Hobbit Enables Personal Communicators

AT&T Announces Complete Chip Set for Low-Power Applications

By Linley Gwennap

AT&T has unveiled its Hobbit CPU along with a set of four chips that provides the core logic for a complete Hobbit system. The four chips are the system controller, a PCMCIA controller, a peripheral controller, and a video display controller. AT&T Microelectronics previously announced a low-voltage modem chip set that interfaces directly to the system controller. The entire chip set is currently sampling, with full production expected by next January. AT&T would like to establish the Hobbit design, in combination with the PenPoint operating system from GO Corporation, as a standard for personal communicators; the first such system is expected to be from EO, a joint venture between AT&T, Matsushita, and Mirubeni. Rae McLellan, one of the original CRISP architects, presented the Hobbit chips at the Microprocessor Forum.

The CRISP Architecture

The Hobbit CPU implements the CRISP architecture *(see* ***060201.PDF****)*, which is designed for optimal execution of C programs. CRISP uses both RISC and CISC features and cannot be clearly assigned to either camp. Like RISC designs, CRISP has a small instruction set with few addressing modes, and most instructions are executed in a single cycle. But the RISC model is broken by CRISP's variable-length instructions, which can be either two, six, or ten bytes in length. Bell Labs' architects selected the most commonly used instructions and addressing modes to be encoded into the two-byte instructions. Depending on the application, 60% to 80% of all instructions use the short form.

Another unusual feature is CRISP's register-less design; all instructions are memory-to-memory. Data can be accessed using absolute addresses, but CRISP also provides stack-offset addressing; this mode allows two five-bit offsets to be included in a two-byte instruction, concisely encoding these memory references. To speed program execution, CRISP machines use an on-chip *stack cache* to hold the topmost portion of the stack. The stack cache provides fast access to local and temporary variables and subroutine arguments that, according to McLellan, make up 80% of all data references in typical C programs. This design allows a CRISP subroutine to immediately begin using data from the stack without having to perform numerous loads and stores. A single CRISP instruction allocates stack space for the subroutine and flushes any stack overflow data to main memory.

The stack cache not only eliminates the execution

Address Type	Mnemonic	Function	Address Type	Mnemonic	Function
2 and 2-1/2 Address	ADD	Addition	1 Address	JMP	Unconditional jump
	SUB	Subtraction		JMPTY	Jump if true, predicted taken
	MUL	Multiplication		JMPTN	Jump if true, predicted not taken
	DIV	DIvision		JMPFY	Jump if false, predicted taken
	REM*	Remainder		JMPFN	Jump if false, predicted not taken
	AND	Bitwise logical AND		~~ifCjmp~~	Jump if carry (replaced by TESTC)
	OR	Bitwise logical OR		~~ifOjmp~~	Jump if overflow (replaced by TESTV)
	XOR	Bitwise logical exclusive OR		CALL	Call function
	SHL	Unsigned shift left		KCALL	Call kernel
	SHR	Arithmetic shift right		ENTER	Allocate stack space for function
	USHR	Unsigned shift right		RETURN	Free stack space and return from function
2 Address	~~UMUL~~	Unsigned multiplication (deleted in Hobbit)		POPN*	Free stack space (N entries)
	UDIV	Unsigned division		CATCH	Fill stack cache
	UREM	Unsigned remainder		LDRAA*	Load PC-relative address into accumulator
	CMPEQ	Equality comparison		FLUSHPBE*	Flush a prefetch buffer cache entry
	CMPGT	Signed greater-than comparison		FLUSHPTE*	Flush a page table entry
	CMPHI	High comparison (unsigned greater-than)	0 Address	NOP	No operation
	MOV	Move		KRET	Return from kernel call
	MOVA	Move address		CRET*	Return from kernel call with new stack
	DQM*	Double- or quad-word move (block move)		TESTC*	Copy carry bit to flag bit
	ADDI	Addition with interlock		TESTV*	Copy overflow bit to flag bit
	ANDI	Logical AND with interlock		CLRE*	Clear PSW enter guard bit
	ORI	Logical OR with interlock		FLUSHI*	Flush the decoded instruction cache
	TADD*	Tagged addition		FLUSHP*	Flush the prefetch buffer cache
	TSUB*	Tagged subtraction		CPU	Read internal CPU register

Table 1. Hobbit instruction set summary. (*indicates additions to the original CRISP instruction set; ~~strikethrough~~ indicates deletions.)

