

Critical Analysis of a Commercial Processor

Transmeta's Crusoe TM5600

EE6810

Advanced Computer Architecture

Curt Nelson

December 11, 2000

Prologue

What you are about to read has been researched and written in the spirit of Advanced Computer Architecture. It is likely to depart from the technical requirements for two reasons: lack of access to technical details, and the fact I was having more fun reading about the issues involved in the company and processor development. Nonetheless, I trust you can sit back and read this as much for technical "entertainment" as for insights into a complex processor trying to find landing room in a highly competitive and mature field.

Introduction

The Transmeta Corporation was founded in 1995 by David Ditzel, previously the chief technical officer at Sun Microsystems. The five year, highly secretive development of the Crusoe processor resulted in Transmeta making an offering to the public of its line of processors early in the year 2000. About a year prior to this time, Transmeta signed on several noteworthy companies to design their processors into new products, including Sony, Nec, Gateway, and others working in web-based appliances and/or mobile computing applications. The goal of the company was to develop a new hardware processor core surrounded by a software process that translated X86 code into a form that the core could execute (See figure next page, courtesy of Transmeta and a Canon scanner). The hardware core is based on a Very Long Instruction Word (VLIW) architecture. The VLIW word itself is packed with 4 x86 type instructions that operate in parallel. The packing of the instruction is the job of the translation software termed Code

