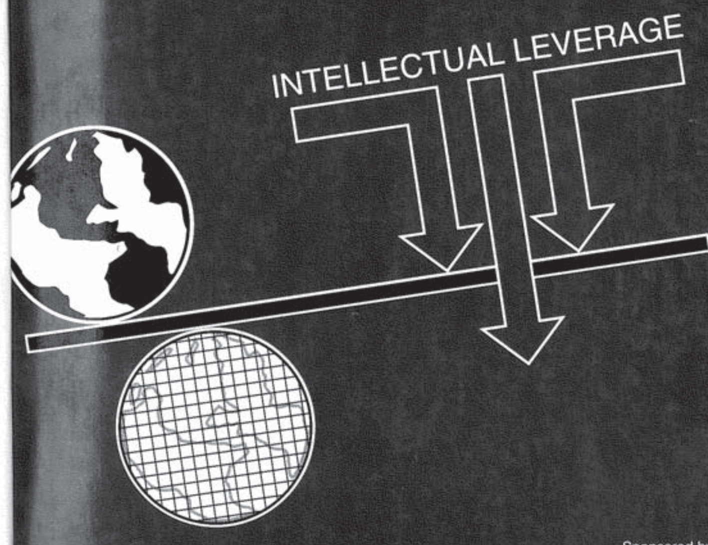
digest of papers

February 25–28, 1996

spring COMPCON 96

RTY-FIRST IEEE COMPUTER SOCIETY INTERNATIONAL CONFERENCE



Sponsored by
IEEE Computer Society

IEEE COMPUTER SOCIETY
50 YEARS OF SERVICE • 1946-1996

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

digest of papers

COMPCON '96

Technologies for the Information Superhighway

Forty-First IEEE Computer Society International Conference

Sponsored by — The IEEE Computer Society

February 25 – 28, 1996
Santa Clara, California

IEEE Computer Society Press
Los Alamitos, California

Washington • Brussels • Tokyo

IEEE Computer Society Press
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1264

IEEE Computer Society Press Order Number PR07414
ISBN 0-8186-7414-8
ISSN 1063-6390

IEEE Order Plan Catalog Number 96CB35911
Order Plan ISBN 0-8186-7415-6
Microfiche ISBN 0-8186-7416-4

Additional copies may be ordered from:

IEEE Computer Society Press
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1264
Tel: +1-714-821-8380
Fax: +1-714-821-4641
Email: cs.books@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: +1-908-981-1393
Fax: +1-908-981-9667
mis.custserv@computer.org

IEEE Computer Society
13, Avenue de l'Aquilon
B-1200 Brussels
BELGIUM
Tel: +32-2-770-2198
Fax: +32-2-770-8505
euro.ofc@computer.org

IEEE Computer Society
Ooshima Building
2-19-1 Minami-Aoyama
Minato-ku, Tokyo 107
JAPAN
Tel: +81-3-3408-3118
Fax: +81-3-3408-3553
tokyo.ofc@computer.org

Editorial production by Mary E. Kavanaugh
Cover by Joseph Daigle
Printed in the United States of America by KNI

The Institute of Electrical and Electronics Engineers, Inc.

Table of Contents

Message from the General Chair xi
Message from the Program Chair xiii
Organizing Committees xiv

Session 1: Wireless Interconnects
Chair: John Barr — *Motorola*

CDPD and Emerging Digital Cellular Systems 2
T. Melanchuk, P. Dupont, and S. Backer

Wireless Network Extension Using Mobile IP 9
R.L. Geiger, J.D. Solomon, and K.J. Crisler

The Bay Area Research Wireless Access Network (BARWAN) 15
R.H. Katz, E.A. Brewer, E. Amir, H. Balakrishnan, A. Fox, S. Gribble, T. Hodes, D. Jiang, G.T. Nguyen, V. Padmanabhan, and M. Stemm

Session 2: ATM Networks
Chair: Anujan Varma — *University of California, Santa Cruz*

Performance of Explicit Rate Flow Control in ATM Networks 22
L.G. Roberts

MPEG-2 Over ATM: System Design Issues 26
S. Varma

FAST: A Simulation Testbed for ATM Networks 32
D. Stiliadis and A. Varma

Session 3: Broadband Interactive Data Services
Chair: Ilja Bedner — *Hewlett-Packard*

HP BIDS — Broadband Interactive Data Solution 39
I. Bedner and A. Ranous

Design Considerations for a Hybrid Fiber Coax High-Speed Data Access Network 45
D. Picker

Session 4: Agent Languages
Chair: Adam Hertz — *General Magic*

Mobile Telescript Agents and the Web 52
P. Dömel

Mobile Agent Security and Telescript 58
J. Tardo and L. Valente

Session 5: World Wide Web

Chair: Robert Hagmann — *Oracle*

People, Places, and Things: The Next Generation Web 65
J. Gwertzman and M. Seltzer

An Internet Difference Engine and its Applications 71
T. Ball and F. Douglis

Don't Get Caught in the Web: A Fieldguide to Searching the Net 77
W.R. Tuthill

Session 6: World Wide Web Servers

Chair: Winfried Wilcke — *HAL Computer Systems*

A Scalable and Highly Available Web Server 85
D.M. Dias, W. Kish, R. Mukherjee, and R. Tewari

Session 7: Performance Characterization and Analysis

Co-Chairs: Nasr Ullah and Marianne Hsiung — *Motorola*

The Capture, Characterization, and Performance Analysis of Macintosh® Traces 94
S. McMahon

A Measurement Study of Memory Transaction Characteristics on a PowerPC-Based Macintosh 100
T. Adams

Load Miss Performance Analysis Methodology Using the PowerPC™ 604 Performance Monitor for OLTP Workloads 111
E.H. Welbon, R.S. Moore, F.E. Levine, and C.P. Roth

Workload Effects on SMP Scaling in AIX Version 4 117
K. Dixit, J. Van Fleet, and B. Olszewski

Session 8: Panel — Networking Virtual Environments

Chair: Michael Zyda — *Naval Postgraduate School*

Panelists: M. Zyda — "Networking Large-Scale Virtual Environments"
T. Meyer — "The Future of VRML"
M. Macedonia — "A Taxonomy for Networked Virtual Environments"
W. Katz — "Defense and Entertainment Industry Efforts in Networking Virtual Environments"

Session 9: PowerPC Microprocessors and Systems

Co-Chairs: Nasr Ullah — *Motorola*
Kaivalya Dixit — *IBM*

Design of the PowerPC 604e™ Microprocessor 126
M. Denman, P. Anderson, and M. Snyder