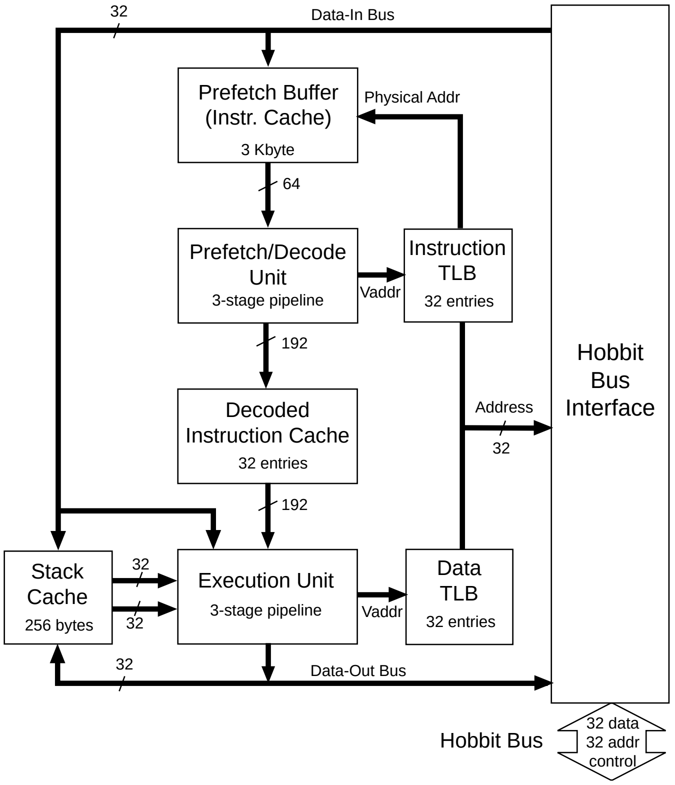


Figure 1. Hobbit CPU block diagram.

of load and store instructions, but it also reduces the code size of programs because these instructions are not needed. Compared to a RISC program, the only loads and stores that remain are accesses to global variables or other information not stored on the stack. These accesses may be coded as "moves" from main memory to the stack, or they may use absolute addressing instructions that require six- or ten-byte encoding. By reducing the number of loads and stores, eliminating branch delay slots, and making frequent use of two-byte instructions, CRISP programs are much more compact than typical RISCs. For example, the compiled image of the portable C compiler for CRISP is 42% smaller than the MIPS version and 48% smaller than SPARC, and it is even slightly smaller than the 386 version. McLellan did not have data to confirm if this reduction is typical for other CRISP applications.

The Hobbit CPU implements several changes to the original CRISP instruction set (see Table 1). *Jump if carry* and *jump if overflow* have been replaced by *test carry* and *test overflow*, so that the only bit tested by condition jumps is the flag (true/false) bit. The carry and overflow bits are now "unsticky," meaning that they can be tested, but they don't affect the results of future calculations. Two new "tagged" instructions write their destination only if the low two bits of both operands are zero, allowing for run-time type checking in object-oriented languages.

The POPN instruction deallocates stack space without executing a return, providing support for tail recursion. The DQM instruction performs generalized block moves. The UMUL instruction was deleted; MUL can be used to generate unsigned as well as signed results. The coprocessor instructions were eliminated; the Hobbit chip does not support an external floating-point processor. Unlike the original big-endian CRISP, the Hobbit CPU operates in either big- or little-endian modes.

The two-byte instruction encodings were changed to increase the percentage of instructions that use them. Three encodings were changed and two were added. Also, some of the short operands in the two-byte encodings are now scaled; for example, stack pointer offsets are multiplied by four. These and other changes make Hobbit code incompatible with previous CRISP implementations, but that shouldn't bother anyone outside of AT&T.

The Hobbit CPU

The Hobbit CPU is being marketed as the ATT92010. Figure 1 shows a block diagram of the CPU. Instructions are loaded into a 3K, three-way set-associative on-chip cache. The associativity, along with the greater code density of the CRISP architecture, increases the hit rate of this cache to roughly that of an 8K direct-mapped cache in a typical RISC architecture. A semi-autonomous prefetch unit further improves the hit rate, although portable systems may disable this feature to reduce memory accesses.

