

# M I C R O P R O C E S S O R

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

## XAP3 TAKES THE STAGE

*Synthesizable 32-Bit Processor Targets Deeply Embedded Applications*

*By Tom R. Halfhill {6/13/05-01}*

From ARM's hometown of Cambridge, England, comes a new licensable embedded-processor core—except it's not from ARM. It's from Cambridge Consultants, a 250-person engineering firm that's been around since 1960. For decades, this company has been

designing electronic gadgets for customers all over the world, but it's a newcomer to selling 32-bit microprocessors.

Cambridge Consultants entered the processor business to gain additional control over its projects, retain more of the intellectual property (IP), accelerate development, and reduce project risk. Owning the processor IP also recaptures some of the revenue escaping to other suppliers. In 1994, Cambridge Consultants introduced its first embedded-processor core, the 16-bit XAP1 (pronounced "zap one"). The 16-bit XAP2 followed in 1999. And at last month's **Spring Processor Forum**, the 32-bit XAP3 entered the spotlight. Now Cambridge Consultants is competing, to some degree, with 32-bit embedded-processor cores from heavy hitters like ARM, ARC International, MIPS Technologies, and Tensilica.

Although Cambridge Consultants is drawing upon its experience with 16-bit processors, the XAP3 represents an entirely new 32-bit RISC architecture. The first implementation, XAP3a, is a simple von Neumann design with a unified memory bus and two-stage instruction pipeline. A future core, the XAP3b, will adopt a Harvard-style bus architecture with separate I/O buses for instructions and data, plus a deeper pipeline and additional features.

Technically, there's little to distinguish the XAP3a from other 32-bit synthesizable processor cores for deeply embedded applications. It's optimized for good code density and low power, not high throughput, and it's similar to the ARM7TDMI-S, ARC 600, MIPS32 4K family, and small configurations of Tensilica's Xtensa LX. In general, competing

processors deliver higher performance and/or have smaller cores than the XAP3a does.

However, one undeniable attraction of the XAP3a is its relatively low price, especially for high-volume designs. After forking over the licensing fee (\$380,000 for a single-core, single-project license), customers don't have to pay royalties, no matter how many chips they manufacture. Other processor-IP vendors usually demand royalties in addition to costlier upfront license fees.

### **Flexible Instruction-Set Architecture**

Many engineers dream of starting with a blank slate and creating a new instruction-set architecture (ISA) that rights all the wrongs they've suffered from whatever architecture they're using now. The engineers at Cambridge Consultants used their opportunity to create an economical RISC architecture specifically for deeply embedded systems. (Over the past 45 years, the company has designed everything from utility meters and circuit breakers to Bluetooth radios and electronic contraceptive aids.) Their design goals were low power, low cost, high code density, position-independent code, and easy software porting to XAP3 from other CPU architectures. Overall, they have achieved those goals.

Not surprisingly, the XAP3 ISA leans more toward ARM's ascetic version of RISC than toward the canonical MIPS version, which originally targeted high-performance workstations. For instance, the XAP3 architecture has only 16 general-purpose registers (GPR), not the customary