The Performance and PowerPC Platform™ Specification Implementation of the MPC106 Chipset 132
C.D. Bryant, M.J. Garcia, B.K. Reynolds, L.A. Weber, and G.E. Wilson

PowerPC Platform: A System Architecture 140
S. Bunch, R. Hochsprung, and T. Moore

Motorola PowerPC™ Migration Tools — Emulation and Translation 145
T. Afzal, M. Breternitz, M. Kacher, S. Menyhert, M. Ommerman, and W. Su

Session 10: PA-RISC Evolution

Chair: Ruby Lee — *Stanford University*

64-bit and Multimedia Extensions in the PA-RISC 2.0 Architecture 152
R. Lee and J. Huck

Mid-Range and High-End PA-RISC Computer Systems 161
R. Elsbernd

PA7300LC Integrates Cache for Cost/Performance 167
D. Hollenbeck, S.R. Undy, L. Johnson, D. Weiss, P. Tobin, and R. Carlson

Session 11: Having it your Way — High-Code-Density, High-Integration, and High-Performance ARMs

Chair: Allen Baum — *Apple Computer*

Thumb: Reducing the Cost of 32-bit RISC Performance in Portable and Consumer Applications 176
L. Goudge and S. Segars

ARM7100 — A High-Integration, Low-Power Microcontroller for PDA Applications 182
G. Budd and G. Milne

StrongARM: A High-Performance ARM Processor 188
R. Witek and J. Montanaro

Session 12: MPEG2

Chair: Vivian Shen — *Hewlett-Packard*

A Scalable Chip Set for MPEG2 Real-Time Encoding 193
A. Ngai, J. Sutton, C. Boice, and C. Gebler

Performance Comparison of MPEG1 and MPEG2 Video Compression Standards 199
S. Liu

Mediaprocessing in the Compressed Domain 204
V. Bhaskaran

Session 13: Interactive Television

Chair: Robert Hagmann — *Oracle*

A Distributed System Client/Server Architecture for Interactive Multimedia Applications 211
S. Rege

Dynamic Bandwidth Allocation for Interactive Video Applications over Corporate Networks 219
C.J. Beckmann

The Tiger Shark File System 226
R.L. Haskin and F.B. Schmuck

Session 14: Interactive TV Settop

Chair: Deven Kalra — *Hewlett-Packard*

Interactive Television Settop Terminal Architectures 233
A.N. Nair

Multimedia Transmission Link Protocol — A Proposal for Digital Information Transmission in HFC Cable Systems 239
R-F. Chiu and R. Hutchinson

DAVID® System Software v2.0 for Interactive Digital Television Networks 241
A. Davidson

Session 15: Scalable Clusters

Chair: Marco Annaratone — *DEC Western Research Laboratory*

Overview of Memory Channel Network for PCI 244
R. Gillett, M. Collins, and D. Pimm

Digital's Clusters and Scientific Parallel Applications 250
R. Kaufmann and T. Reddin

Overview of Digital UNIX Cluster System Architecture 254
W.M. Cardoza, F.S. Glover, and W.E. Snaman, Jr.

Session 16: HAL Computer Systems

Chair: Winfried Wilcke — *HAL Computer Systems*

A 9.6 GigaByte/s Throughput Plesiochronous Routing Chip 261
A. Mu, J. Larson, R. Sastry, T. Wicki, and W.W. Wilcke

Performance Limiting Factors in Http (Web) Server Operations 267
F. Prefect, L. Doan, S. Gold, T. Wicki, and W. Wilcke

Session 17: Exploiting New Storage and Network Technologies

Chair: Norman J. Pass — *IBM Almaden Research Center*

SSA: A High-Performance Serial Interface for Unparalleled Connectivity 274
A. Wilson

Redundant Arrays of Independent Libraries (RAIL): A Tertiary Storage System 280
D.A. Ford, R.J.T. Morris, and A.E. Bell

Randomized Data Allocation for Real-Time Disk I/O 286
S. Berson, R.R. Muntz, and W.R. Wong

Services and Architectures for Electronic Publishing 291
D.M. Choy and R.J.T. Morris

Session 18: Multimedia Authoring

Chair: Michael A. Harrison — *University of California, Berkeley*

Graphical Object-Oriented Multimedia Application Development: Technology and Market Trends 299
H. Steger

Graphical Containment in Multimedia Authoring 300
H. Epelman-Wang, S. Markowitz, and B. Roddy

User Interfaces for Authoring Systems with Object Stores 305
B. Roddy, S. Markowitz, and H. Epelman-Wang

Session 19: Competing Architectures for Multimedia Processing

Chair: Cary Kornfeld — *consultant*

The Mpact™ Media Processor Redefines the Multimedia PC 311
P. Foley

An Architectural Overview of the Programmable Multimedia Processor, TM-1 319
S. Rathnam and G. Slavenburg

Improving Performance for Software MPEG Players 327
D.F. Zucker, M.J. Flynn, and R.B. Lee

Session 20: The MicroUnity Mediaprocessor

Chair: Steve Manser — *MicroUnity Systems*

Architecture of a Broadband MediaProcessor 334
C. Hansen

MicroUnity Software Development Environment 341
R. Hayes, G. Loyola, C. Abbott, and H. Massalin

Broadband Algorithms with the MicroUnity Mediaprocessor 349
C. Abbott, H. Massalin, K. Peterson, T. Karzes, L. Yamano, and G. Kellogg

Session 21: DRAM Technologies

Chair: S. Peter Song — *Samsung*

Burst and Latency Requirements Drive EDO and BEDO DRAM Standards 356
A. Mormann

Synchronous DRAM Evolutionary Changes Bring Cost/Performance Advantages in Memory Systems 360
A.B. Cosoroaba

High Bandwidth RDRAM Technology Reduces System Cost 365
R. Crisp

Multi-Gigabyte/sec DRAM with the MicroUnity MediaChannel™ Interface 378
T. Robinson, C. Hansen, B. Herndon, and G. Rosseel

Session 22: Pentium®Pro System Architecture

Chair: Konrad Lai — *Intel*

An Overview of the Pentium®Pro Processor Bus 383
N. Sarangdhar and G. Singh

Pentium®Pro Processor Workstation/Server PCI Chipset 388
M. Bell and T. Holman

Multiprocessor Validation of the Pentium®Pro Microprocessor 395
D. Marr, S. Thakkar, and R. Zucker

