Hitachi, ST Extend SuperH to 64 Bits

New SH-5 Architecture Aims for Multimedia Systems on a Chip

by Tom R. Halfhill

Building muscle without gaining fat—it's the dream of every weight lifter, football player, and embedded-processor architect. In the last case, the goal isn't physical strength, but processing power without code bloat. That's what drove Hitachi Semiconductor and STMicroelectronics to collaborate on the SH-5, a 64-bit extension to Hitachi's SuperH architecture that's compatible with SuperH's svelte 16-bit instructions.

Hitachi's Jim Slager and ST's Jean-Marie Rolland described the new RISC architecture at this week's Microprocessor Forum. The SH-5 is a clever combination of power and efficiency that preserves existing investments in SuperH software. It also keeps Hitachi and ST in the accelerating race against other high-performance embedded processors, such as those based on PowerPC, MIPS, and StrongArm cores.

Everyone, it seems, is aiming at the same target: system-on-a-chip (SOC) solutions for information appliances, digital set-top boxes, automotive navigation systems, and other futuristic embedded applications. Although the much-anticipated "post-PC era" hasn't arrived yet—by all appearances, we're still struggling through the pre-post-PC era—virtually every major semiconductor company is competing to supply the silicon that will make it possible. If the market flops, it won't be because the embedded-system engineers of the 21st century will have trouble finding high-performance embedded processors.

To rise above this noise, each company needs at least one point of differentiation. Hitachi and ST claim to have several: high code density, small cores, frugal power consumption, and advanced on-chip debugging features. That's in addition to the features now commonly expected of architectures and cores in this class: wicked-fast floating point, multimedia extensions, digital-signal-processing (DSP) capabilities, and single-instruction, multiple-data (SIMD) instructions.

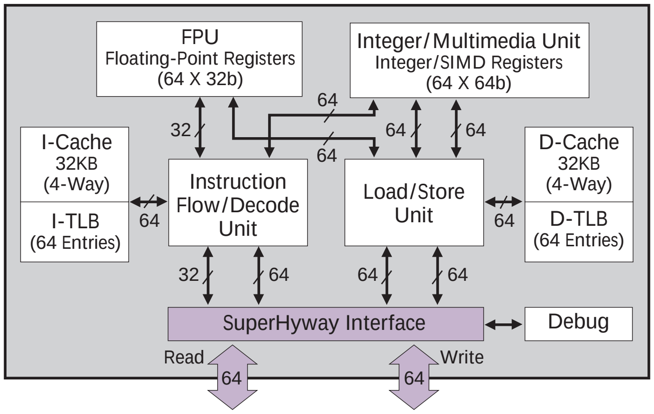

Figure 1. The first SH-5 core is designed for systems on a chip. The SuperHyway bus connects to on-chip peripherals.

64-Bit Core, 16/32-Bit Instructions

As Figure 1 shows, the first implementation will be a scalar core with dual 32K caches, dual TLBs, an integer/multimedia unit, an FPU, and a load/store unit. A high-speed on-chip bus known as the SuperHyway links the core to integrated peripherals on an SOC. Hitachi refers to this jointly designed core as the SH8000, while ST calls it the ST50. It's the foundation for a series of SOCs that each company will independently design, manufacture, and market.

Both companies plan to build the first SH-5 chips in a 0.15-micron IC process with copper interconnects, which means production probably won't commence until at least 2001. (Hitachi and ST are mum on this point.) Although each partner will manufacture the parts in its own fabs, they will standardize the IC process to minimize porting costs. In the target process, the core is tiny: 14 mm^2 with caches, TLBs, and an FPU. Without the FPU, it's 11 mm^2, and without the caches and TLBs, it's only 3 mm^2.

Based on simulations, Hitachi and ST expect SH-5 chips to execute 604 Dhrystone 2.1 MIPS, 2.8 GFLOPS, or 1.6 billion integer multiply-accumulate (MAC) operations per second at the target frequency of 400 MHz.

