

Cradle Chip Does Anything

“Universal Microsystem” Is Most Flexible Chip Ever, But Is the Market Ready?



by Peter N. Glaskowsky

Imagine that you want to design a full-custom processor to meet the exact needs of your application. You don't have time to design an ASIC, and you don't want to learn a hardware description language like Verilog or VHDL. A conventional microprocessor would be too slow, and a field-programmable gate array (FPGA) can't run your C code. You can't even find a media processor with the right I/O interfaces. What you want is an off-the-shelf chip that can be programmed with the software *and* hardware elements of your design.

You are the target market for Cradle Technologies' new Universal Microsystem (UMS). Cradle was established in 1995 as a project at Cirrus Logic and became independent in 1998. It plans to introduce the first UMS chip early next year.

Cecil Kaplinsky, Cradle's chief technical officer, announced the UMS at this week's Microprocessor Forum. The UMS combines a parallel array of microprocessor and signal-processor cores with programmable logic and protocol engines to handle all I/O. A simple DRAM controller is the only hardwired peripheral. As Figure 1 shows, all elements of the UMS design communicate over a single global bus. This bus moves code and data among external memory, the local memory blocks within each processing element, and the I/O protocol engines at a peak rate of 5.1 GBytes/s.

Programmable I/O Offers Unique Flexibility

The most interesting aspect of the UMS architecture is the inclusion of reprogrammable logic for I/O. All of the pins on

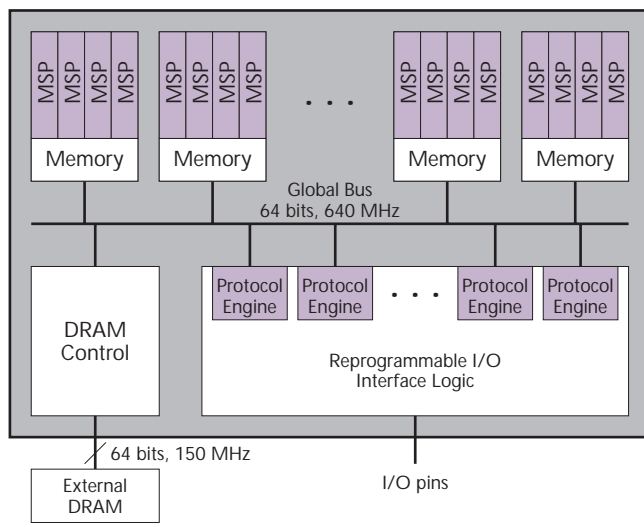


Figure 1. Only the DRAM controller in Cradle's Universal Microsystem is hardwired. Everything else is programmable.

the device, except for those used by the DRAM controller, are connected to a single seamless logic array. Customers will divide the array into sections for each I/O interface according to the number of pins, or the amount of logic, required to implement the interface. If the desired interface requires more logic than is associated with the necessary number of pins, the section can be expanded to use additional logic from nearby pins.

According to Cradle, each group of eight I/O pins is associated with FIFOs, a clock-distribution system, and a RAM-based complex programmable logic device (CPLD) to handle low-level handshaking. Altogether, about 32,000 gates of logic are associated with each eight-pin group.

The programmable logic is assisted by protocol engines distributed around the periphery of the chip. Protocol engines are not directly connected to specific pins, but there will be one protocol engine for every 10–30 pins. Cradle says that in the first UMS chips, the protocol engines will provide more than 2.5 billion operations per second of total I/O processing power and 1 GByte/s of aggregate I/O bandwidth.

The protocol engines are connected to the on-chip global bus, which gives them access to memory and additional processing power. The protocol engines are responsible for most of the UMS's I/O customization. The protocol engines will be programmed in assembly language or C; the programmable logic will be designed using BOOL, a logic-design language.

I/O protocols too complex to be handled solely on a protocol engine, such as the multilayer Ethernet-TCP/IP stack, can be shared with the UMS's processor array.

Cradle Provides I/O Library

Cradle has built a library of predefined I/O interfaces that includes IEEE-1394, Ethernet, SCSI, and PCI blocks. Customers can purchase these library elements for use in their products or design their own, using Cradle's tools. Cradle will broker sales of customer-designed elements to other UMS customers, and it will encourage third parties to design additional I/O blocks.

The unique Cradle architecture is not compatible with any existing silicon intellectual property (IP). Before Cradle can hope to win over chip architects accustomed to buying design elements in VHDL or Verilog form, it must find a way to achieve compatibility with existing IP libraries—or to redesign them for the UMS architecture, which seems impractical, given the wide range of IP available.

