

Hyperstone Merges CPU and DSP Cores

Fixed-Point DSP Shares Instructions and Registers With Integer Core

by Jim Turley

Hyperstone Electronics is renewing its push into the embedded processor market with its combination RISC processor and digital signal processor (DSP). Hyperstone's hybrid CPU/DSP design is creative and innovative but has languished due to lack of interest. Now, with a new licensing partner and new staff, the company hopes to tackle the market for cable and ISDN modems, digital cameras, and wireless infrastructure.

Hyperstone sells its microprocessors under its own name and recently signed LG Semicon as a second source. LG, which also holds an ARM license, will offer the E1 core to its ASIC customers. Hyperstone is seeking at least two more licensees before the year is out, broadening its base of supply.

A small company on the Swiss/German border, Hyperstone began operations in 1988 with the E1, a 16-bit design with no DSP features (see MPR 9/19/90, p. 6). The company signed two licensees: Zilog in 1990 and Alps Electric in 1992. Although the latter company shipped about 500,000 chips, Hyperstone continues to toil in obscurity. The E1 underwent a major overhaul in 1995, adding DSP capability, on-chip

DRAM, and a new pipeline. The integer portion of the new E1-32, however, is largely unchanged from the original vintage design.

On-Chip DRAM and DRAM Controller

Hyperstone's design has 32-bit registers and internal data paths. A 4K block of on-chip memory allows code and data to be stored on chip. The pipeline has only two stages: fetch/decode and execute/writeback. Although the short pipe limits clock speed, it gives the E1 an almost painless, one-clock penalty for taken branches.

The E1 programmer's model includes 32 global registers, G0-G31, plus 64 local registers (described later). Of these, only G0-G15 are directly addressable; G16-G31 are treated as control/status registers, accessible only through privileged instructions.

Hyperstone ships two versions of the chip: the E1-32 and the E1-16. Internally, the two are identical; the suffix indicates the width of the off-chip bus. Both have 4K of on-chip RAM and a memory controller that drives RAS, CAS, WE, and parity signals for page-mode DRAMs. An enhanced

Mnemonic	Description	Mnemonic	Description	Mnemonic	Description
SHLI	Shift left, immediate count	LD	Load from memory	ADD	Add registers, unsigned
SHLDI	Shift double left 0...31	ST	Store to memory	ADDS	Add registers, signed
SHL	Shift left, register count	MOV	Copy register to register	ADDC	Add registers with carry
SHLD	Shift double left, register count	MOVI	Copy immediate32 to register	ADDI	Add imm32 to register
SARI	Shift right, imm count, signed	MOVD	Copy double registers	ADDSI	Add imm32 to register, signed
SARDI	Shift dbl right, imm cnt, signed	CHK	Trap if destination > source	SUM	Add register and immediate
SAR	Shift right, reg count, signed	CHKZ	Trap if destination = 0	SUMS	Add register to imm32, signed
SARD	Shift dbl right, reg count, signed	TESTLZ	Count leading zeros	SUB	Subtract registers, unsigned
SHRI	Shift right, immediate count	CMP	Arithmetic compare (subtract)	SUBS	Subtract registers, signed
SHRDI	Shift dbl right, imm count	CMPI	Arithmetic subtract, immediate	SUBC	Subtract registers with borrow
SHR	Shift right, reg count	CMPB	Compare and set Z flag	MUL	Multiply (32 × 32 → 32)
SHRD	Shift dbl right, reg count	CMPBI	Compare, set Z flag, imm	MULS	Multiply (32 × 32 → 64)
ROL	Rotate left, register count	SETcc	Set/clear register on condition	MULU	Multiply (32 × 32 → 32), uns
AND	Logical AND	Bcc	Branch on condition cc	DIVS	Divide (64 ÷ 32 → 32)
ANDN	Logical AND NOT	DBcc	Delayed branch on condition	DIVU	Divide (64 ÷ 32 → 32), uns
OR	Logical OR	CALL	Call subroutine	NEG	Two's complement
XOR	Logical exclusive OR	RET	Return from subroutine	NEGS	Two's complement with sign
ANDNI	Logical AND NOT immediate32	TRAPcc	Trap on condition cc	Floating-Point	
ORI	Logical OR immediate32	FRAME	Allocate stack frame space	FADD{D}*	Add, single/double
XORI	Logical exclusive OR imm32	SETADR	Set frame pointer	FSUB{D}*	Subtract, single/double
NOT	Invert bits	FETCH n	Force prefetch of 2n bytes	FMUL{D}*	Multiply, single/double
MASK	Copy with logical AND	DO*	Initiate subroutine	FDIV{D}*	Divide, single/double
XM1/2/4/8	Copy and shift by 1/2/4/8 bits	NOP	No operation	FCVT{D}*	Convert single/double
XX1/2/4/8	Copy, shift 1/2/4/8 bits, check	DSP		FCMP{D}*	Compare single/double
		(See Table 2)		FCMPU{D}*	Compare single/double, uns