It's apparent from the design of this architecture that backward compatibility with existing SuperH processors was paramount. The SH-5 can execute all 208 instructions in the current SuperH instruction set, as well as 209 new instructions in the SH-5 instruction set. Thus, it preserves the software investment of Hitachi's customers, simplifies the porting of operating systems and development tools, and doesn't sacrifice the efficient code density for which embedded processors in the SH-1 through SH-4 series are justly famous.

Still, when stretching the SuperH architecture to 64-bit datapaths and registers, something had to give. Inevitably, there will be some code expansion. The current SuperH has 16-bit fixed-length instructions, while the new SH-5 instructions are 32 bits long. The architects were forced to double the instruction length because there simply wasn't enough room in the tight 16-bit opcode space to add the new instructions and registers they believe are necessary for a competitive 64-bit architecture. Table 1 lists all of the new SH-5 instructions.

The new instruction format has 10 bits for opcodes, 6 bits for register addresses, up to three operands, and 4 bits reserved for future features, such as predication or specula-

Instruction	Description
Flow-Control Instructions	
PTA, PTB	Prepare target, immediate
PTABS, PTREL	Prepare target, abs/relative
BEQ, BEQI,BGE, BGEU	Conditional branches
BGT, BGTU, BNE, BNEI	Conditional branches
BLINK	Unconditional branch/mode switch
GETTR	Move from target register
Integer Instructions	
MOVI, SHORI	Move immediate, shift then OR immed
ADD, ADD.L, ADDI, ADDI.L	Add register/immediate 32/64b
ADDZ.L	Add with zero-extend 32b
SUB, SUB.L	Subtract 32/64b
MULU.L	Multiply 32 ×32b to 64b unsigned
CMPEQ, CMPGT,CMPGTU	Compares
CMVEQ, CMVNE	Conditional moves
AND, ANDI, ANDC	AND, AND-complement
OR, ORI, XOR, XORI	OR, exclusive-OR, immediates
SHARD, SHARD.L, SHARI	Shift-right dynamic 32/64b, immed 64b
SHLLD, SHLLI	Shift-left logical dynamic/immed 64b
SHLRD, SHLRI	Shift-right logical dynamic/immed 64b
SHLLD.L, SHLLI.L	Shift-left logical dynamic/immed 32b
SHLRD.L, SHLRI.L	Shift-right logical dynamic/immed 32b
BYTEREV	Byte reversal, count sign bits, no op
NSB	Count sign bits
NOP	No operation
Integer Load/Store Instructions	
LD.[B,L,W,Q]	Load displacement 8/16/32/64b signed
LDX.[B,L,W,Q]	Load indexed 8/16/32/64b signed
LD.[UB,UW]	Load displacement 8/16b unsigned
LDX.[UB,UW]	Load indexed 8/16b unsigned
LDHI.[L,Q]	Load misaligned high 32/64b
LDLO.[L,Q]	Load misaligned low 32/64b
ST.[B,L,W,Q]	Store displacement 8/16/32/64b
STX.[B,L,W,Q]	Store indexed 8/16/32/64b
STHI.[L,Q]	Store misaligned high 32/64b
STLO.[L,Q]	Store misaligned low 32/64b
Memory Synchronization Instructions	
SWAP.Q	Atomic swap in memory 64b
ICBI, PREFI	I-cache block invalidate/prefetch
ALLOCO	Operand cache block allocate
OCBI, OCBP	Operand cache block invalidate/purge
OCBWB	Operand cache write-back
SYNC[I,O]	Sync instructions or operand data