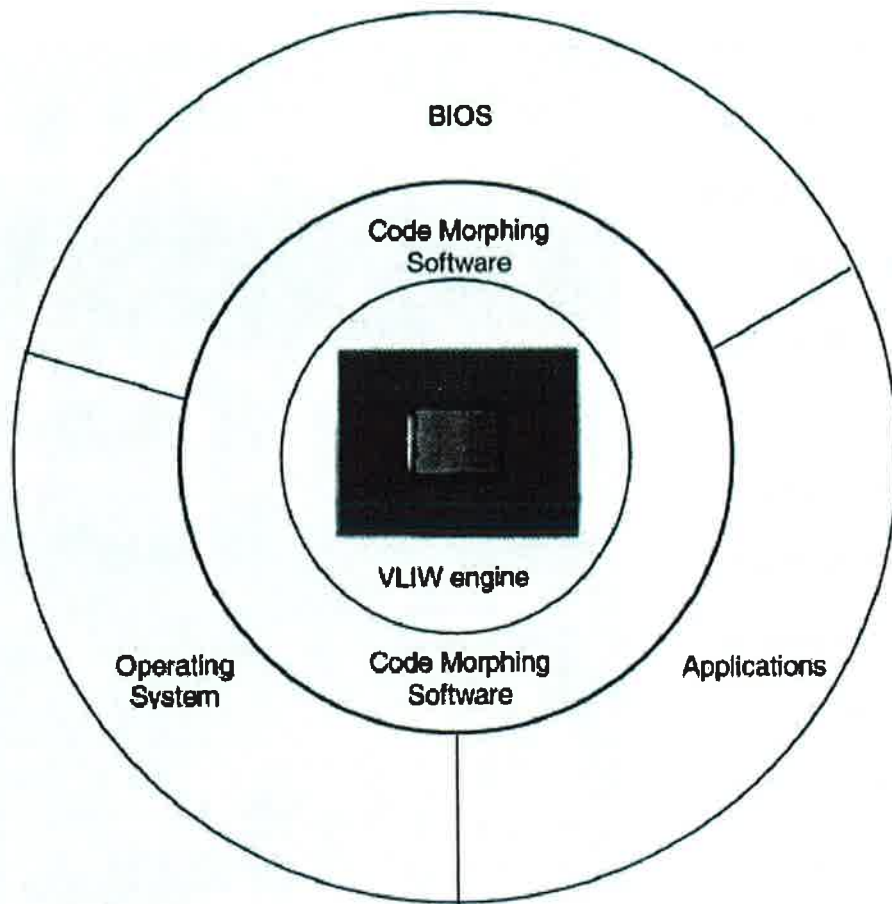


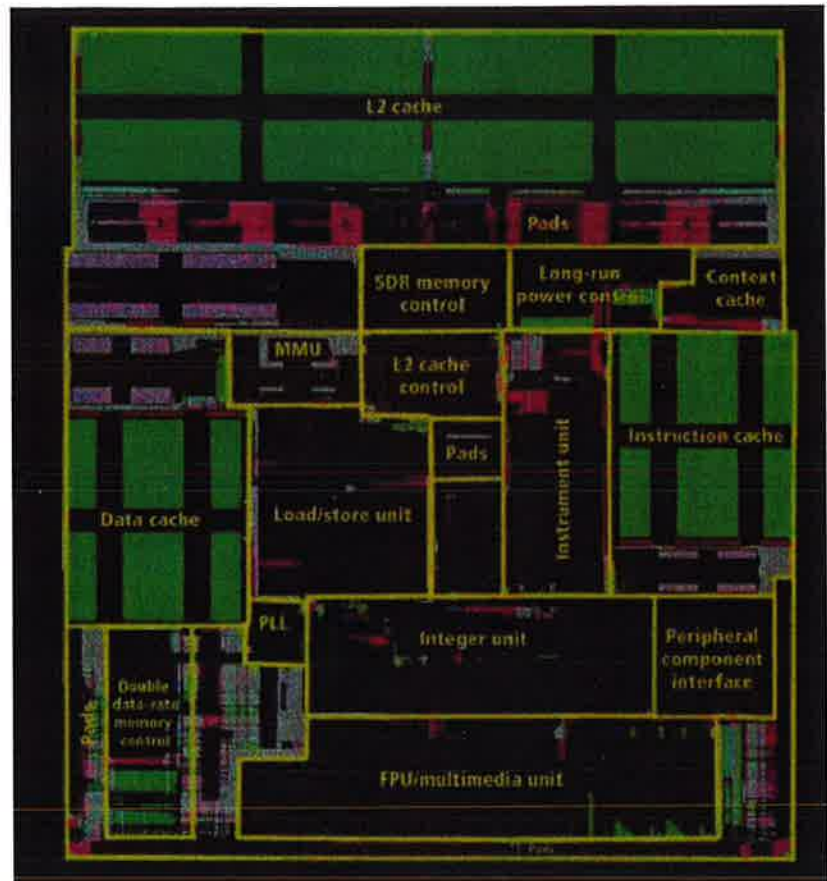
Figure1.

Morphing Software (CMS) by the engineers at Transmeta. While it may appear to the outside world that the Transmeta processor is an x86 clone, in reality it is a completely different processor core surrounded by an intelligent and novel translation process. Since speed is critical to consumer acceptance, the CMS had to be real time and fast, hence an actual hardware translation as done by a PROM or other hardware based circuits.

The overall goal of the design team was to design an x86 compatible processor with significant reduction in power usage without taking a large hit in throughput. Their marketers aimed at the mobile computing industry who was, and still is, thirsty for laptops and notebook computers that have acceptable performance and a battery life of at least 8 hours.

Layout

The photo at the right (courtesy IEEE Spectrum) shows the layout of the TM5400 chip. The chip measures 73 mm² and is manufactured by IBM in a 0.18μ process with 5 copper interconnect layers. The chip contains separate 64K level 1 data and instruction caches in addition to a 256KB combined level 2 write-back data/instruction cache. Perhaps the most significant feature to the



Crusoe processor was the power management circuit, deemed Long-run power control.