Session 23: Storage Technology

Chair: Harry S. Gill — *IBM*

Data Storage IC Technolgy 402
J. Kovacs and R. Kroesen

Session 24: UltraSPARC and Java

Chair: Robert Garner — *Sun Microsystems*

UltraSPARC™: Compiling for Maximum Floating-Point Performance 408
P. Tirumalai, D. Greenley, B. Beylin, and K. Subramanian

UltraSPARC-II™: The Advancement of UltraComputing 417
G. Goldman and P. Tirumalai

Java™ and HotJava: A Comprehensive Overview 424
S. Shaio, A. van Hoff, and H. Jellinek

Session 25: Desktop Color — From Eye to Paper

Chair: Allen Baum — *Apple Computer*

Digital Cameras and Electronic Color Image Acquisition 431
J. Dalton

Electronic Color Printing Technology 435
G.K. Starkweather

ColorSync™: Synchronizing the Color Behavior of Your Devices 440
W-L. Chu and S. Swen

Session 26: Architecture of Workflow Management Systems

Chair: Berthold Reinwald — *IBM Almaden Research Center*

Object-Oriented Workflow Technology in InConcert 446
S.K. Sarin

Structured Workflow Management with Lotus Notes Release 4 451
B. Reinwald and C. Mohan

An Architecture for Large-Scale Work Management Systems 458
M. Beizer

Session 27: "Toy Story"

Chair: Darrell Long — *University of California, Santa Cruz*

The Making of *Toy Story* 463
M. Henne, H. Hickel, E. Johnson, and S. Konishi

Additional Paper: *The following paper was presented as the last paper in Session 12*

Single Chip MPEG2 Decoder with Integrated Transport deocder for Set-top Box 469
J. Fandrianto

Author Index 473

An Architectural Overview of the Programmable Multimedia Processor, TM-1

Selliah Rathnam, Gert Slavenburg

Philips Semiconductors
811 E. Arques Avenue, Sunnyvale, CA 94088

ABSTRACT

TM-1 is the first in a family of programmable multimedia processor from the Trimedia product group of Philips Semiconductors. This "C" programmable processor has a high performance VLIW-CPU core with video and audio peripheral units designed to support the popular multimedia applications. TM-1 is designed to concurrently process video, audio, graphics, and communication data. The VLIW-CPU core is capable of executing a maximum of twenty seven operations per cycle, and the sustained execution rate is about five operations per cycle for the tuned applications. The audio unit easily handles different audio formats including the 16-bit stereo data. The video unit is capable of processing different YUV and RGB pixel formats with horizontal and vertical scaling and color space conversion. TM-1 applications can range from low-cost, stand alone systems such as video phones to programmable, multipurpose plug-in cards for traditional computers.

1.0 INTRODUCTION

TM-1 is a building-block for high-performance multimedia applications that deal with high-quality video and audio. TM-1 easily implements popular multimedia standards such as MPEG-1 and MPEG-2, but its orientation around a powerful general-purpose CPU makes it capable of implementing a variety of multimedia algorithms, whether open or proprietary.

More than just an integrated microprocessor with unusual peripherals, the TM-1 microprocessor is a fluid

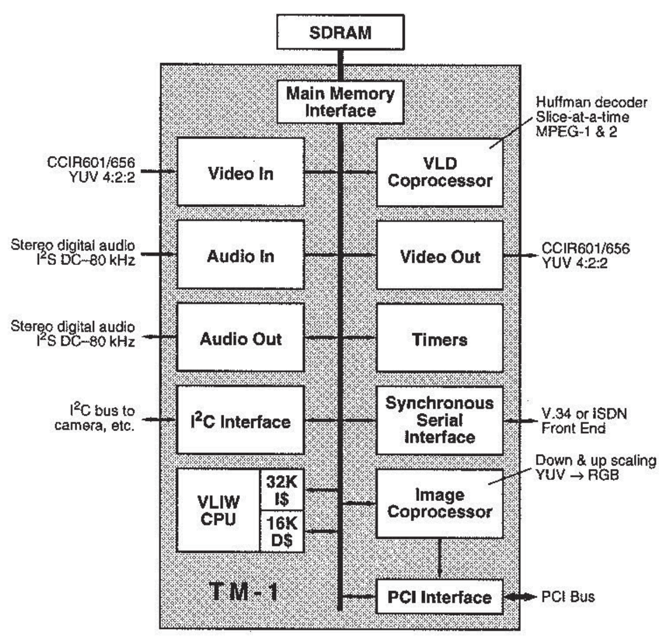


Figure 1. TM-1 block diagram.

1063-6390/96 $5.00 © 1996 IEEE
Proceedings of COMPCON '96

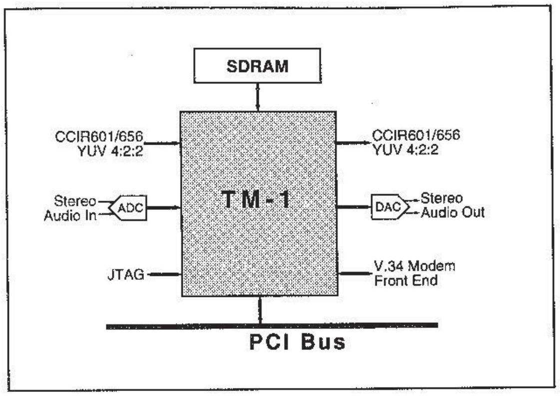


Figure 2. TM-1 system connections. A minimal TM-1 system requires few supporting components.

computer system controlled by a small real-time OS kernel that runs on the VLIW processor core. TM-1 contains a CPU, a high-bandwidth internal bus, and internal bus-mastering DMA peripherals.

TM-1 is the first member of a family of chips that will carry investments in software forward in time. Compatibility between family members is at the source-code level; binary compatibility between family members is not guaranteed. All family members, however, will be able to perform the most important multimedia functions, such as running MPEG-2 software.

Defining software compatibility at the source-code level gives Philips the freedom to strike the optimum balance between cost and performance for all the chips in the TM-1 family. Powerful compilers ensure that programmers seldomly need to resort to non-portable assembler programming. Programmers use TM-1's powerful low-level operations from source code; these DSP-like operations are invoked with a familiar function-call syntax. Trimedia also provides hand-coded and tuned multimedia libraries which can be used to increase the performance of the multimedia applications.