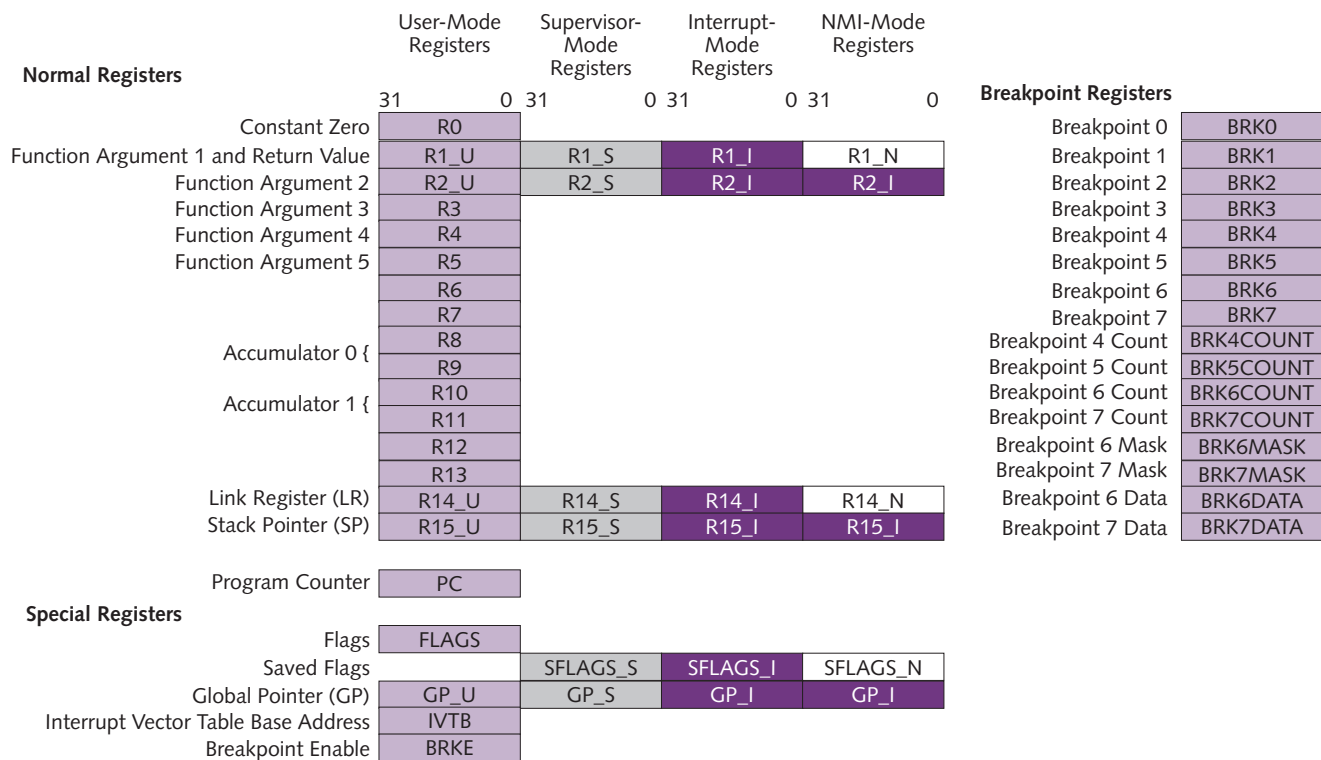
complement of 32 GPRs found in the MIPS ISA and most other RISC architectures. The smaller register file requires only four bits for register addressing instead of five bits, allowing XAP3 instructions to use longer immediate values and providing other benefits that improve code density. Although a smaller register file increases the chance of a resource dependency that could reduce throughput, it's not important for a simple embedded processor like the XAP3. Figure 1 shows the layout of the register files.

Another, more significant example of the XAP3's economy is its cleverly optimized instruction set. To improve code density, the ISA defines 16-bit-long instructions in addition to the standard 32-bit-long RISC instructions. Of course, all other licensable 32-bit processors also have ISAs with 16-bit instructions, so that alone isn't new. But in the XAP3 ISA, hybrid 16/32-bit instructions were designed into the architecture from the start, not retrofitted later, so they have interesting characteristics.

One interesting feature is that Cambridge Consultants carefully defined the subset of 16-bit instructions to duplicate the most common 32-bit operations. Although 16-bit instructions account for only 25% of the instruction set (owing to the limited number of bits available for encoding the opcodes), they typically account for 65–70% of the assembled or compiled program code. Such a heavy mix of 16-bit instructions should reduce overall code size by about one-third.

Better yet, programmers needn't explicitly choose between the duplicate 16- and 32-bit instructions when coding in assembly language or ANSI C. Mnemonics for duplicate versions of a 16/32-bit instruction are identical. The ANSI C compiler generates these mnemonics and feeds them into the XAP3 assembler, which automatically substitutes the 16-bit version of an instruction for a 32-bit instruction whenever possible. Result: programmers get excellent code density with virtually no extra effort.