Table 1. The Hyperstone E1 instruction set is rich with shift, rotate, and logical operations. Most arithmetic operations (ADD, MUL, etc.) can operate on both 32-bit and 64-bit quantities. The 14 floating-point operations are trapped for emulation. *emulated instructions.

E1-32X chip is due midyear, expanding the on-chip memory to 8K and adding EDO support to the DRAM controller.

Unlike most microprocessors, the E1's on-chip memory is DRAM, not SRAM, and needs to be refreshed periodically. As with normal DRAMs, each refresh cycle refreshes only one row. In Hyperstone's case, a row is only 16 bytes, so 256 refresh cycles are required to refresh the entire 4K array. There is a one-cycle load-use penalty for internal memory. Refresh happens automatically and has virtually no impact on performance.

The DRAM itself is built around an unusual three-transistor cell of Hyperstone's own design. The company chose DRAM over SRAM for density reasons, and developed its 3T cell to be portable across different vendors' logic processes.

Instruction Set Mixes CPU, DSP Operations

The E1's instruction set, listed in Table 1 and Table 2, is an eclectic mix of integer, floating-point, and signal-processing instructions. The opcode map is rich with control-oriented operations to set, clear, mask, and invert bits. The chip has very good support for 64-bit operations, including extended addition, subtraction, multiplication, and even division.

Hyperstone was ahead of its time in abusing the term RISC. Instructions vary in length from 16 bits to 48 (1-3 halfwords). The longer forms are generally used to encode 32-bit immediate values for arithmetic operations or to specify absolute memory addresses. Smaller immediate values can be encoded in shorter instruction words.

Mnemonic	Operation	
EMUL	Multiply	$(32 \times 32 \rightarrow 32)$
EMULS	Multiply	$(32 \times 32 \rightarrow 64)$, signed
EMULU	Multiply	$(32 \times 32 \rightarrow 32)$, unsigned
EMAC	Multiply-add	$(32 \times 32 + 32 \rightarrow 32)$
EMACD	Multiply-add	$(32 \times 32 + 64 \rightarrow 64)$
EMSUB	Multiply-subtract	$(32 \times 32 - 32 \rightarrow 32)$
EMSUBD	Multiply-subtract	$(32 \times 32 - 64 \rightarrow 64)$
EHMALC	Multiply-add	$((16 \times 16) + (16 \times 16) + 32 \rightarrow 32)$
EHMALD	Multiply-add	$((16 \times 16) + (16 \times 16) + 64 \rightarrow 64)$
EHCMLD	Complex MUL	$((16 \times 16) - (16 \times 16) \rightarrow 32)$ $((16 \times 16) + (16 \times 16) \rightarrow 32)$
EHCALC	Complex MAC	$((16 \times 16) - (16 \times 16) + 32 \rightarrow 32)$ $((16 \times 16) + (16 \times 16) + 32 \rightarrow 32)$
EHCALD	Complex add/sub	$(16 + 16 \rightarrow 16)$ $(16 + 16 \rightarrow 16)$ $(16 - 16 \rightarrow 16)$ $(16 - 16 \rightarrow 16)$
EHCFFTD	FFT kernel	$((16 + (32 \gg 15) \rightarrow 16)$ $(16 + (32 \gg 15) \rightarrow 16)$ $(16 - (32 \gg 15) \rightarrow 16)$ $(16 - (32 \gg 15) \rightarrow 16)$

Table 2. Thirteen "extended" DSP instructions use the 32-bit multiply-accumulate (MAC) hardware in the E1-32. These instructions are pipelined, with a 1-4-cycle repeat rate. EHCMLD and EHCALC carry out two operations simultaneously; EHCALD and EHCFFTD execute four.

