

Matsushita Premieres 32-Bit Processors

MN10300 Series Offers Little Not Found in Competitors' Chips

by Jim Turley

Like the last kid left on the playground after choosing teams, Matsushita seems to be the only company that hasn't developed its own processor or sided with one of the licensed designs. Rather than take its ball and go home, the company worked up its courage, put on its game face, took a mighty swing—and hit a weak ground ball.

In Matsushita's case it doesn't matter, because the company is in a league of its own. Being the world's largest maker of consumer-electronics goods (sold under the Panasonic brand name) means you can buy enough chips from yourself to make them volume leaders. Even if the company never sells its new CPUs to another customer, the chips will still probably achieve economic success.

The lyrically named MN10300 family provides yet another spin on the low-cost/low-power/high-performance formula pursued by so many companies working to gain a technical advantage in high-volume consumer products. The 10300 architecture is a mixture of old microcontroller features and newer RISC-inspired thinking. The first chips in the series are already in production, and new devices—including one with support for Windows CE—are in the works for 1999. Matsushita, naturally, will be using the chips in future Panasonic audio and video gear.

Cramped Register Set Collides With C

Matsushita emphasizes the 10300 family's advantages in code density, low power consumption, and high-level language support, particularly C. Matsushita expects few of its customers will do any assembly-language programming, preferring to use Matsushita's own C compiler.

That C compiler will have a tough time allocating registers, because there are so few. As Figure 1 shows, the 10300 has only eight registers: four for addresses and four for data. The programmers' model is like that of a 386, or half of a 68000.

Also like an x86, the 10300 has many variable-length instructions. Instructions can be as short as one byte or as long as eight. Very few of the 46 instructions are inherently more than one or two bytes long; longer instruction words are almost always a side effect of long-displacement addressing modes.

Loads are usually of 32-bit quantities. The 10300 can zero-extend bytes and halfwords; to create sign-extended loads, compilers must pair a zero-extended load with a sign-extension (EXTB or EXTH) instruction.

Apart from loads and stores, no instructions can reference memory directly; the 10300 is a load/store machine. With only four data registers, assembly-language programmers will have to juggle operands almost continuously while making heavy use of the stack. Seeming to recognize this, Matsushita provides the 10300 with good parameter-passing ability for calling (and returning from) subroutines.

For example, the MOV_M (move multiple) instruction is the quickest and most general way to push or pop more than one operand on or off the stack at a time. (Other instructions are better suited for function calls.) The peculiar syntax of MOV_M allows the compiler to pick and choose a few of the general-purpose registers to copy; the others are pushed as a group. Specifically, MOV_M can specify only D2, D3, A2, and A3 individually; the chip's other seven registers (D0, D1, A0, A1, MDR, LIR, and LAR) are handled *en masse*, identified with an "other" bit in the instruction encoding. Thus, programmers are encouraged to use the lower half of the register file for their most volatile operands.

Math Skills Remain Basic

Integer addition and subtraction (ADD and SUB) work as one might expect. Address and data registers may be added to or subtracted from each other, and any immediate value (8-, 16-, or 32-bit) can be added to any register. Add with carry (and subtract with borrow) works only on data registers, so there's no signed arithmetic on addresses, which is not unusual.

Integer multiplication is iterative, as the 10300's designers skimped on the hardware multiplier to save space. One

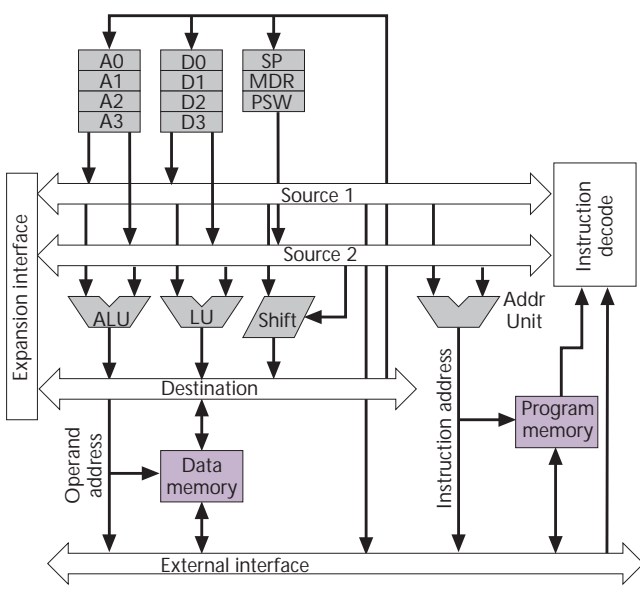


Figure 1. The MN10300 has a simple microarchitecture with a dual-ported register file, separate arithmetic and logic units, barrel shifter, and address generation. The core CPU is extensible with additional ALU and instructions.

hopes they saved a lot of space: the multiplier is slower than most, calculating only one bit of result per clock cycle, with an additional five cycles of overhead. On the plus side, the 10300 automatically gauges the magnitude of the second operand (the multiplier) and terminates the process early for small multipliers. Thus, a 32×24-bit multiply takes 29 cycles, but a 32×8-bit operation takes 13 cycles. Both signed and unsigned multiplication is supported, with the upper half of the result deposited in the MDR (multiply/divide register).