Multimedia Instructions	
MABS.[W,L]	Absolute signed 16/32b, w/saturation
MADD.[W,L]	Add 16/32b
MADDS.[B,W,L]	Add 8/16/32b, w/saturation
MCMPEQ.[B,W,L]	Compare equal 8/16/32b
MCMPGT.[UB,W,L]	Compare greater-than 8/16/32b
MCMV	Bitwise conditional move
MCNVS.[WB, WUB, LW]	Convert word to byte, long to word
MEXTR[1,2,3,4,5,6,7]	Extract 64b from 128b
MMACFX.WL	Fractional multiply-accumulate 16b
MMACNFX.WL	Fractional multiply-subtract 16b
MMUL.[W,L]	Multiply 16/32b
MMULFX.[W,L]	Fractional multiply 16/32b
MMULFXRP.W	Fractional multiply 16/32b, round
MMULHI.WL	Full multiply signed 16b high part
MMULLO.WL	Full multiply signed 16b low part
MMULSUM.WQ	Multiply and sum signed 16b
MPERM.W	Permute 16b
MSAD.UBQ	Sum of absolute diffs 8b unsigned
MSHALDS.[W,L]	Shift arithmetic saturate-left 16/32b
MSHARD.[W,L]	Shift arithmetic saturate-right 16/32b
MSHARDS.Q	Shift arith right, saturate signed 16b
MSHLLD.[W,L]	Shift logical left 16/32b
MSHLRD.[W,L]	Shift logical right 16/32b
MSHFHI.[B,W,L]	Shuffle upper half 8/16/32b
MSCHFLO.[B,W,L]	Shuffle lower half 8/16/32b
MSUB.[W,L]	Subtract 16/32b
MSUBS.[UB,W,L]	Subtract 8/16/32b, w/saturation
Floating-Point Instructions	
FABS.[S,D], FNEG.[S.D]	Absolute/negate 32/64b
FSQRT.[S,D]	Square-root 32/64b
FADD.[S,D], FSUB.[S,D]	Add/subtract two 32/64b
FCMPEQ,[S,D]	Compare equality 32/64b
FCMPGE.[S,D]	Compare ≥ 32/64b
FCMPGT.[S,D]	Compare > 32b/64b
FCMPUN.[S,D]	Compare unordered 32/64b
FCNV.[DS,SD]	Convert 64b to 32b or 32b to 64b
FGETSCR, FPUTSCR	Move from/to status-control reg
FLOAT.[LS,LD]	Convert int32 to FP32, int32 to FP64
FLOAT.[QS,QD]	Convert int64 to FP32, int64 to FP64
FMOV.[S,D]	Move FP32 to FP32, FP64 to FP64
FMOV.[SL,LS]	Move FP32 to int32, int32 to FP32
FMOV.[DQ,QD]	Move FP64 to int64, int64 to FP64
FMUL.[S,D], FDIV.[S,D]	Multiply/divide two 32b or two 64b
FTRC.[SL,DL]	Convert FP32 to int32, FP64 to int32
FTRC.[SQ,DQ]	Convert FP32 to int64, FP64 to int64
Special Floating-Point Instructions	
FMAC.S	Fused multiply-accumulate
FIPR.S	Vector dot-product
FTRV.S	Transform vector by matrix
FCOSA.S, FSINA.S	Approximate cosine/sine
FSRRA.S	Reciprocal square-root
Floating-Point Load/Store Instructions	
FLD.[S,P.D]	Load displacement 32b/2x32b/64b
FLDX.[S,P,D]	Load indexed 32b/2x32b/64b
FST.[S,P,D]	Store displacement 32b/2x32b/64b
FSTX.[S,P,D]	Store indexed 32b/2x32b/64b
System-Control and Configuration Instructions	
BRK, TRAPA	Cause debug exception or trap
RTE	Return from exception
SLEEP	Power-saving sleep mode
GETCFG, PUTCFG	Move from/to configuration register
GETCON, PUTCON	Move from/to control register

Table 1. The SH-5 has 209 new instructions, known as the SHmedia instruction set. SH-5 processors can also execute all 208 existing SuperH instructions (not shown), which is known as SHcompact mode. SH-5's BLINK (branch and link) instruction can switch modes.

tion. The old instruction format has only 8 bits for opcodes, 4 bits for register addresses, and two operand fields, as Figure 2 shows.