To pack instructions into memory more efficiently, the XAP3 ISA allows 16- and 32-bit instructions to reside at any half-word memory address, with no restrictions on memory alignment. Any 32-bit word of memory can hold one 32-bit instruction, two 16-bit instructions, two halves of different 32-bit instructions, or a 16-bit instruction plus half of a 32-bit instruction. Nor must the processor switch modes to execute either type of instruction. The XAP3a can transparently fetch, decode, and execute any blended stream of 16- and 32-bit operations. Internally, the processor expands 16-bit instructions into 32-bit constructs before decoding them. In all these respects, the XAP3 ISA is more like ARC's 16/32-bit ARCompact ISA than ARM's 16-bit Thumb extensions for the ARMv5T ISA. (See the sidebar, "ARCompact: An Elegant 16/32-Bit ISA," in *MPR 2/18/03-06*, "Soft Cores Gain Ground," and *MPR 3/27/95-01*, "Thumb Squeezes ARM Code Size.")



**Figure 1.** The XAP3 32-bit RISC architecture defines a relatively small set of general-purpose registers. Note that two pairs of 32-bit registers can serve as accumulators for instructions that return 64-bit results or manipulate 64-bit operands. The XAP3 ISA has several such instructions for accelerating simple signal-processing tasks.

A minor oddity of the XAP3 architecture is that it's little-endian. ARC, ARM, MIPS, and Tensilica are biendian (configurable at design time). Cambridge Consultants says little-endian byte ordering is more convenient when fetching instructions from a branch-target address, because the least significant bit in each opcode is a flag indicating whether the instruction is 16 or 32 bits long. Therefore, the processor instantly discovers the length of the first instruction when branching to a new instruction stream. In the deeply embedded applications for which the XAP3a is intended, endianness has no significant effect on performance.

### DSP Instructions and Multiple Privilege Levels

Several instructions in the XAP3 ISA can accelerate digital-signal processing, although Cambridge Consultants is the first to note that XAP3 has no ambitions of replacing a DSP. It's merely becoming commonplace for embedded RISC processors to add DSP-type instructions or extensions. (See the sidebar, "MIPS24KE: Better Late Than Never" in *MPR 5/31/05-01*, "White Paper: The MIPS32 24KE Core Family.")

Although the XAP3a lacks a true multiply-accumulate (MAC) instruction, it can perform a 32- × 32-bit multiply that returns a 64-bit result in a pair of GPRs addressed as a single 64-bit accumulator. (As Figure 1 shows, the XAP3a has two of these accumulators.) A subsequent add instruction can then operate on the 64-bit result. Another instruction can perform a 32- × 32-bit multiply and shift the 64-bit result by up to 63 bits. Of course, these are multicycle operations, but the XAP3a executes most other instructions in a single cycle.

Other DSP-like features include an instruction for finding the most significant bit in a word (useful for normalization), an absolute-value instruction, some byte-swapping instructions, and six flip instructions (which, among other things, are useful for inverting those little-endian data words). In all, the XAP3 instruction set has 175 32-bit instructions, 47 of which are duplicated as 16-bit operations. Programmers familiar with other RISC architectures should be comfortable with the XAP3 ISA.

