

MPR'S ANALYSTS' CHOICE AWARDS

By Max Baron {1/22/02-01}

Each year *Microprocessor Report* analysts review and evaluate more than 100 microprocessors, digital-signal processors, and application-specific digital machines. The most interesting and innovative products make it into our newsletter, some soon after their proud designers have

presented them to the world from the stage of *MPR's* Microprocessor or Embedded Processor Forum. Then, in the first month of each following year, *MPR's* analysts gather for the exciting and difficult task of selecting the best of the best.

Looking back at 2001, one of the hardest years for the industry, we at *MPR* are encouraged by the number of innovative designs that have been brought to successful completion; this feeling is further strengthened by the numerous submittals of abstracts for new-product presentations at *MPR's* upcoming 2002 Embedded Processor Forum.



The analyst team long ago gave up picking one single best processor, because so many are designed for, and excel in, specific applications that range from desktop computers through handsets and MP3 players. This year's *MPR* awards will recognize the best processor in nine categories, each category having three or more nominees competing for the top spot.

Microprocessor Report is proud to present the nominees for its annual awards honoring 2001's best processors.

Outstanding Technology in the Field of Digital Processing

- Intel Hyper-Threading Technology
- Proceler Dynamically VArIable Instruction seT Architecture (DVAITA)
- Sun Microsystems Laboratories Asynchronous Design Technology
- Theseus Logic NULL Convention Logic (NCL)

Best DSP Cores:

- 3DSP UniPHY
- BOPS WirelessRay
- Infineon Carmel 1000
- LSI Logic ZSP400
- Siroyan OneDSP

Best Digital Signal Processors:

- Analog Devices Blackfin 21535
- Analog Devices TigerSHARC TS101S
- LSI LSI402ZX
- Motorola 8102
- Texas Instruments C6414

Best Gaming Chip Set:

- Microsoft Xbox: Intel Pentium III, Nvidia XGU/MCPX
- Nintendo GameCube: IBM Gekko processor, ATI Flipper
- Sony PlayStation 2: Sony Emotion Engine and Graphics Synthesizer
- PC desktop: AMD Athlon XP, VIA Apollo KT2 66A, Nvidia GeForce3

Continued on page 4

AT A GLANCE

◆ PROCESSORS

MPR's Analysts' Choice Awards1
Microprocessor Report announces the Analysts' Choice Award nominees, recognizing the best processors in nine categories, each category having three or more nominees competing for the top spot.

Server Battles Heat Up in 200111
 The year 2001 was a watershed year for servers. See which one was the analysts' choice as best of 2001.

◆ EMBEDDED

ARM Shakes Hands With DSP5
 Dual-core devices are becoming increasingly popular, especially for use in low-power systems. The new C547x devices from TI are the first in a family of devices from the company.

2001: A Graphics Odyssey7
 Video-game consoles from Microsoft, Nintendo, and Sony competed among themselves and against the PC in 2001. We pick a winner.

DCT Marches Into Java Processors14
 Another Java processor vendor steps up to bat. Does it have what it takes to compete?

◆ DEPARTMENTS

Intel and Microsoft: Together Forever?3
 Intel and Microsoft have been partners for more than 25 years. Despite efforts by both companies to find other arrangements, today we have four Wintel platforms to choose from, not just one.

Most Significant Bits23
 Gigahertz UltraSPARC III SPEC Surprise; AMD Starts 2002 With Model 2000+; BOPS Announces New Performance Levels; Intel's 2.2GHz P4 Pulls Ahead

Literature Watch24

Patent Watch25

Chart Watch: PC Processors26

Resources28

MICROPROCESSOR REPORT WWW.MPRONLINE.COM

Editor in Chief

MAX BARON
mbaron@mdr.cahners.com

Senior Editor

PETER N. GLASKOWSKY
png@mdr.cahners.com

Senior Editor

KEVIN KREWELL
kkrewell@mdr.cahners.com

Senior Editor

MARKUS LEVY
mlevy@mdr.cahners.com

Senior Editor

CARY D. SNYDER
[cgs@mdr.cahners.com](mailto:cds@mdr.cahners.com)

Managing Editor

LESLIE FISH

Production Editor

PIA WALKER

Editorial Board

DENNIS ALLISON, ANDY BECHTOLSHEIM,
 RICH BELGARD, BRIAN CASE, JEFF DEUTSCH, DAVE EPSTEIN,
 DON GAUBATZ, JOHN NOVITSKY, BERNARD PEUTO,
 NICK TREDENNICK, JOHN WAKERLY

Published by **MICRODESIGN**
RESOURCES A member of the Cahners Electronics Group

1101 S. Winchester Blvd., Building N, San Jose, CA 95128
 Sales/Customer Service (in Scottsdale Arizona): 480.609.4551
 Fax: 480.609.4523; Email: emckeighan@instat.com

Microprocessor Report is published weekly at www.MPRonline.com
 and monthly as a paper publication (ISSN 0899-9341).
 See back cover for subscription information.

Computer Press Award, Best Newsletter:

1999, 1998, 1997, 1994, 1993

Copyright ©2002, Cahners MicroDesign Resources.
 All rights reserved. No part of this newsletter may be reproduced,
 stored in a retrieval system, or transmitted in any form or
 by any means without prior written permission.



THE EDITORIAL VIEW

INTEL AND MICROSOFT: TOGETHER FOREVER?

By Peter N. Glaskowsky {1/28/02-02}

How many times has Microsoft tried to help create a non-Intel computing platform? At least five that I can think of. Over the years, Intel has invested millions of dollars into supporting non-Microsoft operating systems. None of these efforts has ever seemed to matter. Despite

all their efforts to escape the “Wintel” moniker, the two companies seem fated to remain bound together for eternity.

Microsoft and Intel have been together since the dawn of the microprocessor. Intel’s 8080 was one of the first widely used microcomputer CPUs, and Microsoft’s BASIC was one of the first popular high-level languages for microcomputers. In its early days, however, Microsoft had no special relationship with Intel. For example, Microsoft worked with Apple and Radio Shack, which used non-Intel CPUs.

It was Microsoft’s historic 1981 action that tied Microsoft and Intel together. Microsoft agreed to provide MS-DOS for the IBM PC, and, over the next several years, Microsoft established parallel deals with PC-clone makers such as Compaq. The IBM deal gave Microsoft control over two-thirds of the critical software running on the PC—the operating system and development software.

Through the 1980s, Microsoft built up the third leg of this strategic triad: application software. At the same time, however, Microsoft was also involved in deals to create alternatives to the PC. The first of these was the MSX system, a home computer codeveloped in 1983 by Microsoft, the Japanese company ASCII, and major Japanese consumer-electronics companies that included Matsushita and Sony. The Z-80-based MSX machines were much less expensive than PCs, had comparable software, and were much more popular in some parts of the world—but not in the United States.

Microsoft provided a version of BASIC compatible with the PC’s GW-BASIC; an 8-bit version of MS-DOS called, predictably, MSX-DOS; and limited application software. MSX machines were used primarily as game consoles, but Microsoft was not yet a major player in game software. Although the MSX platform evolved through the 1980s, it could not evolve sufficiently to keep pace with the PC, which eventually became the world standard for home and personal computing. Simultaneously, despite stiff competition, Intel processors became the standard choice of PC vendors.

In 1985, just two years after the debut of MSX, Microsoft’s Bill Gates flirted briefly with the notion of throwing Microsoft’s considerable weight behind the Macintosh platform. Gates recognized that the Mac’s sophisticated combination of hardware and software was technically superior to that of the IBM PC architecture, but his efforts to get Apple to open up the Mac to outside hardware and software developers were rebuffed.

In the early 1990s, Microsoft resolved to try even harder to break Intel’s grip on the personal-computer industry. Microsoft hired David Cutler, an architect of Digital Equipment Corporation’s VMS operating system, to create what would become Windows NT. Microsoft decided to make NT platform neutral: that is, it would not be tied to the x86 architecture. All early NT development, in fact, took place on MIPS-based workstations. By writing and testing all the NT code on MIPS processors, Cutler’s team could be sure that no undesirable x86-specific code existed in NT.

It was Microsoft’s plan to support NT on multiple processors and let the market decide which implementations would succeed. Most of the senior NT team members believed that MIPS and Alpha would dominate their most important target markets: servers and workstations. Indeed, when NT finally shipped, the fastest and most capable NT machines had inside them RISC processors—not Intel. These systems came with very high prices, mandated by their high development costs and low sales volumes, and the NT market quickly moved to standardize on x86 once more.

Ironically, while Intel insisted throughout the early days of NT that there was no need to leave x86 behind, internally, it knew better. Just as x86 was winning the battle for NT, Intel announced it would develop its own non-x86 processor for servers and workstations. The new Itanium architecture is years behind schedule now, and its ultimate fate is uncertain, but Intel has already been more successful in offering an alternative to x86 than have all the RISC NT vendors put together.

MPR's ANALYST CHOICE AWARDS

(Continued from Page 1)

Best High-Performance Embedded Processor:

- Broadcom BMC1250
- IBM PowerPC 750FX
- Motorola Apollo chip
- NEC VR 5500
- PMC-Sierra RM9000X2

Best High-Performance Processor Soft Cores:

- ARM 1020E Core
- MIPS Technologies MIPS64 20Kc Core
- Tensilica Xtensa Core

Best Network Processor:

- Agere Payload Plus
- AMCC nP7250
- IBM PowerNP NP4GS3
- Motorola C-Port C-5
- Vitesse IQ2000

Best PC Processor:

- AMD Athlon XP
- AMD Duron
- Intel Northwood (Pentium 4)
- Intel Tualatin (Mobile Pentium III-M)

Best Security Processor:

- Broadcom BCM5840
- Corrent CR7020
- Hifn 8154
- Securelink PCC-ISES

Best Server/Workstation Processor:

- AMD Athlon MP
- Compaq Alpha 21264C 1,001MHz
- IBM Power4/Regatta
- Intel Itanium
- Intel Xeon MP

THE EDITORIAL VIEW (continued)

This success has simply cemented Microsoft's dependence on Intel at the high end of the market.

Perhaps inspired by NT's promise of processor independence, Microsoft embarked in the mid-1990s on another effort to create a CPU-neutral operating system. This effort led to Windows CE and several generations of handheld and pocket-size systems. CE machines have been built around ARM, MIPS, PowerPC, SuperH, and even x86 processors. Unlike NT, CE succeeded in breaking loose from x86.

CE, however, did not lead to the diversity of solutions Microsoft sought. The latest generation of Pocket PC systems is based solely on one processor architecture, StrongARM, originally developed by Digital. In the greatest irony of all, StrongARM is now an Intel product.

Microsoft's most recent attempt to foster non-Intel processors never really had a chance. The Xbox video-game console shipped with an Intel processor, but for most of the early days of the project, AMD was tipped to be the

front-runner. If Xbox had come along five or six years earlier, it might even have used a RISC processor. Instead, Xbox is just another Intel x86 machine.

Today, despite years of effort, Microsoft's strategic planning remains Intel focused. The vast majority of Microsoft software is run on Intel processors. AMD makes good CPUs, but AMD has no meaningful influence on Microsoft's strategies. Microsoft makes good money on Macintosh application software, but these products simply parallel the company's own Windows products.

Will Microsoft keep looking for Intel alternatives? Almost certainly, but it's likely to be a few years before the next such effort emerges. In the meantime, we'll have four Wintel platforms—Windows XP on x86, Windows XP on Itanium, Pocket PC, and Xbox—to choose from, not just one. ♦

Peter W. Fleury

ARM SHAKES HANDS WITH DSP

New TI Devices Combine ARM7 and C54x

By Markus Levy {1/7/02-01}

Two new dual-core chips from Texas Instruments won't break any performance records, but the chips do provide a tidy and compact solution for low-end connected applications. The TMS320C5470 and 'C5471 combine a 100MHz 'C54x DSP with a 47.5MHz ARM7TDMI,

a host of microcontroller-type peripherals, and a 10/100 media access controller (MAC). (See Figure 1.)

Rather than elaborating on the chip's peripherals, we shall consider some interesting design implementations. First, and probably most important in a dual-core design, is the intercore communication mechanism. This mechanism is handled by the ARM port interface (API). Within the 'C54x subsystem, there are four 8K x 16-bit data RAM blocks. The API provides the ARM with access to one of those blocks, with certain limitations: if the DSP and the ARM7TDMI try to perform an access at the same time, the microprocessor has access priority and the DSP waits one cycle. In a properly designed system, this simultaneous access will occur during the bootstrap process, when the ARM7 transfers code to the DSP, an event that is not performance critical. A software handshake mechanism informs the DSP when a block of code is ready for relocation. When communication is going in the other direction, the DSP has access to a 2K x 16-bit shared-memory interface within the ARM subsystem. This shared memory is enabled during the API boot mode and is aliased to the upper 2K x 16 bits of DSP program space to allow the DSP to begin fetching code at the reset vector area when the ARM7 releases the DSP reset. The memory map changes when the ARM7 disables the API boot mode.

Another interesting, but certainly common, aspect of the dual-core approach is that each core has its own PLL (phase-locked loop), allowing each core to run at different clock speeds (the ARM7 at 47.5MHz, the DSP at 100MHz). The API has programmable timing to allow for wait states, and the clocks need not be integer multiples.

There is one peripheral in this chip that deserves a bit of explanation, and that is the Ethernet state machine (ESM) module responsible for packet routing (not depicted in Figure 1). The ESM's main task (to offload the ARM processor) is to wait for an Ethernet packet to become available in a receive queue, look at its destination address, and pass it to a transmit queue corresponding to the destination address value.

