides documentation and tools for those who want to write their own low-level functions in C or assembly language.

The MRC6011's DSP cores are distributed in two identical modules, each with its own datapaths for internal and external I/O. Each module has three DSP cores, a 32-bit unidirectional data-input bus, a 128-bit bidirectional DMA bus to a shared memory subsystem, and a 32-bit bus for the RISC controller. The shared memory subsystem includes the RISC controller and DMA interfaces for both modules and connects to a pair of slave I/O bus controllers, which provide 64-bit interfaces to external DSPs. The modules also share an interrupt controller and PLL. Figure 1 is a chip-level block diagram of the MRC6011.

Interestingly, the MRC6011 isn't 100% home-grown Motorola technology. Motorola licensed the DSP cores and RISC controller from Morpho Technologies, an intellectual-property (IP) provider that specializes in DSP cores. The cores in the MRC6011 are based on Morpho's MS1 rDSP (reconfigurable digital signal processor), which is available as synthesizable IP or as a hard macro. An MS1 rDSP core can include four times as many function units as Motorola implements in the MRC6011, which opens another potential door to future scalability.

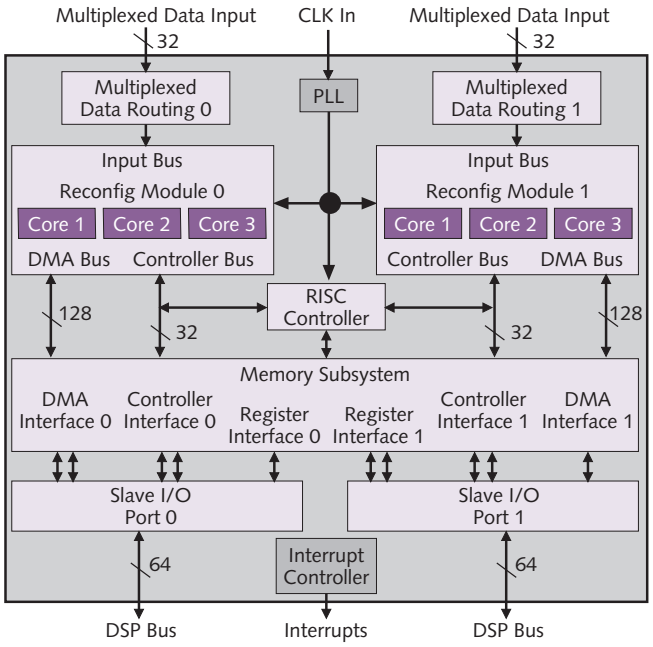


Figure 1. Motorola's MRC6011 has six DSP cores distributed in two independent modules. Separate I/O buses and datapaths provide high data throughput, with few opportunities for bus conflicts. The microarchitecture is so symmetrical and scalable that it's easy to imagine future chips in this series having only one module or additional modules.

Motorola says the MRC6011's maximum core speed will be 250MHz at 1.2V (3.3V I/O) in a 0.13-micron process. All off-core buses run at 100MHz. The packaging is a 31- × 31mm tape ball-grid array (TBGA), and power consumption is less than 3W (typical). General availability is scheduled for mid-2004.

Inside the DSP Modules

The chip-level block diagram in Figure 1 doesn't do justice to the MRC6011's complexity. Each of the six DSP cores is a miniature 16-bit processor in its own right, with all the architectural elements of a standalone microprocessor: an instruction set, multiple register files, function units, instruction/data caches, buffers, and program memory—even some peripherals. In addition, the DSP cores can communicate with each other and with other DSP cores in their companion module over a sophisticated data fabric.

Figure 2 is an inside look at a single DSP core within a module. Note that the core can receive data over the unidirectional input bus—which is fed by the multiplexed data router seen in Figure 1—or over the bidirectional DMA bus, which is the link to the shared memory subsystem. Normally, data from a communication link arrives over the data-input bus. An internal 128-bit-wide bus connects the input buffer to a 40K frame buffer, which in this context is local memory that temporarily holds incoming data, representing a brief time slot in the communication stream. Another 128-bit datapath connects the frame buffer to the processing elements.