The code bloat resulting from doubling the instruction length should be considerably less than 2×, however. The SH-5 executes all of the old 16-bit instructions, so programs that don't need the new SH-5 features won't expand at all. Even those programs that take advantage of the SH-5's new features can segregate their code into sections of old and new instructions and dynamically switch between them.

For instance, a programmer can write control code in the 16-bit format and isolate the 32-bit multimedia, DSP, and SIMD instructions in separate subroutines. (Special directives or pragmas tell a compiler which binaries to gen-

For More Information

Hitachi and STMicroelectronics aren't saying when the first SH-5 chips will be available or how much they will cost. The first core—which Hitachi refers to as the SH8000 and ST calls the ST50—will be manufactured in a 0.15-micron copper IC process, so it probably won't enter production until 2001 at the earliest. For more information, go to *http://semiconductor.hitachi.com/superh* and *http://eu.st.com*.

erate for each code block.) If the often-quoted 80/20 rule holds up—that 20% of the code does 80% of the work, and vice versa—the resulting code expansion should be significantly less than the doubled instruction length implies.

Single Bit Switches Modes

To encourage programmers to write code in this fashion, the SH-5 uses branch and return instructions to switch between the old and new instruction formats. The 16-bit format is now called SHcompact mode, and the 32-bit format is called SHmedia mode. The lowest bit of an instruction address—previously unused because the fixed-length instructions are always aligned in memory—distinguishes between the modes. If the low bit is zero (the default for existing SuperH code), it indicates SHcompact mode. If the low bit is one (which previously triggered a trap), it indicates SHmedia mode.

The five existing branch instructions in SHcompact mode can switch to SHmedia mode because they can set the low address bit. In SHmedia mode, a single new instruction—BLINK, for branch and link—handles all unconditional branches and returns. BLINK can call another SHmedia-mode routine by setting the low bit, or it can switch into SHcompact mode by clearing the low bit. Mode switches don't require any more cycles than a regular unconditional branch.

In a sense, the SH-5 doesn't really execute SHcompact instructions natively. Instead, it diverts them to an extra decoding stage in the pipeline, where a lookup table translates SHcompact instructions into equivalent SHmedia instructions. The nominal seven-stage pipeline for SHmedia instructions thus becomes an eight-stage pipeline for SHcompact instructions.

This doesn't affect code size in memory, of course, because the translation from 16 to 32 bits happens dynamically in the processor. Theoretically, it could affect performance, because it adds an extra stage to the execution of every SHcompact instruction. Of course, pipelining hides that extra cycle unless a branch breaks the flow of execution.

The impact is probably insignificant. SH-5 processors will run at higher clock rates, so they should still deliver more performance with SHcompact code than do existing SuperH processors. And the most performance-critical code—such as multimedia and DSP operations—will be compiled with the new SHmedia instructions, which bypass the extra translation stage.

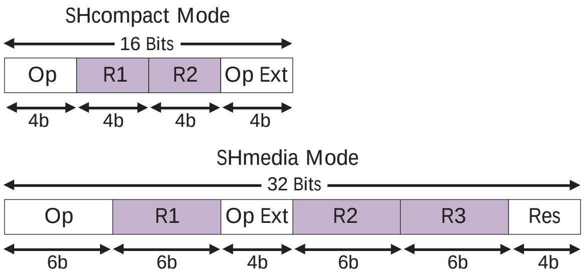

Figure 2. The SH-5's new instructions (SHmedia mode) are twice as long as the existing SuperH instructions (SHcompact mode).

Expanded Register File

As Figure 3 shows, each instruction set has its own logical view of the same physical register file, with SHcompact registers mapped onto the SHmedia registers. This allows programs to share data between modes without copying values among registers.

SHcompact code sees the same general-purpose registers (GPRs) it does today: 16 integer and 32 floating point, all 32 bits wide. SHmedia instructions see an expanded register file that has 64 integer GPRs and 64 floating-point GPRs. The integer registers are 64 bits wide, but the floating-point registers still hold only 32 bits. Double-precision floating-point instructions use pairs of these registers to hold 64-bit values.