The decode unit converts the variable-length instructions into 192-bit control words. The decoded instruction cache holds 32 of these values. Using branch prediction bits from the opcode, the decode unit includes *predicted next instruction* and *alternate next instruction* addresses in each control word. This technique, called *branch folding*, eliminates the execution of branch instructions and allows zero-cycle branching. If the branch is mispredicted, the worst-case penalty is three cycles if the actual target is in the decoded instruction cache, or six cycles if it is in the undecoded cache. Because CRISP uses separate condition-setting and branch instructions, this penalty can be reduced by placing one or more instructions between the condition-setting instruction and the branch.

The stack cache holds the top 256 bytes of the stack, providing twice the on-chip storage of a typical 32-entry register file. The chip also includes separate 32-entry TLBs for data and instructions. No floating-point unit is available, although the integer unit performs multiply and divide operations. Multiplies take 4 to 20 cycles using an "early out" Booth algorithm, while divides take 38 cycles. The die is just 92 mm^2 and incorporates 419,000 transistors. It uses a 0.9-micron (drawn), two-layer-metal process. *(For more information on the Hobbit CPU, see* ***060201.PDF****, which gave a detailed and accurate description of this chip eight months before it was announced.)*

Like the entire Hobbit chip set, the CPU can operate at 20 MHz with 3.3V power, or up to 30 MHz at 5V. Even at the lower voltage, the chip is rated at 27K Dhrystones (version 2.1), or 13.5 VAX MIPS. Performance will suffer on larger benchmarks due to the small cache. No SPECmark data is available because AT&T has not ported UNIX to Hobbit. In general, SPECint ratings are somewhat less than VAX MIPS ratings. With minimal (10 pF) loading, the CPU consumes only 250 mW at 3.3V when running at top speed. AT&T has priced the chip at $35 in quantity, about the same as a 16-MHz 386SL.

The Hobbit System

The complete Hobbit chip set implements all the core functions needed by a personal communicator, as shown in Figure 2. The AT&T components are shown in gray. Additional functions (such as SCSI, LAN, or floppy disk controllers) can be added using standard ISA chips or PCMCIA cards.

At the core of the system is the Hobbit Bus, with 32 data lines and 32 address lines (non-multiplexed), running at the same frequency as the CPU. The bus supports up to five bus masters, with arbitration resolved by the System Management Controller (SMC) chip. To facilitate this arbitration, each potential bus master has its own bus request (BREQ) and bus grant (BGRANT) pins; the SMC receives all bus requests, selects one, and asserts BGRANT to that device. Only single-, double-, and quad-word transfers are supported, but a device may continue to issue transactions once it has gained control of the bus, as long as its BGRANT continues to be asserted. Including the overhead cycles, the sustainable bandwidth of the bus is over 40 Mbytes per second at 20 MHz.

The Hobbit Bus supports interlocked transactions to allow for the read-modify-write operations of the CRISP instruction set. Bus retry, bus error, and reset signals provide for exception handling. The separate address and data lines make it simple to add ROMs or other devices to the bus without latches or buffers; the specification allows long-latency devices to operate in a tri-state mode so that they do not tie up the bus.

The bus arbitration logic uses a round-robin algorithm to select a new bus master after a device relinquishes the bus. The SMC will terminate a device's bus access if the device reaches its transaction limit, preventing any one device from taking over the bus for too long. The transaction limit for each device can be independently programmed by software. The SMC takes control of the bus after reset, honoring requests only from the CPU until the rest of the system has been initialized.