Power consumption will directly benefit from any level of integration, and this dual-core system-on-chip is no different. TI claims for the 'C5471 a 27% power reduction from the power consumed by a discrete system implementation (175mW versus 240mW). This claim was made for the 'C5471 running in nominal conditions (1.8V core, 3.3V I/O), with the DSP executing program code from internal SRAM consisting of 50% NOP/50% MACD instructions at 100MHz, and the ARM7TDMI executing the antiquated Dhrystone program from external SRAM at 47.5MHz. Although *MPR* considers this operation very atypical, the real-world power

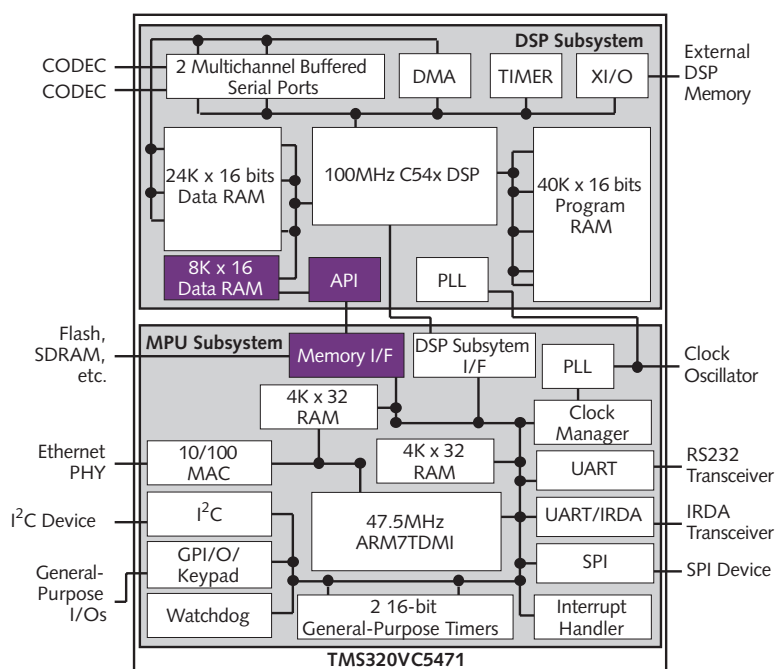


Figure 1. Dual-core devices, such as this C5471, provide significant benefits for performance, power consumption, system design, and manufacturing. Of course, the system designer may find that there are certain limitations, such as being "stuck" with a 47.5MHz ARM7 core.

Pricing and Availability

The C5470 and C5471 DSPs are available today in production quantities. The C5470 DSP is priced at \$15.50 and the C5471 DSP at \$17.57, both in 10,000-unit quantities.

saving is probably still within the same order of magnitude. (Note: As a reference point for the discrete system implementation, TI used power numbers for Atmel's ARM-based AT91M40800, Analog Devices' ADSP-2189, and a simple peripheral subsystem.)

A fairer comparison of the two implementations would probably be the saving in board space (as measuring package size and board layout is considerably more straightforward). The benefit is an approximate 40% saving in board space, which also translates to a less expensive system design.

Other DSPs With ARMs

The new 'C547x devices are not the only devices that have an ARM processor integrated with a DSP (or, depending on your perspective, a DSP integrated with an ARM processor). Analog Devices (ADI) offers its AD6522 GSM digital processor, which combines a 65MHz ADSP218x DSP core, a 39MHz ARM7TDMI core, and 1Mb SRAM. Combining this processor with ADI's AD6521 voiceband/baseband codec produces the company's msp430 SoftFone chip set for cellular phones. Pricing for the chip set is \$15 in 100,000-unit quantities.

TI also offers other devices that include an ARM processor plus a DSP. For example, the company's OMAP710 for GSM includes a C54x DSP plus an ARM9 processor. Pricing for this device is unavailable, but we would expect something in the \$20 range, to be competitive. The OMAP710 and C547x devices will help support TI's recent agreement with Palm to power its next-generation handheld computers using OMAP. ♦

2001: A GRAPHICS ODYSSEY

Games Get the Spotlight, But PCs See More Progress

By Peter N. Glaskowsky {1/28/02-03}

Video games dominated media coverage of graphics technology in 2001. Sony's PlayStation 2 had its first full year of sales, and both Microsoft and Nintendo shipped their own consoles in time for the critical Christmas season. The new machines did as well as could be expected in

a weak U.S. economy, but PS2 systems and games did even better.

Consoles don't seem to be much of a threat to the PC, however, despite predictions to that effect by Sony and other companies. In fact, technology developed for video games is now making PCs faster. For example, Nvidia's integrated-graphics chip set, designed for Microsoft's Xbox, formed the basis of its nForce Athlon chip set. Similarly, ATI is migrating elements of the ArtX 3D core in the Nintendo GameCube Flipper graphics chip into its own PC products.

To help resolve this controversy, we at *Microprocessor Report* have decided to give an Analysts' Choice Award for Best Gaming Chip Set of 2001. Console systems are represented by three nominees: Microsoft's Xbox, with Intel's Pentium III and Nvidia's XGPU/MCPX chip set; Nintendo's GameCube, with IBM's PPC405-based Gekko processor and the ATI/Nintendo Flipper system controller; and Sony's PlayStation 2 with Sony's own Emotion Engine processor and Graphics Synthesizer chip. We also considered the PC chip set most highly regarded by gamers: AMD's Athlon XP 2000+ processor, VIA's Apollo KT266A core-logic chip set, and Nvidia's GeForce3 graphics accelerator. (The Athlon XP 2000+ processor was available in sample quantities during 2001, although it was not announced until the first week of 2002.)

Microsoft Raises Bar for Console Gaming

The introduction of Microsoft's Xbox in November set a new standard for video-game console features, quality, and performance. Xbox was the first console to ship with an internal hard disk as standard equipment, and its unified-memory system architecture gave the system capabilities unmatched by the competition, such as full-time antialiased graphics and support for high-definition video output.

Most Xbox games match the visual quality of the best PlayStation 2 titles. Where the same title is available on both platforms, such as *SSX Tricky* from Electronic Arts, Xbox produces distinctly superior graphics. There are excellent games



on both platforms, of course; hardware considerations are still secondary to the effort applied by game developers. Xbox offers two key advantages over PlayStation 2 for game developers: a simpler, yet more powerful, programming model and significant compatibility with the Microsoft Windows platform.

The Xbox programming model is already familiar to most PC software developers. It is conceptually simple: high-level application code runs on Xbox's 733MHz Pentium III-based custom processor, while low-level audio and 3D functions are handled by dedicated silicon. Table 1 shows the basic specifications of Xbox, along with those of GameCube and PlayStation 2. Published reports claim the Xbox CPU has just 128K of L2 cache, making it more like a Celeron product, but full details of the chip's configuration have not been officially released.

Microsoft and Nvidia codeveloped the Xbox graphics processing unit (XGPU), which acts as a memory controller, PCI bridge, and graphics accelerator. The XGPU is connected by a HyperTransport link to the Media/Communications Processor for Xbox (MCPX), designed by Nvidia and including a pair of high-performance MediaStream DSPs sourced from Parthus Technologies.

Microsoft, of course, provided the Xbox system software and software-development platform. Xbox is designed to run a customized version of the kernel from Windows 2000, but, in principle, any OS could be used; we expect Microsoft will eventually move developers to a Windows XP-derived kernel. Device drivers support the OS kernel and application programming interfaces (APIs) derived from those used in Windows 2000, most notably the DirectX multimedia API set. Game developers can share the majority of the code used for an Xbox game with a Windows game, and vice versa, because of this software architecture.

Although the same potential for portability applies to PC productivity software, we don't expect to see Microsoft Office on Xbox. Microsoft is unlikely to risk its revenue from

Feature	Microsoft Xbox	Nintendo GameCube	Sony PlayStation 2
Processor	Custom Intel Pentium III	Custom IBM PowerPC 405	Emotion Engine
Processor Speed	733MHz	485MHz	295MHz
Architecture	Superscalar x86 core 64-bit integer SIMD 128-bit FP SIMD	Superscalar PowerPC core 64-bit FP SIMD	Superscalar MIPS core with 128-bit integer SIMD Two 128-bit vector units
Processor Cache	L1: 16K I + 16K D L2: 128K unified	L1: 32K I + 32K D L2: 256K unified	L1: 16K I + 8K D
System Memory	64M 128-bit 200MHz DDR	24M 64-bit 325MHz 1T-SRAM 16M 8b 81MHz DRAM	32M RDRAM 2 16-bit 800MHz channels
3D Engine	Custom Nvidia GeForce3	Custom ATI/ArtX	Graphics Synthesizer
Clock Rate	250MHz	162MHz	147MHz
Pixels/cycle	4	4	16
Texels/cycle	8	4	8
Polygons/second	60M (theoretical)	33M (theoretical)	75M (theoretical)
Graphics Memory	Unified in main memory	Frame buffer: 2M 1T-SRAM Texture buffer: 1M 1T-SRAM Integrated	4M embedded DRAM Integrated
Audio	Dual 200MHz DSPs	81MHz DSP	Handled by CPU
Mass Storage	DVD-ROM 8G hard disc	8cm optical disc (1.5G)	DVD-ROM
I/O options	4 controller ports Ethernet	4 controller ports 2 serial, 1 parallel	2 controller ports USB, 1394

Table 1. Specifications of the major video-game consoles vary widely, but all produce roughly the same level of effective performance for running game software and rendering 3D graphics.

PC operating-system and application-software sales. A cheap Xbox running cheap productivity software would surely pose such a threat.

It's worth noting that Xbox is not Microsoft's first foray into gaming, as some reports have claimed. It's actually the company's second hardware platform for gaming and its third software platform. The first game console designed with Microsoft's help was the 8-bit MSX machine of the 1980s, which had some success in Asian markets but only limited sales in the United States. Much like Xbox, MSX was intended as a home-entertainment computer system. MSX machines were made by several vendors and were offered with games and some limited personal productivity tools.

Microsoft's Xbox software strategy is even more directly comparable with the company's effort to promote a derivative of Windows CE as a development and runtime environment for Sega's Dreamcast console. Sega determined Dreamcast's hardware architecture, however, and offered its own software environment, which gave better access to the features of the system's PowerVR-based graphics core.

Microsoft learned much from both these prior experiences, likely explaining why the company retained complete control of the critical elements of Xbox: hardware, software, and marketing. Because of this increased control, Xbox will easily surpass MSX and Dreamcast as contributors to Microsoft's revenue stream—if it has not already done so.

Nintendo Goes It Alone

Much less is known about the hardware and software that underlies Nintendo's GameCube. We know that GameCube's

CPU was designed by IBM and Nintendo, and that it uses a PowerPC 405 core. This core runs at 485MHz, achieving a nominal 1,125 Dhrystone mips at that speed. The chip's 64-bit, 162MHz bus connects to the Flipper system controller codigned by ATI and Nintendo.

Flipper includes memory and I/O controllers, as well as a graphics core based on the ArtX technology ATI acquired in 2000, an audio DSP core from Macro-nix, and two banks of integrated DRAM. These banks of DRAM—2M of frame buffer and 1M of texture cache—are implemented with the MoSys 1T-SRAM technology, giving them (nearly) the speed of SRAM with (nearly) the density of conventional DRAM. GameCube, like PlayStation 2, uses integrated DRAM to reduce the bandwidth demands on off-chip memory.

This approach also limits the resolution of the display; with most game consoles connected to low-resolution TV sets, however, this limited resolution is not a severe handicap.

Software development for GameCube uses a mix of tools from Nintendo as well as third-party tools, including CodeWarrior from Metrowerks, and middleware such as Numerical Design Labs' NetImmerse and Criterion Software's RenderWare. These third-party tools simplify porting titles among the various gaming platforms. Versions of the CodeWarrior tools are available for the PC and PS2; both NetImmerse and RenderWare also support PC, PS2, and Xbox development.

Nintendo is exclusively focused on gaming; the company chose not to make GameCube capable of playing DVD movies, for example, believing the portability allowed by small physical size is more important to gaming than the ability to play DVDs. GameCube is less than half the size and weight of Xbox—with PS2 in between—and has a built-in carrying handle the others lack. The downside to GameCube's small size is the fact that it's too small to accept a DVD movie disc. Although the machine contains all the electronic hardware needed to play DVD movies, it is physically unable to do so.

Sony Settles In as Number One

PlayStation 2 is only a little more than a year old, but it has already sold more than 23 million units worldwide, according to Sony—more than 10 times the sales volume of either Xbox or GameCube. Game sales are also running at a brisk clip, with each console buyer picking up four to five games on average. During the 2001 Christmas holidays, PS2 games

dramatically outsold those for Xbox and GameCube, owing to Sony's larger installed base.

Even the PS2 console itself outsold the new arrivals in 4Q01. Although Xbox and GameCube arrived midquarter, initial sales represented significant pent-up demand that presumably more than compensated for the smaller sales window. The PS2's volume is all the more impressive, considering that the year-old machine still sells in the United States for the same price it fetched at its debut. Sony considered, and ultimately rejected, a U.S. price cut before the holidays, knowing it would still sell, at the full price, all the systems it could make; Japanese buyers did get a 15% discount, to about \$220.

PlayStation 2 sits somewhere between the other two consoles in overall hardware sophistication. PS2's Emotion Engine offers more raw computing horsepower than either competing CPU, but reports from game developers suggest this potential is difficult to realize in real games. The complexity of the Emotion Engine's dual-vector engines, with their asymmetric connections—one paired with the processor core, the other attached to the graphics interface—does not readily lend itself to easy software development.

PS2's Graphics Synthesizer doesn't match the display quality of Xbox or GameCube, but it still leads all contenders in at least one metric—bandwidth to its integrated-DRAM frame buffer. The chip's multiported DRAM array has a 2,560-bit bus running at 150MHz for 48GB/s of peak throughput, some 7.5 times faster than the interface to Xbox's external DDR SDRAM array. These numbers make for impressive specifications, but in the low-resolution world of television monitors, the Graphics Synthesizer's bandwidth goes mostly unused.