Architecture

General

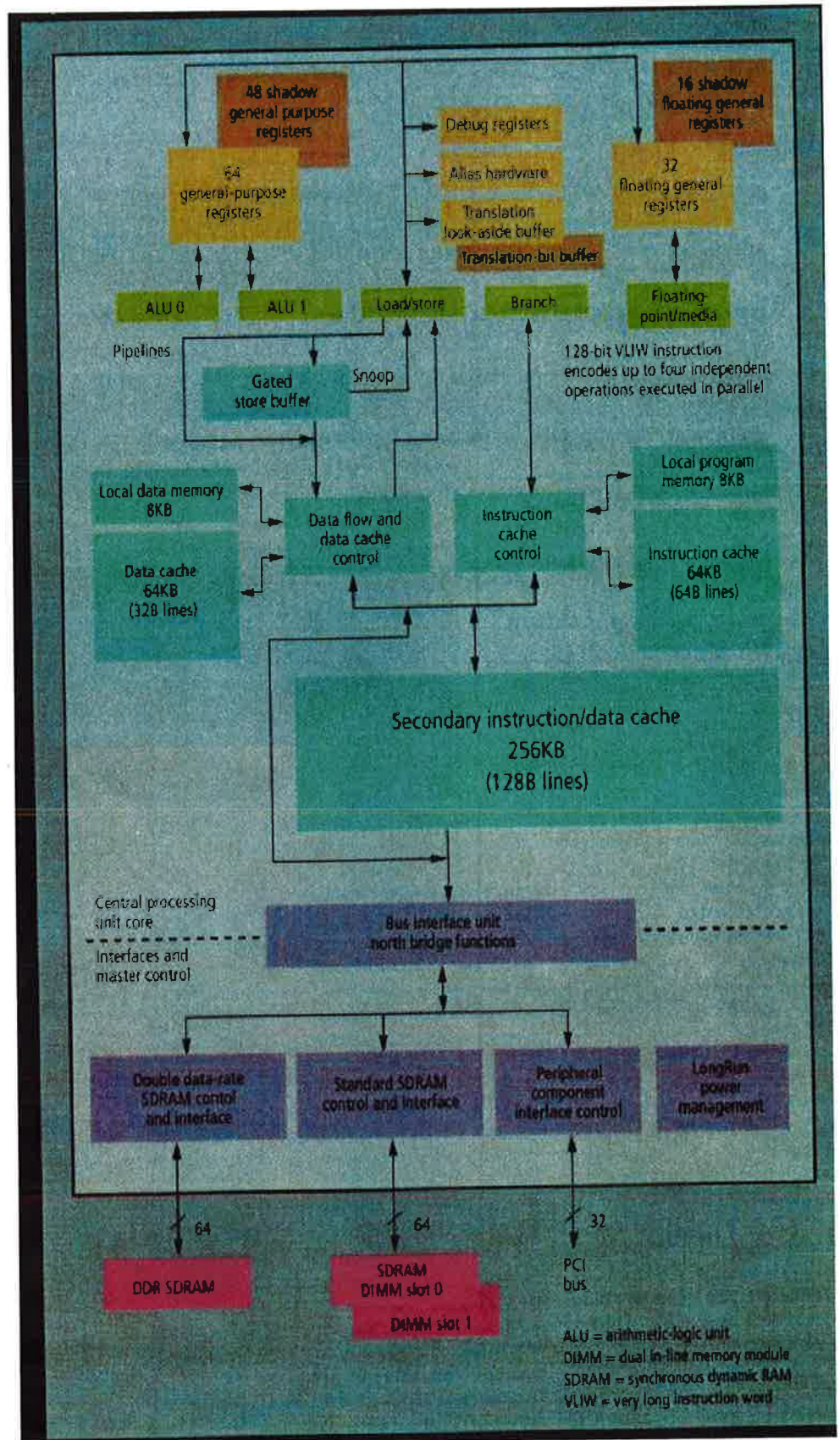
See figure next page (courtesy IEEE Spectrum). The actual hardware core contains about 1/2 the transistors of an x86 processor. It includes 5 execution units (two ALU's, a load/store, a branch, and a floating point unit). It contains 64 general purpose registers and 32 floating-point registers, shadowed by 48 general purpose and 16 FP registers. In addition, the Crusoe integrates a double data-rate SDRAM controller, a standard data rate SDRAM controller, a PCI bus controller, and a

serial ROM interface controller. The integration of these last four items removes some of the need for peripheral circuits in a system solution.

The instruction execution sequence is scheduled by software. This provides for a simpler hardware implementation that includes an in-order 7-stage integer pipeline and a 10-stage floating point pipeline.

Caches

Crusoe includes a 64K 8-way set-associative Level 1 instruction cache, a 64K 16-way set associative



L1 cache, and 512K L2 write back cache. The Translation Lookaside Buffer (TLB) has the same protection bits and address mapping as x86 processors.

Code Morphing Software

The VLIW word (called a molecule) is packed with four X86 type instructions (called atoms). It was felt by the design team that if the primary goal is to save power, then the burden of performance must be placed upon the software wherever possible, read compiler. To match throughput of an AMD or Intel processor would require a superscaler architecture with its associated complexity, increasing hardware and eventually the anticipated power savings would erode.

An example of VLIW packing is shown at the right. The typical behavior of the CMS is to execute a loop of code. The first few times that a code sequence is executed, CMS dispatches a VLIW by packing x86 instructions one byte at a time into the 128-bit VLIW. Once the



instructions have been executed several times, the CMS translates the x86 instructions into a highly optimized VLIW and caches the native instructions translations for future use. If the

same code is repeated, (i.e. a loop) the already optimized VLIW instructions are executed and no re-translation is necessary.

Exception Handling

Foremost on the mind of the engineers was how their architecture was going to handle exceptions. Their approach was termed a commit and rollback and was implemented by creating an extra set of registers called shadow registers. The purpose of the shadow registers is to essentially duplicate the data in the working registers. While the main working registers could update their data as needed, the shadow registers would hold their data until the instruction associated with the data committed. In the event of an exception, the data is copied back from the shadow registers to the working registers and the instructions causing the fault are reissued, with slightly more conservative scheduling.

Store buffer

For an exception, data to be stored in memory would also have to be rolled back. This is accomplished through a gated store buffer which keeps track of the stores between commit points. If an exception occurs during this time, the system can roll back to the previous state and discard those stores. The store buffer is divided into a committed and uncommitted sides. Once the instruction associated with the data commits, the data is moved from the uncommitted side to the committed side and then stored in memory. This can involve a substantial amount of data as one x86 instruction can modify up to 130 bytes of memory.