The processing section of each DSP core has its own instruction and data caches (4K each), context memory (for storing program instructions), controller unit (with a 32-bit interface to the on-chip RISC controller), array of 16 compute nodes (function units), and local register files (one for each

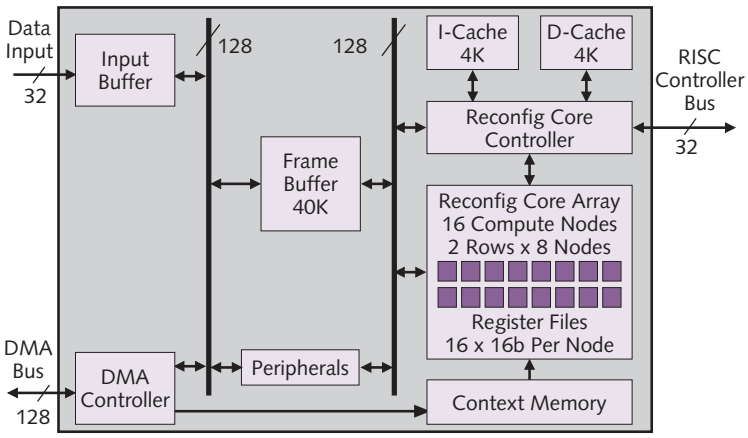


Figure 2. This block diagram shows the inner mechanism of one DSP core; there are three identical cores in each module, and two modules in the MRC6011. Once programmed, each core has enough resources to locally execute a substantial portion of a baseband-processing task without wasting any I/O bandwidth on fetching instructions from outside the core.

Defining Reconfigurable Processing

By Nick Tredennick

There is no industry-standard definition of "reconfigurable," so vendors are free to use the term as they see fit. However, *MPR* prefers a more specific definition that describes a processor or system whose logic is dynamic.

Consider a problem's solution to be composed of two elements: structure and procedure. A microprocessor manufacturer provides the structure (the chip's resources), and a programmer provides the procedure in the form of a program. These two elements cooperate in the system. If the structure is fixed and the programmer supplies only procedure, the system is not reconfigurable.

ARC International, MIPS Technologies, and Tensilica offer configurable—but not reconfigurable—microprocessor cores. An engineer determines the processor's structure when designing a chip with those cores, but once the chip is manufactured, the structure remains static. Programs run as instruction-based procedures on the fixed structure, just as they do on conventional microprocessors.

With a true reconfigurable processor, an engineer can change the run-time structure and procedure after the chip is manufactured, possibly even at run time. The fundamental requirement is that the structure must vary with time. FPGA-based systems are configurable, but rarely reconfigurable, because chips from Altera and Xilinx do not yet support run-time configuration well. FPGA-based systems could be reconfigurable.

Using a reconfigurable processor, an engineer might configure a hardware MPEG decoder for one portion of a problem and later use the same resources to build several rake receivers. In contrast, a conventional processor with fixed resources would use instruction-based programs to implement both functions.

Nick Tredennick is a long-time member of the MPR *editorial board, the former editor of a newsletter about reconfigurable processing, and an editor of the* Gilder Technology Report.

node). This mini-DSP has about 50 instructions, with heavy emphasis on the bit manipulation and arithmetic operations required for communications. For example, one instruction can perform a complex correlation that would normally require several instructions on a general-purpose processor.

Basically, the compute nodes are specialized, homogeneous ALUs tailored for communications. They can perform routine 16-bit integer operations as well as more-difficult tasks, such as complex correlations and multiply-accumulate (MAC) instructions. The 16 nodes are arranged in an 8 × 2 logical array. At most, they can execute 16 tasks, using eight different instructions per cycle (with both nodes in each column executing the same instruction). In another configuration, all eight nodes in a row can execute the same instruction while the other eight nodes in the second row are executing a different (but common) instruction in the same clock cycle.

For instance, if the first compute node in the first row executes a MAC instruction, the first node in the second row will be idle unless it also fetches a MAC. Therefore, to achieve the best-case scenario of 16 instructions per cycle, each row of nodes would have to execute an identical sequence of eight instructions. In a general-purpose processor, this would almost never happen. In a special-purpose processor with such a small instruction set that is performing repetitive operations on a datastream, it's more likely that a compute node in one row will encounter the same type of instruction as its companion node in the other row.

Each compute node has its own file of 16 programmer-visible registers, 16 bits wide. (The registers are visible only to programmers who write their own routines in the DSP core's assembly language; normally they are invisible to programmers calling the function libraries from a C program.) The compute nodes fetch their instructions from the local context memory, which can hold up to 128 different "context planes" or instruction sequences for the 16 nodes. Motorola says the MRC6011 can perform spectrum-despreading operations on as many as 192 simultaneous datastreams from 3G cellular handsets while using only about 40 context planes, leaving about two-thirds of the context memory free for other tasks.