The GS chip can render up to 1.25 billion pixels per second, about the same rate at which Nvidia's NV25 core generates pixels in Xbox. Even an HDTV set, however, can accept only about 62 million pixels per second.

However difficult software development may be for PlayStation 2, the market does not lack PS2 titles. Popular gaming Web site www.gamespot.com lists, for the U.S. market alone, 449 PS2 titles, some of which are still in development. This figure compares to a few dozen titles currently shipping for Xbox and GameCube. PS2's advantage in title availability will keep it the system of choice for most customers for some time to come.

Sony is likely to drop the price of the PS2 console at some point this year, which will help maintain system sales and ultimately lead to more game sales. Sony says it is producing more than 1.5 million PS2 systems per month. It will be quite some time before Xbox or GameCube can match this sales rate, and potentially years before either can achieve a larger installed base.

PCs Still Outsell Game Consoles

Because they are useful for so many other purposes, PCs outsell game consoles by about 12:1—and at much higher

system prices. Total revenue from PC software sales similarly outstrips that from console games. We see no signs that this status quo will be reversed anytime soon.

Nevertheless, PC games do not generate the kind of revenue that console games do. There are tens of thousands of PC games on the market. Indeed, shovelware distributors offer CD-ROMs that each contain more than 500 (old) PC games. No single game on the PC, however, can match the popularity of the best console games. A hot PC game might sell a few hundred thousand copies, whereas some console games sell millions.

PCs provide a very different environment for game play than do consoles, and many of the differences favor the PC. Most PCs are desktop or laptop systems designed to be used by one person at short range. PCs run general-purpose operating systems and are equipped with general-purpose hardware. Top-of-the-line PCs generally have faster CPUs and more capable graphics subsystems than any game console has.

Today's best PC processors deliver about three times the performance of the fastest game-console CPUs. The PC's marginal advantage in graphics is particularly slim right now, since Nvidia offers comparable cores in its Xbox and PC 3D accelerators. The company's PC-oriented GeForce3 is slightly faster than Xbox's NV25 core, but only because it has its own dedicated DDR SGRAM memory array running faster than the DDR SDRAM used for both graphics and processor operations in Xbox. The greater memory bandwidth available to GeForce3 gives it the ability to support higher-resolution displays and better rendering quality, principally through superior antialiasing.

On the flip side of the equation, PC games can't be written to run exclusively on top-of-the-line systems. PC games, instead, are written to run on some large fraction of the installed base of systems. PC games can't try to use all the available performance of the CPU, memory, or hard disk, because they must ensure adequate playability, even when the system is running background tasks: soft-modem codecs for Internet connections, file sharing, and so on.

Xbox Wins

After considering the technology underlying these platforms, as well as performing considerable hands-on testing, we have decided to give the *Microprocessor Report Analysts' Choice Award for Best Gaming Chip Set of 2001* to the Xbox team of Intel, Microsoft, and Nvidia. The Xbox hardware offers performance very close to that of the best PCs, and its software environment offers easy game development and reliable game play.

Xbox may lag significantly behind Sony's PlayStation 2 in overall sales and title availability, but these factors do not count against the chip set. Any game that can be run on a PC can be adapted to run on Xbox, and we expect that, in the coming years, most PC games will be offered on both platforms.

Notable 3D Events of 2001

ARM licenses two PowerVR graphics cores from **Imagination Technologies** for high-performance and low-power embedded applications (*MPR 2/20/01-01*).

ATI rolls out Mobility Radeon, a low-power version of the Radeon architecture for laptop computers. Two versions include 8M and 16M of SGRAM integrated in the chip package to reduce physical size and power consumption (*MPR 3/12/01-02*).

Sony, IBM, and Toshiba announce a joint-development agreement for “Cell,” an advanced multiprocessor architecture that could be used in future Sony PlayStation video-game consoles (*MPR 3/19/01-02*).

Separately, Toshiba spins off the group that developed PS2’s Emotion Engine to create **ArTile Microsystems**. (*MPR 4/23/01-01*). In September, ArTile announces the TMPR7901XB microprocessor, its first system-on-chip product.

VIA ships the ProSavage KN133, an integrated-graphics chip set for AMD’s Athlon and Duron processors (*MPR 6/11/01-05*).

Nintendo reveals the final configuration and release schedule for GameCube (*MPR 7/16/01-03*), although actual release is delayed by two weeks.

Microsoft reveals details of the DirectX version 8.1 multimedia application programming interface included with Windows XP (*MPR 8/6/01-02*). DX8.1 enables 3D features supported by ATI’s Radeon 8500 graphics chip, announced in August at Siggraph (*MPR 9/24/01-01*). Also at Siggraph, **Nvidia** announces the Personal Cinema (a video input/output solution meant to compete with ATI’s All-In-Wonder series) and the Quadro2 Go mobile-workstation 3D chip.

Transmeta announces the TM6000 integrated processor at Microprocessor Forum 2001 (*MPR 10/15/01-01*). The chip, intended primarily for embedded systems, includes a 2D-only graphics core.

National rolls out the Geode GX2 integrated processor, aimed at the same market as Transmeta’s chip (*MPR 11/5/01-02*). Although the GX2 adds 3DNow to the original Geode design, they share the same old 2D-only graphics core.

An Xbox game is inherently more stable than a PC game. The Xbox platform may be extended, and improved models are sure to arrive eventually, but its base hardware functionality will never change, whereas PC gamers risk losing the ability to play their favorite games every time they upgrade some hardware or software element of their system.

Xbox’s standard Ethernet port provides a more convenient connection for multiplayer gaming than the proprietary interfaces on the other consoles. The advantage of the standard port is perhaps even more significant than Microsoft expected: enthusiasts have already figured out how to enable multiplayer gaming over the Internet on one title—Microsoft’s Halo combat game—meant to support LAN connections only.

Finally, Microsoft’s decision to give every Xbox a hard disk shows that the company understands how to deliver value to customers, even at some increased cost to itself. The Xbox console must be the best value in computing devices

on the market today—\$300, for what amounts to a complete 733MHz desktop PC (sans display and keyboard), is a great deal. Devoting all that value to gaming was a gutsy call, but it is likely to pay off for Microsoft in the long run.

Other analysts have criticized Microsoft for pricing the Xbox console below its manufacturing cost. The general complaint is that Microsoft will “lose” more than \$100 per sale when hardware and marketing costs are taken into account. These criticisms show a profound lack of business sense. When a company spends money today in expectation of generating a larger income stream in the future, we do not call it “losing money”; we call it “making an investment.”

It may take a year or two for Microsoft’s investment in the Xbox platform to pay off, but with \$36 billion in the bank, Microsoft can afford to take the long view. In the meantime, the rest of us get to enjoy the best gaming experience on the market. This sounds like a good deal to us. ♦

SERVER BATTLES HEAT UP IN 2001

Itanium Faces Off With Power4 and US III

By Kevin Krewell {1/28/02-01}

The year 2001 was an interesting one for servers. Intel released the long-awaited Itanium processor; Sun delivered improved hardware and significantly improved benchmark scores for the UltraSPARC III; and, with the release of the Regatta server with the Power4 processor, IBM

turned our award winner for 1999 Technology of the Year into real hardware. Server wannabes AMD and Transmeta also made some news in 2001, with each offering unique value. The year 2001 was also the one in which the industry's first 64-bit processor and the first 64-bit processor to reach 1GHz, Compaq's Alpha, received its death sentence from Compaq's management. The HP PA-RISC family lingered on as HP showed great ingenuity in keeping the processor competitive without significantly changing the core microarchitecture.

The Upstarts

The newest player on the block was Transmeta. Although the primary market for the Crusoe processor is lightweight, low-power mobile computers, the low-power nature of Crusoe also allows system designers to put significantly more processors into the fixed-size and -power envelope of a server rack. RLX Technologies was Transmeta's highest-profile design win, but this new form factor was released just as the dot-com bubble burst, and many potential customers for the product went out of business or downsized. The Crusoe processor has some significant limitations as a server processor: specifically, it doesn't support multiprocessing, it doesn't have ECC protection for main memory, and it supports only a 32-bit, 33MHz PCI bus. The Crusoe processor's performance also has been controversial. The processor allocates some of its main memory for code-morphing operations, and system performance has been shown to vary significantly, depending on application and code-caching history.

AMD had been talking about entering the server market for quite some time, but not until 2001 did it introduce its first multiprocessor solution. The AMD 760MP chip set supports up to two Athlon MP processors and DDR SDRAM memory. AMD recently upgraded the chip set with a new south bridge that now allows the north bridge to support 64-bit, 66MHz PCI, giving the new AMD 760MPX sufficient



bandwidth to support Gigabit Ethernet (see *MPR 12/26/01-01*, "AMD Maps Servers to 2003").

We nominated the Athlon MP in this category because it has proved to be an excellent, scalable, and well-balanced processor, with very good floating-point performance (for an x86 processor). Despite Athlon MP's very good price and performance, it has been difficult for AMD to attract a top server OEM to the processor. Part of the company's problem is that it lacks a multiprocessing solution

beyond two processors. Clustered computing is making good progress in the server market, but major OEMs also need a 64-bit, scalable, fault-tolerant multiprocessing solution for enterprise applications. At Microprocessor Forum 2001, AMD revealed details of its 64-bit, scalable, fault-tolerant solution—the SledgeHammer processor (see *MPR 11/26/01-02*, "AMD Takes Hammer to Itanium"). Unfortunately, SledgeHammer will not ship until 1H03, leaving AMD in the so-called white-box server market until then. When Hammer ships, AMD will still have an uphill battle against Intel's Itanium and Xeon processors, but it will offer something Intel cannot—one architecture for both 32- and 64-bit computing.

Sun Stumbles, Recovers

Last year's winner, Sun's UltraSPARC III, had some difficulty delivering on its frequency promises. The fastest version, at 900MHz, experienced manufacturing difficulties and was delayed, eventually shipping after a semiconductor process shrink. That process improvement also recently gave us a new 1,050MHz US III and, along with it, some highly improved benchmarks. The nominee for most improved compiler is the Forte 7 compiler for delivering incredible improvements in the US III SPEC scores: with the new compiler, the 1,050MHz US III delivers scores of SPECint2000(base) 537 and SPECfp2000(base) 701 (see *MPR 1/14/02-01*, "Gigahertz UltraSPARC III SPEC Surprise"). The new scores are competitive but not good enough to put the US III in the lead on either SPECint or SPECfp benchmarks.

Itanium Rolls, Prepares the Way for McKinley

One of the big stories of 2001 was Intel's Itanium processor. While the SPECint2000(base) score of 358 was disappointing, the SPECfp2000(base) score of 703 was the leading score in 1H01. Because of the strong SPECfp score and its impact on the market, we nominated the Itanium as Best Workstation/Server Processor of 2001.

Itanium was quickly surpassed as the traditional 64-bit RISC competitors delivered new products in 2H01. It also became apparent that the initial Merced-based Itanium was mostly useful as a development vehicle and that the "real deal" was the second-generation EPIC design, the McKinley processor (see *MPR 10/01/01-01*, "Intel's McKinley Comes into View"). McKinley was revealed to have additional computational resources, a revised pipeline, 3MB of on-chip L3 cache, a new socket with more bandwidth, and a future higher clock frequency when produced in the same 0.18-micron process as Merced. But while other high-end server processor designs are moving to glueless multiprocessing, simultaneous multithreading, chip-level multiprocessing, and integrated memory controllers, Itanium system architecture is beginning to show its age. Perhaps the design has been in development too long and has had too many cooks. The shared-bus system design may be cost-efficient, but it does not offer the dedicated bandwidth of competitors' solutions.

Intel's Xeon Gets Hyper

Intel's 32-bit solutions continued to dominate server volumes with a variety of cache configurations, multiprocessing capability, and a couple of microarchitectures. The most impressive news for Intel's 32-bit server processors in 2001 was HyperThreading technology (see *MPR 09/17/01-01*, "Intel Embraces Multithreading"), an Intel-branded version of simultaneous multithreading (SMT). HyperThreading will formally appear in the FosterMP processor, which began sampling in 2001. HyperThreading is embedded in the Pentium 4 microarchitecture but is not currently enabled.

Public information today indicates that HyperThreading seems to be a good first attempt at adding SMT to a processor, using a minimum of die overhead. The FosterMP processor is based on the Pentium 4 microarchitecture, and Intel has validated the concepts on the Willamette processor. HyperThreading will be able to extract more efficiency out of the processor and therefore deliver more processing performance on multithreaded software. The FosterMP processor should offer clock speeds exceeding 1.5GHz and excellent processor front-side bus bandwidth. The 2GHz Xeon processor also produces excellent SPEC scores, surpassing the 800MHz Itanium on SPECfp. We nominated the Intel's Xeon processors with HyperThreading technology because it will be the first commercially available server processor with SMT and will continue the x86 legacy of excellent price/performance and acceptable scalability.

To address the nascent blade-server market, Intel took its 0.13-micron mobile Pentium III processors and adapted

them (sans SpeedStep) for blade-server designs. The lower power requirements and power dissipation of the low-voltage processors make them a good fit in blade designs. The new 0.13-micron Pentium III processor, code-named Tualatin, also offers a 512KB L2 cache and clock speeds up to 1.4GHz.