As the first member of the family, TM-1 is tailored for use in PC-based applications. Because it is based on a general-purpose CPU, TM-1 can serve as a multi-function PC enhancement vehicle. Typically, a PC must deal with multi-standard video and audio streams, and users desire both decompression and compression, if possible. While the CPU chips used in PCs are becoming capable of low-resolution real-time video decompression, high-quality video decompression—not to mention compression—is still out of reach. Further, users demand that their systems provide live video and audio without sacrificing the responsiveness of the system.

TM-1 enhances a PC system to provide real-time multimedia, and it does so with the advantages of a special-purpose, embedded solution—low cost and chip count—*and* the advantages of a general-purpose processor—reprogrammability. For PC applications, TM-1 far surpasses the capabilities of fixed-function multimedia chips.

Other Trimedia family members will have different sets of interfaces appropriate for their intended use. For example, a TM-1 chip for a cable-TV decoder box would eliminate the video-in interface.

2.0 TM-1 CHIP OVERVIEW

The key features of TM-1 are:

- A very powerful, general-purpose VLIW processor core that coordinates all on-chip activities. In addition to implementing the non-trivial parts of multimedia algorithms, this processor runs a small real-time operating system that is driven by interrupts from the other units.
- DMA-driven multimedia input/output units that operate independently and that properly format data to make processing efficient.
- DMA-driven multimedia coprocessors that operate independently and perform operations specific to important multimedia algorithms.
- A high-performance bus and memory system that provides communication between TM-1's processing units.

Figure 1 shows a block diagram of the TM-1 chip. The bulk of a TM-1 system consists of the TM-1 microprocessor itself, a block of synchronous DRAM (SDRAM), and minimal external circuitry to interface to the incoming and/or outgoing multimedia data streams. TM-1 can gluelessly interface to the standard PCI bus for personal-computer-based applications; thus, TM-1 can be placed directly on the PC mainboard or on a plug-in card.

Figure 2 shows a possible TM-1 system application. A video-input stream, if present, might come directly from a CCIR 601-compliant digital video camera chip in YUV 4:2:2 format; the interface is glueless in this case. A non-standard camera chip can be connected via a video decoder chip (such as the Philips SAA7111). A CCIR 601 output video stream is provided directly from the TM-1 to drive a dedicated video monitor. Stereo audio input and output require external ADC and DAC support. The operation of the video and audio interface units is highly customizable through programmable parameters.

The glueless PCI interface allows the TM-1 to display video via a host PC's video card and to play audio via a host PC's sound hardware. The Image Coprocessor provides display support for live video in an arbitrary number of arbitrarily overlapped windows.

Finally, the V.34 interface requires only an external modem front-end chip and phone line interface to provide remote communication support. The modem can be used to connect TM-1-based systems for video phone or video conferencing applications, or it can be used for general-purpose data communication in PC systems.

3.0 BRIEF EXAMPLES OF OPERATION

The key to understanding TM-1 operation is observing that the CPU and peripherals are time-shared and that communication between units is through SDRAM memory. The CPU switches from one task to the next; first it decompresses a video frame, then it decompresses a slice of the audio stream, then back to video, etc. As necessary, the CPU issues commands to the peripheral units to orchestrate their operation.

The TM-1 CPU can enlist the ICP and video-in units to help with some of the straightforward, tedious tasks associated with video processing. The function of these units is programmable. For example, some video streams are—or need to be—scaled horizontally, so these units can handle the most common cases of horizontal down- and up-scaling without intervention from the TM-1 CPU.

3.1 Video Decompression in a PC

A typical mode of operation for a TM-1 system is to serve as a video-decompression engine on a PCI card in a PC. In this case, the PC doesn't know the TM-1 has a powerful, general-purpose CPU; rather, the PC just treats the hardware on the PCI card as a "black-box" engine.

Video decompression begins when the PC operating system hands the TM-1 a pointer to compressed video data in the PC's memory (the details of the communication protocol are typically handled by a software driver installed in the PC's operating system).

The TM-1 CPU fetches data from the compressed video stream via the PCI bus, decompresses frames from the video stream, and places them into local SDRAM. Decompression may be aided by the VLD (variable-length decoder) unit, which implements Huffman decoding and is controlled by the TM-1 CPU.

When a frame is ready for display, the TM-1 CPU gives the ICP (image coprocessor) a display command. The ICP then autonomously fetches the decompressed frame data from SDRAM and transfers it over the PCI bus to the frame buffer in the PC's video display card (or the frame buffer in PC system memory if the PC uses a UMA (Unified Memory Architecture) frame buffer). The ICP accommodates arbitrary window size, position, and overlaps.

3.2 Video Compression

Another typical application for TM-1 is in video compression. In this case, uncompressed video is usually supplied directly to the TM-1 system via the video-in unit. A camera chip connected directly to the video-in unit supplies YUV data in eight-bit, 4:2:2 format. The video-in unit takes care of sampling the data from the camera chip and demultiplexing the raw video to SDRAM in three separate areas, one each for Y, U, and V.

When a complete video frame has been read from the camera chip by the video-in unit, it interrupts the TM-1 CPU. The CPU compresses the video data in software (using a set of powerful data-parallel operations) and writes the compressed data to a separate area of SDRAM.

The compressed video data can now be disposed of in any of several ways. It can be sent to a host system over the PCI bus for archival on local mass storage, or the host can transfer the compressed video over a network, such as ISDN. The data can also be sent to a remote system using the integrated V.34 interface to create, for example, a video phone or video conferencing system.

Since the powerful, general-purpose TM-1 CPU is available, the compressed data can be encrypted before being transferred for security.

4.0 VLIW CORE AND PERIPHERAL UNITS

4.1 VLIW Processor Core

The heart of TM-1 is its powerful 32-bit CPU core. The CPU implements a 32-bit linear address space and 128, fully general-purpose 32-bit registers. The registers are not separated into banks; any operation can use any register for any operand.

The core uses a VLIW instruction-set architecture and is fully general-purpose. TM-1 uses a VLIW instruction length that allows up to five simultaneous operations to be issued. These operations can target any five of the 27 functional units in the CPU, including integer and floating-point arithmetic units and data-parallel DSP-like units.