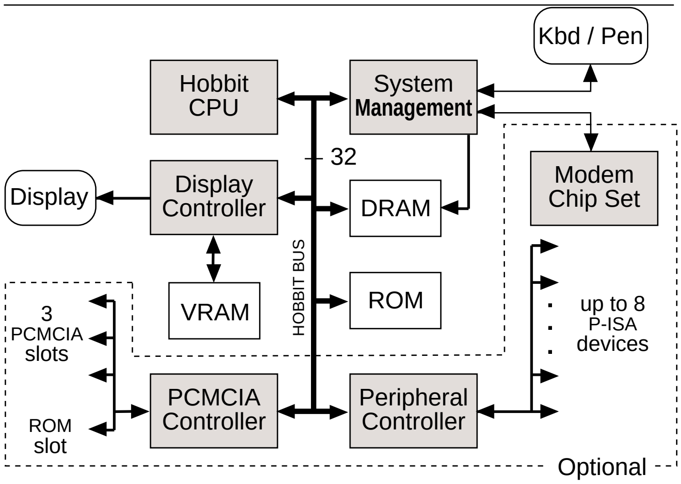


Figure 2. Hobbit system block diagram. (AT&T parts in gray.)

System Design for Low Power

The Hobbit chip set includes several features to reduce the overall power consumption. Operating at 3.3V reduces power by 56% from standard 5V chip sets, since power is proportional to the square of the voltage. The CPU integer core (everything except the instruction cache) uses just 250K transistors, compared to the 400K to 500K used by most high-performance processors, and the floating-point unit is left out entirely, further reducing power consumption.

The chip designs make frequent use of gated clocks, which dynamically disable portions of the chips that are not being used. Power-hungry precharge logic and sense amps are virtually eliminated from the chip, taking advantage of the relatively leisurely minimum clock cycle of 33 ns. The CPU uses a loopback technique to eliminate the need for pull-up resistors, which continually use power; after reading data from the Hobbit Bus, the CPU immediately drives that data back onto the bus to prevent signal transitions. The DRAM controller uses the same trick when driving the memory signals.

The entire Hobbit chip set uses a fully-static design—the clocks can be completely stopped (on clock phase 1) and restarted later without any loss of state. With its clock stopped, each chip consumes an insignificant 0.05 mW of power. The SMC takes advantage of this feature by selectively disabling the clock of any device in the system (including the CPU) that is not being used. The SMC can monitor interrupt signals and restart the clocks when an event requires attention.

To further reduce power, the SMC can shut down the entire system, including itself. All clocks are stopped, but power is maintained to the chips. The SMC is able to (optionally) refresh the DRAM in this mode so that no memory contents are lost. During shutdown, the SMC can restart the system after a real-time clock interrupt or if its ONOFF signal is asserted.

The core chip set (not including the modem or memory) requires about 1 Watt of power when the clocks are running, assuming a reasonable system bus load of 100 pF. If clocks are disabled during periods of system inactivity, the average power consumption will be even lower.

System Management Controller

Along with controlling Hobbit Bus arbitration and the chip clocks, the SMC includes a set of functions intended to reduce the part count of a Hobbit system. It provides a memory controller, two serial ports (one asynchronous, one synchronous), 256 bytes of battery-backed memory, a real-time clock, and eight parallel I/O pins. It also contains the interrupt control logic and generates the quadrature system clocks. (The Hobbit chips use a pair of 20-MHz clocks that are 90° out of phase, so the chips can internally divide each system clock cycle into four phases.)

The memory controller timing is fully programmable with an accuracy of one-quarter clock period, and can handle two different types of memory simultaneously. For DRAMs, the SMC can handle fast-page-mode reads and writes, and it takes care of memory refresh. The chip supports up to 32 Mbytes of DRAM using sixteen 16-Mbit parts in two banks. Some (or all) of main memory may be implemented using SRAM, but the SMC does not allow the SRAM to be configured as a second-level cache.

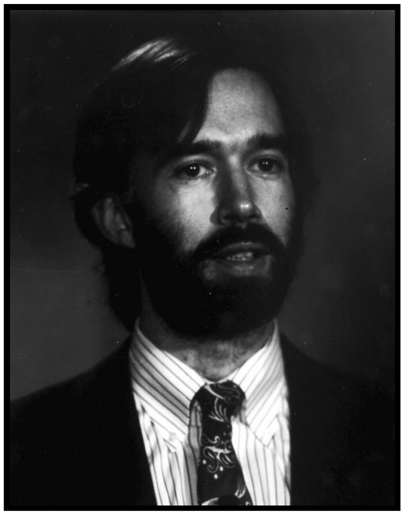


PHOTO BY ANNE HAMERSKY

"The architectures of the more RISC-like machines [Hobbit, ARM] have much more headroom... for performance enhancements. The Intel x86 architectures are pretty much at their limits.... The extension of their performance is taking a good bit of parallel hardware to achieve higher performance."