The E1 has an innovative approach to encoding short, 5-bit constants. Rather than simply interpret the numbers 0-31 literally, it treats many 5-bit immediate values as code words for commonly used constants. As Table 3 shows, values from 0x11 through 0x1F are actually shorthand notation for a number of useful coefficients and scaling factors. This clever technique saves code space by avoiding 32-bit immediate values in some instances.

Most instructions use a destructive two-operand addressing form. An exception is the SUM instruction, which is distinct from ADD in that it places the sum of a register and an immediate into a separate destination register.

Although the E1 defines more than a dozen floating-point instructions, all of them invoke a trap handler and must be emulated in software (which Hyperstone provides). Execution time for these instructions varies widely, depending on the operation and the magnitude of the operands.

DSP Operations Go Beyond Simple MAC

Hyperstone pays more than just lip service to DSP operations. The chip has a hardware multiply-accumulate (MAC) unit that is separate from its conventional ALU. Any of the 13 DSP instructions listed in Table 2 can execute in parallel with the conventional instructions from Table 1. The latency for DSP operations is 1-4 clock cycles, but the MAC unit is fully pipelined. Once a DSP instruction is launched, the E1 can begin executing more integer instructions immediately.

The E1 uses registers G14 and G15 for its accumulator. All DSP instructions deposit their results in one (for 32-bit results) or both (for 64-bit results) of these registers. Any other register can hold source data. Split operations, and operations on 16-bit data, pack two operands into a single register. For example, EHMALC multiplies the upper halves and the lower halves of the two source registers and adds the results to G15. As in x86 chips with MMX, Sun's VIS, and others, the packed data types double throughput without increasing the size of the MAC unit.

Some of the more ambitious DSP operations, such as EHCALD and EHCFFTD, perform up to four operations at once. With single-cycle throughput on some operations, the E1 can crank out 400 DSP MIPS at 100 MHz, better than many fixed-point DSPs. Hyperstone claims its 66-MHz E1-32 compares favorably with DSPs from Zoran, Lucent, Motorola, and Analog Devices on simple FFTs and FIR filters.

Code	Value ₁₀						
00	0	08	8	10	16	18	-8
01	1	09	9	11	imm32	19	-7
02	2	0A	10	12	+imm16	1A	-6
03	3	0B	11	13	-imm16	1B	-5
04	4	0C	12	14	2 ⁵	1C	-4
05	5	0D	13	15	2 ⁶	1D	-3
06	6	0E	14	16	2 ⁷	1E	-2
07	7	0F	15	17	2 ³¹	1F	2 ³¹ -1

Table 3. Five-bit immediate values from 0x00-0x10 are used directly; 0x11-0x13 indicate an immediate value follows; larger values encode one of several fixed constants.

Merged CPU/DSP No Longer Unusual

There are strong parallels between the design of the E1 and other recent incarnations of the merged CPU/DSP concept. The nearest comparison is to ARM's Piccolo (see MPR 11/18/96, p. 17), which began development about the same time as the E1-32 but is approximately a year behind the E1 in reaching the market.

The Piccolo DSP has a repertoire of more than four dozen instructions, while the E1 has only 13 for its DSP unit. Many of those Piccolo instructions, however, duplicate functions found in the E1's conventional instruction set. Unlike Piccolo, the E1 executes from a single code stream, so the DSP unit does not require its own conditional, logical, and flow-control instructions. In that regard, the E1 is more similar to Hitachi's SH7410 (see MPR 3/31/97, p. 4).

The E1 allows more independent parallelism than the 7410 but less than Piccolo and far less than Motorola's 68356 (see MPR 6/20/94, p. 9). The Hitachi chip keeps its integer and DSP units in lock step, executing one instruction at a time. ARM lets Piccolo run on its own, with some cooperation between the two. The E1 allows some DSP and integer operations to run side by side but maintains close ties between them. The 68356 is basically two independent processors sharing a die and a plastic package.

The internal architectures of these parts are also very different. The 68356 and 7410 maintain separate X and Y data memories; Piccolo and the E1 do not, relying on their register files. Within that grouping, Piccolo has 16 registers that are separate from ARM's, while the E1 uses a single register set, limiting the number of operands. Both the E1 and ARM/Piccolo share their internal address and data buses.

In Piccolo's case, spare ARM bandwidth is used to access data in memory. The E1 has a similar strategy, hiding memory references under the latency of DSP instructions. All four architectures—ARM's, Hitachi's, Hyperstone's, and Motorola's—can stream back-to-back DSP operations while the integer unit handles transfers to and from memory.