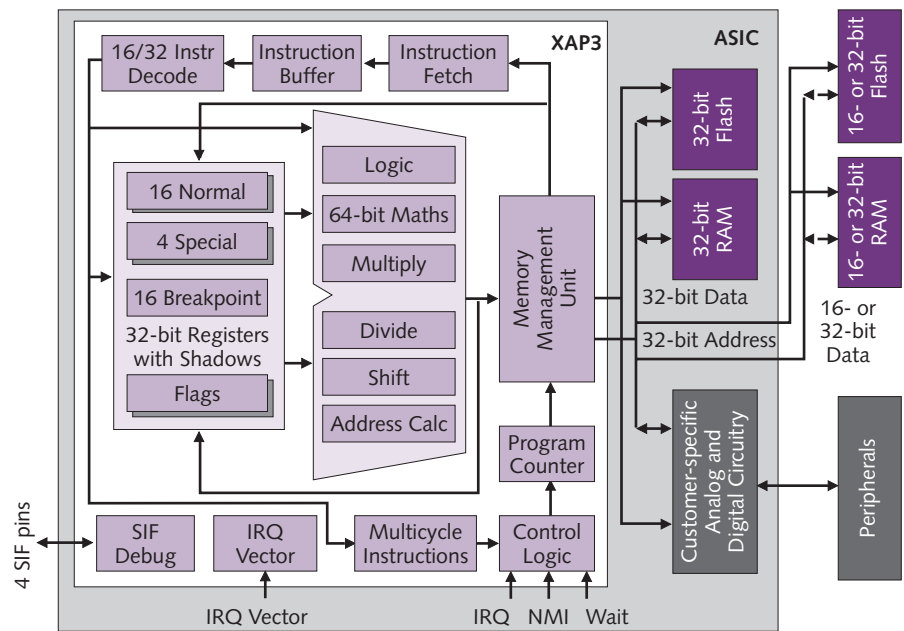
For applications requiring more-powerful signal processing, Cambridge Consultants has offered a DSP coprocessor core since 1993 that also works with the XAP3a. Called the Arithmetic Processing Engine (APE), it's more like a customizable state machine than a programmable DSP. An APE-generator tool lets customers specify the word width, the configuration of datapaths, the numbers of adders and shifters, and other features.

### Price & Availability

Cambridge Consultants offers a single-project, single-core, royalty-free XAP3a license for \$380,000. That price includes a synthesizable Verilog model of the processor and a one-seat license for the *xIDE* software-development tools. Available at extra cost are additional tools, FPGA development boards, and the APE DSP coprocessor. Everything is available now. For more information, see [www.cambridgeconsultants.com/ASIC](http://www.cambridgeconsultants.com/ASIC).

Using this input, the generator creates an application-specific signal-processing engine in register-transfer-level (RTL) Verilog, as well as compatible software-development tools. APE can integrate its own local memory for program code and data, and it attaches to the XAP3a's 32-bit memory bus. Figure 2 shows a block diagram of the XAP3a.

Thanks to 32-bit addressing, the XAP3a can access up to 4GB of linear memory without segments or bank switching—an advantage over the 8- and 16-bit embedded processors it may replace. Data types from 8 to 32 bits long can reside at any byte address, and programs are position independent, because code structures are located at relative offsets from the program counter and global pointer. Unlike some simple microprocessors, the XAP3a supports multiple privilege modes, so the system can isolate user tasks from an operating system running in supervisor mode. (The XAP3a currently



**Figure 2.** XAP3a block diagram. Although the diagram shows a memory-management unit (MMU), it's really an interface controller, not a true virtual-memory MMU. The 32-bit memory interface is a straightforward proprietary design that doesn't support any standards such as AMBA, CoreConnect, or OCP. All memory, peripherals, application-specific logic, and coprocessors attach to the main I/O bus.

Feature	Cambridge XAP3a	ARC ARC 600	ARM ARM7TDMI-S	MIPS 4KE	Tensilica Xtensa LX
Architecture	XAP3	ARCompact	ARMv4T	MIPS32	Xtensa
Arch. Width	32 bits	32 bits	32 bits	32 bits	32 bits
Configurability	None	High	None	Medium	High
Instr Lengths	16/32 bits	16/32 bits	16/32 bits	16/32 bits	16/24 bits
Endianness	Little endian	Biendian	Biendian	Biendian	Biendian
Core GPRs	16	32	16	32	16
Pipeline Depth	2 stages	5 stages	3 stages	5 stages	5 or 7 stages
Branch Predict	—	Static	—	—	—
Privilege Levels	4	1	3	2	4
DSP Instructions	Few	Optional	—	Yes	Optional
Caches (I / D)	—	0–32K	—	0–64K	0–32K
MMU	—	—	—	Yes	—
FPU	—	Optional	—	—	Optional
Size*	40K–50K gates	27K gates	0.32mm <sup>2</sup>	1.0–1.7mm <sup>2</sup>	20K gates
Max Frequency	100MHz <sup>†</sup>	290MHz <sup>‡</sup>	133MHz <sup>‡</sup>	233MHz <sup>‡</sup>	350MHz <sup>‡</sup>
Availability	Now	Now	Now	Now	Now