Rae McLellan, AT&T Bell Laboratories

The serial port provides an interface equivalent to the National 16450 UART. It includes 64-byte FIFOs for send data, receive data, and receive status. The serial data pins can be configured to operate at 5V even when the rest of the SMC is running at 3.3V. The baud rate clock uses an external crystal and can be disabled when not in use to save power. The UART has a single interrupt for a variety of error conditions, along with separate interrupts for transmit buffer empty and receive buffer full.

A synchronous serial port provides a PC/AT keyboard interface. It can also be used to connect to a digitizer for pen-position input. The maximum clock rate is 2 MHz. It has a single-character output buffer and a 16-character input buffer. Like the UART pins, the clock and data pins can operate at 5V while the rest of the SMC runs at 3.3V. Only a single interrupt is provided, for the receive buffer.

A battery-backed portion of the chip maintains 256 bytes of static memory for storing system configuration and other information. This area also contains the 47-bit real-time clock, which measures up to 2^{32} seconds (136 years) with a resolution of about 30 µs. An alarm register will generate an interrupt at any specified value of the real-time clock. Three interval timers can also be programmed to generate interrupts on a regular basis.

Besides the interrupt sources listed above, the SMC has eight programmable I/O pins that can also be configured to generate interrupts, or can be used for simple I/O. The SMC contains a set of five interrupt registers that can map any of the 24 interrupt sources to one of the five interrupt signals on the Hobbit Bus.

Video Display Controller

The Video Display Controller (VDC) connects directly to either color or grayscale LCD displays and, with an external RAMDAC, can simultaneously drive a CRT as well. The latter mode allows a portable computer to display slides for a presentation, for example. Three display sizes are supported directly: 640 × 240, 640 × 480, and 1024 × 768. Smaller sizes can be handled by mapping the physical display into a larger logical size.

The VDC displays up to nine gray levels (3-bit grayscale plus an overlay bit) at a time. These can be selected from a palette of 17 gray dither patterns (4 bits plus an overlay bit). The overlay bit creates a separate "ink" plane that can quickly display pixels drawn by a pen without using the slower rendering software.

For color displays, the 17 patterns can be applied to each of the three channels, resulting in a total of 4,913 color dither patterns, from which nine colors can be used at any one time. The chip supports LCD displays with four, eight, or twelve data lines. It can handle double-tiered LCD displays, which use two physical panels to create a larger logical display. Both interlaced and non-interlaced CRTs are supported.

The video memory must be implemented with 1 Mbit VRAMs, providing up to 512 Kbytes of memory with four chips. Only a single VRAM is needed for a 640 × 240 display. The VRAM timing is programmable and fast page mode is used whenever possible.

The VDC connects directly to the Hobbit Bus and can be accessed as either a slave I/O device or a memory-mapped device. The VDC cannot become a master on the Hobbit Bus. No graphics acceleration is included; the chip provides a simple frame-buffer interface to the CPU.

Like the serial interface on the SMC, the VDC's display interface and VRAM interface can operate at 5V while the rest of the VDC runs at 3.3V. Three reduced-power modes are available: the screen can be frozen (no VRAM writes), the screen can be turned off, or the entire video subsystem can be turned off.

PCMCIA Controller

The PCMCIA Controller (PMC) connects the Hobbit Bus to up to three PCMCIA slots. A fourth PCMCIA device can also be attached, but it must be hardwired because no card-detect signals are provided for the fourth slot. These slots can provide additional RAM, applications stored in ROM, or I/O devices. Each slot can support up to 64M. Because the current PCMCIA specification only allows 5V devices, the PMC drives these signals to 5V while the rest of the chip can operate at 3.3V.

The CPU accesses the PMC as a slave device, gaining direct access to all memory and I/O spaces of the four PCMCIA cards. The PMC can also act as a bus master to perform block move accesses (BMA). BMA is similar to normal DMA except that there are no provisions for flow control between the PMC and the PCMCIA device; instead, transfers are controlled by a 20-bit counter.