Cradle offers a 1394 link-layer controller design as an example of a UMS interface block. The design uses 13 pins and their associated logic plus one protocol engine. In

comparison, Cradle says, a conventional ASIC would need about 70,000 gates of logic to perform the same function.

The 1394 example illustrates one of the few limitations of the UMS architecture. In its first implementation, the UMS cannot be used to build a 1394 physical-layer interface chip, because its pin drivers support only digital I/O.

Logic levels may be defined by the customer, but there are hardware limits to the achievable voltage swing and current drive. More advanced physical-layer signaling standards, including 1394, Gigabit Ethernet, Rambus's Direct RDRAM, and USB, are outside the range of the UMS's capabilities. These interfaces will require external devices to translate between UMS I/O logic levels and the specific requirements of each interface.

Analog I/O is not supported at all. The UMS cannot be used to make a single-chip graphics controller, for example, because it does not offer an integrated digital-to-analog converter. The absence of this capability puts the UMS at a disadvantage for set-top box designs and other consumer-electronics products, many of which must interface to external analog devices. For now, customers needing analog I/O must also use external converter chips that will add cost to UMS-based designs.

Cradle says it is developing technology that will allow it to provide more-flexible I/O drivers and limited mixed-signal capability in future chips.

As valuable as the UMS programmable I/O architecture is for some applications, it may not be a good fit for others. For new I/O standards with low pin counts but highly complex protocols, such as System I/O (see MPR 9/13/99, p. 4), Cradle's design may not offer the right balance of logic and pins. In applications that require only one or two narrow interfaces, the unneeded pins and I/O circuitry will waste a significant portion of the chip area and add unwanted cost. Cradle, therefore, will need chips with a fairly wide range of I/O capabilities.

DRAM Control Handled by Hardwired Logic

Only one external device was considered ubiquitous enough to deserve a dedicated interface: memory. The UMS architecture includes a memory controller for off-chip DRAM. The controller can be configured to manage various types of SDRAM, including DDR SDRAM. It does not support synchronous graphics RAM, Enhanced's ESDRAM, or Direct RDRAM. Typical configurations will use a 64-bit interface at speeds up to 150 MHz.

The 1.2 GBytes/s of bandwidth available from such a memory array should be enough for the majority of applications that Cradle contemplates for its early UMS chips. More bandwidth will be required if Cradle hopes to compete with

3D-graphics chips in video-game consoles, mainstream PCs, or professional workstations. Existing 3D accelerators already have 3 GBytes/s—or more—of bandwidth to their local-memory arrays.

In normal operation, the memory controller services transfer requests from the UMS's global bus. These requests may be initiated by dozens of independent sources within the chip. Cradle has not provided a global coherency scheme for the UMS. Applications that require coherency must manage it in software.

The UMS has an independent controller for flash-memory devices. These devices are used to store the initial configuration of the I/O logic as well as code for the various programmable cores on the chip. The flash-memory operates in parallel and I²C serial modes, using the programmable I/O pins, but these pins may be reused for other purposes once the UMS is configured.

Global Bus Runs Fast

The UMS's global bus is the sole communications channel for all on-chip resources. The bus is 64 bits wide and operates at 640 MHz, providing 4.2 GBytes/s of sustained bandwidth. This is well below the potential demand from the chip's processors and I/O, but Cradle believes that most applications can be written to concentrate bandwidth demands in the memories local to each processing element.

Cradle chose a shared bus to connect PEs and I/O rather than multiple point-to-point interconnects. This decision simplifies the UMS's programming model, reduces cost, and speeds the development of

new UMS devices. Additional members of the UMS family can be created simply by connecting additional processors or I/O units to the global bus. Point-to-point communication schemes, by comparison, could require much more wiring as complexity increases.

The shared bus also creates a limit to scalability, however. As additional processors and peripherals are connected to the bus, arbitration delays and other overhead make the bus less efficient. Cradle is evaluating other connection schemes for future products.

Quads Contain Processors, Local Memory

The fundamental unit of processing in the UMS design is a "quad"—four RISC processors combined with eight signal processors, memory, and support logic. A complete UMS chip consists of the I/O and memory controllers plus multiple quads.

Figure 2 shows the internal configuration of a quad. Though Cradle defines one RISC processor element (PE) plus two digital signal engines (DSEs) as a multistream processor (MSP), there is no physical association of these



Cecil Kaplinsky, Cradle's chief technical officer, describes the Universal Microsystem.

processors within the quad. Each PE can direct tasks to any of the eight DSEs within the quad.

There are separate local buses for program and data traffic. PEs connect to both, while DSEs use only the data bus. PEs transfer instruction words to the local memory in the DSEs over the data bus. Execution of these instructions is controlled by the PEs, using semaphores and interrupts.