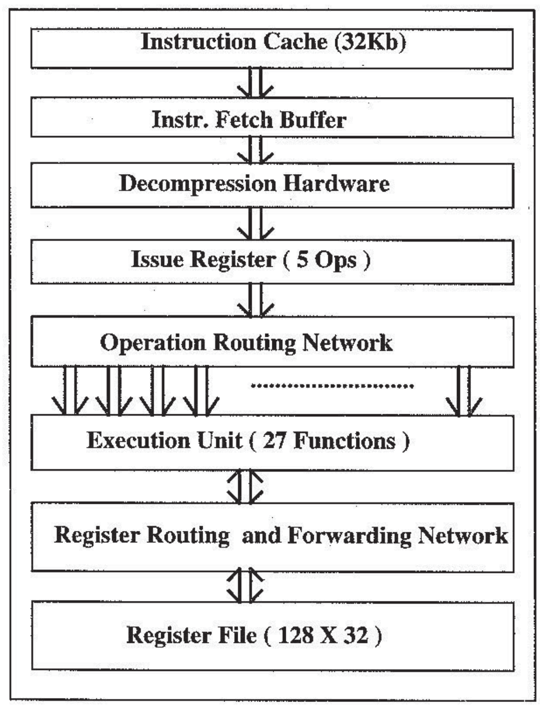


Figure 3. VLIW Processor Core and Instruction Cache.

Although the processor core runs a tiny real-time operating system to coordinate all activities in the TM-1 system, the processor core is not intended for true general-purpose use as the only CPU in a computer system. For example, the processor core does not implement virtual memory address translation, an essential feature in a general-purpose computer system.

TM-1 uses a VLIW architecture to maximize processor throughput at the lowest possible cost. VLIW architectures have performance exceeding that of superscalar general-purpose CPUs without the extreme complexity of a superscalar implementation. The hardware saved by eliminating superscalar logic reduces cost and allows the integration of multimedia-specific features that enhance the power of the processor core.

The TM-1 operation set includes all traditional microprocessor operations. In addition, multimedia-specific operations are included that dramatically accelerate standard video compression and decompression algorithms. As just one of the five operations issued in a single TM-1 instruction, a single special or "custom" operation can implement up to 11 traditional microprocessor operations. Multimedia-specific operations combined with the VLIW architecture result in tremendous throughput for multimedia applications.

4.2 Internal "Data Highway" Bus

The internal data bus connects all internal blocks together and provides access to internal control registers (in each on-chip peripheral units), external SDRAM, and the external PCI bus. The internal bus consists of separate 32-bit data and address buses, and transactions on the bus use a block-transfer protocol. Peripherals can be masters or slaves on the bus.

Access to the internal bus is controlled by a central arbiter, which has a request line from each potential bus master. The arbiter is configurable in a number of different modes so that the arbitration algorithm can be tailored for different applications. Peripheral units make requests to the arbiter for bus access, and depending on the arbitration mode, bus bandwidth is allocated to the units in different amounts. Each mode allocates bandwidth differently, but each mode guarantees each unit a minimum bandwidth and maximum service latency. All unused bandwidth is allocated to the TM-1 CPU.

The bus allocation mechanism is one of the features of TM-1 that makes it a true real-time system instead of just a highly integrated microprocessor with unusual peripherals.

4.3 Memory and Cache Units

TM-1's memory hierarchy satisfies the low cost and high bandwidth requirement of multimedia markets. Since multimedia video streams can require relatively large temporary storage, a significant amount of DRAM is required.

TM-1 has a glueless interface with synchronous DRAM (SDRAM) or synchronous graphics RAM (SGRAM), which provide higher bandwidth than the standard DRAM. As the SDRAM has been supported by major DRAM vendors, the competition among those vendors will keep the SDRAM price in par with that of the standard DRAM. TM-1's DRAM memory size can range from 2Mbytes to 64 Mbytes.

The TM-1 CPU core is supported by separate 16-KB data and 32-KB instruction caches. The data cache is dual-ported in order to allow two simultaneous load/store accesses, and both caches are eight-way set-associative with a 64-byte block size.

4.4 Video-In Unit

The video-in unit interfaces directly to any CCIR 601/656-compliant device that outputs eight-bit parallel, 4:2:2 YUV time-multiplexed data. Such devices include direct digital camera systems, which can connect gluelessly to TM-1 or through the standard CCIR 656 connector with only the addition of ECL level converters. Non-CCIR-compliant devices can use a digital decoder chip, such as the Philips SAA7111, to interface to TM-1. Older front ends with a 16-bit interface can connect with a small amount of glue logic.

The video-in unit demultiplexes the captured YUV data before writing it into local TM-1 SDRAM. Separate data structures are maintained for Y, U, and V.

The video-in unit can be programmed to perform on-the-fly horizontal resolution subsampling by a factor of two if needed. Many camera systems capture a 640-pixel/line or 720-pixel/line image; with subsampling, direct conversion to a 320-pixel/line or a 360-pixel/line image can be performed with no CPU intervention. Further, if subsampling is required eventually, performing this function during data capture reduces initial storage requirements.

4.5 Video-Out Unit

The video-out unit essentially performs the inverse function of the video-in unit. Video-out generates an eight-bit, multiplexed YUV data stream by gathering bits from the separate Y, U, and V data structures in SDRAM. While generating the multiplexed stream, the video-out unit can also up-scale horizontally by a factor of two to convert from CIF to native CCIR resolution.

Since the video-out unit likely drives a separate video monitor—not the PC's video screen—the PC itself cannot be used to generate the graphics and text of a user interface. To remedy this, the video-out unit can generate graphics overlays in a limited number of configurations.

4.6 Image Coprocessor (ICP)

The image coprocessor (ICP) is used for several purposes to off-load tasks from the TM-1 CPU, such as copying an image from SDRAM to the host's video frame buffer. Although these tasks can be easily performed by the CPU, they are a poor use of the relatively expensive CPU resource. When performed in parallel by the ICP, these tasks are performed efficiently by simple hardware, which allows the CPU to continue with more complex tasks.

The ICP can operate as either a memory-to-memory or a memory-to-PCI coprocessor device.

In memory-to-memory mode, the ICP can perform either horizontal or vertical image filtering and resizing. The ICP implements 32 FIR filters of five adjacent pixel input values. The filter coefficients are fully programmable, and the position of the output pixel in the output raster determines which of the 32 FIR filters is applied to generate that output pixel value. Thus, the output raster is on a 32-times finer grid than the input raster. The filtering is done in either the horizontal or vertical direction but not both. Two applications of the ICP are required to filter and scale in both directions.

In memory-to-PCI mode, the ICP can perform horizontal resizing followed by color-space conversion. For example, assume an $n \times m$ pixel array is to be displayed in a window on the PC video screen while the PC is running a graphical user interface. The first step (if necessary) would use the ICP in memory-to-memory mode to perform a vertical resizing. The second step would use the ICP in memory-to-PCI mode to perform a horizontal resizing (if necessary) and colorspace conversion from YUV to RGB.