Integer division is likewise iterative, with latencies similar to those of the MUL and MULU instructions.

Logical Instructions Round Out Functions

As Table 1 shows, the usual assortment of logical operations is included: shifts, rotates, comparisons, and inversions. Arithmetic shifts to the right or left work on the whole destination register; the shift count can be any 8-bit immediate value. Rotates, on the other hand, are by only a single bit in either direction. Large-scale rotates need a series of single-bit ROL or ROR codes.

Matsushita supplied the 10300 with a completely separate ASL2 instruction that shifts the operand two bits to the left, a near-duplication of the more general ASL instruction. ASL2 is encoded in a single byte, versus the two-byte ASL. Its purpose is to compress pointer calculations, particularly within inner loops, by efficiently shifting array indexes with a single instruction. In a similar vein, the 10300 has separate INC (increment) and INC4 (add four) instructions.

Most processors implement their comparison instructions as arithmetic subtracts that don't store the result or alter the condition flags. On the 10300, CMP is more flexible than SUB, because the former allows the use of 8- and 16-bit immediate values, while the latter does not.

Loop Constructs a Bonus

Matsushita makes much of the 10300's loop constructs in its promotional literature, and the architecture is unusual in this regard. With its traditional five-stage pipeline, the 10300 would typically suffer a two- to three-clock penalty for taken branches. In real-time systems or media applications (where data might be supplied isochronously) the unpredictability of branch delays can complicate coding. Enter the branch-target buffer.

The 10300 implements a much-simplified version of the branch-target buffer seen in many processors and DSPs. The chip's LIR (loop instruction register) and LAR (loop address register) hold the instruction at the top of a loop and the address of the instruction that follows it, respectively. When the 10300 reaches the bottom of a conditional loop, it executes the instruction contained in the LIR while it fetches the instruction pointed to by the LAR. The cached instruction hides the latency involved in flushing the pipeline and fetching the next instruction. The result is a zero-overhead loop.

This interesting construct works only for innermost loops, as there is only one LIR and LAR pair. These two registers are loaded simultaneously through the SETLB (set loop buffer) instruction and come into play when the 10300 executes the conditional loop-termination instruction, Lcc. Strangely, although branches can be made conditional on all 16 permutations of the condition codes, Lcc can branch on only 12; looping on the condition of the V flag or the N flag is not supported.

The LIR is 32 bits wide; if the target instruction is longer than 32 bits, the 10300 requires an extra cycle to fetch and decode the rest of the instruction. Either way, the LAR points to the beginning of the next instruction, even if it's within the LIR.

Mnemonic	Description	Mnemonic	Description	Mnemonic	Description
Arithmetic		Transfer		Flow Control	
ADD	Add	MOV	Move data	Bcc	Branch on condition
ADDC	Add with carry	MOVBU	Move byte, zero-extend	Lcc	Loop on condition
SUB	Subtract	MOVb	Move byte, sign-extend*	SETLB	Set loop buffer LIR, LAR
SUBC	Subtract with borrow	MOVHU	Move halfword, zero-extend	JMP	Jump unconditional
MUL	Multiply, signed 32×32→64	MOVH	Move halfword, sign-extend*	CALL	Call and allocate stack
MULU	Multiply, unsigned 32×32→64	EXT	Sign extend 32→64	CALLS	Call, no stack allocate
DIV	Divide, signed 64÷32→32, 32	EXTB	Sign extend 8→32	RET	Return from subroutine, deallocate
DIVU	Divide, unsigned 64÷32→32, 32	EXTBU	Zero extend 8→32	RETS	Return, no deallocate
INC	Increment by 1	EXTH	Sign extend 16→32	RETF	Return via pointer in MDR
INC4	Increment by 4	EXTHU	Zero extend 16→32	RTI	Return from interrupt
Bit Manipulation		MOVm	Move multiple registers	TRAP	Branch to 0x40000010
BTST	Bit test	CLR	Clear register Dn	Logical	
BSET	Bit test and set	Logical		ASR	Arithmetic shift right
BCLR	Bit test and clear	CMP	Compare	LSR	Logical shift right
Miscellaneous		AND	Logical AND	ASL	Arithmetic shift left
NOP	No operation	OR	Logical inclusive-OR	ASL2	Arithmetic shift left by 2
UDFnn	User-defined function nn	XOR	Logical exclusive-OR	ROR	Rotate right by 1
UDFUnn	User-defined function nn, unsigned	NOT	One's negation	ROL	Rotate left by 1

Table 1. The MN10300's 46 instructions include the usual logical and arithmetic operations, plus basic operations to set, clear, and test individual bits. All data must be stored in a data register; the MN10300 follows a load/store model.