The PMC fully supports release 2.0 of the PCMCIA protocol. Both 8- and 16-bit devices are allowed. The PMC detects when a card is inserted and interrupts the CPU, which typically turns on power to that slot and configures the new card. If there is an error on the PCMCIA side, the PMC simply does nothing until the CPU times out and notices the problem. If a PCMCIA device hangs the bus, the CPU can reset it through the PMC.

P-ISA Peripheral Controller

The P-ISA Controller (PIC) connects the Hobbit Bus to as many as eight devices using what AT&T calls the Peripheral ISA (P-ISA) bus, a subset of the ISA bus. Portable systems will probably not include full ISA slots, but P-ISA allows standard ISA chips to be included, so that the system designer can quickly and easily add new functions. P-ISA is similar to ISA except that bus masters are not supported, and only a subset of the DMA and interrupt functions is used. P-ISA can provide faster transfer rates than ISA by using programmable set-up, access, and hold-off times. The P-ISA definition has no physical specification; system designers can implement non-ISA form factors or no backplane at all if they wish.

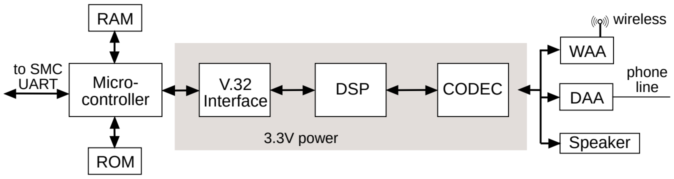


Figure 3. "Data Pump" low-power modem chip set.

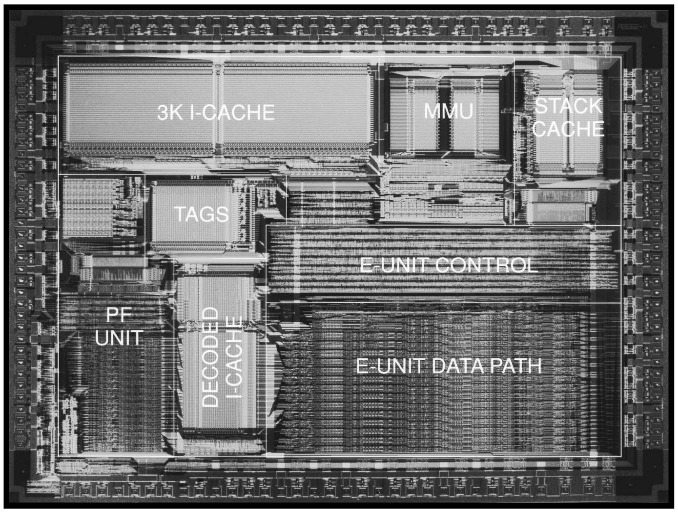
When accessed as a slave device, the PIC allows the CPU to directly access each P-ISA device through its own 16M address space. Both memory and I/O devices are allowed. The PMC can also act as a Hobbit Bus master to perform DMA accesses. Four DMA channels are available for use at the same time. Each channel permits DMA chaining, which can automatically reload and restart DMA activity. Alternatively, the PIC will interrupt the CPU when a DMA transfer completes. As with PCMCIA, only rudimentary error handling is provided for the P-ISA bus.

The PIC also includes eight general-purpose inputs (GPI) and four general-purpose outputs (GPO). The GPI signals can be edge- or level-sensitive and can be used as status or interrupt inputs. The GPO lines can be used as status outputs. They can also be individually programmed as clock outputs running at the system clock frequency or at integer fractions from 1/2 to 1/6. The GPI and GPO signals, as well as all the P-ISA signals, can operate at 5V while the rest of the PIC runs at 3.3V.

Modem Chip Set

AT&T also announced a low-power version of its V.32bis "Data Pump" chip set. The new chips are compatible with the 5V Data Pump, which is already in production. The chip set can send and receive data at 14,400 bits per second over normal telephone lines. It implements the V.17 fax standard and V.32bis for data. With appropriate interfaces added, the chip set supports either wired or wireless data transmission.