The new integer registers have three read ports and one write port, so they can handle three-operand instructions and destructive MACs. The floating-point registers have eight read ports and two write ports.

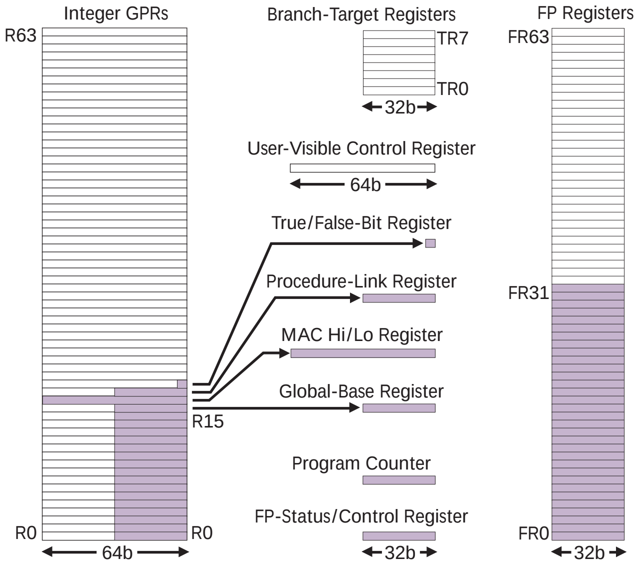

Figure 3. The SH-5 maps existing SuperH registers (shown in purple) onto its much larger integer and floating-point register files.

This situation allows the FPU to perform four single-precision multiplies and three adds every clock cycle—2.8 GFLOPS at 400 MHz. These vector operations are critical for matrix multiplication, which is commonly used for geometry transformations in 3D graphics. The SH-5 inherits this prodigious ability from the SH7750, an SH-4 processor that powers Sega's Dreamcast video-game console. The vital difference is that the SH-5 has twice as many FP registers as SH-4, so programmers and compilers have much more breathing room.

To make programming easier, Hitachi and ST created a SIMD function library for C compilers that can supplant in-line assembly routines and automatically handle register allocation. The library allows statements like this:

```
v5 = MSUBS_W (MMULFXRP_W(v1,v2),
MMULFXRP_W(v3,v4));
```

where v1–v5 are C variables automatically allocated to registers by the compiler; MSUBS_W is a multimedia-subtract instruction that operates on four word-sized integers with saturation; and MMULFXRP_W is a multimedia-multiply instruction that operates on fixed-point fractional words with positive rounding. Hitachi and ST have standardized these functions so they work the same with tools from both companies and from third parties. The binary formats are standardized, too, so developers can mix object files from any SH-5 compiler.

Some of the three-operand SIMD instructions can operate on eight byte-sized values in each 64-bit integer register, performing 24 primitive operations with a single instruction. An example is MSAD.UBQ, which computes the sum of absolute differences for eight bytes at a time. This is extremely useful for MPEG encoding, which works by finding the closest match between groups of one-byte pixels. The SH-5 can also perform four 16 × 16 → 32-bit MACs per cycle (with accumulation to 64 bits). All of these capabilities make the SH-5 a strong architecture for advanced set-top boxes, game consoles, and other multimedia-rich systems.

Split Branching Reduces Delays

Another important difference between the SH-5 and earlier SuperH architectures is a new split-branching scheme. The goal is to minimize branch-induced bubbles in the pipeline without the complexity of dynamic branch prediction.

The SH-5 has a new instruction called PTA (prepare target address) that preloads a branch-target address into one of eight special registers. An optional PTA/L form of the instruction allows the programmer or compiler to set a "likely" bit, which hints that the program will probably branch to that address. Setting this bit advises (but does not compel) the processor to prefetch instructions beginning at that target address into a buffer. The SH-5 core has a buffer large enough for two instructions per branch-target register, or 16 instructions total; future SH-5 implementations could have more buffers.

Jim Slager of Hitachi discussed the SH-5's backward compatibility with SuperH at the Forum.