Itanium Fallout: RISC Pioneers Cut Short

The year 2001 was a bittersweet one for Compaq's Alpha processor. It was the year that Alpha became the first 64-bit RISC processor to ship at 1GHz (see *MPR 8/13/01-03*, "Alpha Quietly Reaches 1GHz"). The next-generation core, the EV7, with glueless scalability and on-chip Rambus memory interface, delivered first silicon—and it worked. But 2001 was also the year Compaq announced the EV7 was the final generation of the Alpha processor, and the EV8 program had been canceled (see *MPR 7/02/01-02*, "Itanium Consumes Alpha"). On top of that, Compaq announced it was transferring the technology and designers to Intel and committing to Itanium for the future. Intel gained access to Compaq's highly regarded compiler technology and developments on scalable multiprocessing and SMT. As the first 64-bit RISC processor to reach the 1GHz mark and for its excellent performance, we nominated the 1GHz Alpha Processor.

Compaq's potential merger partner and Itanium code-developer HP continued to milk the PA-RISC core with a die shrink and more cache. The PA-8700 shipped at 750MHz and 2.25Mb of L1 cache. Although HP does not have plans for a new PA-RISC core, it continues to enhance the existing core with more L1 cache, higher frequencies, and fine-tuning. At Microprocessor Forum 2001, HP also announced that the next-generation PA-RISC processor, called Mako, will place two 1GHz PA-8700 cores, along with 1.5MB of L1 cache, on one chip. Mako will also have a custom 32MB off-chip L2 cache. Mako will be a transition processor to Itanium, because it will use the same 100MHz double-pumped 128-bit processor bus as McKinley.

The Power to Be the Best

Considering all the server technology introduced in 2001—and considering also system scalability, bandwidth, chip-level multiprocessing, fault-tolerance, and performance—it is impossible to ignore the accomplishments of the IBM Power4 architecture. The first release of the architecture is in the pSeries 690 server, which can be equipped from 8- to 32-way processor configurations. It offers two 1.3GHz Power4 processors and 1.5MB of L2 cache on one die. IBM can place four such die in a single module for the base 8-way configuration, and four modules can be connected to form a 32-way multiprocessing system.

One 1.3GHz Power4 processor, with 1.5MB of L2 cache and 128MB of external L3 cache, delivers a SPECint2000 (base) score of 790 and a SPECfp2000(base) score of 1,098. These scores lead all other server processors by a significant margin. Some vendors have complained that the score is unfair: while only one processor was enabled to run the

benchmarks, it had access to the cache and bandwidth resources of two processors. We believe, however, that this arrangement was still within the SPEC rules and therefore acceptable; the results also show the effect of the tremendous system bandwidth available with the Power4.

Because it has produced industry-leading benchmark scores and industry-leading clock frequencies (for 64-bit

processors); actually met shipment schedules publicized two years ago; and generally lived up to the promise shown when we gave the Power4 the Best New Technology award two years ago (see *MPR 2/07/00-01*, "Best New Technology: POWER4"), we give IBM's Power4 the **Microprocessor Report Analysts' Choice Award for Best Workstation/Server Processor** of 2001. ♦

Is Your Company Interested in Obtaining Reprints?

Have you seen an article in MPR that you would like to distribute to colleagues and prospective customers?

Cahners In-Stat/MDR can provide paper or electronic article segments or entire articles at a reasonable price. These reprints can be used for sales presentations, trade shows, and corporate libraries. PDFs may also be purchased for use on your company's intranet or for access on your public Internet site. We can include company logos, part numbers, and special formatting to fit your specific project needs.

Please contact Erin McKeighan at emckeighan@instat.com or 480.609.4551 for more information.



DCT MARCHES INTO JAVA PROCESSORS

Lightfoot and Bigfoot Processors Offer New Twist to Java Execution

By Markus Levy {1/28/02-04}

The year 2002 is the year for embedded Java. It has passed the stage of marketing hype and is settling into a wider variety of embedded applications than ever before. Java will allow wireless providers and manufacturers to dynamically deliver applications and services. Java also opens

the world of smartcards, and major financial institutions (VISA and MasterCard, for example) agree. Like many others, these vendors further demonstrate the need to break away from the desktop environment in a secure and portable way with e-commerce, on-the-fly banking, and remote networking.

There are several ways to process the Java bytecodes associated with these applications, but at the highest level, these can be categorized as software based or hardware based. According to the In-Stat/MDR report *Java Hits the Road: Accelerators in Mobile Applications* (#DE0102MF), in the current generation of mobile phones and other portable applications, 97% of Java support is derived from pure software-based solutions (with only 3% attributed to hardware-based Java accelerators). This scenario is changing, however, as Java applications become more performance hungry, and system designers look for ways to execute these applications more efficiently.

DCT Ltd. is one of the newest processor vendors to enter the embedded Java market with a hardware-based solution. To date, DCT is offering two product lines: Lightfoot and Bigfoot. Lightfoot is a home-grown architecture that combines basic RISC features with an innovative approach to Java execution. Bigfoot, on the other hand, is an ARC Cores processor with modifications to turn this configurable RISC processor into an efficient Java engine. The company initially plans to target the hardware security market for the e-commerce space by focusing on security products (e.g., smartcards and smart-card terminals, network security).

Lightfoot's Architectural Features

With Lightfoot, DCT has devised an architecture that has explicit support for Java. However, Lightfoot's architecture will also be extremely beneficial for embedded applications that, in general, require high-level language support. The most distinctive feature of Lightfoot is its instruction format, which provides a soft bytecode layer to give a system a particular application-specific personality. Lightfoot's soft bytecode feature is elegant in its simplicity, yet it can handle the most complex bytecodes in an efficient and timely manner.

In some respects, the Lightfoot architecture resembles other processor architectures. It implements a modified Harvard architecture; however, it has an 8-bit instruction width, a 32-bit-wide internal architecture, and a 32-bit-wide data memory. While the Harvard architecture feature is also implemented in most modern processors, Lightfoot's approach is rare in that the instruction width is four times shorter than the data width. In fact, it's likely that the Lightfoot is the only Harvard architecture that uses an 8-bit instruction width and a 32-bit datapath width. The benefit represents the potential for a significant code reduction, the extent of which will depend on the application.

Although the general instruction format is 8 bits wide, some instructions can be followed by a single 8-bit immediate operand. If an instruction (including a variety of loads, stores, branches, and constant pool accesses) that uses an immediate operand is prefixed by the WIDE opcode, the immediate operand is taken to be 16 bits wide. The resulting 16-bit value is interpreted as a signed or unsigned value, depending on the particular instruction. Lightfoot instructions (assuming that WIDE is a part of the following instruction) can thus be 8, 16, 32, or even 40 bits wide (for example, using the combination of the unsigned prefix, the WIDE prefix, the *cnsti* instruction, and the 16 bits of immediate data).

Another familiar feature of Lightfoot is its load/store organization. This feature implies no register-memory operations and eliminates use of complex addressing modes, as seen in many 8- and 16-bit microcontrollers. The load/store architecture helps simplify the instruction decode portion of the design and minimizes the number of operation codes required to support a variety of addressing modes. Lack of a flexible addressing system shouldn't restrict Lightfoot's capabilities compared with those of the traditional microcontroller; as a matter of fact, the *ldw* and *stw* instructions can be used to directly access I/O devices (or memory-mapped peripherals). Furthermore, DCT selected the addressing modes of Lightfoot for efficient implementation of high-level languages.

All data accesses must be aligned in their native data-type boundaries to minimize the complexity of the memory interface circuitry. For example, words (32 bits) must be word aligned, and halfwords (16 bits) must be halfword aligned. The memory interface detects illegal accesses and signals a bus error trap, but this can easily be avoided by using proper programming techniques. For many applications, unaligned accesses will not be an issue, but certain applications are notorious for irregular data arrays (networking packet headers, Ethernet frames, etc.). This could result in data bloat in an effort to keep data properly aligned. On the other hand, this is not entirely a Lightfoot issue; it is a common characteristic of a good many microcontrollers and microprocessors.

The capability to support unaligned accesses may become more popular in processors that target networking applications. An example is the new ARMv6, which includes support for the architecture to handle 32-bit accesses within data structures that are not aligned on 32-bit boundaries (see *MPR 11/26/01-03*, “ARM Drives V6 to Microprocessor Forum”). ARMv6 expands on the functionality of its basic load and store instructions, making this new feature transparent to the user. However, the operation still requires two bus transactions.

In many embedded applications, a processor’s real-time performance and deterministic behavior is very important. In support of this, Lightfoot’s maximum interrupt latency is equal to the interrupt latency of the longest machine instruction. Although the average instruction length is only two cycles, the longest uninterruptible instruction in the Lightfoot ISA is the *PARS* instruction. This instruction dumps stack elements into data memory (pointed to by the IX register). With zero-wait-state memory, each element to be dumped consumes a single clock cycle. The *PARS* instruction is provided to make function and method prologues more efficient. (Methods and C functions rarely have more than five parameters.)

Another factor for meeting the demands of a real-time system is the amount of time associated with performing a thread switch. The number of registers that must be saved represents a significant portion of the thread-switch time. For Lightfoot, 24 registers (8 data stack, 4 return stack, and 12 special) must be saved on a context switch.

Lightfoot, like the ARM7, implements a simple three-stage pipeline: fetch, decode, and execute. Although the pipeline is short, DCT claims the processor will be able to reach speeds of 100MHz. (ARM7 implementations might reach 75MHz.) The short pipeline will impose a three-cycle latency for branch operations, a small price to pay for the tradeoff of avoiding the overhead of branch-prediction hardware support.

Programmable, Code-Saving Instruction Set

The most unusual feature of the Lightfoot architecture is its use of three different instruction formats, called IF0, IF1, and

Format	Bit Fields							
IF0: Soft Bytecodes	1	x	x	x	x	x	x	x
IF1: Nonreturnable	0	1	n	n	n	n	n	n
IF2: Returnable	0	0	r	r	r	r	r	r

Table 1. The instruction format for the Lightfoot architecture consists of three instruction categories.

IF2 (see Table 1). Furthermore, the *EXT* (extension) operation code is reserved for extending the instruction length beyond eight bits. Rather than use this operation code to put the processor into an extended mode (à la ARM’s Thumb), *EXT* is a prefix for each special instruction. It could be applicable for a division instruction, for a multiply-accumulate instruction, or for multiprecision arithmetic, for example. Avoiding the extended-mode paradigm avoids the extra hardware complexity required to support it and simplifies DCT’s development-tool strategy.

While the 8-bit instruction width limits the instruction space to 256 encodings, 128 entries are available in that space for customizable instructions that DCT calls soft bytecodes. These instructions are represented by the IF0 instruction format. The 64 IF1 instructions fall into a category of non-returnable instructions. IF2 is used by the 32 single-byte instructions, which can be folded in with a return operation.

The soft bytecode software layer gives a Lightfoot-based system a particular application-specific personality. This approach allows Lightfoot to support Java Virtual Machine (JVM) variants (such as JavaCard), and the provision for JVMs is optimized according to different performance criteria (such as maximum execution speed or minimum memory usage).

The soft bytecode software also allows Lightfoot to support other high-level languages, such as C/C++, by allowing definition of language-specific soft bytecodes. The soft bytecodes consist of a sequence of the IF0, IF1, and IF2 instructions. Ultimately, using fundamental instructions to make up the more-complex soft bytecode helps reduce the complexity of the architecture that would normally be required; in turn, this will allow the processor to run at higher clock speeds.

Soft bytecode invocation, decoded within the fetch unit, consumes one cycle. During this instruction cycle, the processor pushes the program counter onto the return stack, calculates the address of the soft bytecode, and loads the new program counter value. Unlike other branch operations, IF0 suffers only a one-cycle performance penalty that is translated into a NOP in the processor’s pipeline. When the fetch unit encounters an IF0 instruction, it notifies the processor to branch to one of 128 locations in low program memory, where the implementation of the soft bytecode resides. The soft bytecode numbers (0–127) are mapped to a program memory address by shifting the bytecode number left by three bits (in effect multiplying the IF0 instruction operation code by eight). This scheme allocates eight bytes of program memory for each soft bytecode implementation. If more than eight bytes of program memory are required to implement a soft

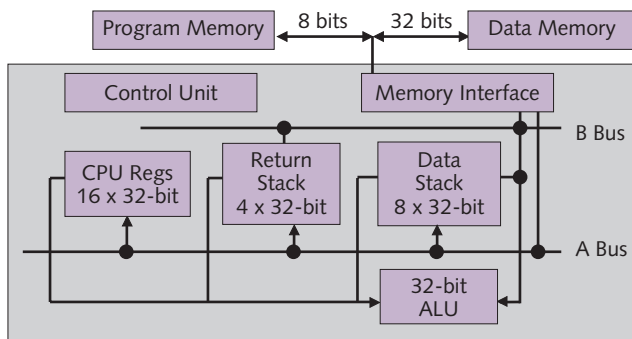


Figure 1. Block diagram of Lightfoot shows a simple, yet effective, architecture.

bytecode, a standard subroutine can be used, and the processor will take a three-cycle hit. The alternative would be to use 16 bytes per soft bytecode slot, but, in general, this would decrease memory efficiency.

The user-programmable soft bytecodes essentially implement a subroutine call by means of a jump table that is, in effect, a hardware implementation of a bytecoded virtual machine's dispatch code. This capability is useful for efficiently implementing dynamic method dispatching in an object-oriented programming language.

These subroutines are basically an efficient means of inlining short code sequences, with the sole benefit of minimizing the code size of an application. (It's not necessarily a performance benefit.) As an example, Lightfoot's JVM implements the complex instruction *invokevirtual* as a soft bytecode.

Another useful feature of Lightfoot is associated with its *decrement and branch on non-zero* (DBNZ) instruction. DBNZ is used to perform single-cycle looping (compared with three cycles for other branch instructions)—on processors, a useful feature for executing tight code loops without incurring the overhead of the loop-variable maintenance and testing. With DBNZ, the processor has dedicated hardware (i.e., the operation does not go through the CPU datapath) to decrement the counter register (CTR) and branches if the result is not zero. The branch address is taken from the top of the return stack. If the branch is not taken (the CTR register after decrementing is 0), the branch address value is popped from the return stack.