All the compute nodes, registers, caches, and memories within a DSP core can move data among themselves in one clock cycle. Moving data to another core within the same module or another module imposes a one-cycle delay if there are no bus conflicts. If there is a conflict, bus-arbitration hardware automatically holds up the data for another clock cycle until the traffic jam clears.

Bus contentions should be comparatively rare in the MRC6011, because Motorola designed the DSP cores and modules to operate independently, using local resources whenever possible. Tolerating a one-clock-cycle penalty for an occasional bus contention is a deliberate design tradeoff. Motorola says that a larger datapath fabric with a direct, single-cycle link between every DSP core in each module would have greatly inflated the chip's size and cost. Indeed, some other array architectures, such as Elixent's D-Fabrix (also presented at **Embedded Processor Forum 2003**), have such complex interconnect fabrics that their arrays resemble checkerboards—the dense webs of datapaths linking the function units seem to occupy as much space as the function units themselves.

To simplify programming on the MRC6011, it's unnecessary for programmers to specify a datapath when moving

Price & Availability

Motorola will sample the MRC6011 to key customers in 4Q03, with general availability scheduled for mid-2004. Pricing has not been announced. For more information about the MRC6011, see *http://e-www.motorola.com/*. For more information about Morpho Technologies, see *www.morphotech.com/*.

data anywhere on the chip, either locally within a DSP core or globally among the other cores and modules. Even when programming the cores in assembly language, a move instruction implicitly selects the best available datapath between the source and destination. The instruction syntax resembles a register-to-register move in a conventional microprocessor.

Where Are the Benchmarks?

The MRC6011 isn't sampling yet, so it's too soon to demand certified EEMBC benchmarks that quantify the chip's performance. Even when the chip becomes available, measuring its performance will be a challenge; Motorola will probably have to rewrite the benchmark code to take advantage of the MRC6011's parallelism. A device as complex as the MRC6011 defies simple expressions of performance in MIPS, BOPS, or GOPS.

Theoretically, if the 8 × 2 array of compute nodes in each of the six DSP cores achieves its ideal of executing 16 operations per clock cycle, the peak performance would be 24,000 MIPS at 250MHz. (This is the maximum data-plane performance; the on-chip RISC core is for control-plane processing.) However, this might understate the MRC6011's performance as measured by other chip vendors that count a MAC instruction, for example, as two operations. If the MRC6011's compute nodes are executing single-cycle MACs and complex correlations, the total operation count would be even higher.

Motorola prefers to express the MRC6011's performance in terms of instruction throughput, not atomic operations. By that measure, the MRC6011 can execute 24,000 16-bit MACs or 48,000 4-bit correlations at 250MHz. What's more relevant to customers is that the MRC6011 can replace an ASIC or conventional DSP to perform the chip-rate functions of a 2.5G or 3G baseband processor in a cellular base station.

By pairing three MRC6011 chips with two conventional 16-bit DSPs (Motorola recommends the Motorola MSC8126, naturally) and a network-interface chip (such as Motorola's PowerQuicc II or III), developers can build a WCDMA (wideband code-division multiple access) base station that would be capable of handling the uplink, downlink, chip-rate, symbol-rate, and random-access channel (RACH) processing for 96–128 voice users or 20 data users at 384kb/s. The same base station could also handle multiple HSDPA (high-speed downlink packet access) users.

No doubt a custom ASIC could do better, but it would require a risky 12- to 18-month development project with higher up-front costs, and it would be less flexible after deployment. A programmable DSP would offer more flexibility than an ASIC, but Motorola says the MRC6011 can do the chip-rate processing tasks of multiple DSPs, because it's more integrated and is optimized for communications. In addition, Motorola provides an integrated development environment (Metrowerks CodeWarrior) that allows programmers to write their software in C or assembly language for both the MRC6011 and the DSPs required for symbol-rate processing—with compilers, assemblers, debuggers, and simulators for both architectures.

The stiffest competition for the MRC6011 in wireless base stations will be the DSPs and off-the-shelf ASICs from Texas Instruments, which offer high performance and can eliminate the need to develop a custom ASIC. To beat the competition, the MRC6011's (yet-to-be-determined) price will have to be more attractive. Customers will also appreciate the MRC6011's flexibility, and the 3W chip will probably consume less power than a less-integrated solution based on conventional DSPs and ASICs.