Transmeta Exposed

U.S. Patent Reveals Details of Secret x86/VLIW Processor

by Rich Belgard

For the casual patent watcher, U.S. Patent 5,832,205 (*www.patents.ibm.com*) is just another processor patent. From the title and claims, no one would guess that it tells much of the heretofore secret Transmeta story. But in the disclosure—the portion not available on public Web sites—Transmeta reveals architectural features that could be at the heart of its new x86-compatible chip. If what Transmeta has patented is what it has built, this is no conventional x86 processor.

Transmeta was founded in 1995 by former SPARC architect Dave Ditzel, its current CEO. The company has collected an impressive cadre of CPU designers, mostly from Sun and MIPS, and a large contingent of software gurus including, among others, Linus Torvalds, creator of Linux. Transmeta has not previously disclosed its intentions, but it has long been rumored to be developing an x86 processor. Sources indicate that the company received first silicon several months ago.

Code Morpher Translates x86 Code to VLIW

The claims of the '205 patent are narrow and cover only one aspect of the Transmeta processor. The disclosure, however, describes a "morph host" designed to run x86 programs in conjunction with "code-morphing software," as Figure 1 shows. The code-morphing software translates x86 instruction sequences into native host instructions on the fly—as they are called for execution—storing the translated sequences in a 2M buffer, probably in main memory.

The idea to use code morphing may have originated with Ditzel and Bob Cmelik, one of the co-inventors named on the patent, who is well known for his work on dynamic translators. Colin Hunter, a Transmeta founder and former head of Hunter Systems—who developed XDOS, a static translator like Alpha's FX!32—may also have played a part.

Transmeta's code morphing seems akin to the dynamic compilers used by Insignia in SoftPC and Connectix in Virtual PC to run x86 code on generic RISC processors. Transmeta's code morpher applies increasingly aggressive optimizations to code that executes more frequently, causing programs to speed up the longer they run. Transmeta's code morpher is different than previous translators, however, in that it targets a morph host that has dedicated hardware to reduce the emulation overhead that sapped the performance of these earlier translators.

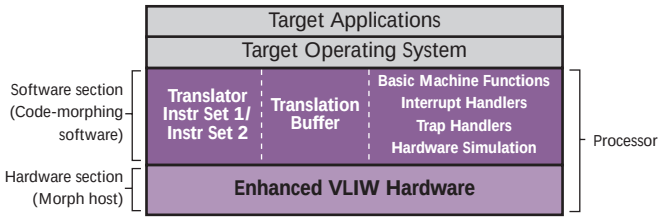

Figure 1. U.S. patent reveals Transmeta's new processor to be a VLIW engine that relies on code-morphing software to translate x86 code into native VLIW code. (Source: U.S. Patent 5,832,205)

The inclusion of x86-specific hardware on a RISC (or VLIW) core is not a new idea. It is the basis of many modern x86 CPUs, like Intel's P6, and will be used in Merced. But in these processors, x86 instruction translation is performed in the processor's execution pipeline, whereas Transmeta does it in software. Much of this existing technology is cited in the patent, so presumably Transmeta's invention is different enough to avoid prior-art claims.

The patent states that the morph host is a classical, statically scheduled VLIW engine with an instruction format capable of controlling one load, one store, two integer operations, an FP add, an FP multiply, and a branch. The processor has a large number of registers to support software register renaming; the specification suggests 64 integer and 32 FP registers. Special target-context registers hold the user-visible state, which is updated by committing the working registers. With this feature, the machine can be rolled back to a precise x86 architectural state on an exception. The processor also implements a gated store buffer that is similar in function to the working registers, but for memory data.

All x86 competitors face the problem of Intel's patent portfolio. Many solve the problem with cross-licenses or by using cross-licensed foundries. Considering the nature of the Transmeta design, however, the company may skirt this problem. Most of Intel's patents have "apparatus claims" describing specific hardware critical to implementing an x86 microprocessor. Since most of Transmeta's x86 functionality is in software, the company could avoid most of Intel's best x86 patents. Intel's patents with "method claims" could be more problematic.

Transmeta is probably placing its hopes on the simplicity of VLIW hardware to gain both cost and performance advantages. The patent states that its gate count would be about 25% that of a Pentium Pro. If true, Transmeta's die would be substantially smaller, and cheaper, than those of other x86 products. With much of the decode and issue complexity foisted onto software, the simple VLIW hardware might also achieve higher clock rates than conventional x86 processors, allowing it to overcome the emulation overhead.

If Transmeta can achieve full x86 compatibility and good performance at low cost, its unique approach could change the future of microprocessor design. Future "processors" could become part hardware and part software. M