Foldable Subroutine Returns

Most programmers are familiar with the return operations associated with high-level languages. These are typically inefficient and require many clock cycles to restore the processor's state. This operation, if applied to Lightfoot's soft bytecode mechanism, would result in a serious performance penalty, because the return overhead would represent a minimum of 12% of the soft bytecode (assuming a bytecode length of eight). DCT has devised a zero-overhead return operation for IF0 instructions.

When an IF0 instruction is executed, the address of the following instruction is automatically pushed onto the on-chip return stack. When returning from the soft bytecode execution (or any other subroutine call), an IF2 instruction can be folded with the return operation (if the "R" bit is set within the IF2 instruction), and the program counter value is loaded with the value popped from the return stack. This effectively delivers a zero-overhead return operation. (It also prevents the soft bytecode subroutine from using one of its eight precious instruction slots for a separate return instruction.)

Lightfoot's Functional Units

Lightfoot's main functional units consist of the control unit, the ALU, the data stack, the return stack, and the register bank (Figure 1). At first glance, these are features common to many other architectures, but, digging into the details, one realizes that Lightfoot incorporates some unique features that will benefit high-level language programmers.

The control unit is for fetching, decoding, and sequencing execution of instructions in the processor. It also contains modules for implementing run-time checks and handling traps that are used by the JVM.

The ALU features a 32-bit barrel shifter. Shifts of 1, 2, 4, and 8 are accomplished in a single cycle; other shift amounts require combinations of those. For example, to shift by 17 bits takes three cycles: two 8-bit shifts plus a 1-bit shift. The ALU also contains a 2-bit-per-clock step multiplier unit (allowing a 32×32 -bit multiply to execute in 16 cycles, for example). The multiplier performs 8×8 -bit, 16×16 -bit, and 32×32 -bit multiplies. DCT implemented Lightfoot's multiplier in this manner to help reduce the complexity of the design and gate count. However, this slow multiplier will impose limiting factors for basic communication applications that process even the simplest type of DSP filtering algorithm.

The data stack is used to hold program variables—not to implement the stack frame, for which special support is provided. The data stack, a significant component of any Java-based architecture, consists of a bank of eight 32-bit registers plus an extension pointer (EP) register. An eight-deep stack minimizes the amount of processor state that must be saved when switching threads. The top three elements of the data stack are connected to the inputs of the ALU, and there are instructions that specifically manipulate these elements.

A fill/spill circuit handles the data stack overflows and underflows. When a spill occurs, the EP register points to the data memory location for the write operation. It takes two cycles to perform a spill, on the basis of using zero-wait-state memory. Because Lightfoot, unlike other Java-processing solutions, implements separate operand and parameter stacks, the data stack will spill infrequently. Furthermore, function and method parameters are pushed onto the data stack before they are called, but the parameters are moved to the stack frame when the function or method starts executing. This will prevent function calls from increasing the data stack. (Note:

Method parameters, part of a stack frame, are accessed using the parameter pool pointer and parameter registers.)

The data stack, discussed above, is enhanced by the data-stack-limit register, which constrains the data stack extension to a user-defined size. The processor executes a trap subroutine whenever this size is exceeded or whenever a stack underflow occurs. Similar mechanisms (using a return-stack-limit register) are used for the return stack.

The return stack, as its name implies, holds return addresses for subroutines. It is similar to the data stack in that it consists of four 32-bit registers, a return extension pointer (REP), and a fill/spill circuit. An attempt to pop an address from an empty stack causes the processor to generate a “return stack underflow” trap to prevent the processor from entering an illegal state. Within the return stack, the top-of-stack element is used as an index register to access program memory. The return stack can also be used as an auxiliary stack for programs (to hold temporary values).

Lightfoot has a 256-word register space; the 16 CPU registers reside at the bottom, and the remaining registers are for peripherals. This register bank contains four parameter cache registers, which hold the first four method parameters. Any of these registers can be read or written by the processor in a single cycle.

Java-Related Features

While many of Lightfoot’s features will inherently support both embedded microcontroller and Java-specific functions, some are explicitly geared to benefit Java applications (and other high-level languages). At a minimum, Lightfoot’s primitive operations (arithmetic, branch and conditional, variants of load/store, etc.) can be translated into JVM bytecodes. The class loader translates between JVM bytecode encodings and Lightfoot VM bytecode encodings. (Actually, this activity includes translation to the IF1 and IF2 formats, and even to the IF0 format for the more-complex JVM instructions, such as *invokevirtual*, and multiprecision integer and floating point.)

Although Lightfoot lacks the myriad addressing modes available on many microcontrollers, it has all the addressing modes needed to implement high-level languages. On the other hand, despite the elaborate addressing modes found on legacy microcontrollers, compilers are typically unable to take advantage of them. (That is, the addressing modes can be used only with assembly language programming, which programmers are trying to escape.)

Unlike many other Java processing solutions, Lightfoot has dedicated hardware-based instructions for managing stack frames, method invocation, return protocols, and constant pool handling. For example, an array-bounds cache (ABC) unit is one feature of Lightfoot designed to support the JVM. Specifically, it is used to implement the mandatory array-bounds checking operation.

ABC consists of two register pairs, each of which contains *base address* and *vector length* fields. When ABC operation is

enabled, execution of an indexed instruction, such as the JVM *baload* (or Lightfoot’s *LXB*), causes the ABC to be scanned for the presence of the base address of the data array. (Essentially, this action compares the second stack element to the value in the register.) If the base address is in one of the registers, the vector-length field is compared with the index. The processor removes the base and index from the stack and replaces them with the sign-extended value stored in the array.

This is where the Java-related part comes into play: If the base address is not found, then the array-bounds-miss trap automatically executes. This trap fills the least-recently-used register pair with the size and base address of the vector and restarts the indexed instruction. If the index is greater than the length or equal to it, an array-bounds trap is generated, and the Java security features are used.

Without the array-bounds detection feature, every array access would require the program to perform an explicit, and time-consuming, inspection. This feature provides a general-purpose benefit in a processor without an MMU. Furthermore, an MMU will have little benefit as an array-bounds detector. First, a program may have too many arrays, making MMU usage impractical. Additionally, an MMU will not provide the appropriate level of granularity to effectively monitor array bounds.

Lightfoot has support for thread-stack overflow detection. This feature, not explicitly implemented for Java, allows each thread to run in its own hardware-protected area of memory. Absent an MMU, it provides protection against the overwriting of system memory by “runaway” threads.

Because Java programs heavily use the first few local variables, Lightfoot provides four 32-bit parameter registers (P0 to P3) and eight associated instructions to make program execution more efficient. The instructions move data between the data stack and the parameter registers, a general feature of stack architectures. The method-invoking protocol of Java requires the caller to deposit a method’s parameters on the data stack before calling the method. When the method is invoked, the parameters are popped off the stack and stored in the stack frame. The combination of the *SSS* (Setup Stack Frame), *PARS* (Parameter Store), and *REGS* (Register Store) instructions make this process more efficient.

Lightfoot has no hardware support for garbage collection; this function is supported by the JVM in software. The system developer can use any mixture of the following strategies in DCT’s JVM: no collection; mark and sweep; and incremental, using a low-priority thread.

Noah Let Bigfoot on the ARC

Bigfoot is a combination RISC and Java processor built upon the ARC Cores base architecture. DCT has used the extensibility of the ARC architecture to emulate a stack-based machine. Unlike its coprocessor competitors (e.g., Jazelle and JStar) which “force” a Java implementation on a RISC processor, DCT was able to bend the ARC processor with relatively simple logic modifications and additions.

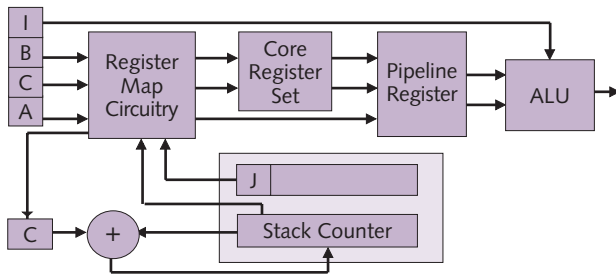


Figure 2. DCT devised simple circuitry to allow a RISC architecture to emulate a stack-based architecture.

With the design of Bigfoot, DCT took a fundamentally different approach than it used for the Lightfoot architecture. Lightfoot is a ground-up design, whereas Bigfoot is a combination RISC and Java processor built upon the ARC Cores base architecture (called ARCTangent-A4 but referred to throughout this analysis as ARC). This is a perfect model for ARC Cores, which is designed to have its base instruction-set architecture extended by any customer. In more traditional applications, ARC Cores customers would add custom instructions to accelerate applications such as MPEG2, network routing, or a variety of telecommunications algorithms. DCT has taken a different, and simpler, approach, adding hardware support to convert the processor's register bank into a stack.

Getting a RISC architecture to behave like a stack-based architecture is a challenge, but this is an essential ingredient for efficiently running the Java environment. Typically, there is an architectural mismatch between a stack-based execution model and a RISC (register-based) architecture. In a register-based processor, the high-level language compiler uses its register allocator to assign variables and partially evaluated operands to particular registers. The number of registers is finite and the register allocator must use sophisticated algorithms, which are costly in terms of time spent and the amount of code occupied by the allocator (especially costly if the compiler is a Java just-in-time [JIT] compiler).

Stack architectures, such as the JVM, do not require register allocation, because the stack is an extensible structure. Therefore, to run a stack-based program on a register machine, either the stack must be emulated in memory, which can be inefficient, or the program must be converted (using register-allocation techniques) to the native register-based code. This action results in code bloat over the original stack-based program and requires a complex translator to be part of the runtime environment if dynamic class loading is required.

Going Into J-Mode

The key to efficient execution of Java on a RISC-like architecture is to provide a means of efficiently mapping the JVM stacks onto the register bank. With modifications to the ARC processor, DCT created a processor that seamlessly switches between a stack-based and a register-based architecture (Figure 2). The first modification is the addition of a J-mode

bit to the processor's program status word (PSW). The J-mode bit enables and disables operation of the register map (RM) circuit, in effect turning the augmented ARC+ mode on or off. (Note: In addition to allowing instructions to be added to the base architecture, ARC's PSW, condition-code flags, and register set can be augmented.)

In the ARC processor, the second stage of its pipeline (operand fetch) uses fields encoded in the instruction word to select the two source operands (B and C) and the destination operand (A). In the unmodified processor, the fields address the core register bank (using a six-bit register address). DCT's modifications involve dynamic remapping of the register fields. The RM mechanism allows DCT to treat the first 16 registers of the ARC core as a "rotating" register file. When the J bit is enabled, registers r0–r63 are partitioned into two groups. The first 16 registers are mapped dynamically into "physical" registers r0–r15 on the basis of the current value of the stack counter (SC) register. The mapping is the sum (modulo 16) of the register number and the value of SC. Registers r16–r63 are mapped directly into the corresponding registers r16–r63 (except for the phantom registers described below).

With the J mode active, it is possible to interleave stack and nonstack instructions without suffering any mode-switch penalty. However, if a branch to C code is required, the program must deactivate the J mode and save any registers used by the C function. In an alternative embodiment, the Bigfoot Java functionality can be implemented without a J-mode bit by using separate register windows for the Java mode and the C mode. Although this implementation adds an extra cost of approximately 5,000 gates, it allows the two modes to operate completely independent of one another.

The SC register is a four-bit register, allocated in the ARC auxiliary register bank along with a four-bit adder circuit and a stack counter control circuit. To convert registers r0–r15 into a dynamic stack, some means of automatically incrementing and decrementing the SC register must be provided. To accomplish this, DCT assigned three phantom register numbers: r0+, r1-, and r1—. These registers are allocated out of the extended core register range r32–r63 of the ARC processor. (The macro facility of the assembler translates these aliases into real register names.) These registers are phantom because they are not mapped into distinct "physical" registers and are used as aliases for other registers. The RM circuit detects the phantom register numbers and substitutes the phantom register number with r0 or r1, depending on the exact phantom register (r0 for r0+ and r1 for r1- and r1—). The RM circuit also generates the control signal for use by the SC controller. (It increments SC by 1 for r0+; decrements SC by 1 for r1-; and decrements SC by 2 for r1—.) When an instruction does not contain a phantom register number, the value of the SC register is not modified.

Unlike DCT's Lightfoot architecture (and other Java processors), Bigfoot lacks dedicated stack fill/spill circuits. Having them would have required major overhauls to the

ARC core. However, each JVM method definition in a class file contains information about the maximum number of elements it uses on the data stack and the number of local variables and parameters. (Bigfoot uses a unified operand/local variable stack. If the combined stack size is less than 16 (the number of registers available inside ARC), these elements can be stored in the register bank. On the other hand, if the combined stack size exceeds 16, the overflow is stored (by the method's prologue code) and maintained in a memory-resident stack frame.

Bigfoot Does the Bytecode Shuffle

To use the ARC's new stack-emulating circuitry, DCT also had to devise two software-based bytecode-translation schemes: TR1 and TR2. The combined use of TR1 and TR2 allows JVM programs to be translated into native code for the augmented RISC processor. TR1 is the baseline translation scheme that provides mapping between each JVM bytecode and a sequence of one or more ARC+ machine instructions. DCT's architectural enhancements allow the translation to use the ARC's native instructions in a register-independent manner. Furthermore, the translation, which is a simple table lookup, can be performed during class loading. When this translation is being performed, it is assumed that register *r0* represents the "next" free element on the stack, *r1* is the current top-of-stack element, and *r2* is the second stack element. The RM circuitry dynamically determines the physical registers these stack element pointers represent.