While sending the final, resampled and converted pixels over the PCI bus to the video frame buffer, the ICP uses a full, per-pixel occlusion bit mask—accessed in destination coordinates—to determine which pixels are actually stored in the frame buffer for display. Conditioning the transfer with the bit mask allows TM-1 to accommodate an arbitrary arrangement of overlapping windows on the PC video screen.

Figure 3 illustrates a possible display situation and the data structures in SDRAM that support the ICP's operation. On the left in Figure 3, the PC's video screen has four overlapping windows. Two, Image 1 and Image 2, are being used to display video generated by TM-1.

The right side of Figure 3 shows a conceptual view of SDRAM contents. Two data structures are present, one for Image 1 and the other for Image 2. Figure 3 represents a point in time during which the ICP is displaying Image 2.

When the ICP is displaying an image (i.e., copying it from SDRAM to a frame buffer), it maintains four pointers to the data structures in SDRAM. Three pointers locate the Y, U, and V data arrays, and the fourth locates the per-pixel occlusion bit map. The Y, U, and V arrays are indexed by source coordinates while the occlusion bit map is accessed with screen coordinates.

As the ICP generates pixels for display, it performs horizontal scaling and colorspace conversion. The final RGB pixel value is then copied to the destination address in the screen's frame buffer only if the corresponding bit in the occlusion bit map is a one.

As shown in the conceptual diagram, the occlusion bit map has a pattern of 1s and 0s that corresponds to the shape of the visible area of the destination window in the frame buffer. When the arrangement of windows on the PC screen is changed, modifications to the occlusion bit maps may be necessary.

It is important to note that there is no preset limit on the number and sizes of windows that can be handled by the ICP. The only limit is the available bandwidth. Thus, the ICP can handle a few large windows or many small win-

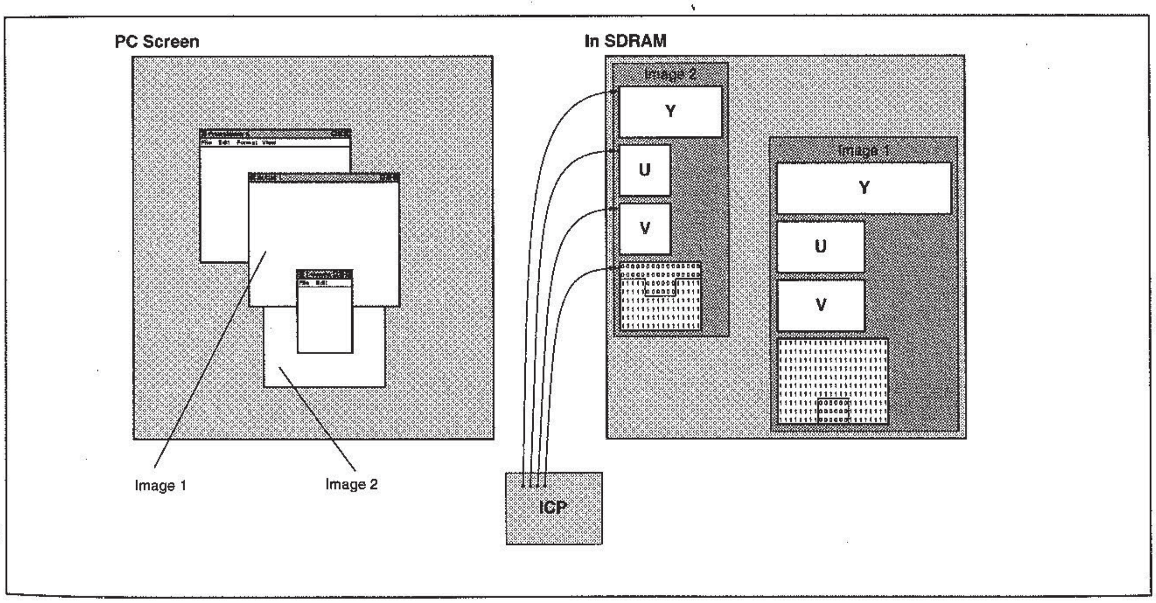


Figure 4. ICP operation. Windows on the PC screen and data structures in SDRAM for two live video windows.

dows. The ICP can sustain a transfer rate of 50 megapixels per second, which is more than enough to saturate PCI when transferring images to video frame buffers.

ICP has a micro-programmable engine. All ICP operations such as filtering, scaling and color space conversions and their formats are programmable. The ICP's micro programs loads itself from the SDRAM memory.

4.7 Variable-Length Decoder (VLD)

The variable-length decoder (VLD) is included to relieve the TM-1 CPU of the task of decoding Huffman-encoded video data streams. It can be used to help decode MPEG-1 and MPEG-2 video streams.

The VLD is a memory-to-memory coprocessor. The TM-1 CPU hands the VLD a pointer to a Huffman-encoded bit stream, and the VLD produces a tokenized bit stream that is very convenient for the TM-1 image decompression software to use. The format of the output token stream is optimized for the MPEG-2 decompression software so that communication between the CPU and VLD is minimized.

As with the other processing-intensive coprocessors, the VLD is included mainly to relieve the CPU of a task that wastes its performance potential. When dealing with the high bit rates of MPEG-2 data streams, too much of the CPU's time is devoted to this task, which prevents its special capabilities from being used.

4.8 Audio-In and Audio-Out Units

The audio-in and audio-out units are similar to the video units. They connect to most serial ADC and DAC chips, and are programmable enough to handle most reasonable protocols. These units can transfer MSB or LSB first and left or right channel first.

The sampling clock is driven by TM-1 and is software programmable within a wide range from DC to 80 kHz with a resolution of 0.02 Hz. The clock circuit allows the programmer subtle control over the sampling frequency so that audio and video synchronization can be achieved in any system configuration. When changing the frequency, the instantaneous phase does not change, which allows frequency manipulation without introducing distortion.

As with the video units, the audio-in and audio-out units buffer incoming and outgoing audio data in SDRAM. The audio-in unit buffers samples in either eight- or 16-bit format, mono or stereo. The audio-out unit simply transfers sample data from memory to the external DAC; any manipulation of sound data is performed by the TM-1 CPU since this processing will require at most a few percent of the CPU resource.

4.9 PCI Bus Interface Unit (BIU)

This unit connects the internal Data Highway Bus to an external PCI bus. It has a PCI master to initiate memory read/write cycles for TM-1-CPU requested read/write transactions including burst read/write DMA transactions. The PCI target within the BIU responds to the transactions initiated by external PCI master devices to read/write the TM-1's memory space, and it satisfies their requests. External devices can access the TM-1's MMIO registers through this unit.