Subroutine Calls Are Streamlined by Compiler

With its tiny register set and Matsushita's emphasis on C programming, the 10300 is likely to spend much of its time making function (subroutine) calls while pushing and popping parameters on and off that stack.

The chip provides two forms of CALL and three forms of RET for just this purpose. The CALL and CALLS instructions differ in that the former allocates stack space and pushes registers onto the stack. Likewise, RET and RETS return from subroutines, with RET popping the registers and deallocating stack space. The third form, RETF, returns to an address stored in the MDR. Using the multiply/divide register as a return pointer saves stack space and speeds up the return function, as long as no multiply or divide instructions appear in the subroutine.

With variable-length instructions, the instruction set can be arbitrarily large. Matsushita has blocked out 48 major opcodes for application-specific extensions to the instruction set. The UDF_n (user-defined function) and UDFUn (user-defined function, unsigned) mnemonics map to extended-length instructions that will be defined in future versions of the 10300 architecture. Depending on the value of *n*, these instructions may update the destination register and flags or create a nondestructive operation.

Two Chips Now, More on the Way

There are four members defined in Matsushita's lineup, with at least one more in development. All will share the same CPU core; as Table 2 shows, these chips differ mainly in the amount and type of on-chip memory and the level of peripheral I/O integration. The first two are in production now and achieve 60 MHz in Matsushita's 0.35-micron process.

Matsushita made a high-profile announcement with Microsoft in July regarding the planned AM33. This chip will include the MMU modifications to run Windows CE, making Matsushita's the sixth CPU architecture (after ARM, MIPS, PowerPC, SuperH, and x86) to gain a port of this popular operating system. As with the other architectures, gaining Windows CE compatibility involved adding an MMU that suited Microsoft's paging requirements. Apart from that, the AM33 will not be significantly different from other 10300 chips. Matsushita is not revealing the schedule for the AM33 or pricing for any of the 10300-series chips.

Matsushita has announced plans to use WinCE in some, but not all, of its Panasonic consumer-electronics equipment. Specifically, the company envisions Windows CE in audio, video, and computer products, but not in industrial systems or in traditional white goods such as refrigerators.

Another Tool for Consumer Electronics

Matsushita's 10300 family has similarities to NEC's V800 series, both superficially and technically. Both are proprietary 32-bit designs that are almost unknown in North America. Both have variable-length instructions (although the V800 quantizes its instructions 16 bits at a time), limited

Price & Availability


Matsushita's AM30 and AM31 processors are currently in production; prices are not available. The production schedule for the AM33 has not been announced.

For more information, contact Matsushita Cupertino (Calif.) at 408.252.9890 or visit www.psd.com/micros/microcon.htm

development tools, and owners with a large presence in consumer electronics. Clock rates (about 60 MHz) and performance ratios (1 MIPS/MHz) are also similar.

NEC has a big lead, and its V830 and V850 chips are already established in various industrial, consumer, and peripheral designs. That company has also begun expanding the capabilities of its family with hardware media accelerators, such as in the V853 (see MPR 6/2/97, p. 22). With its extensible instruction set, Matsushita seems prepared to take this same path, but NEC has a head start of several years.

As they stand today, the 10300 chips have very basic instruction sets, with no particular advantages for bit-manipulation, image compression or decompression, audio processing, or network handling. The groundwork may be there, but the offerings to date are unremarkable. There's no media acceleration, no floating-point ability, no DSP extensions, and almost no registers. On purely technical grounds, they're backward, and with no prices provided by Matsushita, it's impossible to judge their worth.

The new architecture's crowded register set and two-operand functions make these chips appear like microcontrollers, albeit ones with wide registers. They seem neither fish nor fowl; neither low-end microcontrollers nor general-purpose RISC processors. Perhaps after Matsushita fleshes out the family with the AM33 and other chips with specific enhancements, this new line will begin to find its niche in the embedded marketplace. 

	MN103000 (AM30)	MN103001G (AM30)	MN1030F01K (AM30)	MN103002A (AM31)
Clock Freq	60 MHz	60 MHz	60 MHz	66 MHz
Inst Memory	16K RAM	128K ROM	256K flash	4K cache
Data Memory	16K RAM	8K RAM	256K RAM	4K cache
I/O Pins	89	72	72	26
A/D	8 chan	4 chan	4 chan	None
DMA	4 chan	None	None	4 chan
DRAM Ctrl	Yes	Yes	Yes	Yes
Voltage	3.3 V	3.3 V	3.3 V	3.3 V
Power (typ)	500 mW	300 mW	300 mW	500 mW
Package	PQFP-160	LQFP-100	LQFP-100	PQFP-160
Price	n/a	n/a	n/a	n/a
Availability	1Q99	Now	1Q99	Now

Table 2. Matsushita's MN10300 family includes three devices in production and the planned AM33 Windows CE controller. n/a=not available from vendor. (Source: Matsushita)