As an example of the TR1 translation, consider the JVM *iadd* instruction. This instruction takes two parameters from the top of the stack, removes them, and replaces them with their sum. The translation replaces this with the ARC+ instruction:

```
add r2,r1-,r2
```

where the second stack word (*r2*) is replaced by the sum of the top-of-stack word (*r1*) and the second stack word (*r2*). The phantom register *r1-* causes the SC register to be decremented after the operand fetch phase, and the second stack element then becomes the new top of stack.

In another example, consider the JVM *iload* *<n>* instruction, which loads the value of local variable *<n>* onto the stack. Assuming the variable is in the register "window," this instruction becomes *mov r0+,r<n>*. This instruction replaces the next "free" stack slot with the contents of the stack frame register *<n>*. Phantom register *r0+* causes the SC to be incremented after the operand fetch phase, and the "old" *r0* becomes the new top-of-stack register *r1*. The exact value of *<n>* depends on the current depth of the operand stack and is calculated statically during translation. This translation produces a "normal" ARC instruction, with the significant exception that it is essentially context free; this is similar to Java bytecodes, a zero-address instruction format.

Optimizing Bytecode Blocks

The limitation of the TR1 scheme is that it translates only one bytecode at a time. In some cases, this may not yield the most efficient code sequence. Consider the Java code sequence:

```
iload x
iload y
iadd
istore z
```

If the operands and the result of the operation are within the register window (the program already placed them in the register stack by a previous instruction), this instruction sequence can be translated into the following ARC instruction:

```
add rz,rx,ry
```

This action results in a saving of 12 bytes and three clock cycles that would have been required if TR1 alone were used. To derive these optimizations, DCT created translation scheme TR2 by using a simple pattern-matching program, which replaces particular sequences of JVM bytecodes with their ARC+ equivalents (and defaults to TR1 whenever the pattern matching fails).

The sizes of the TR1 and TR2 translation tables depend on their exact implementation. For TR1, assuming an average of 10 bytes per translation, the table would be approximately two kilobytes. The size of the TR2 table depends on the number of patterns (different configurations having a different number of patterns), but it is probably in the same size range as the TR1 table. For Bigfoot, which has an external memory subsystem, this should have a minimal impact on the memory footprint.

Enhancing Conditional Branches

On most RISC processors, the conditional branch instructions use condition code flags set by a previous instruction. (Alternatively, certain architectures, such as the ARC and ARM, can perform predicated execution on most of their instructions and thereby avoid excessive branching.) However, this is in contrast to the behavior of the JVM conditional branch instructions that pop one or two of the top stack elements during execution of the conditional instructions. (In other words, the comparison is performed during the branch instruction itself.) For example, the *ifeq* JVM instruction pops the top stack element, compares it with zero, and branches if the value is zero. Similarly, the *if_icmpeq* instruction pops two elements from the operand stack and branches if their values are equal.

The ARC+ equivalent of *ifeq <markus>* is the following:

```
sub.f r1,r1-,0
br.eq <markus>
```

This translation takes two ARC+ instructions. Because branches are relatively common in JVM programs, this translation will negatively affect both the execution time and the

size of translated programs. DCT's scheme to make branches more efficient on the ARC+ consists of six additional condition codes (using the unallocated condition code patterns on the ARC). These conditions are as follows:

- SZ - stack top zero
- SNZ - stack top non-zero
- SGZ - stack top greater than zero
- SLZ - stack top less than zero
- SGEZ - stack top greater than or equal to zero
- SLEZ - stack top less than or equal to zero

Instructions that write to the top of the stack set these condition-code flags whenever possible. The SCC circuit accepts from the decode circuit an extra control input, which detects a branch instruction code and one of the extended condition codes and causes the SC register to be decremented by one at the end of the cycle.

Proprietary and Commercial Tool Support

DCT's software tools strategy is different for Bigfoot and for Lightfoot. For example, Bigfoot takes advantage of the ARC Cores tools suites. Specifically, ARC's Metaware subsidiary produces a very good compiler and integrated development environment (IDE) technology. Alternatively, Lightfoot support consists of proprietary but standard tools (i.e., ANSI standard C compiler, debugger, macro assembler, linker/librarian, and simulator) as well as some value-added software (i.e., a TCP/IP stack, a C runtime library, and an RTOS). DCT has ported Lightfoot and Bigfoot to a Xilinx FPGA and offers these in the form of evaluation boards.

The Bigfoot core contains debug support in the form of JTAG as part of ARC's general debug strategy. Bigfoot requires no special debug extensions, so the standard ARC debug tools can be used. The first device will have simple debugging capability, using a wire protocol with a daemon process running. This is similar to what can be used for 8051 or rudimentary ARM7 designs.

How the 'Foot's Stack Up

Competitive analysis of DCT requires a three-pronged approach: first, a comparison with other Java processors; second, a comparison with other "C-based" processors; third, a comparison with other processors and/or systems that include Java and C support.

Analyzing the Java competition presents a wide range of challenges (see In-Stat/MDR's report *Spilling the Beans on Java Accelerators*, #DE0103DE). In short, it's the job of any Java processing product to perform the functions of the JVM. One step better than the software-only just-in-time compiler from a performance and memory footprint perspective, is the hardware-based JIT. An example of this comes from Parthus Technologies, with its MachStream coprocessor engine and Java module (see *MPR 3/26/01-04*, "Java to Go: Part 3"), which essentially "forces" the Java bytecodes into the instruction format of the host processor. MachStream will work with any host

processor, including ARM and MIPS. To overcome some of the inefficiencies of stack semantics on a register-based CPU, MachStream uses special accumulation techniques to identify related operations that may be combined and performed with a single instruction. With stack semantics, an operation implicitly pops values from the stack; these values must sometimes be pushed back onto the stack to be operated upon again. DCT's Lightfoot and Bigfoot products both directly solve the stack semantics problem, as they implement a stack-based architecture. Furthermore, the Bigfoot method simplifies the translation from JVM bytecodes to augmented RISC code.

The Java hardware interpreter is essentially an on-the-fly interpretation engine that generates native code from byte codes. Two products are available in this category: namely, ARM's Jazelle and Nazomi's JSTAR (see *MPR 2/12/01-01*, "Java To Go: Part 1"). These products reside between the instruction cache and the processor core. Some limitations are associated with these interpreters. For example, the processor/interpreter must perform the translation every time the code is run. In other words, there are no caching benefits. Contrast this with Lightfoot's and Bigfoot's bytecode translation during class loading.

The JSTAR interpreter can perform some of the same optimizations as Bigfoot and Lightfoot. For example, JSTAR can perform a simple folding of three Java instructions into one native atomic CPU instruction (iload x, iload y, iadd). As with the DCT products, the data must be localized within the Java execution engine (or the CPU's register file). This optimization happens on the fly, once the prefetch unit aligns and buffers bytecodes. On the other hand, Jazelle does not implement any optimization features (although this is not a limitation of the architecture).

When Jazelle or JSTAR encounters a Java bytecode that is not in its repertoire, it passes a pointer from a call-back table to the CPU, indicating where to execute from. The handoff to the CPU and back to the interpreter mode takes one to two cycles (but this doesn't account for the number of cycles used for saving the processor state). Lightfoot approaches this same problem with its soft bytecodes. Although the equivalent handoff requires an extra clock cycle, Lightfoot can make this transition without changing state and without flushing the pipeline. Both Jazelle and JSTAR have automatic fill/spill mechanisms for their Java stacks, but, like Lightfoot, each spill/fill consumes at least one processor cycle to complete the memory transaction.

JSTAR consumes between 27,000 and 30,000 gates (depending on the host CPU), roughly the same size as the entire Lightfoot core. On the basis of brute force, JSTAR will have a performance benefit over Lightfoot or Bigfoot because this coprocessor can scale to 400MHz. Jazelle requires an additional 12,000 gates and, when accompanying the ARM926EJ, will run at between 180MHz and 200MHz (although Jazelle itself will scale with the ARM core frequency).

The most direct competition for DCT (in the form of Java processing) comes from aJile, Imsys, and Zucotto Wireless.

Each of these companies offers standalone Java processors. The aJile aJ-100 microprocessor is a stack-based architecture running on a writable microcode store (see *MPR 8/7/00-02*, “Embedded Java Chips Get Real”). Because this is a microcoded engine, aJile can create customized instructions (in addition to the standard Java instructions) for specific applications. Sounds like Lightfoot, except that Lightfoot’s approach to custom instructions is cleaner and easier to implement, because it’s done in software.

The aJ-100 has 16K of microcode ROM, 16K of microcode SRAM, 40 pins of general-purpose I/O, two UARTs, a serial port interface (SPI), and three timer/counters. This chip sells for \$15 (10,000-unit quantities), which actually puts it in a different ballpark from Lightfoot. In April 2001, aJile announced the aJ-80, with fewer I/Os and 80MHz operation, selling for \$12. Nevertheless, it appears that Lightfoot offers higher performance, has more on-chip memory, and has roughly the same peripheral set while costing half the price of the aJ-100.

Zucotto Wireless has good relationships with many Tier One OEMs (including Nokia) and offers a core (Xpresso) and a silicon chip (Xpresso 100). Zucotto’s Xpresso isn’t the fastest Java processor on the market, but the company focuses more on efficient operation and low power consumption (see *MPR 6/4/01-01*, “Java To Go: Part 4). The Xpresso 100 features 16KB instruction and data caches, an eight-bit GPIO port, and a Bluetooth baseband controller (with the XJB 100 Bluetooth protocol upper stack and API for Java applications). *MPR* still can’t obtain any public pricing for this device, but we’ve been assured that “it is sampling to key customers.”

The Xpresso core contains a four-stage pipeline and a memory-management unit (MMU). Similar to the aJile product, Xpresso is a microcoded machine that will provide flexibility for future implementations but will be performance limiting. Zucotto’s software-based class loader, built into the company’s SLICE software layer, monitors small sequences of operations to determine if they can be optimized (which sounds similar to Bigfoot’s TR2 translation scheme). Zucotto’s class loader recognizes optimizable sequences and replaces them with a custom instruction. Furthermore, because the “folding” is performed at load time, the operations are translated only once—as opposed to at run time, when the operations would be required to fold every time the processor came across the particular sequence. Zucotto includes hardware support for garbage collection, which appears to be one distinct competitive advantage over Lightfoot or Bigfoot. Its patent-pending garbage collection function is spread across a number of instructions related to memory referencing.

A big portion of Lightfoot’s competition comes from established suppliers of microcontrollers and ASSPs (including companies such as Atmel, Hitachi, Infineon, Microchip, Motorola, NEC, STM, and Philips). Although none of these companies has announced plans for Java-based microcontrollers, they will more than likely license the technology from a third party. Philips’s plans may differ, because the

company is strongly focused on the consumer market (a potential sweet spot for Java).

DCT has several challenges and strengths in the microcontroller market. The biggest challenge is that it is a newcomer to this industry. Going up against the incumbents is always a challenge, no matter how great your technology is. Three other challenging factors in embedded development are tools, tools, and tools. Most companies listed in the preceding paragraph have been selling microcontrollers for many years and have developed a wide range of development tools to support those microcontrollers. One advantage that DCT has to combat this, however, is the C language benefits of both Lightfoot and Bigfoot. Although most, if not all, the vendors have C language support, their architectures are generally not very programmer friendly.

Getting Down to Business

DCTL is a fabless semiconductor vendor. The company has manufacturing relationships with Fujitsu, TSMC, and others. Although DCT’s primary business is selling chips, it will offer the Lightfoot and Bigfoot architectures as cores (in VHDL and Verilog format) for customers’ inclusion in systems on a chip (SoC). DCT will grant technology licenses on a per-end-user-product basis, structured with an initial license fee and a small royalty per commercial deployment. DCT will also provide an up-front pricing agreement with the customer for a follow-on license.

With regard to Bigfoot, the company will take advantage of its relationship with ARC Cores to supply ASIC development tools. ARC will license Bigfoot to customers wishing to add Java support to ARC’s base architecture. This action will boost DCT’s business, as ARC Cores is a well-established company that has a good list of licensees and potential customers. ARC will provide an IP sales channel for Bigfoot, with Java-related revenues shared with DCT (a nice benefit for newcomer DCT).

In its minimum configuration, ARC uses only 16,500 logic gates, including the 32x32 register file. The Bigfoot extensions add approximately 5,000 logic gates. ARC’s DSP options quadruple the gate count but provide capabilities that include saturating arithmetic, modulo (circular) addressing, longer accumulators, dual 16-bit multipliers, and several new instructions. Combined with Bigfoot’s Java features, this repertoire should yield a processor suitable for many wireless and wired applications. Broadly speaking, an ARC license costs about \$300,000; adding Bigfoot’s capability will double the cost.

Shera International has acquired a license to use DCT’s Lightfoot core for inclusion in a security-related SoC. The SoC project, which will be complete by mid-2002, will deliver a single-chip microcontroller for embedded applications requiring secure communications. The chip will integrate DCT’s 32-bit Lightfoot processor core with industry-standard communications ports, analog and digital I/O, on-chip program and data flash, and hardware-assisted encryption processing, plus proprietary I/O to support the particular design

Device	Nonvolatile Memory	SRAM	Ext Memory Interface	Price (10k units)	Avail.
LFJ1101	128KB ROM, 128KB flash	32KB	No	\$10	2Q02
LFJ0011	None	32KB	Yes	\$8	2Q02
LFJ0102	128KB flash	64KB	No	\$10	1Q03
LFJ0112	128KB flash	64KB	Yes	\$12	1Q03
LFJ1201	128KB ROM, 256KB FeRAM	None	No	\$10	3Q02
LFJ0211	256KB FeRAM	None	Yes	\$10	1Q03

Table 2. Initial product offerings from DCT will include a timer module and a serial port.

requirements of Shera's client base. This appears to be a unique product offering.