Instruction Prefetch Substitutes for Cache

The E1 has 4K of on-chip RAM (8K for the E1-32X) but no cache, at least not in the usual sense. The chip's instruction "cache" is actually a 128-byte buffer that holds as many as 64 consecutive instructions. The buffer is managed as a circular queue with head and tail pointers. The chip attempts to keep ahead of the program flow by prefetching into the buffer. Prefetches happen opportunistically, between memory accesses. There is no cache management or replacement algorithm, *per se*; the chip merely fills the buffer sequentially and updates the head and tail pointers as it moves along. For linear code, this scheme works well.

Prefetching pauses when the chip decodes an impending memory reference or a branch. This avoids wasting bus bandwidth with potentially unnecessary instructions. Taken branches reset the head and tail pointers to the target address, effectively invalidating the buffer. The exception is a

short branch to a target already in the buffer; this leaves the buffer unchanged and allows the E1 to cache short loops.

Ideally, the E1 prefetches 32 bytes ahead of the program counter, which is only one-fourth of the prefetch buffer's capacity. Barring a recent branch, the rest of the buffer holds code that has already been executed. By retaining at least 96 bytes of "old" code, the E1 can hold moderately sized loops in its prefetch buffer, eliminating fetch cycles during DSP loops, which are typically data intensive.

Hyperstone's instruction buffer is essentially a small loop cache with a 100% hit rate. It performs better than a conventional cache in some cases because it anticipates (prefetches) code instead of holding already used code.

On the other hand, the prefetch buffer has a 0% hit rate for long branches or when returning to previously executed code. Branching outside the buffer resets its pointers; even if the program branches back immediately, the instructions in the buffer cannot be reused. With no cache tags and no persistence, the instructions in the buffer are lost after a branch outside its range. Likewise, subroutine calls or exceptions wipe out buffered loop code. On returning to the loop, the E1 must load the buffer again.

Hyperstone uses this buffer instead of a cache to save die area and simplify the design. With no cache tags, replacement algorithm, or update policy, the circular buffer is considerably easier to manage than even a small cache. For only 128 bytes, the buffer provides a significant performance boost, though not in all cases.

A modest cache, say 1K, would be more effective, at least in the general case. On the other hand, Hyperstone's buffer behaves more predictably than a cache, a plus for most DSP programmers.

Programmer Can Schedule Code, Data References

Memory references slow the prefetch mechanism for at least two cycles while the memory access is fulfilled. This pause puts the buffer two words (1–4 instructions) behind where it would ideally be. Back-to-back memory accesses completely stop the prefetch mechanism; even accessing memory on every alternate instruction effectively halts prefetching. As a consequence, a long sequence of memory references can run the prefetch buffer dry.

The `FETCH` instruction can be used to alleviate this situation. With it, programmers can force up to the next 30 bytes of code into the prefetch buffer before strangling the E1 with memory references. Although using `FETCH` does not reduce the number of cycles required, it does allow programmers to rearrange code and data references, which may help alleviate bus-contention problems.

The E1's `FETCH` capability is unique. There are many embedded RISC chips that consume one instruction per cycle. With no cache, chips like the ARM710 fully saturate all the available bus bandwidth just by fetching instructions. Any data reference to memory—or worse, memory latency—slows instruction fetching and chokes performance considerably.

Price & Availability

Hyperstone's E1-16 and E1-32 are shipping now at 66 MHz. Pricing for either chip in 10,000-unit quantities is approximately \$20.

For more information, contact Hyperstone GmbH (Konstanz, Germany) at 49.7531.980.30 or Hyperstone US (Cupertino, Calif.) at 408.257.1057, or via the Web at www.hyperstone.com.

Stack Cache Helps Speed Shallow Subroutines

The E1 maintains an on-chip stack cache to accelerate stack operations. Two other recently developed CPUs, PicoJava and ShBoom, also have stack caches, although those two designs are strongly stack-oriented, while the E1 is not.

Physically, the stack cache is a 64-word circular buffer, like the instruction buffer. When the stack contains fewer than 64 words, stack references stay on chip. When the 65th push operation overflows the stack cache, the E1 moves the oldest entry to memory. Underflow is handled by retrieving items from memory while popping items off the cache.

For deeply nested subroutines or operating systems, the effect of the stack cache is minimal; each push still forces a memory reference, although 64 items can be popped before the stack cache empties and the E1 pulls from memory. But for code with few subroutines and few parameters, the on-chip stack will eliminate many memory accesses entirely.