The ICP unit has a direct connection to the BIU unit in order to transfer the pixel image data efficiently from TM-1 to the graphics device or host memory through the PCI bus.

The DMA transactions are considered as background transactions. To reduce the latency of the single word read/write transactions on the PCI bus, the BIU interleaves the burst read/write DMA cycles with single word read/write transactions.

5.0 CUSTOM OPERATIONS

Custom operations in the TM-1 CPU architecture are specialized, high function operations designed to dramatically improve performance in important multimedia applications. Custom operations enable an application to take advantage of the high performance VLIW-CPU core.

Important multimedia applications, such as the decompression of MPEG video streams, spend significant amounts of execution time dealing with eight-bit data items. Using 32-bit operations to manipulate small data items makes inefficient use of 32-bit execution hardware in the implementation. There are custom operations designed to operate on four eight-bit data items simultaneously in order to improve the performance about four to ten times compared with that of the general purpose CPU. Furthermore, some custom operations are defined to combine multiple arithmetic and control instructions into a single custom operation. These custom operations can be used easily in the C language as function calls.

Custom operation syntax is consistent with the C pro-

```
unsigned char A[16][16];
unsigned char B[16][16];
        .
        .
        .
for (row = 0; row < 16; row += 1)
{
    for (col = 0; col < 16; col += 1)
        cost += abs(A[row][col] - B[row][col]);
}
```

Figure 5. Match-cost loop for MPEG motion estimation.

gramming language, and just as with all other operations generated by the compiler, the scheduler takes care of register allocation, operation packing, and flow analysis.

The multimedia application development has been additionally improved by providing hand coded and well tuned multimedia code in the form of 'C' library functions.

5.1 Example: Motion-Estimation Kernel

One part of the MPEG coding algorithm is motion estimation. The purpose of motion estimation is to reduce the cost of storing a frame of video by expressing the contents of the frame in terms of adjacent frames.

A given frame is reduced to small blocks, and a subsequent frame is represented by specifying how these small blocks change position and appearance; usually, storing the difference information is less expensive than storing a whole block. For example, in a video sequence in which the camera pans across a static scene, some frames can be expressed simply as displaced versions of their predecessor frames. To create a subsequent frame, most blocks are simply displaced relative to the output screen.

The code in this example is for a match-cost calculation, a small kernel of the complete motion-estimation code. This code provides an excellent example of how to transform source code in order to make the best use of TM-1's custom operations.

Figure 5 shows the original source code for the match-cost loop. The code is not a self-contained function. At some location early in the code, the arrays A[][] and B[][] are declared; At some location between those declarations and the loop of interest, the arrays are filled with data.

We start by noticing that the computation in the loop of Figure 5 involves the absolute value of the difference of two unsigned characters (bytes). TM-1 operation set includes, several operations that process all four bytes in a 32-bit word simultaneously. Since the match-cost calculation is fundamental to the MPEG algorithm, it is not surprising to find a custom operation—ume8uu—that implements this operation exactly. The definition of ume8uu operation is shown in Figure 8.

If we hope to use a custom operation that processes four pixel values simultaneously, we first need to create four parallel pixel computations. Also, to use the ume8uu operation, however, the code must access the arrays with 32-bit word pointers instead of with 8-bit byte pointers.

Figure 6 shows a parallel version of the code from Figure 5. By unrolling the loop and simply giving each computation its own cost variable and then summing the costs all at once, each cost computation is completely independent.

Figure 7 shows the loop recoded to access A[][] and B[][] as one-dimensional instead of as two-dimensional arrays. We take advantage of our knowledge of C-language array storage conventions in order to perform this code transformation. Recoding to use one-dimensional arrays prepares the code for the transformation to 32-bit array accesses.

Figure 7 also shows the loop of Figure 6 recoded to use ume8uu. Once again taking advantage of our knowledge of the C-language array storage conventions, the one-dimensional byte array is now accessed as a one-dimensional 32-bit-word array.

Of course, since we are now using one-dimensional arrays to access the pixel data, it is natural to use a single 'for' loop instead of two. Figure 9 shows this streamlined version of the code without the inner loop. Since C-language arrays are stored as a linear vector of values, we can simply increase the number of iterations of the outer loop from 16 to 64 to traverse the entire array.

The recoding and use of the ume8uu operation has resulted in a substantial improvement in the performance of the match-cost loop. In the original version, the code executed 1280 operations (including loads, adds, subtracts, and absolute values); in the restructured version, there are only 256 operations—128 loads, 64 ume8uu operations, and 64 additions. This is a factor of five reduction in the number of operations executed. Also, the overhead of the inner loop has been eliminated, further increasing the performance advantage.

```
unsigned char A[16][16];
unsigned char B[16][16];
        .
        .
        .
for (row = 0; row < 16; row += 1)
{
    for (col = 0; col < 16; col += 4)
    {
       cost0 = abs(A[row][col+0] - B[row][col+0]);
       cost1 = abs(A[row][col+1] - B[row][col+1]);
       cost2 = abs(A[row][col+2] - B[row][col+2]);
       cost3 = abs(A[row][col+3] - B[row][col+3]);

        cost += cost0 + cost1 + cost2 + cost3;
    }
}
```

Figure 6. Unrolled and Parallel version of Figure 5.

```
unsigned char A[16][16];
unsigned char B[16][16];
        .
        .
        .
unsigned int *IA = (unsigned int *) A;
unsigned int *IB = (unsigned int *) B;

for (row = 0, rowoffset = 0; row < 16;
                 row += 1, rowoffset += 4)
{
    for (col4 = 0; col4 < 4; col4 += 1)
        cost += UME8UU(IA[rowoffset + col4],
                       IB[rowoffset + col4]);
}
```

Figure 7. Using the custom operation ume8uu to speedup the loop of Figure 6 resulted in a performance speedup of about

```
ume8uu            Sum of absolute values of
                  unsigned 8-bit differences

'C' function prototype:
    unsigned int
    ume8uu(unsigned int a, unsigned int b );

Function of ume8uu:
abs(zero_extto32(a<31:24>) - zero_extto32(b<31:24>))+
abs(zero_extto32(a<23:16>) - zero_extto32(b<23:16>))+
abs(zero_extto32(a<15:8>)  - zero_extto32(b<15:8>)) +
abs(zero_extto32(a<7:0>)   - zero_extto32(b<7:0>))
```

Figure 8. Custom Operation ume8uu

```
nsigned char A[16][16];
unsigned char B[16][16];
    .
    .
unsigned int *IA = (unsigned int *) A;
unsigned int *IB = (unsigned int *) B;

for (i = 0; i < 64; i += 1)
    cost += UME8UU(IA[i], IB[i]);
```

Figure 9. The loop of Figure 7 with the inner loop eliminated.