When the processor encounters the branch later in the program, the target address is already loaded in a target register, and the target instructions may be loaded as well. The processor can execute the branch with no delay if the hint is correct. If the hint is incorrect, the penalty is only one cycle.

Programmers and compilers can append a "likely" hint to the branch instructions, too, in addition to the PTA hint. Together, these hints reduce overall branch penalties without the higher cost of dynamic branch-prediction logic. But they do place a little more burden for branch prediction on programmers and compilers.

As mentioned above, hinting that a branch is likely doesn't compel the processor to prefetch the target instructions. It's friendly advice, not a command. At run time, the processor decides what to do, based on whether it can schedule the prefetches without stalling the program. Prefetching effectively hides the latency of the instruction cache, which is 1.5 cycles in the SH-5 and will probably be longer in future implementations that have bigger caches.

A SuperHyway for Bus Traffic

To link the core with integrated peripherals on SOCs, the SH-5 has SuperHyway, a new on-chip bus. There's already such a surplus of on-chip buses that pretty soon they'll be appearing for sale in eBay auctions, but Hitachi and ST think there's room for one more. At least it's compatible with the Virtual Socket Interface (VSI) protocols.

In the 400-MHz SH8000/ST50, SuperHyway runs at 200 MHz. It has dual 64-bit read/write channels for 3.2 GBytes/s of peak bandwidth, and it uses a split-transaction packet-based protocol with memory-mapped addressing. When a device wishes to communicate with another device on the bus, it first sends a request packet to the second device. The bus is free to handle other transactions until the second device responds by sending a reply packet back to the first device. This keeps the bus available, no

Jean-Marie Rolland of STMicroelectronics described his company's contribution to the SH-5.

matter how much time the second device needs to respond, which prevents slower devices from hogging the bus.

As Figure 4 shows, SuperHyway links the SH-5 core to a 32-bit 66-MHz PCI interface; a 64-bit double-data rate SDRAM interface; a 16-bit interface for slower external devices, such as flash memory; a DMA controller; and the SHdebug interface. There's also a bus bridge that connects SuperHyway to a pair of lower-bandwidth on-chip buses for slower peripherals. The P bus accommodates real-time clocks, timers, interrupts, serial ports, and the power-management unit (PMU). The ST bus allows designers to integrate additional peripherals from ST's macro library.

The PMU has three power-reduction modes. In sleep mode, only the CPU shuts down, allowing peripherals to continue operating and performing DMA transfers. In standby mode, the clocks stop, too, and the PLL can be on or off. In module standby mode, the PMU can individually turn off any device on the SuperHyway bus. Hitachi and ST say the 400-MHz SH-5 core will typically consume less than 1 W at 1.5 V in its normal mode; they don't have estimates for a complete SOC in normal or power-reduction modes.

Debug isn't exactly a headline-grabbing feature, but Hitachi and ST say their SHdebug interface is unusually versatile and will help distinguish SH-5 chips from other SOCs. Their assertion is based on the growing complexity of SOCs and the increasingly difficult task of tracking down bugs. Only a high-speed (100 MHz) debugger interface that's invisible to other software will allow designers to diagnose problems inside a highly integrated chip, they claim.

SHdebug allows developers to single-step through their programs or set as many as 12 different breakpoints or watchpoints. When a program meets a watchpoint's condition, the processor sends a trace packet over the SHdebug link rather than stopping execution as it does at breakpoints. This permits real-time debugging.

Because the SHdebug interface attaches to the SuperHyway bus, developers can also inject data packets and observe bus traffic to test the operation of on-chip devices.