RISC Engines Handle Conventional Code

Cradle defined its own 32-bit RISC architecture for the PEs. The company wanted better code density than was available from existing RISC designs, so it created a new instruction set that uses 16-bit instruction words with opcode compression. Instruction types include two-operand integer and floating-point operations compatible with IEEE-754 (except for some exception-handling issues). The PE includes a 32-entry 32-bit register file.

The first implementations of this new architecture will use a four-stage nonoverlapped pipeline, completing one instruction every four clocks at 320 MHz for a peak processing rate of 80 MIPS. This simple design increases the effective performance per transistor and simplifies the programming model. With no pipeline stalls or other hazards, program execution is very predictable but slow.

Digital Signal Engines Excel at Simple Tasks

For improved performance, inner loops of signal-processing applications can be executed on the eight DSEs within each quad. These DSEs may be assigned to separate tasks or lumped together to work like a simple eight-wide vector processor.

Most DSE instructions use a two-operand format. A fully pipelined multiply-accumulate operation takes a third two-bit operand to specify one of four accumulators. DSEs can handle either integer or floating-point data, but floating-point support is limited to a subset of IEEE values—exponents range from -32 to +46 only.

Most instructions (including the multiply-accumulate operation) are completed in one clock cycle at a nominal

320 MHz, giving each DSE up to 640 MFLOPS of floating-point throughput. A sum-of-absolute-differences operation is also available that computes four eight-bit integer differences and one summation per cycle.

Each DSE has 96 32-bit registers. Instructions are 20 bits wide and are stored in a local SRAM array containing 384 20-bit words.

Quads Have Local Memory, DMA Engine

Each quad is also equipped with 16K of data memory and 12K of program memory. Each is 64 bits wide and operates at 320 MHz, ensuring that all local resources can access the local memory without contention. These memories can be configured as SRAM or as one, two, or four independent caches assigned to the PEs.

Cradle says the PEs typically use less than 35% of the available bandwidth of these memories; the rest remains available for transfers among quads or to main memory. Each quad includes a DMA engine that can handle most of this data movement, allowing the PEs and DSEs to concentrate on data processing. These engines can move data among all the quads, memory, and I/O over the global bus.

The coarse-grained parallelism of the UMS architecture allows Cradle's customers to assign tasks to individual PEs or MSPs, minimizing contention for shared resources. Table 1 shows the number of MSPs needed to handle various applications, according to Cradle.

Because each quad contains about 40 KBytes of local program and data storage (or less, if some local memory is configured as cache), it can take a significant amount of time to save or restore the state of a quad for a specific task. Quads contain enough local memory to store the state of individual PEs or DSEs, however, so applications can perform some task switching within the quad.

Cradle says each quad includes about 3.2 million transistors and occupies about 13.5 mm² in the initial 0.25-micron process. Operating at 320 MHz, each quad is rated at 6 GFLOPS and 320 MIPS, yet it typically consumes less than 450 mW. This unusually low power consumption makes the UMS an attractive choice for portable electronics and networking products that require dense packaging.

Design Tools Are Critical

To design a chip like the UMS may be easier than to configure and program it for a complex application. Some previous attempts to use software-based media processors to replace fixed-function devices have been spectacular failures. For example, MicroUnity's "soft"

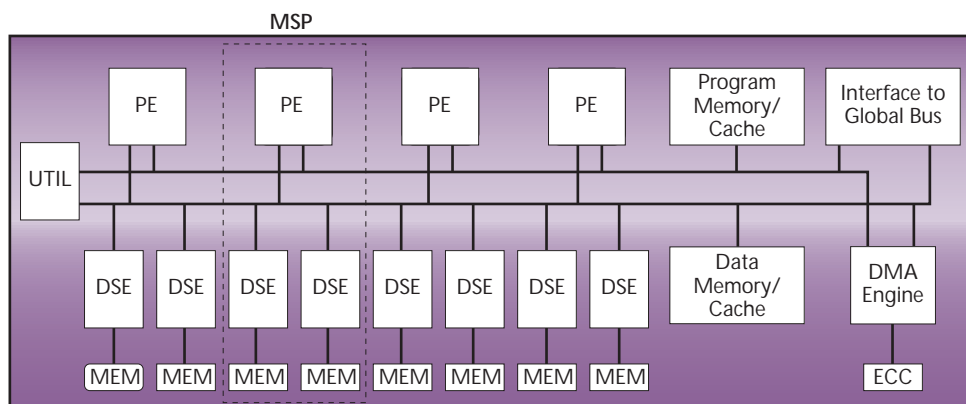


Figure 2. UMS processors are combined in blocks known as quads. Each quad contains four RISC processing elements (PE), eight digital signal engines (DSE), two cache memories, and associated logic. A multistream processor (MSP) comprises one PE and two DSEs.