The E1's subroutine calls (CALL, TRAP, and emulated FP instructions) allocate a fixed amount of space on the stack for parameters. The overall concept is similar to SPARC's overlapping register windows, but with more control over parameter passing. Unlike SPARC, the E1 lets programmers adjust the number of registers allocated and the amount of overlap, hiding or exposing more or fewer stack entries.

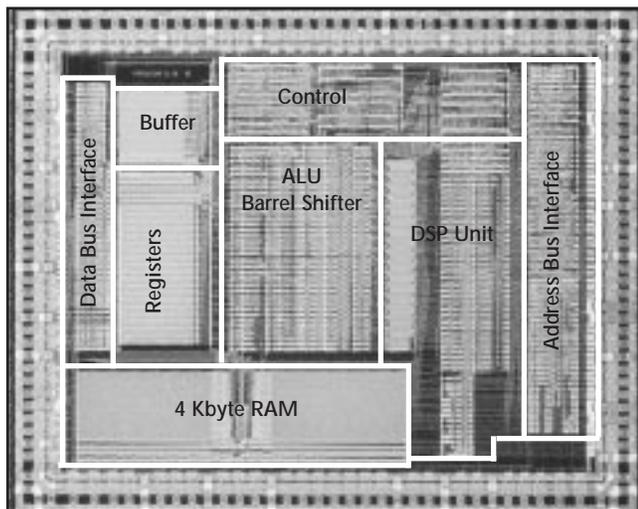


Figure 1. In UMC's 0.8-micron two-layer-metal CMOS process the E1-32's 210,000 transistors measure 55 mm².

Roadmap Calls for 0.35 μ , 120 MHz by Year's End
Pedestrian manufacturing processes and the short pipeline are limiting Hyperstone's clock rates. The current E1-32, shown in Figure 1, is made on UMC's 0.8-micron, two-layer-metal process, where the chip's 210,000 transistors sprawl over 55 mm². About 100,000 transistors are expended on the chip's local DRAM. The chip is housed in a 144-lead TQFP package; with its narrower external bus, the E1-16 comes in a TQFP-100 package.

The fab process allows both 3.3-V and 5-V operation; at the higher voltage, the chip consumes about 950 mW (typical) at 66 MHz. Lowering the voltage drops the top speed to 50 MHz and the power to 230 mW. While these clock rates are unremarkable by current CPU standards, they're pretty fast compared with those of most DSP chips.

Hyperstone's deal with LG gives it a second source for parts and access to more modern manufacturing processes. Slated for July is the E1-32X, a 0.5-micron shrink of the E1-32 combined with some minor enhancements. A larger, 8K on-chip DRAM will help performance scale with clock rate, even as the die size falls to an estimated 21 mm². This new part is expected to reach 100 MHz.

Hyperstone predicts its 0.35-micron version of the E1-32X, due in 3Q97, will reach 120 MHz. Because the chip is pad limited, the die size is expected to remain the same.

Hyperstone Among Few RISC/DSP Vendors

Hyperstone has hit on a clever combination of conventional CPU features and DSP functions with a unified instruction set, register set, and programming model. As other vendors are discovering, these are useful characteristics for a number of emerging applications.

Hyperstone's business model calls for a combination of chip sales and licensing agreements. The company wants to ship one million chips by the end of 1998—a tall order, given last year's sales of 50,000 units. The company hints that current design wins could significantly increase that volume, especially with ASICs destined for ISDN modems, consumer items, or automotive instrumentation.

Hyperstone has stiff competition: ARM is approaching de facto status in the ASIC core business, and with a new licensee *du jour* (see MPR 4/21/97, p. 4), it's nearly ubiquitous. Unless Piccolo falls flat in some area, Hyperstone will have a tough time differentiating its wares.

Hitachi's SH-DSP will also prove tough to overcome as that vendor expands both its standard products and its ASIC presence through VLSI (see MPR 8/26/96, p. 4). Once a dark horse like Hyperstone, Hitachi has now made the short list for many customers.

All companies start small; some succeed sooner than others. With the interest in merged CPU/DSP chips growing rapidly, Hyperstone's prospects look better than ever. The perennial startup has already survived for nearly ten years; if it can move ahead with its fabrication plans and meet its volume projections, the next ten should be far better. \square