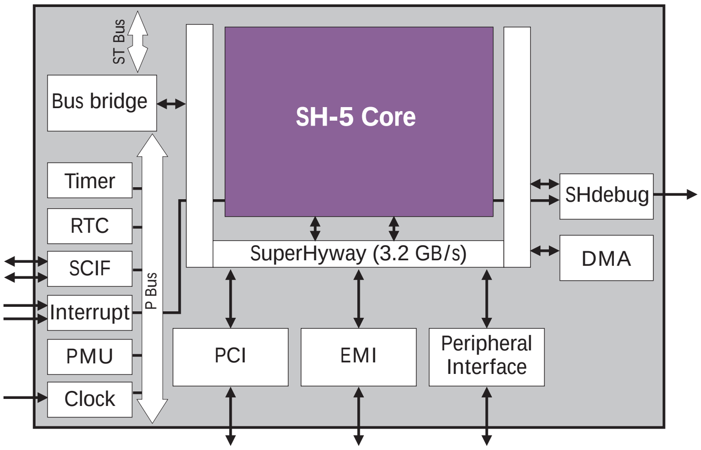

Figure 4. Hitachi's name for the first SH-5 system on a chip is the SH8000; STMicroelectronics calls this same device the ST50.

This could indeed be a valuable feature as developers integrate more and more intellectual property from numerous sources on SOCs. If something goes horribly wrong, it could lead them directly to the guilty party.

Competing in the SOC Derby

The SH-5 is yet another contender in an extremely crowded field. It's racing against numerous vendors of ARM- and MIPS-compatible cores; formidable competition from giants such as IBM and Motorola; Intel's reinvigorated Strong-Arm 2 (see MPR 5/10/99, p. 1); trendy newcomers such as Sun's MAJC (see MPR 8/23/99, p. 13); and even some x86-based challengers, such as National Semiconductor's Geode (see MPR 8/2/99, p. 16).

One of the SH-5's best selling points is code density. The ability to use 16-bit instructions in SHcompact mode conserves memory, which translates into lower system costs and less power consumption—critical factors in embedded products. As memory costs decline, this advantage may seem less important. And it's not an exclusive advantage, because rival architectures also have ways of compressing code (e.g., Arm's Thumb, IBM's Codepack, and the MIPS-16). But those code-compression schemes aren't always implemented, while the SHcompact instructions will always be available in the SH-5.

When we compare performance—or at least estimated performance, since most of the other chips aren't available either—it appears that Hitachi and ST aren't pushing the SH-5 very hard. Even in a costly new 0.15-micron copper process, their target clock speed is only 400 MHz. Other companies are hitting that frequency in 0.18-micron aluminum processes, while still others are coming close in fully amortized 0.25-micron fabs.

Of course, the SH-5 gets a lot of work done at 400 MHz. At 2.8 GFLOPS, it's almost twice as fast as SandCraft's SR1-GX (see MPR 7/12/99, p. 10) at the same frequency, which bodes well for 3D graphics.

The Emotion Engine chip (see MPR 4/19/99, p. 1) in Sony's future PlayStation 2 will execute 6.2 GFLOPS at 300 MHz. That sets a new benchmark for video gaming that the SH-5 will find tough to beat. But for any other embedded application, the Emotion Engine would be overkill, even if it were available on the merchant market—not to mention that it burns 15 W and is large enough to have its own ZIP code.

Hitachi and ST are probably clocking the SH-5 rather conservatively to improve yields and reduce power consumption. Low power is important, even for a line-powered appliance like a set-top box, because it improves reliability and may eliminate the need for a fan. It also allows Hitachi and ST to use less-expensive chip packaging. But the SH8000/ST50 still burns too much power for many mobile applications, such as smart cell phones, although it's within the envelope for Windows CE devices.

The SH-5's strengths are high code density, backward compatibility with existing SuperH software, multivendor sourcing, and excellent performance on floating-point and multimedia tasks. It is well suited for set-top boxes that include gaming or 3D-graphics capabilities, and the architecture has a modern instruction set with room for expansion. Although Hitachi and ST haven't discussed pricing, previous SuperH processors have always been competitive in this regard. Having two vendors will prevent price gouging.

As the embryonic markets for information appliances and SOCs evolve, all those qualities will make the SH-5 a strong contender. It would be an even stronger contender if it appeared in 2000 instead of 2001. With so many competitors working on high-performance embedded processors, Hitachi and ST would be well advised to accelerate their production schedules.