The Data Pump includes three chips (see Figure 3). The CODEC converts analog telephone-line signals to a digital bit stream, which is processed by the DSP. The V.32 interface chip controls the data flow and administers the V.32 protocol. A fourth chip, the microcontroller, connects the V.32 chip to a UART and allows the use of V.42 and V.42bis compression proto-



Die photo of the Hobbit CPU. The chip uses 419,000 transistors on an 11.1 mm × 8.5 mm die.

cols. In a Hobbit system, the microcontroller connects directly to the UART port on the SMC chip.

At this time, AT&T does not have a 3.3V microcontroller available for this chip set, but the 5V part can be used instead. The company expects to introduce a full 3.3V solution in the future, and it also plans to reduce the number of chips required from four to two. The Data Pump chips are currently available in standard PQFPs or, to reduce board-space requirements, small-outline packages (SQFPs).

Design Support and Testability

AT&T provides a range of design support tools for the Hobbit chip set. Documentation and training courses are currently available. Software development tools are available for either DOS or Solaris (Sun) platforms; the DOS version includes an integrated debugging environment. In November, GO will offer hardware and software developers' kits for the Hobbit version of PenPoint.

AT&T currently sells a development board that includes the Hobbit CPU along with 128K of secondary-cache SRAM and 8M of DRAM. (The cache, which uses a kludged 386 cache controller, is a relic from a previous design.) This board can be plugged into an ISA-bus system and uses standard ISA peripherals. In November, AT&T will have a second development board that uses the complete Hobbit chip set. The development boards come with a symbolic debugger that runs under Microsoft Windows, providing a graphical interface to all CPU registers and caches.

Manufacturing support is provided by a JTAG port on each chip. This port allows full scan-in and scan-out of all internal chip states. Basic self-test code is provided with the development board.

Price and Availability

The Hobbit CPU (ATT92010) uses a 132-pin JEDEC PQFP and is priced at $35. The system controller (ATT92011), PCMCIA controller (ATT92012), P-ISA peripheral controller (ATT92013) and video display controller (ATT92014) all use the same 208-pin EIAJ PQFP. The price for the four-chip set (P-ISA chip not included) is $100. With the P-ISA chip, the price is $124. All pricing is in quantities of 10,000. All chips are currently sampling and are expected to be in production by January. For more information, contact AT&T Microelectronics, Personal Communication Systems, 4995 Patrick Henry Drive, Santa Clara, CA 95054; 800/372-2447.

The 3.3V V.32bis "Data Pump" chip set includes the V32-INTFC interface chip, the DSP16A digital-signal processor, and the CSP1027 CODEC. Both the V32 and the DSP16A use either an 84-pin PQFP or a 100-pin SQFP. The CODEC uses a 44-pin PQFP or a 48-pin SQFP. The new Data Pump chip set is expected to sample in December, with volume production in 1Q93 and is priced at $79 in quantities of 10,000. For more information, contact AT&T Microelectronics, Dept. AL-520404200, 555 Union Blvd, Allentown, PA 18103; 800/372-2447, fax 215-778-4106.

Competitive Products

The Hobbit chips, aimed at the portable market, compete primarily against low-power x86 chips such as Intel's 386SL and AMD's 386SXLV. Apple's selection of the ARM610 for its handheld "Newton" product makes that chip another important competitor. None of the major RISC vendors has designed a chip set to offer the functions and low power needed for portable applications, although system vendors such as Tadpole and Sony have built laptop RISC systems by slowing the clock on standard SPARC and MIPS processors.

According to McLellan's figures, the Hobbit CPU is four to five times faster than either the 386SXLV or the 386SL on the Dhrystone benchmark, when all three processors are running at their maximum frequency at 3.3V. Furthermore, the AMD part uses 485 mW at that speed, while the Intel processor uses more than a Watt. The Hobbit CPU, by comparison, uses about 360 mW to drive the same load (50 pF). Thus, the Hobbit chip has a 5× to 10× advantage in Dhrystones per Watt.

It is not clear how this advantage will translate into real application performance. GO, which supports PenPoint on both x86 and Hobbit, reports that Hobbit systems run only about twice as fast as 386SL systems. One reason for the discrepancy is that PenPoint is object-oriented and thus has poor code locality, reducing the effectiveness of the small Hobbit cache. AT&T believes that system performance is also being limited by the slow 3.3V ROMs that are currently available and expects faster ROMs to be used in final products.