Algorithm	Type	MSPs Required
MPEG-2 Decode	6 Mbits/s	4
	15 Mbits/s	6
AC-3 Audio Decode	5 channels	2
	V.90	0.5
Modems	G.Lite ADSL	3
	per 100Base-T	0.5
Ethernet Router with QoS Support	per Gigabit Ethernet	4
	for 1.6 Mpolys/s	4
DV Codec	Camcorders	8

Table 1. Cradle offers these estimates of the number of MSPs (one-fourth of a quad) required by various algorithms.

broadband communications controller was responsible for some of the most impressive performance projections of its time—but no products.

Cradle hopes to avoid a similar fate by limiting the scope of its plans to specific applications it is sure it can implement. The UMS architecture is also much more modular than early media processors, which shared a single high-performance processor core among all the algorithms required for the entire system. These early designs had high theoretical performance but low effective throughput, because of resource contention and task-switching delays. Cradle believes its approach makes these problems much easier to prevent.

UMS development won't be easy, however. Customers must develop code to manage four different types of programmable elements (PEs, DSEs, DMA engines, and the I/O protocol engines), each of which is optimized for specific tasks. Data movement must be conducted over a single shared bus, and no cache-coherency hardware is available.

Cradle will offer a unified software-development environment for the UMS, complete with the necessary compilers and simulators. The software simulator can also be linked to a real UMS chip via a special back-door interface that makes visible all features of the hardware.

Cradle plans to make the development system available at no charge to interested parties. Some minimum level of developer support (probably email based) will be included at no extra cost, and Cradle will offer extra-cost support options for developers who need more help.

A beta version of this development system has been in the hands of 30 developers at 13 companies for about six months, and Cradle reports these companies are making good progress on several representative applications.

Can UMS Be All Things to All Users?

No single UMS chip will be suitable for every application. Some applications will always be a better fit on hardwired ASICs or conventional microprocessors. Even applications well suited to the UMS architecture will need only a certain number of pins and processing elements. Nevertheless, a reasonable variety of UMS implementations, covering a wide range of cost and capacity, could meet the needs of a large fraction of today's chip designs.

For More Information

Cradle expects to introduce the first chips in its Universal Microsystem family in 1Q00. The company says the family will include chips priced below \$50 with over 30 GFLOPS of floating-point performance. For more information, visit the company's Web site at www.cradle.com.

Among the first markets being pursued by Cradle are Ethernet switches and cellular base stations. A single UMS chip would provide all the processing needed to handle 24 100Base-T ports plus two Gigabit Ethernet ports. Services such as virtual-LAN management, high-level filtering, and quality-of-service monitoring could be provided at a lower cost than with current solutions, Cradle believes.

Cradle has high hopes for the future of UMS. The company speaks of building a \$100 billion "dynasty" of UMS-based products by 2010 and proclaims, "The ASIC as we know it is dead."

Cradle's only hope for such success lies in its ability to adapt to the needs and design styles of the world's chip designers. It must offer analog I/O options, Verilog and VHDL compatibility, and comprehensive libraries of low-cost IP for the UMS before it can get a foot in the door at most fabless semiconductor companies. It must also demonstrate, with real chips in production systems, that the UMS architecture can deliver on its promise of high performance at low cost.

If it is truly faster, easier, and cheaper to implement sophisticated chip designs in a UMS device than in a conventional ASIC, Cradle's success is virtually guaranteed. With ten times the theoretical performance of a high-end desktop PC microprocessor at one-tenth the price—and reconfigurable I/O to boot—the UMS is sure to attract attention.

If, however, the complex UMS mix of processors and programmable logic is not as easy to configure as Cradle promises, the performance potential may not be realized. The total cost of a UMS-based solution could greatly exceed the cost of the chip. In such an event, the company's four-year effort will fall short of expectations.

More likely, Cradle will fall into the market niche currently occupied by high-end programmable logic devices. These chips, from vendors such as Altera and Xilinx, are often used for prototype and limited-volume hardware and find occasional design wins in expensive high-margin systems as well. This niche offers a reasonable opportunity for moderate success, though volumes are low.

Competition is much more intense in the high-volume markets Cradle seems focused on today. Networking equipment and consumer electronics consume hundreds of millions of complex ASICs each year. Most of these chips are highly optimized for specific tasks. It won't be easy for Cradle to displace any of these suppliers, but its Universal Microsystem could be just the right tool for the job. \square