Long-run Power control

The CMS was employed as a means to reduce power consumption. The CMS monitors applications as they are running and then instructs the hardware to adjust both the clock frequency and the supply voltage to the necessary circuits. This results in double the power

savings. The CMS determine will determine how often a given application is in sleep mode and adjusts the clock frequency accordingly. The benchmark results show a power reduction of three to thirty times, depending on the application.

The Crusoe processor operates from 1.2V - 1.6V core voltage supply. This voltage is adjusted on the fly to support current processor demands. Crusoe also supports standard power management models by incorporating support for five distinct power states: Normal, Auto Halt, Quick Start, Deep Sleep, and Off. These states may be used to further reduce power consumption through system level inactivity. The Crusoe architecture is notable for what it does not include. There is no superscaler decode, grouping, or issue logic. There is no register renaming, segmentation, or memory management hardware. The goal and result is fewer transistors, saving power.

Input / Output

Transmeta incorporated many features normally found off-chip in their TM5600 processor. The DDR SDRAM controller provides a direct interface to double data-rate RAMS, removing one layer of off-chip interfacing. The SDR SDRAM controller is used for slower main memory. These interfaces run between 100 and 133 MHz.

The PCI bus controller is PCI 2.1 compliant. It is 32 bits wide and is compatible with 3.3V signal levels. The controller provides a host bridge, a bus arbiter, and a DMA controller. It can sustain 132 Mbytes/sec bursts for reads and writes.

The serial ROM interface is a five-pin interface and is used to read data from a serial flash ROM. The ROM is 1M-byte in size and stores the CMS. The contents of the ROM are copied to system memory at boot-up.

Analysis

The actual core instruction set architecture is a complete mystery to me. This is due to a lack of available data. From the outside, the TM5600 looks all the world like any other CPU executing native x86 code. However, the CMS layer between the external program and the internal ISA hides what exactly the ISA is.

Transmeta's approach has some advantages:

- The core hardware can be changed if a design flaw is detected or a new architectural feature is needed. The CMS interface must then be changed to incorporate this revision. This change is transparent to the x86 code. In essence, Transmeta has made their processor "software upgradable". Since the CMS resides in flash memory, the processor can be updated over the web.
- Debugging is enhanced by allowing the CMS programmers to change the way they pack a molecule. For simpler debugging, one could place a single atom into the VLIW and execute accordingly. Or one could try various combinations of instructions to see which execute faster/slower, or to see what dependencies tend to slow things down.
- The core hardware can be kept simpler by placing more burden on the compiler side of the equation.
- Obviously, the core hardware of Crusoe is not limited to executing x86 instruction sets. Although it has been steered that direction and optimized for that purpose, the same hardware can execute any instruction set simply by changing the CMS layer. The entire processor itself seems more like a highly optimized emulator that can be used to implement any instruction set that can be translated into the VLIW instruction format.

- Crusoe's CMS requires a translation cache for storing packed molecules. This operates much like a TLB for virtual memory. Essentially, it is generating "code locality".

I can see some disadvantages to this architecture as well, but not as many:

- The additional run-time translation imposed by the CMS is an extra layer that must be traversed during dynamic execution. If not designed carefully, this could slow performance. The designer's hope was that the additional intelligence built into the CMS would allow the clock and power management circuits to dynamically adjust to the task at hand. Initial benchmarks are not showing the kind of throughput numbers that Transmeta has promised.
- The decision where to draw the hardware/software line in the design process is always debatable. It seems like Transmeta has focused their efforts around the laptop / PC application arenas, and rightly so. What about servers or other high end systems? Dual or multi-processor? These systems have some unique features that may be better suited for other processors.

Epilogue

Even as you read this, Transmeta is dealing with a recall announced last week (Dec 1, 2000). It seems there is a design or manufacturing flaw in their high end processor, the TM5600. As it stands, the only affected commercial product is produced by NEC which has recalled about 2500 laptop computers to have the CPU replaced.

Transmeta also faces the challenge of finding a new manufacturer of its chips as IBM has decided not to be a fab house any longer. This shows the vulnerability of a fabless semiconductor company. If the past is any indication, Transmeta will no doubt find a way to meet these challenges head on. I'm pulling for them, as I feel the design goals and architecture

warrant serious consideration for portable applications. And like my digital camera and GPS unit, it would be cool to download new versions of my processor over the web.

I have had fun tracking down what information I could and assembling it into this report. I learned something I wouldn't otherwise have known, and will continue to follow the development of the Crusoe processor family as well as the company itself with renewed interest.

Bibliography

Transmeta's Magic Show, IEEE Spectrum, May 2000, Pages 26-33

Transmeta breaks x86 low power barrier, The Microprocessor Report, 14 February, 2000, pages 9-18

Web pages

www.transmeta.com

www.pcworld.com

www.eetimes.com

www.mdronline.com (Embedded Processor Forum)

www.hotchips.org (IEEE's Hot Chips Conference)