The ARM610, which is a cost-reduced version of the ARM600 (see µPR 12/18/91, p. 8), can match the Dhrystone performance of the Hobbit chip and probably has similar application performance. To do so, however, the ARM chip must run at 25 MHz using 5V power. At this voltage, the chip requires about 525 mW with no load; ARM does not quote power for a loaded device, which is generally significantly higher. The company is currently requalifying their part for a 3V supply; assuming that the ARM610 can still reach 25 MHz at the lower voltage, power consumption could be about the same as the Hobbit CPU. If the lower voltage reduces the ARM's clock rate (which is likely), the chip may not reach the overall Hobbit performance level. Even so, its Dhrystones per Watt could still be similar to the AT&T processor's.

The ARM610 incorporates similar functions as the Hobbit CPU. It has an integer core, a 4K cache, and a 32-entry TLB. Both the cache and the TLB are unified instruction and data. The ARM cache is slightly bigger than the total of the Hobbit CPU's stack cache and instruction cache, and its 64-way associativity improves its hit rate. These advantages are balanced by the better code density of CRISP as compared to ARM, which uses fixed-length 32-bit instructions.

One of the reasons that Apple prefers ARM over Hobbit is its flexible MMU, which Apple helped specify. Along with mapping a variety of page sizes, the ARM MMU also has an extensive protection mechanism. An unusual feature, which the Hobbit MMU does not implement, is called "domain access control." This feature lets Apple's object-oriented operating system quickly change the access rights of groups of objects.

The Hobbit chip set offers a much more complete range of functions than ARM's system chips. (The ARM system chips have been integrated into the ARM250, which is described in the following article.) The ARM chips are not qualified for low voltages. Important functions such as an LCD display controller are missing, as well as standard peripheral buses such as ISA or PCMCIA. The power management features of the Hobbit SMC are not available from ARM. Although Apple may be developing chips to bridge ARM's functionality gap, these system chips are unlikely to be offered on the merchant market.

For low-power x86 chips, several chip sets provide many of the functions of the Hobbit chips. LSI/Headland, HK Technology, VLSI Technology, and others currently sell low-voltage system-logic chip sets, with a number of other vendors soon to enter the market *(see 061304.PDF)*. AT&T may be the first to offer a 3.3V PCMCIA interface.

At $35 (in quantities of 10,000), the Hobbit CPU is roughly the same price as the 16-MHz 386SL, the 25-MHz 386SXLV, and the ARM610. A low-voltage x86 system-logic chip set costs $40 to $60, depending on the vendor and the functions that are included, bringing the price near the Hobbit chip set price of $100. The ARM system chips cost about $38.

Conclusions

The next wave of portable computing devices will require new technology and new standards, driven by the need for low power, low cost, small size, wireless communication, and ease of use. The Hobbit chip set addresses many of these needs. It is optimized for low voltage and low power consumption, both in its chip design and its software power-management capabilities. The cost is reasonable and will certainly decline over time. The CPU and SMC integrate a range of functions, allowing for physically small systems. The performance of the current chip set is adequate for a pen-based GUI such as PenPoint, but future versions are needed to allow advanced ease-of-use features such as handwriting recognition, voice recognition, and real-time video.

AT&T claims the Hobbit chips are "communications-oriented," but the only point that backs this claim is the ability to connect to modem and other communication chips. As the company moves to more aggressive IC technologies, it plans to incorporate more functions in the core chip set, eventually evolving to a single-chip personal communicator with on-board DSP capability.

The Hobbit chips deliver much higher performance, as measured by Dhrystone, than current x86 chips at roughly the same price, and they use less power than the x86. Intel's forthcoming 486SL will close the performance gap, but it is expected to cost much more than the Hobbit CPU and will probably need more power. The ARM610 CPU matches the Hobbit chip in price and performance, but at its current 5V specification, the ARM chip uses much more power and it lacks merchant availability of a full set of support chips. A next-generation ARM chip running at 3V with improved system logic may be able to match the Hobbit chip set, but by that time, AT&T will have improved their product as well. ♦

Our next issue will include a report on the personal communicator market, including a look at Hobbit-based products (see 061509.PDF).