6.0 APPLICATIONS

TM-1 has the potential to be used in many multimedia applications and only few of them are discussed.

6.1 Video Teleconferencing/Digital White Board

Businesses are increasingly turning towards interactive computing as a means of becoming more efficient. Collaborative computing, for instance, involves sharing applications amongst multiple personal computers and multipoint video teleconferencing.

TM-1 is a single chip video teleconferencing solution that runs all current video codecs across all common transport mechanisms. This may also includes H.324 (POTS), H.320 (ISDN) and H.323 (LAN).

6.2 Multimedia Card for Consumer Multimedia Applications

The achievement of true computer based realism is only possible with a fully integrated approach to multimedia -- one that permits the smooth flow of audio, video, graphics and communications. Today's computer user wants a highly interactive and realistic experience. The Trimedia processor makes this possible.

TM-1 is a low-cost, programmable processor for the consumer multimedia market. This product provides the additional processing power required for a true-to-life computer based experience. The Trimedia processor concurrently processes multiple data types including audio, video, graphics and communications. The first version of this chip, designated TM-1, is targeted for the PC market.

7.0 SUMMARY

The TM-1 is the first programmable multimedia processor from the Trimedia division of the Philips Semiconductors. The TM-1 has high performance VLIW CPU core, efficient 'C' compiler with multimedia library functions, glueless logic to high-bandwidth SDRAM, standard PCI bus interface, and standard interfaces to video and audio stream that make the TM-1 the next generation multimedia processor for stand-alone systems such as the video phone, video conferencing system and plug-in multimedia cards for the PC systems.

8.0 REFERENCES

[1] J. Labrousse, G. A Slavenburg. "A 50MHz Microprocessor with a VLIW Architecture." ISSCC, 1990.

[2] J. Labrousse, G. A. Slavenburg. "CREATE-LIFE: A Design System for High Performances VLSI Circuits" ICCD-88. 1988.

[3] J. Labrousse, G. A. Slavenburg. "CREATE-LIFE: A Modular Design Approach for High Performances ASIC's." Compcon Conference 1990.

[4] Brian Case. "Philips Hopes to Displace DSPs with VLIW" Microprocessor Report, December 5, 1994.

[5] Brian Case. "First Trimedia Chip Boards PCI Bus." Microprocessor Report, November 1995.

[6] Gert Slavenburg. "The Trimedia VLIW-Based PCI Multimedia Processor" In Microprocessor Forum, October, 1995.

[7] A.S. Huang, G. Slavenburg, J.P. Shen. "Speculative Disambiguation: A compilation Technique for Dynamic Memory Disambiguation". In 21st Annual International Symposium on Computer Architecture, April, 1994.

[8] R.P. Colwell, R.P. Nix, J.J O'Donnell, D.B. Papworth, P.K. Rodman. " A VLIW Architecture for a Trace Scheduling Compiler." Proc. of ASPLOS II. October, 1987.

[9] J.A. Fisher. "Trace Scheduling: A Technique for Global Microcode Compaction." IEEE Trans on Computers. July 1981.

[10] P.Y.T. Hsu and E.S. Davidson. "Highly Concurrent Scalar Processing." Proc. of the 13th Symposium on Computer Architecture, 1986

Offices of the IEEE Computer Society

Headquarters Office
1730 Massachusetts Avenue, N.W.
Washington, DC 20036-1903
Phone: (202) 371-0101 — Fax: (202) 728-9614
E-mail: hq.ofc@computer.org

Publications Office
P.O. Box 3014
10662 Los Vaqueros Circle
Los Alamitos, CA 90720-1264
Membership and General Information: (714) 821-8380
Publication Orders: (800) 272-6657 — Fax: (714) 821-4010
E-mail: cs.books@computer.org

European Office
13, avenue de l'Aquilon
B-1200 Brussels, BELGIUM
Phone: 32-2-770-21-98 — Fax: 32-2-770-85-05
E-mail: euro.ofc@computer.org

Asian Office
Ooshima Building
2-19-1 Minami-Aoyama, Minato-ku
Tokyo 107, JAPAN
Phone: 81-3-408-3118 — Fax: 81-3-408-3553
E-mail: tokyo.ofc@computer.org

IEEE Computer Society

IEEE Computer Society Press Publications

CS Press publishes, promotes, and distributes over 20 original and reprint computer science and engineering texts annually. Original books consist of 100 percent original material; reprint books contain a carefully selected group of previously published papers with accompanying original introductory and explanatory text.

Submission of proposals: For guidelines on preparing CS Press books, write to Manager, Press Product Development, IEEE Computer Society Press, P.O. Box 3014, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-1264, or telephone (714) 821-8380.

Purpose

The IEEE Computer Society advances the theory and practice of computer science and engineering, promotes the exchange of technical information among 100,000 members worldwide, and provides a wide range of services to members and nonmembers.

Membership

All members receive the monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.

Publications and Activities

Computer Society On-Line: Provides electronic access to abstracts and tables of contents from society periodicals and conference proceedings, plus information on membership and volunteer activities. To access, telnet to the Internet address info.computer.org (user i.d.: guest).

***Computer* magazine:** An authoritative, easy-to-read magazine containing tutorial and in-depth articles on topics across the computer field, plus news, conferences, calendar, interviews, and product reviews.

Periodicals: The society publishes 10 magazines and seven research transactions.

Conference proceedings, tutorial texts, and standards documents: The Computer Society Press publishes more than 100 titles every year.

Standards working groups: Over 200 of these groups produce IEEE standards used throughout the industrial world.

Technical committees: Over 29 TCs publish newsletters, provide interaction with peers in specialty areas, and directly influence standards, conferences, and education.

Conferences/Education: The society holds about 100 conferences each year and sponsors many educational activities, including computing science accreditation.

Chapters: Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.

Published by the IEEE Computer Society Press
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1264

IEEE Computer Society Press Order Number PR07414
IEEE Order Plan Catalog Number 96CB35911
ISBN 0-8186-7414-8
ISSN 1063-6390

ISBN 0-8186-7414-8
90000
9 780818 674143