**Table 1.** Cambridge Consultants' XAP3a has much in common with small 32-bit embedded-processor cores from ARC, ARM, MIPS, and Tensilica. All these synthesizable processors are cacheless or have no-cache options. Competing processors tend to have advantages over the XAP3a in core size, clock frequency, and configurability. \*Base configuration. †0.18-micron CMOS. ‡0.13-micron CMOS.

runs Micrium's  $\mu$ C/OS-II real-time operating system, and Cambridge Consultants says  $\mu$ C-Linux will be running soon.)

For software development, Cambridge Consultants provides a graphical integrated development environment called *xIDE*, which runs on multiple platforms (Windows, Linux, Unix, and Mac OS) and supports GNU C/C++ (GCC) and ANSI C compilers. The tool chain includes an assembler, debugger, and instruction-set simulator, but not a cycle-accurate simulator. In addition, the tools support a noninvasive debug interface, multiprocessor designs, and a development board that can emulate a XAP3a in a Xilinx FPGA.

### Not Smallest or Fastest, but Affordable

Unlike ARM, Cambridge Consultants doesn't discourage customers from deploying its processor cores in FPGAs instead of ASICs. Naturally, performance will suffer in an FPGA, because the XAP3a isn't optimized for synthesis in programmable logic, unlike the synthesizable processor cores inexpensively licensed by Xilinx and Altera. (See *MPR 5/17/05-02*, "MicroBlaze Can Float," and *MPR 6/28/04-02*, "Altera's New CPU for FPGAs.")

Even in CMOS, the XAP3a isn't exactly a speed demon. The processor's abbreviated two-stage pipeline is an obvious limitation—even the austere ARM7TDMI-S has a three-stage pipe. Simulations indicate the XAP3a will run at

about 20MHz when optimized for size in a 0.18-micron fabrication process, and it should reach 100MHz in that process when optimized for speed. (Cambridge Consultants hasn't yet estimated the clock frequency of the XAP3a in a 0.13-micron process.) Those metrics put the XAP3a in the lower range of performance when compared with similar processor cores from ARC, ARM, MIPS, and Tensilica. Table 1 summarizes the features of these processors.

Despite its simplicity, the XAP3a isn't the smallest 32-bit synthesizable processor core, either. Cambridge Consultants estimates the processor will require 40,000 gates when optimized for size and 50,000 gates when optimized for speed. (Both estimates include the simple MMU and debug interface.) To further reduce core size, customers can omit the decoding logic that expands 16-bit instructions into 32-bit operations, saving about 5,000 gates. The obvious trade-off is that program code will require more memory. (Programmers can set flags to tell the assembler not to use 16-bit instructions when targeting a 32-bit-only configuration.)

The XAP3a is a little larger than similar processors. For example, the cacheless base configuration of Tensilica's Xtensa LX is about 20,000 gates, and a similar ARC 600 configuration is about 27,000 gates. Of course, the configurable ARC and Tensilica cores grow much larger when extended with their numerous optional features. The XAP3a is about the same size as the cacheless ARM7TDMI-S and is equally unconfigurable. But making these size comparisons is like debating how many gates will fit on the head of a pin, because all these cores occupy mere motes of silicon in a deep-submicron fabrication process.

Cambridge Consultants says it's not trying to compete head-to-head with other processor-IP vendors. The company's main goal is to offer its engineering-consultancy customers the choice of an affordable 32-bit processor in addition to the 16-bit XAP1 and XAP2. (Also in the works is a 16-bit XAP4 based on the XAP3 ISA.) With relatively affordable licensing fees and no downstream royalties, the XAP3a is an attractive option for customers designing small embedded systems on a tight budget. ♦

To subscribe to Microprocessor Report, phone 480.483.4441 or visit [www.MDRonline.com](http://www.MDRonline.com)