Shera International also has a Lightfoot-based product that will be available during 2002. The device contains an excellent mix of peripherals that support communications and security (USB1.1, 10Mb/s eMAC, DES, AES, RSA encryption hardware, eight timers, and two UARTs).

DCT's first Bigfoot product will be manufactured by Fujitsu on a 0.25-micron process, with samples expected in March 2002. The product includes dual 100MHz ARC cores (one of which will be Java enabled); 16k instruction and data caches; 2Mb of SDRAM; and ATAPI, USB1.1, and Ethernet support. Fujitsu's manufacturing of Bigfoot should be fairly straightforward; the company is one of the original ARC Cores licensees. In addition to being the manufacturing partner, Fujitsu's multimedia group is using the first Bigfoot part as part of its chip sets.

For its Lightfoot products, DCT will offer a variety of devices that have a simple peripheral mixture and different memory configurations (see Table 2). All devices will include a timer module with six 16-bit timers and two serial ports capable of supporting a UART or a SPI. Compared with competing devices in the embedded-controller market, these peripherals are a bare minimum. But the core's minimal size of 26k gates, combined with this minimal feature set, should allow DCT to offer a competitive pricing structure. All devices, on a 0.25-micron process, will operate at 100MHz with a voltage supply of 1.8V. DCT's first Lightfoot product, the LFJ1101, will be manufactured by TSMC on a 0.25-micron flash-based

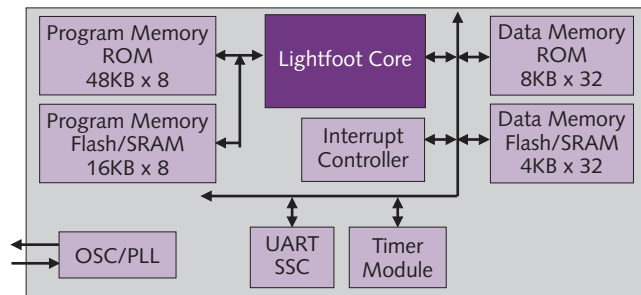


Figure 3. The LFJ1101 is DCT's first commercially available Lightfoot device.

process, with samples expected in April 2002 and production in October 2002 (Figure 3).

The success of Lightfoot will depend on DCT's ability to market its products and convince customers to go with the underdog and promote applications that simultaneously require the features of a microcontroller and a Java processor. The hardware security market (e.g., smartcards and smart-card terminals, network security) is an up-and-coming application area and should provide good opportunities for DCT, especially if the company can deliver its Lightfoot products at the quoted price points and performance levels.

With Lightfoot, DCT also plans to target opportunities in the wireless market not easily covered by the ARM7/9 Jazelle product. In that case, Lightfoot will serve as a self-contained companion chip that links to the existing solution via a serial port. This will provide a Java upgrade without having to redesign and remanufacture the main processor ASIC.

DCT is in discussion with OEMs regarding a Lightfoot-based coprocessor that will provide Java support in a mobile phone. This part, which will be sold directly, will sell for about \$5 (in quantities of 10 million to 40 million pieces per year). Combined with DCT's 46KB CLDC (including a proprietary RTOS) implementation, it will be a compelling product. MPR believes no other competitor can achieve this price point, memory footprint, and performance, although vendors such as aJile and Nazomi are preparing to introduce new silicon products for which we shall soon have details. ♦

MOST SIGNIFICANT BITS

GIGAHERTZ ULTRASPARC III SPEC SURPRISE

By Kevin Krewell {1/14/02-01}

Sun pulled more than one trick out of its hat with the introduction of the latest speed grade for the UltraSPARC III. For its latest 1,050MHz version, Sun used the same 0.15-micron, low-*k* dielectric TI semiconductor process it previously used for the 900MHz US III.

Sun has not put much credence in benchmark results and has not been an exceptional performer on SPEC benchmarks (see *MPR 9/4/01-02*, "900MHz UltraSPARC III Ready to Ship"). However, the latest US III produced surprising SPEC numbers, with one benchmark in particular showing an amazing increase over previous Sun benchmarks. The base score of 9,389 for the SPECfp2000 program 179.art is roughly four times the score of its closest competitor, the 800MHz Itanium. The combined SPECfp2000 (base) result of 701 virtually ties the 703 score achieved by the 800MHz Itanium, although it still trails the 1,098 score produced by the 1.3GHz Power 4. These new US III results are partially owing to the higher frequency and an improved translation look-aside buffer (TLB), but in large part the improvements are owing to a new Forte Developer 7 compiler. The new benchmark results now put the 1,050MHz UltraSPARC III in the middle of the high-performance pack instead of at the end of its tail. The US III at 1,050MHz is scheduled to be available for customer shipments in 1Q02.

AMD STARTS 2002 WITH MODEL 2000+

By Kevin Krewell {1/14/02-02}

Getting the jump on Intel's 0.13-micron Pentium 4 (Northwood) launch, AMD has squeezed one more speed grade out of the 0.18-micron Athlon XP. On January 7, 2002, AMD released the Athlon XP model number 2000+, running at 1.667GHz. The benchmarks AMD released showed the latest Athlon XP running about 10% faster than the 0.18-micron Pentium 4 (Willamette), but a good part of that performance gap will be eaten away by the performance gains in the Northwood processor, with its significantly larger 512KB L2 cache.

The Athlon XP model 2000+ operates at a nominal 1.75V, and the maximum operating temperature is 90° C. The maximum thermal power specification is 70.0W, and the typical thermal power is listed as 62.5W. In Stop Grant S1 or ACPI Sleep State (low-power state) Athlon XP processors drop to a nominal voltage of 1.30V and a maximum Icc of 1.54A.

The processor is shipping now, and systems based on it are expected to be available immediately from Compaq and later from manufacturers such as HP and MicronPC. The AMD Athlon XP model 2000+ is priced at \$339 in 1,000-unit quantities. Prices for the slower Athlon XP processors are \$269

for the Athlon XP model 1900+ (1.60GHz); \$223 for the Athlon XP model 1800+ (1.53GHz); \$190 for the Athlon XP model 1700+ (1.47GHz); and \$160 for the Athlon XP model 1600+ (1.4GHz).

BOPS ANNOUNCES NEW PERFORMANCE LEVELS

By Markus Levy {1/14/02-03}

On December 21, 2001, BOPS Inc. announced EEMBC benchmark scores that indicated improved capabilities of the company's new breakthrough Halo compiler (www.bops.com). The primary factor in this performance increase is a new global optimization component that BOPS has brought into its tools flow. Specifically, this component is the VLIW Instruction Memory Allocation (VIMA) tool, which performs a call-graph analysis of the entire program and globally allocates slots in the VLIW instruction memories (VIM). In other words, this analysis can identify potential VIM optimizations that require spanning a set of code modules and may not be visible from within a module. Furthermore, the tool lifts "Load VLIW" instruction sequences out of their position in a function and promotes these sequences to the highest safe location in the call graph (as part of the initialization sequence). This action allows the processor to run these instructions only once for each benchmark. (This action works similarly for real applications, such as 802.11A.)

The complexity of a VLIW processor combined with distributed processing elements requires either a highly skilled assembly programmer or an extremely robust compiler. Relying on the latter tool, BOPS continues to pour significant resources into its compiler; the result of this activity is evidenced by its latest release of EEMBC benchmark scores.

Specifically, the company released a certified EEMBC Telemark of 181.3, a marked contrast to a score of 139.8 only three months earlier. It derived this 30% improvement by using benchmark code that included only C optimizations (i.e., no assembly coding). The Telemark is designed to allow a quicker comparison between devices benchmarked in the Telecomm benchmark suite of EEMBC. It is calculated by applying a geometric mean of the scores in the Telecomm suite and dividing by 785.138 (each application suite has a unique normalization factor). The Telemark assumes equal weighting for all benchmarks in the benchmark suite.

A more interesting analysis can be applied by examining the scores for the individual benchmarks (www.eembc.org/benchmark). This examination shows that the biggest performance gains were realized from benchmarks with a larger percentage of overhead. For example, with the Auto-correlation benchmark, the implementation with the smaller dataset (pulse) achieves a 44% improvement. (A smaller

Continued on page 24

LITERATURE WATCH

AUDIO/VIDEO

Cutting-edge consoles target the television. The latest video-game consoles from Microsoft, Nintendo, and Sony package state-of-the-art technology at rock-bottom prices. Each system offers high-performance 3D graphics and interfaces to broadband networks. Brian Dipert, *EDN*, 12/20/01, p. 47, 8 pp.

Expanding options bring surround sound to the forefront...and the back...and the sides. Ever-evolving technology, in conjunction with DSPs and memory, creates an

immersive audio experience, no matter where you hear it. New DVD-Audio and Super Audio CD standards raise the quality bar for digital audio and demand higher-quality playback components and systems. Brian Dipert, *EDN*, 1/10/02, p. 34, 7 pp.

EMBEDDED SYSTEMS

Diagnose what ails your auto. Automotive onboard diagnostics help your engine perform at peak efficiency, reduce emissions, and even help you fix your car. Onboard diagnostic version 2 (OBD-2) systems, standard in U.S.-market

cars since 1996, process inputs from multiple sensors and control every aspect of engine operation. OBD-2 systems interface with external diagnostic equipment to speed vehicle repair. Greg Vrana, *EDN*, 12/20/01, p. 37, 5 pp.

INDUSTRY HISTORY

Artifacts: An Archeologist's Year in Silicon Valley by Christine A. Finn. This book provides a take on Silicon Valley and its impact on American culture as seen by a British journalist and archeologist. Finn talks with Valley locals, especially those who

have a historical perspective. 288 pp., MIT Press, \$24.95, ISBN 0-2620-62240.

SECURITY

Improving Security, Preserving Privacy. Securing public places depends on the right mix of technology, well-trained personnel, and, eventually, security-enhanced building design. Measures such as access control, surveillance, and automatic face recognition may have social costs that go beyond their benefits. Stephen Cass, Michael J. Riezenman, *IEEE Spectrum*, 1/02. ♦

MOST SIGNIFICANT BITS *Continued from page 23*

dataset implies that the inner benchmark loop is executed fewer times, which tends to exaggerate the time spent in the loop setup.) Compare this with the Autocorrelation implementation having the larger dataset (speech), which realizes an improvement of only 3%.

An additional factor in the compiler's performance increase was instruction-combining, which is a scalar optimization. This optimization finds potential instruction sequences that can be merged into a single instruction. For example, a LOAD followed by an INCR can be replaced with a LOAD/Post-INCR instruction.

MPR analysts need to see equally impressive performance enhancements in other market segments (in other words, the Telemark tells only part of the story). The BOPS approach is paralleled by the approaches of other companies, which have added flow-tracing capability to DSP arrays in an effort to improve the efficiency of their chips. Other benchmark areas that will be of interest include consumer applications such as digital imaging and MPEG. We look forward to seeing how well the Halo compiler handles these.

INTEL'S 2.2GHZ P4 PULLS AHEAD

By Kevin Krewell {1/22/02-03}

On January 7, 2002, Intel launched the 0.13-micron Pentium 4 (Northwood)—the same day AMD released Athlon XP model 2000+. Intel has used the process shrink to lower the power dissipation on the new Pentium 4 while increasing

clock frequency by 10%. Northwood also increases the L2 cache to 512KB, from 256KB in the original 0.18-micron Pentium 4 (Willamette) die. Intel released the new Pentium 4 at 2.2GHz and 2.0GHz with TDP of 55.1W and 52.4W, respectively, significantly cooler than the 67W TDP for the 2.0GHz Willamette. It also released even cooler versions of the 0.13-micron Pentium 4 at 2.0-, 1.8-, and 1.6GHz for TDP designs below 45W. All the 0.13-micron processors operate at a nominal core voltage of 1.5V.

Benchmarks on various PC-enthusiast Web sites indicate that the extra 256KB of L2 cache generally improves performance 4–9%, depending on the benchmark. The combination of the larger L2 cache and a 533MHz frequency lead pushes the 2.2GHz Pentium 4 ahead of AMD's Athlon XP model 2000+ on most benchmarks. Although the race is still relatively close, Intel will have the performance edge until AMD ships a faster 0.13-micron Athlon XP later in 1Q02. Intel published a SPECint2000(base) score of 771 and a SPECfp2000(base) score of 766 for the 2.2GHz Pentium 4 processor. The only processor with a higher SPECint score (at 790) is the 1.3GHz Power4 processor with 1.5MB of L2 cache and 128MB of L3 cache. The SPECfp score trails only the Power4 and the 1GHz Alpha processor.

The 0.13-micron Pentium 4 is shipping now, and systems based on it are expected to be available immediately from leading manufacturers. The Intel Pentium 4 processor at 2.2GHz is priced at \$562 in 1,000-unit quantities, and the 2.0GHz version is \$364. ♦

PATENT WATCH

By Rich Belgard, Contributing Editor

The following U.S. patents related to microprocessors were issued recently. Please send email to belgard@arithmetic.stanford.edu with comments or questions.

6,237,081

Queuing method and apparatus for facilitating the rejection of sequential instructions in a processor

Filed: December 16, 1998

Issued: May 22, 2001

Inventors: Hung Qui Le et al.

Claims: 20

Assignee: IBM

A processor includes an issue unit having an issue queue for issuing instructions to an execution unit. The execution unit may accept and execute the instruction or produce a reject signal. After each instruction is issued, the issue queue retains the issued instruction for a critical period. After the critical period, the issue queue may drop the issued instruction unless the execution unit had rejected it, in which case it is re-marked as available for issue.

6,237,064

Cache memory with reduced latency

Filed: February 23, 1998

Issued: May 22, 2001

Inventors: Harsh Kumar et al.

Claims: 19

Assignee: Intel

The invention provides methods and a data processing system for accessing memory of a data processing system, including a first-, and, at least a second-level cache. The method includes issuing a memory request to the first- and second-level caches simultaneously. If both caches hit, it retrieves data from both caches and ignores the information from the second-level cache.

6,233,690

Mechanism for saving power on long latency stalls

Filed: September 17, 1998

Issued: May 15, 2001

Inventors: Lynn Choi et al.

Claims: 16

Assignee: Intel

To improve power saving in a microprocessor, disclosed is a method for gating a clock signal to an execution unit on long-latency memory stalls. The method monitors an external stall signal, a data hazard signal, a resource hazard signal, and a data return signal. The clock signal is decoupled from the execution unit when the stall and data hazard signals are asserted for a selected interval and the data return and resource hazard signals are not asserted for a selected interval.

6,233,675

Facility to allow fast execution of and, or, and test instructions

Filed: March 25, 1999

Issued: May 15, 2001

Inventors: Kenneth Munson et al.

Claims: 32

Assignee: Rise Technology

Improvements are made in how microprocessors execute AND, OR, and TEST instructions when the operand registers or addresses of the two operands are equal. AND/OR/TEST instructions with equal operands are used to set flags based on the contents of only one of the operands without explicitly performing the actual AND/OR/TEST command. By setting these flags directly, this mechanism allows these instructions to be paired with preceding dependent instructions simply by using the flags set by the AND/OR/TEST for the previous instruction.

6,233,657

Apparatus and method for performing speculative stores

Filed: September 17, 1999

Issued: May 15, 2001

Inventors: H.S. Ramagopal et al.

Claims: 26

Assignee: AMD

An apparatus and methods for performing speculative stores in a microprocessor that reads the original data from a cache line that is being updated by the speculative store and stores the read data into a re-store buffer. The speculative data is then written into the cache line. If the speculative store is canceled, the original data is written back from the re-store buffer into the cache line, thereby re-storing the correct data.

6,230,261

Method and apparatus for predicting conditional branch instruction outcome based on branch condition test type

Filed: December 2, 1998

Issued: May 8, 2001

Inventors: Glenn Henry et al.

Claims: 23

Assignee: I.P. First

A static branch predictor in a microprocessor having an instruction set that uses the test condition of a conditional branch to statically predict the branch outcome. The methods and apparatus rely on types of test conditions, which are presumably biased toward either a true or false result. The methods further include using the displacement of the branch to predict the outcome if the type of the condition being tested does not fall into a biased type.

OTHER ISSUED PATENTS

6,237,077 *Instruction template for efficient processing clustered branch instructions*

6,237,074 *Tagged prefetch and instruction decoder for variable length instruction set and method of operation*

6,237,021 *Method and apparatus for the efficient processing of data-intensive applications*

6,233,679 *Method and system for branch prediction*

6,233,671 *Staggering execution of an instruction by dividing a full-width macro instruction into at least two partial-width micro instructions* ♦

CHART WATCH: PC PROCESSORS

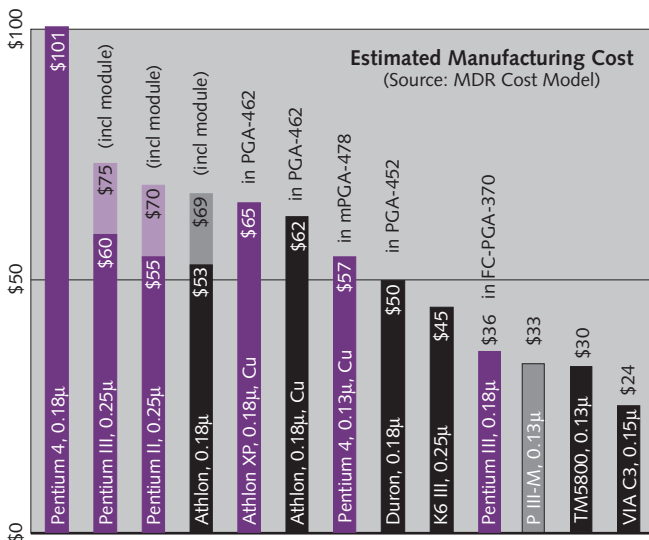
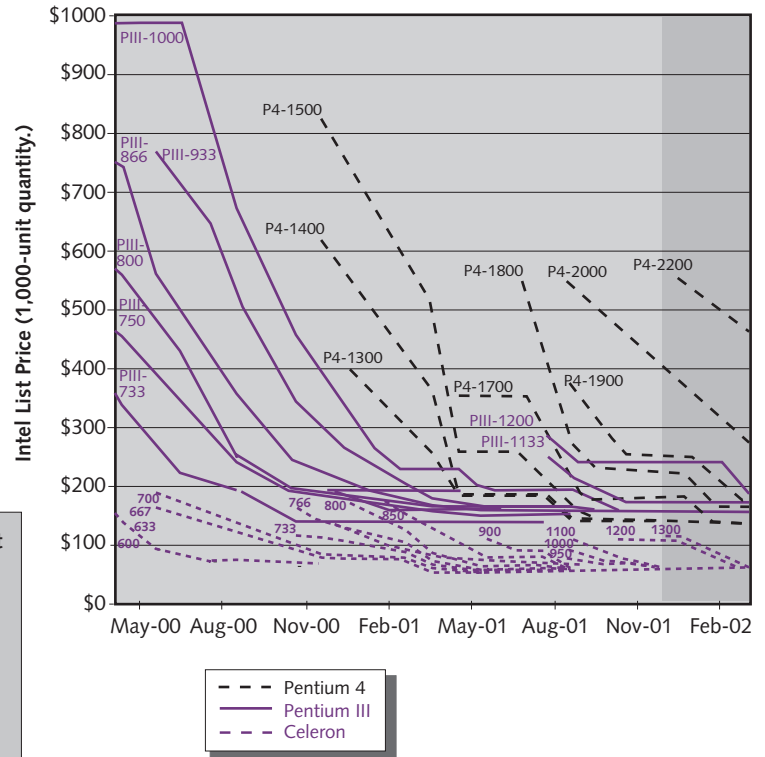
Processors	10/14/01	1/20/02	% Chg
Pentium 4-2200	—	\$562	—
Pentium 4-2000	\$562	\$364	35%
Pentium 4-1800	\$256	\$225	12%
Pentium 4-1700	\$193	\$193	0%
Pentium 4-1600	\$163	\$163	0%
Pentium 4-1500	\$133	\$133	0%
Celeron-1300	—	\$118	—
Celeron-1200	\$103	\$103	0%
Celeron-1100	\$89	\$89	0%
Celeron-1000	\$74	\$74	0%
Mobile Pentium III-M-1200	\$722	\$508	30%
Mobile Pentium III-M-1133	\$508	\$401	21%
Mobile Pentium III-M-1067	\$401	\$294	27%
Mobile Pentium III-M-1000	\$294	\$241	18%
Mobile Pentium III-M-933	\$241	\$198	18%
Mobile Pentium III-700 ULV	—	\$209	—
Mobile Celeron-933	\$134	\$134	0%
Mobile Celeron-866	\$107	\$107	0%
Mobile Celeron-800	\$91	\$91	0%
Mobile Celeron-600 ULV	\$144	\$118	18%

Vendor	Part/Number	List Price	Avail
AMD	—	—	—
	Athlon XP model 2000	\$339	Now
	Athlon XP model 1800	\$223	Now
	Athlon XP model 1700	\$190	Now
AMD	—	—	—
	Duron/1300	\$118	Now
	Duron/1200	\$103	Now
	Duron/1100	\$89	Now
AMD	—	—	—
	Duron/1000	\$74	Now
	Mobile Athlon 4/1200	\$525	Now
	Mobile Athlon 4/1100	\$425	Now
AMD	Mobile Athlon 4/1100	\$425	Now
	Mobile Athlon 4/1000	\$290	Now
	Mobile Athlon 4/900	\$230	Now
Transmeta	TM-5800/800	\$198	Now
AMD	Mobile Duron/900	\$100	Now
VIA	Mobile Duron/850	\$90	Now
VIA	VIA C3/800	n/a	Now
Transmeta	TM-5500/667	\$85	Now

n/a = not available (Source: vendors)

This edition of Chart Watch covers x86 processors for PC systems. The first table shows the latest pricing for Intel processors. The second table provides comparable pricing for other x86 processors.

The figure at the right shows historical Intel list pricing back to 2Q00 and MDR's projected Intel pricing through 1Q02. The figure below graphs the manufacturing costs of these chips as estimated by the MDR Cost Model.



EMBEDDED

The conference for embedded technology

PROCESSOR FORUM

April 29 – May 2, 2002
The Fairmont, San Jose, CA

The Embedded Industry's Most Important Week of the Year

■ The newest chips ■ The sharpest analysis ■ The freshest insights

Register Before MARCH 1 and Save up to \$600

Whether your application is in information appliances, digital audio, or networking, whether it requires low power, high performance, or DSP technology, the Embedded Processor Forum gives you the in-depth technical information you need to make winning embedded-design decisions. Embedded Processor Forum is the industry's premier event for new embedded-processor introductions and for full-day technical seminars on today's hottest embedded-design topics.

First disclosures of the newest processors and cores

Insightful seminars on the latest chips and applications

The opportunity to network with the industry's leading players

You don't want to be left behind, so register before March 1 to take advantage of special savings. Go to www.mdronline.com/epf/register or call **480.483.4441** before March 1, 2002.

For updated information on all the new technology announcements, seminars, and keynote speakers planned for Embedded Processor Forum 2002, please visit us on the Web at www.mdronline.com/epf.

Presented by
 **Cahners** IN-STAT / MDR

RESOURCES

◆ WINHEC MOVES CLOSER TO HOME

From hot and humid to cold and wet. After the 2001 show in New Orleans, **WinHEC 2002** will be in Seattle. The show, to be held April 16–18 in the Washington State Convention and Trade Center, features keynote presentations by Bill Gates and Intel VP Paul Otellini during the morning of April 18. The rest of the show will include multiple tracks of technical presentations on hardware design, driver development, and the Windows roadmap. *MPR* analyst Peter N. Glaskowsky will reprise his special session on PC Platform Technology, which won an award in 2001 as the highest-rated presentation by a non-Microsoft speaker. Register for the full conference by March 5 for just \$1,195, or pay \$1,595 at the door. More information is available online at www.microsoft.com/winhec.

◆ NETWORK AT NETWORLD+INTEROP

Videoconferencing over a fast Internet connection may be a reasonable alternative to some business travel, but there's no substitute for face time at **Networld+Interop 2002 Las Vegas**, the biggest annual show in the networking industry. N+I will be held at the Las Vegas Convention Center May 5–10, 2002; the expo floor is open May 7–9. Key3Media has already signed up more than 350 exhibitors for the expo, which also hosts

innumerable technical presentations, including the Network Processing Summit. Advance registration (through April 25) is free for the expo floor only; the full-week pass is \$2,795–\$2,995 at the door. For more information, network over to www.key3media.com/interop/lv2002.

◆ FOCUS ON SECURITY AT FOSE 2002?

No, that isn't what FOSE stands for. That's a secret. Never mind. If, however, you're involved with information technology in the government sector, you need to attend **FOSE 2002**, March 19–21 at the Washington, D.C. Convention Center. You'll be joining Adobe CEO Bruce Chizen, Intel CTO Pat Gelsinger, and three other CEOs—all giving keynotes at the show—as well as some 17,000 other attendees and hundreds of exhibitors at the event's trade show. Featured topics include homeland security, biometrics, and information accessibility. FOSE must be the best deal in the public sector: it's free to government professionals and just \$50 for the rest of us. The event is run by the Post Newsweek Tech Media Group, a division of the Washington Post Company, which knows more about the government than the government does! You can learn more at www.fose.com. (Confidentially, FOSE used to stand for Federal Office Systems Expo.) ◆

Be in the Know... with In-Stat/MDR's *Electronics Report*

For over 20 years, the *Electronics Report* has provided top executives in the electronics industry with an overview of monthly business trends relevant to their own business outlook. The newsletter provides a comprehensive compilation of high-level WSTS/SIA, SEMI, IPC and U.S. Dept. of Commerce data on semiconductors and their end products.

Each issue delivers the following data, along with a concise, easy-to-digest synopsis of month-to-month variations:

- Semiconductor Revenues, Unit Shipments and ASPs
- Wafer Fab Utilization
- Monthly U.S. Shipments and Bookings for Computer and Communications End Products as well as Semiconductor and Non-Semiconductor Components

For additional information, please contact Chris Kissel at 480.609.4531 or ckissel@instat.com.

 Cahners IN-STAT / MDR
<http://www.instat.com>

SUBSCRIPTION INFORMATION

To subscribe to *Microprocessor Report*, contact our customer service department in Scottsdale, Ariz., by phone, 480.609-4551; fax, 480.609.4523; email, emckeighan@instat.com; or Web, www.MDRonline.com.

One year	U.S. & Canada*	Elsewhere
Hardcopy or Electronic	\$695	\$795
Both Hardcopy and Electronic	\$795	\$895
Two years		
Hardcopy or Electronic	\$1,295	\$1,495
Both Hardcopy and Electronic	\$1,395	\$1,595

*Sales tax applies in the following states: AL, AZ, CO, DC, GA, HI, ID, IN, IA, KS, KY, LA, MD, MO, NV, NM, RI, SC, SD, TN, UT, VT, WA, and WV. GST or HST tax applies in Canada.

Microprocessor Report back issues are available on paper and CD-ROM. Volume reprints of individual articles are also available.

Ship to: