

Proceedings

IEEE COMPCON 97

San Jose, California

February 23 – 26, 1997

Sponsored by
IEEE Computer Society



IEEE Computer Society Press
Los Alamitos, California

Washington • Brussels • Tokyo

SONY 1019

TK7885
C6132
1997



IEEE Computer Society Press
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314

Copyright © 1997 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved

Copyright and Reprint Permissions: Abstracting is permitted with credit to the source. Libraries may photocopy beyond the limits of US copyright law, for private use of patrons, those articles in this volume that carry a code at the bottom of the first page, provided that the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923.

Other copying, reprint, or republication requests should be addressed to: IEEE Copyrights Manager, IEEE Service Center, 445 Hoes Lane, P.O. Box 133, Piscataway, NJ 08855-1331.

The papers in this book comprise the proceedings of the meeting mentioned on the cover and title page. They reflect the authors' opinions and, in the interests of timely dissemination, are published as presented and without change. Their inclusion in this publication does not necessarily constitute endorsement by the editors, the IEEE Computer Society Press, or the Institute of Electrical and Electronics Engineers, Inc.

IEEE Computer Society Press Order Number PR07804
ISBN 0-8186-7804-6
ISBN 0-8186-7805-4 (case)
ISBN 0-8186-7806-2 (microfiche)
IEEE Order Plan Catalog Number 97CB36028
ISSN 1063-6390

Additional copies may be ordered from:

IEEE Computer Society Press
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1-714-821-8380
Fax: + 1-714-821-4641
E-mail: 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 Bob Werner
Art production by Joe Daigle/Studio Productions
Printed in the United States of America by Technical Communication Services



The Institute of Electrical and Electronics Engineers, Inc.

Table of Contents

Proceedings of COMPCON'97

Message from the Conference Chair	xiii
Message from the Program Chair	xiv
Organizing Committees	xv
Distance Health Track Overview	xvi

Monday Morning Sessions

Track 1

New Microsoft Server Technologies Chair: Jim Gray, Microsoft Research

Universal Data Access with OLE DB	2
<i>J. Blakeley</i>	
Windows NT Clusters for Availability and Scalability	8
<i>R. Short, R. Gamache, J. Vert, M. Massa</i>	
Microsoft Transaction Server	14
<i>R. Limprecht</i>	

Track 2

DEC Alphas: Server to Desktops Chair: Allen Baum, DEC

The Alpha 21164PC Microprocessor	20
<i>P. Bannon, Y. Saito</i>	
The Alpha 21264: A 500 MHz Out-of-Order Execution Microprocessor	28
<i>D. Leibholz, R. Razdan</i>	
DIGITAL FX!32: Running 32-Bit x86 Applications on Alpha NT	37
<i>R. Hookway</i>	

Track 3

Image and Video Quality Chair: Ruby Lee, Hewlett-Packard

Color Image Quality Metric S-CIELAB and its Application on Halftone Texture Visibility	44
<i>X. Zhang, D. Silverstein, J. Farrell, B. Wandell</i>	
An Evaluation of Video Fidelity Metrics	49
<i>A. Yu, R. Lee, M. Flynn</i>	
Image Quality Metrics Based on Single and Multi-Channel Models of Visual Processing	56
<i>J. Farrell, X. Zhang, C. van den Branden Lambrecht, D. Silverstein</i>	

LINDA HALL LIBRARY
KANSAS CITY, MISSOURI

Monday Early Afternoon Sessions

Track 1

Trees and Tree Transformations in Interactive Environments

Chair: Michael A. Harrison, University of California at Berkeley

Efficient Self-Versioning Documents	62
<i>T. Wagner, S. Graham</i>	
Presentation by Tree Transformation.....	68
<i>M. Harrison, V. Maverick</i>	
Grendel: A Web Browser with End User Extensibility.....	74
<i>B. Dennis, M. Harrison</i>	

Track 2

New PA-RISC Processors and Systems

Chair: Ruby Lee, Hewlett-Packard

PA-8500: The Continuing Evolution of the PA-8000 Family	
<i>B. Smith</i>	
The Evolution of the HP/Convex Exemplar	81
<i>T. Brewer, G. Astfalk</i>	
Computer Optimizations for the PA-8000	87
<i>A. Holler</i>	

Track 3

Web on Your TV

Chair: Robin Williams, IBM Almaden

Web Browser Intelligence: Opening up the Web.....	122
<i>R. Barrett, P. Maglio, D. Kellem</i>	
Acorn's Technology for Network Computing.....	124
<i>J. Redford, M. Taunton</i>	
The Information Appliance Revolution	
<i>J. Busch</i>	

Monday Mid-Afternoon Sessions

Track 1

Internet Security

Chair: Robert Hagmann, Vitria Technology, Inc.

New Security Architectural Directions for Java.....97
L. Gong

Privacy-Enhancing Technologies for the Internet..... 103
I. Goldberg, D. Wagner, E. Brewer

Track 2

Functional Verification of Microprocessor Architecture and Design

Chair: Sunil Kakkar, Integrated Device Technology

A Comprehensive Diagnostics Software Strategy for IDT's Microprocessors 111
S. Kakkar

Functional Verification of the Superscalar SH-4 Microprocessor 115
P. Biswas, A. Freeman, K. Yamada, N. Nakagawa, K. Uchiyama

Track 3

Thin Web Clients

Chair: Robin Williams, IBM Almaden

Screen Phones
M. Jennings

Odin, A Low Cost Device to Work with TV
J. Salvadore

A Low Cost, High Speed Portable Communications Device: A Case Study using
IDT RV4640 Microprocessor and IDT77903 ATM Card 172
S. Kakkar, A. Hussain

Tuesday Morning Sessions

Track 1

Object-Relational Database Systems

Chair: Robert Hagmann, Vitria Technology

Evolution of Object-Relational Database Technology in DB2 131
D. Chamberlin

Bringing Objects to the Mainstream 136
A. Nori, S. Kumar

DataBlade Extensions for INFORMIX-Universal Server	143
<i>M. Olson</i>	

Track 2

The Origin System

Chair: Forrest Baskett, Silicon Graphics

System Overview of the SGI Origin 200/2000 Product Line	150
<i>J. Laudon, D. Lenoski</i>	

Origin System Design Methodology and Experience: 1M-gate ASICs and Beyond	157
<i>A. Eiriksson, J. Keen, A. Silbey, S. Venkataraman, M. Woodacre</i>	

The SGI Origin Software Environment and Application Performance	165
<i>S. Whitney, J. McCalpin, N. Bitar, J. Richardson, L. Stevens</i>	

Track 3

Digital Photography

Chair: John Meyer, Hewlett-Packard Labs

Visual Realism and Interactivity for the Internet	269
<i>R. Blumberg, P. Hughes</i>	

Output Technologies for Digital Photography	274
<i>L. Hanlon</i>	

Image Processing Considerations for Digital Photography	280
<i>P. Wong, D. Tretter, C. Herley, N. Moayeri, R. Lee</i>	

Tuesday Early Afternoon Sessions

Track 1

Java Enabling Compilers and Translators

Chair: Tariq Afzal, Samsung Semiconductor

A Gcc-based Java Implementation	174
<i>P. Bothner</i>	

Briki: An Optimizing Java Compiler	179
<i>M. Cierniak, W. Li</i>	

Making Java™ Math Libraries by Translating Fortran	185
<i>M. Myszewski</i>	

Track 2

3D Graphics Accelerators

Chair: Dileep Bhandarkar, Intel

GLINT 500TX: Second Generation Silicon for Personal Workstation Graphics

N. Trevett

PERMEDIA: Low Cost Entry-Level Professional 3D Graphics

N. Trevett

RealizM Graphics 192
M. Buckelew

Track 3

Panel on Taxation of the Internet

Chair: Ted Laliotis, Laliotis and Associates

Tuesday Mid-Afternoon Sessions

Track 1

Optimizations and Benchmarking for Java

Chair: Tariq Afzal, Samsung Semiconductor

NetRexx — An Alternative for Writing Java Classes 200
M. Cowlishaw

Java Bytecode Optimizations 206
H. Lambright

A Study of the Cache and Branch Performance Issues with Running Java on
Current Hardware Platforms 211
C. Hsieh, M. Conte, T. Johnson, J. Gyllenhaal, W. Hwu

Track 2

Network Hardware

Chair: Edward Frank, Epigram, Inc.

FX-1000: A High Performance Single Chip Gigabit Ethernet NIC 218
M. Ross, A. Bechtolsheim, M. Le, J. O'Sullivan

The Reality of ATM Networks 224
R. Sansom

Track 3

Information Technology and People with Disabilities

Chair: T.V. Raman, Adobe Systems

Emacspeak — An Audio Desktop	229
<i>T. Raman</i>	
Structure and Representation of Documents and Data	232
<i>M. Krishnamoorthy</i>	
Access to Electronic Information by People with Disabilities	235
<i>M. Paciello, Y. Rubinsky</i>	

Wednesday Morning Sessions

Track 1

The Inferno Operating System

Chair: Darrell Long, UCSC

Inferno	241
<i>S. Dorward, R. Pike, D. Presotto,</i> <i>D. Ritchie, H. Trickey, P. Winterbottom</i>	
Programming in Limbo	245
<i>S. Dorward, R. Pike, P. Winterbottom</i>	
Inferno Security	251
<i>D. Presotto</i>	

Track 2

New Intel Microprocessor Technology

Chair: Dileep Bhandarkar, Intel

Intel's MMX™ Technology — A New Instruction Set Extension	255
<i>O. Lempel, A. Peleg, U. Weiser</i>	
The Pentium® Processor with MMX™ Technology	260
<i>M. Eden, M. Kagan</i>	
Pentium® Processor with MMX™ Technology Performance	263
<i>M. Greene</i>	

Track 3

Distance Health for the General Population

Chair: Leslie Sandburg, CyperOptions

Introduction to Domains, Terms and Players	
<i>L. Sandburg</i>	

Telemedicine becomes Distance Health
L. Lorton

A Guide to the Literature, with Emphasis on the WWW
F. Strange

Wednesday Early Afternoon Sessions

Track 1

New Optical Disc Technology: DVD
Chair: Akito Iwamoto, Toshiba

DVD Overview 287
H. Yamada

DVD-Video Format 291
H. Mimura

DVD-RAM
M. Takenaga

DVD-R
A. Inoue

Track 2

Silicon Entertainment
Chair: Allen Baum, Digital Equipment Corporation

The Silicon Gaming Odyssey Slot Machine.....296
A. Levinthal, M. Barnett

Designing Virtual Reality Systems:
A Case Study of a System with Human/Robotic Interaction.....302
R. Latham

Pippin: Adapting PC Technology for the Technophobe 308
*E. Sirkin, V. Mehra, E. Keshishoglou,
L. Madar, A. Carter, W. Knott*

Track 3

Distance Health for Special Populations
Chair: Bijoy Khandheria, MD, Mayo Clinic

Distance Health for the Underserved
A. Liebgott, P. Hansen, T. Lightner

Home Health Monitoring:
Cost Effective Approaches to Quality Care in Non-Trauma Settings
V. Dragon, F. Russell, R. Byrd, J. Fitch

Early Experiences in ATM-Based Telemedicine and a 10 Gigabit/sec Testbed to Support Additional Distance Health Trials
J. Terdiman; B. Lennon

Wednesday Mid-Afternoon Sessions

Track 1

Data Bases, OLAP

Chair: Larry Stevens, Oracle

Panelists: J. Kraft, M. Adrian

Track 2

The Next-Generation PowerPC™ Microprocessor

Chairs: Nasr Ullah, Motorola and Marianne Hsiung, Apple

A G3 PowerPC™ Superscalar Low-Power Microprocessor.....315
*R. Kennedy, M. Alexander, E. Fiene, J. Lyon,
B. Kuttanna, R. Patel, M. Pham, M. Putrino,
C. Croxton, S. Litch, B. Burgess*

Thermal Management System for High Performance PowerPC™ Microprocessors.....325
*H. Sanchez, B. Kuttanna, T. Olson, M. Alexander
G. Gerosa, R. Philip, J. Alvarez*

Performance Implications of Next Generation PowerPC™ Microprocessor Cache Architectures.....331
J. Reinold

Track 3

Distance Health Futures: How To Proceed

Chair: Paul B. Lee, Pantheon Corp.

Implications of the Kennedy-Kassebaum Bill
J. Scanlon

Mainstreaming Telemedicine into the Healthcare Delivery System
D. Puskin

The California Telehealth Initiative
M. Thompson

Distance Health Technologies to Augment the Capacity of the Military Healthcare System
A. Ckacko

Index of Authors 341

A G3 PowerPC™ Superscalar Low-Power Microprocessor

A. Richard Kennedy, Mike Alexander, Eric Fiene, Jose Lyon, Belli Kuttanna,
Rajesh Patel‡, Mydung Pham‡, Mike Putrino‡, Cody Croxton,
Suzanne Litch, Brad Burgess

Motorola, Inc., ‡International Business Machines Corporation
Somerset Design Center, 6300 Bridgepoint Parkway, Bldg #4, Austin TX. 78730

Abstract

This microprocessor is the third generation PowerPC microprocessor and is a member of the G3 family of PowerPC processor products. Although its high performance (estimated 10.0 SPECint95) makes it suited for high end desktop systems, its low typical power dissipation of 5W and size of 66.5mm² make it very attractive for portable systems as well. This microprocessor is a dual-issue superscalar machine with a four stage pipeline, separate Instruction and Data side L1 caches (32KB each), and full tags and support for up to a 1MB of back-side L2. A Thermal Assist Unit and I-cache Throttling feature are included in the microprocessor as additional tools for thermal management. This microprocessor was designed in a 0.25um CMOS process (0.18um Leff) to operate at a frequency of 250MHz.

1. Introduction

This member of the G3 family of PowerPC microprocessors is the latest in the line of PowerPC microprocessors developed at the Somerset Design Center in Austin, Tx. A large number of features have been implemented in this processor. These include:

- Two independent integer units capable of executing all fixed point shift, rotate, logical, add and compare instructions
- Specialized hardware for integer multiplies and divides
- A floating point unit capable of handling double precision operands
- A load/store unit with two reservation stations
- Control and tags for up to 1MB of back-side L2 cache.
- Separate 32KB, 8-way, instruction and data caches
- Instruction and data cache support for hit-under-miss
- Separate 128-entry instruction and data TLBs

- Hardware Tablewalks
- Support for misaligned and Little Endian accesses
- A 64-entry Branch Target Instruction Cache (BTIC)
- A 512 entry Branch History Table (BHT) for dynamic branch prediction.
- Ability to fetch past two unresolved branches
- A Second Fixed point unit that can execute *all* single cycle fixed point instructions
- 6 GPR and 6 FPR renames
- Store Gathering
- Additional Clock modes
- DLL for L2 synchronization
- Thermal Assist Unit
- Icache Throttling
- Performance monitor

A performance simulator specifically designed to model this microprocessor was used for sensitivity analysis. The simulator is flexible enough to quickly evaluate performance trade-offs. The benchmarks used in the evaluation process include traces from MacSpec (a collection of applications running under the MacOS), Spec92 and Spec95 (both integer and floating point). This analysis helped determine which features gave large enough performance increases to justify their hardware costs and schedule impact. Equally important, the analysis points out what features should *not* be added because their impact on schedule and frequency far outweighs their performance gains. This is elaborated further in Section 13.

2. Overview

A block diagram of this microprocessor is shown in Figure 1.

The branch unit determines the effective addresses of the instructions to be fetched. Requested instructions that miss in the caches are brought into the processor via

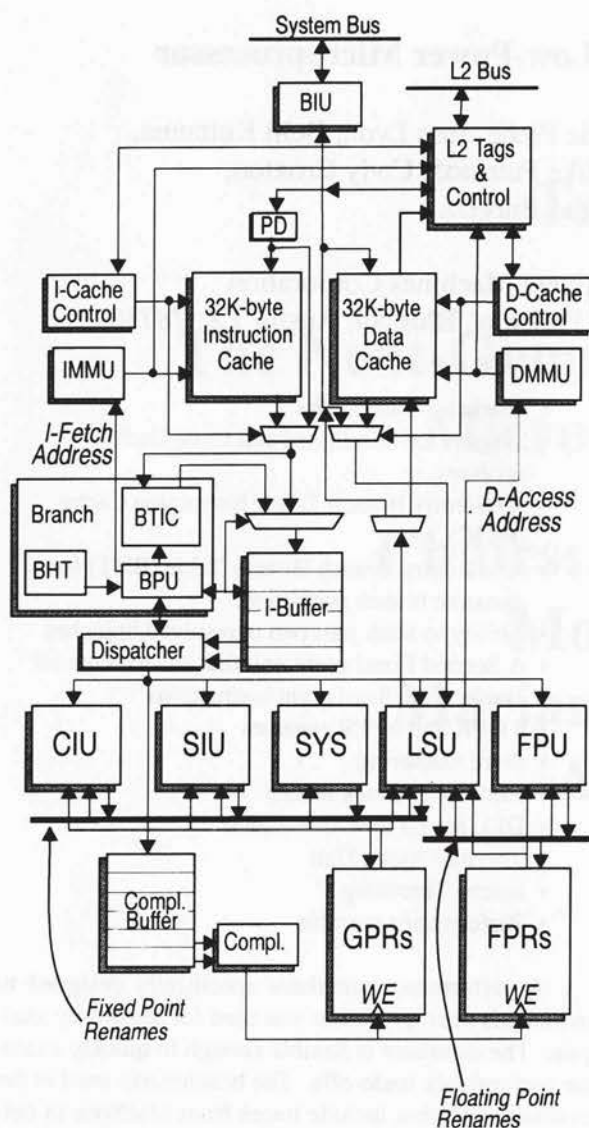


Figure 1: Block Diagram of this microprocessor.

the Bus Interface Unit (BIU). As instructions are being brought on board, the pre-decode (PD) block remaps their original 32-bit opcodes into a 36-bit form that allows for faster decode. The remapped opcodes are loaded into the instruction side L1 cache and the "top" of the Instruction buffer (I-Buffer) FIFO queue while the L2 cache receives the non-remapped opcodes. If a request misses the L1 cache but hits the L2 cache, the line that contains the requested instructions is brought into the L1 cache after being remapped by PD. If the requested instructions are in the L1 cache, the L1 cache can supply up to four instructions per cycle to the I-buffer.

Instructions in the I-buffer are eligible to be decoded and dispatched once they reach one of the bottom two slots. Branches however, are processed as soon as they enter the I-buffer and taken branches can be "folded" out before ever reaching the bottom.

Up to two instructions can be decoded and dispatched in a single cycle. Instructions are dispatched in-order to one of the five functional units; Complex Integer Unit (CIU), Simple Integer Unit (SIU), System Unit (SYS), Load/Store Unit (LSU) and Floating Point Unit (FPU); and are allowed to execute out-of-order. When instructions are dispatched, corresponding information is sent to the completion buffer. Completion logic keeps track of the outstanding instructions and retires them in-order. When the instruction is retired, its results update the architected registers.

When an instruction is dispatched to a functional unit, the unit is directed to put the results onto a specified target rename. (This microprocessor has 6 renames for integer results and another 6 renames for floating point results.) The dispatcher also tells the unit whether to get its source operands from one of the General Purpose Registers (GPRs), Floating Point Registers (FPRs) or the renames. For each source that is to come from a rename, the dispatcher supplies the unit with a tag which tells the unit which rename to get the source operand from. Each unit has at least one reservation station which holds the source operands until they are all available. Once the unit has all the needed operands, the unit executes the instruction. When the instruction finishes execution, the unit writes its result to the specified target rename. Upon retirement of the instruction, the result is taken off of the rename bus and written to the register file.

Load and store Instructions are sent to the LSU which calculates the effective address of the desired data and makes a request to the data cache. Data accesses that can not be satisfied by the caches are brought onto the processor via the BIU. As data is brought on board, it is loaded into the data side L1 cache and the L2 cache. Load data is forwarded to the LSU for writeback whereas store data is merged into the data stream as it is being loaded into the caches.

All caches are physically tagged. Separate I and D Memory Management Units (MMUs) each perform the effective to physical address translation for instruction and data accesses respectively.

3. Pipeline

As shown in Figure 2(a), this microprocessor contains a four stage pipeline: Fetch, Decode/Dispatch, Execute and Complete. Integer adds, subtracts, shifts

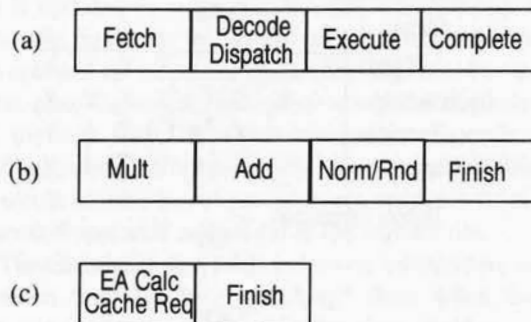


Figure 2: (a) Pipeline of this microprocessor, (b) FPU pipeline (c) LSU pipeline.

and logical instructions execute in a single cycle. Integer multiply and divides have multi-cycle latencies as shown in Table 1.

Instruction Type	max # of exec. cycles
8-bit multiply	2
16-bit multiply	3
24-bit multiply	4
32-bit multiply	5
divides	19

Table 1: Multi-Cycle Integer Unit Latencies

Figure 2(b) shows the pipeline for Floating point instructions. The result is driven on the rename bus during and is available for forwarding during the Norm/Rnd stage; the instruction does not complete however until the finish stage. Table 2 shows floating point operation latencies.

The pipeline for load/store instructions is shown in Figure 2(c). Load instructions which hit in the first level cache have a two cycle latency where as those that miss have a latency dependent on L2 hits and bus timings. LSU instructions can write back and complete in the final stage of the LSU pipe.

Instruction Type	# of exec. cycles
single precision mult-acc	3
double precision mult-acc	4
single precision divides	17
double precision divides	31

Table 2: Floating Point Latencies

4. Branch

The branch unit consists of the branch processing unit (BPU), a 512-entry branch history table (BHT), and a 64-entry, 4-way set associative branch target instruction cache (BTIC). Branches are recognized and loaded into the BPU as they are loaded into the I-buffer. Once in the BPU, the branch is processed and the branch target address is calculated. On taken branches, the branch is folded out by overwriting the branch in the I-Buffer with the target instruction.

Some branch instructions specify the value in either the Counter or the Link register to be used as the target address of the branch. (The PowerPC Architecture specification defines the Counter and Link registers as part of the architecture.[2]) However, there may be instructions that update the Counter or Link registers which have finished executing but have not yet updated the architectural state. For this reason, the BPU maintains a local copy of both the Counter and Link registers so it can select the branch target address from the local copy and not have to wait for the architected registers to be updated.

The Link register can be implicitly updated with "branch-and-link" instructions. For these instructions, if the branch is to be taken, the address of the instruction that would have executed next is saved in the link register while the branch target is fetched. Therefore, branch-and-link instructions can be used as subroutine calls since the return address will be saved in the link register and the link register can be used to supply a branch target address.

This microprocessor can employ either static or dynamic branch prediction for conditional branches that are dependent on a condition which has not yet resolved. (Software selects whether static or dynamic prediction is used). For static prediction, hints are provided in the branch opcodes to specify that a particular branch is to be predicted taken or not-taken. For dynamic predic-

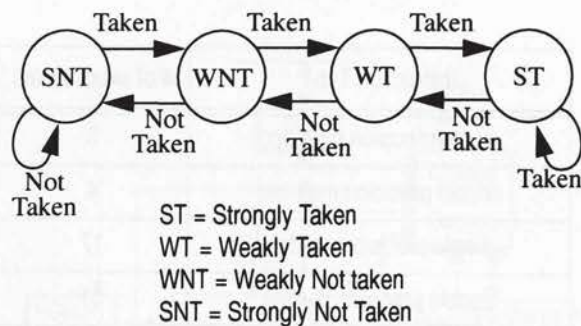


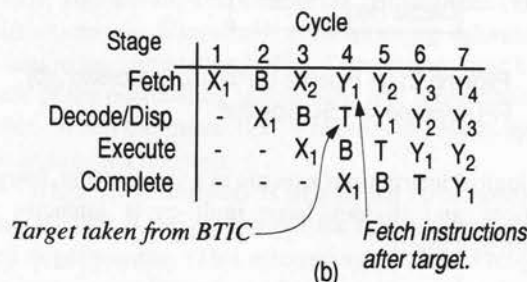
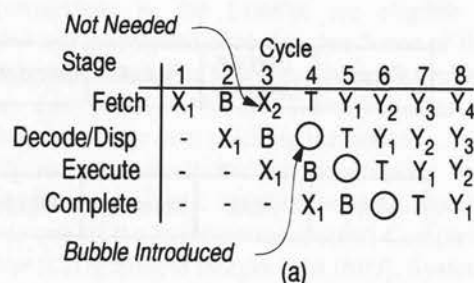
Figure 3: State transition diagram for the 613's dynamic branch prediction.

tion, bits 19-29¹ of the address of the branch are used to look up a two bit history from the BHT which gives one of four history states: strongly taken, weakly taken, weakly not-taken, strongly not-taken. Branch histories with states of strongly taken or weakly taken are predicted taken; those with histories of weakly not-taken or strongly not-taken are predicted not taken. States are updated when the branch resolves using the familiar Lee and Smith[3] algorithm shown in Figure 3.

In this microprocessor's pipeline, a taken branch may cause fetched instructions to be discarded. Normally, this would create a bubble in the pipeline since an extra cycle would be required to get the target of the branch into the decode stage (see Figure 4(a)). In this microprocessor, this bubble is eliminated by using a BTIC which caches the *instructions* at the target of the branch. Figure 4(b) shows how the bubble is eliminated.

The first time a branch resolves taken, the BTIC is loaded with the branch's target instructions as they are brought into the I-buffer from the instruction cache. If the branch gets re-executed, its address is used to reference the BTIC which can supply up to two instructions: the branch's target instruction and the instruction after the target. When instructions are taken from the BTIC, the next fetch address given to the instruction cache by the branch unit is for the instructions after the one(s) supplied by the BTIC.

This microprocessor allows for speculative execution down a predicted path beyond one unresolved branch. The completion unit keeps track of which instructions in execution are speculative and does not allow them to complete. If it is determined the prediction is correct, the instructions that are marked speculative are made non-speculative and execution continues as normal. If the branch is mis-predicted and specula-



X_n - Sequential instructions
 B - Branch
 T - Target of branch
 Y_n - Sequential Instructions after branch target

Figure 4: Fetching of a branch target. (a) Without BTIC, bubble is introduced. (b) This microprocessor uses a BTIC to eliminate the bubble.

tive instructions have been executed, the branch unit waits for all non-speculative instructions to complete and then, in a single cycle, flushes all the speculative instructions out of the machine.

This microprocessor allows for speculative instruction *fetching* beyond two unresolved branches. This means that when a second unresolved branch is encountered, the microprocessor will continue to speculatively fetch down the second predicted branch path. However, none of these instructions will be allowed to execute unless the first unresolved branch resolves correctly.

5. Dispatch

Instructions are decoded and dispatched in-order from the bottom two slots of the I-buffer. An instruction can be dispatched if:

- it is to a unit which is not busy
- there are enough renames for the instruction's result(s)
- the completion buffer is not full.
- dispatching the instruction does not initiate execution down a second speculative path.

1. PowerPC Architecture specification defines the MSB as bit 0 and the LSB as bit 31.

Since execution is out-of-order but the architectural state is updated in-order (i.e. in-order completion), an instruction can have its results on the rename but not have updated the targeted architectural register. For this reason, the dispatcher maintains a register-to-rename map to keep track of which rename corresponds to which register. At dispatch time, this mapping tells a unit which rename bus to get its source operands from if the needed operands are not yet in the register file.

The dispatcher also folds out no-op instructions and non-taken branches by "squashing" them when they reach one of the bottom two I-buffer slots. Folding no-ops enhances performance by keeping them from using functional units and completion buffer resources.

6. Fixed Point Units

This microprocessor has two integer units: a Complex Integer unit (CIU) and a Simple Integer Unit (SIU). Both units are able to execute all fixed point shift, rotate, logical, add and compare instructions in a single cycle. The CIU can also perform multi-cycle multiply and divide operations.

Operands are latched into the unit's reservation station during dispatch and execution starts when all the operands are ready. The reservation station receives inputs from the GPRs and rename buses. All results are written back to the rename buses. The CIU's multiply array is 32 X 8 bits so multiplication takes from one to four clock cycles to execute dependent on the operand width. Integer divides take 19 clock cycles to complete.

Either of the two separate integer units can update the carry bit. The two units can execute out-of-order but the carry value must be updated and used as if execution were in order. The typical approach to such a problem is to provide a shadow copy of the carry, which can be difficult to maintain. Instead, instructions that use the carry bit are kept in the reservation station until the correct carry value is in the architected state. With this solution, the unit can always get the carry bit from the architected register. This has almost no effect on performance because instructions that use the carry bit are very infrequent.

7. Floating Point Unit

This microprocessor includes a floating point unit (FPU), compliant with the *IEEE 754 Standard for Binary Floating-point Arithmetic*, that executes all PowerPC floating-point single and double-precision instructions, including three of the optional instructions for graphics applications. The FPU also includes a non-IEEE mode for improved performance in applications

not requiring denormalized results; in this mode a result requiring denormalization is forced to zero.

The FPU contains three pipeline stages: the MULT stage, the ADD stage, and the Normalize/Round stage. The MULT stage contains a 53x28 bit Booth-encoded Wallace tree, a right shifter and a final 3-to-2 carry-save adder. The shifter and adder perform the proper alignment and addition of an add-type instruction's second operand or a multiply-add type instruction's third operand. Double precision multiply and multiply-add instructions require two passes through the MULT stage while single-precision multiplies and multiply-adds require only one pass. Multiply, add, and subtract instructions which only require two source operands force the third unused operand to a 1 or 0 value as necessary. For example, an add instruction ($A+B$) forces the C operand to a 1 value thereby performing the equivalent operation $((Ax1) + B)$. Likewise a multiply instruction (AxC) forces the B operand to a 0 value thereby performing the equivalent operation $((Ax0) + 0)$.

The ADD stage combines the final partial products of the MULT stage with the use of a combination of carry look-ahead adders (CLAs) and incrementers. The incrementers are used in place of full CLAs for data residing beyond the boundaries of the partial products exiting the MULT stage (such as add operands that have not been fully shifted into or that have been shifted beyond the multiply- or second add-operand-range, in addition to their use in multiple pass operations). A leading zero detector in this stage determines the proper shift-amount used for normalization in the next stage.

The Normalize/Round stage contains a 63-bit normalizer. In most cases, normalization only requires one pass through this stage. For instances when mass cancellation occurs, more than one normalization pass may be required. In addition, IEEE rounding is performed and floating-point status is generated here. The result from the Normalize/Round stage is placed on one of the six rename buses. This data can be used directly from the rename bus the cycle after execution.

The completion unit is notified that the instruction has finished in the cycle following the Normalize/Round stage at which time it can transfer the rename bus data to the architected FPR upon completion of the instruction.

Floating-point, divide-type operations such as single- and double-precision divides and the floating-point reciprocal estimate instruction utilize a non-restoring algorithm (two mantissa bits are generated per clock cycle). Double-precision divides have a latency of 31 execute cycles, single-precision divides have a latency of 17 cycles, and floating-point reciprocal estimate instructions have a latency of 10 cycles.

8. System Unit

The system unit is responsible for moves to and from Special Purpose Registers and for logical operations within the Condition Register. It also contains:

- The Decrementer - A 32-bit counter which is decremented every bus-clock cycle and allows for interrupt generation when the count decrements to zero.
- The Time Base register - A 64-bit counter which increments every bus-clock cycle and provides a mechanism for calculating elapsed time.
- Performance monitor - Four counters which can be specified to count various events such as cache hits and correctly predicted branches. This information can be useful in tuning the system software to achieve higher performance.

9. Completion

The completion buffer is a six-entry FIFO which receives information about instructions as they are dispatched. After dispatch, the completion buffer keeps track of when the instructions finish, which rename (if any) corresponds to the instruction's architected target register, and when instructions are allowed to write their result to the register file (i.e. complete).

Instructions can be completed when they reach one of the bottom two slots of the buffer. Instructions complete if:

- the instruction is finished.
- all instructions before it have completed or are allowed to complete (in-order completion)
- it will not cause more than two GPR writes.
- it will not cause an exception to be reported.
- it is non-speculative.

Except for these restrictions, the completion buffer can complete two instructions on any given cycle. If two instructions eligible for completion update the same GPR, FPR, or CR field, they both are allowed to complete but only the latest one updates the register file.

10. Exception

When an exception occurs, hardware saves the address of either the next instruction to complete (for synchronous interrupts) or the last instruction to complete (for asynchronous interrupts) and gives control to the exception handler. Even though instruction execution is out-of-order, this microprocessor supports a precise exception model so that when control is passed to the exception handler, the architectural state of the machine appears as if instructions were executed in-

order. This is done by halting completion which keeps the instructions after the one that triggered the exception from updating the architected registers.

Exception logic determines what type of exception needs to be reported and directs the branch/fetch unit to go to the appropriate exception vector. If multiple exceptions occur at the same time, the exception logic is responsible for reporting only the one with highest priority. In some cases exceptions can be layered such that if multiple exceptions occur, they must all be reported and handled in order of priority.

11. Load/Store

This microprocessor has a single Load Store Unit (LSU) that executes the fixed- and floating-point load/store instructions. It calculates the 32 bit Effective Address and makes a request to the MMU for address translation and to detect any exception conditions. On an exception, the LSU reports the condition to the completion unit.

The LSU allows for speculative load accesses. These are accesses which may be flushed later due to execution down a mis-predicted path or from a previous instruction having an exception. The data returned from the cache is held on a rename bus until the completion unit allows the data to write the register file.

Stores are not allowed to execute speculatively. The LSU holds the address of the store in a queue until it is safe to commit the store. The LSU then gets the store data directly from the GPR and makes a request to the cache to execute the access.

Store gathering is performed when multiple word size stores in the store queue access the same double-word. This allows for multiple stores to be written in a single cache access.

For floating point loads, the LSU generates a tag which indicates the data type. These tags are used by the floating point unit for data type detection. The LSU also handles the conversion of single-precision data from memory into the internal double-precision format for single-precision loads and visa-versa for single-precision stores.

This microprocessor's LSU can allow accesses to be performed out of order in the case of a hit-under-miss or a load bypassing a store. It can also execute mis-aligned accesses within a double-word boundary in a single cycle and allows misaligned Little Endian accesses.

The LSU can finish one load instruction and one store instruction in the same cycle. It also ensures correct order for load and store instructions to cache inhibited and guarded space and handles all the cache

management instructions.

12. Memory Sub-System

12.1. L1 Caches

The L1 instruction and data caches are split, 8-way set associative and 32K bytes each in size. Each data cache line is 32 bytes long while each instruction cache line contains 8 pre-decoded instructions, 36-bits each in size. Burst reads are performed into both caches in four beats. A burst read is performed critical double-word first with the critical double-word being forwarded to the requesting unit while simultaneously being written into the cache. On a fetch that hits in the instruction cache, a maximum of four instructions can be supplied to the sequencer. On a fetch that misses the instruction cache, the instructions are "streamed" into the I-buffer in parallel with the cache write.

Both caches are non-blocking. This means that the caches can continuously service requests that hit in the cache while a miss is pending. An instruction side hit-under-miss happens when the instruction cache has a miss pending and then the sequencer signals a change of stream (due to a branch or interrupt) and gives a new fetch address which hits in the cache. A pseudo least recently used (PLRU) algorithm is used by both caches for cache line replacement. A line is marked as the most recently used whenever it is accessed. The line chosen for replacement by the PLRU algorithm is replaced with new data on a cache miss when all 8 lines in the set are valid. In the data cache, if the line being replaced contains modified data, that data is written back to the L2 cache or main memory before being replaced.

Memory coherency is maintained by the data cache by using a three state MEI (exclusive modified, exclusive unmodified and invalid) protocol that is a coherent subset of the standard four state MESI protocol. Bus snooping adheres to this protocol. Coherency mechanisms for L2 caches are supported. No sharing of data is permitted by marking all burst reads as if they were "reads with intent to modify". This flushes all external caches prior to the burst read, allowing the processor to be the sole owner of the data once the burst read is completed. A snoop access into the single-ported data cache tags has higher priority than a load, store or tablewalk access. However, if the snoop access coincides with a tag write, which may be due to validation for a burst read, the snoop is retried.

The enabling and disabling of the caches are controlled through writes to a control register. When the caches are disabled, all accesses are sent to the bus interface unit (BIU) for handling. Snooping is not performed

when the data cache is disabled. Invalidation of the entire Instruction and data caches is also permitted through a control register. During a hard reset (power-up), both caches are invalidated and are disabled. The caches can also be "locked". A locked cache supplies data normally on a hit, but cache lines are not replaced on a miss.

Word-line redundancy hardware exists to replace inoperable bit-cells or word-lines at manufacturing test, thereby increasing effective yield.

12.2. L2 Cache

This microprocessor contains an on-chip L2 cache controller which provides fast access to off-chip secondary cache. It also serves to further improve system performance by reducing system bus traffic because a bus request is issued only when the access misses in the L2.

The L2 cache controller includes 8-Kbyte of tag which supports 256K, 512K or 1M of external, 2-way set associative, unified (instruction and data), L2 cache. The tags are set up to support either a two sector cache line (a sector being 32-bytes) for a 256K or 512K L2 cache, or a four sector cache line for a 1M L2 cache. Each tag entry has its own valid and modified status bits to support copyback mode and system cache coherency (the L2 services snoop requests from the PowerPC 60x bus).

An access that misses the instruction or data L1 cache is sent to the L2 cache controller for a tag look-up. If it is determined that the request hits in the L2 cache, the L2 returns an L1 cache line (32-bytes) which contains the requested data. If the L2 tag look-up misses, the request is forwarded on to the PowerPC 60x bus interface for memory access.

The L2 may have to service up to four requests at any one time: one from the instruction cache, one load request and two store requests. The L2 cache is a non-blocking cache which supports hit-under-miss (but not miss-under-miss).

This microprocessor supports three L2 SRAM types:

- *flow-through* (register-buffer) synchronous burst SRAMs which clock addresses in and flow data out
- *pipelined* (register-register) synchronous burst SRAMs which clock addresses in and clock data out
- *late-write* synchronous SRAMs which are required by the microprocessor to be a pipelined (register-register) configuration.

The settings of the L2 enable, L2 size, L2 clock ratio, L2 RAM type, and several L2 test support features

are configured in an L2 cache control register. This microprocessor generates the clock for the external L2 synchronous data RAMs. The clock frequency for the RAMs is divided down from this microprocessor core clock frequency. Generally, the divisor must be chosen according to the frequency supported by the external RAMs, the frequency of the internal core, and the phase adjustment range that the L2 delay-lock loop (DLL) supports. The L2 RAM frequency can be divided down by 1, 1.5, 2, 2.5, or 3 from the core operating frequency.

Several L2 test support features are provided to facilitate testing of the L2 cache. Testing involves performing a general memory test on the external data RAMs, performing a general memory test on the on-chip tag RAMs, or performing a general test of the overall L2 cache operation.

12.3. MMUs

This microprocessor implements a virtual memory management scheme compliant with the PowerPC Architecture specification for 32-bit microprocessors. It has dual instruction and data MMUs and supports up to 2^{52} bytes of virtual memory and 2^{32} bytes of physical memory.

The organization of the MMU hardware consists of:

- 128-entry, two way set-associative Instruction and Data Translation Lookaside Buffers (TLBs).
- 4 entry, fully associative Instruction and Data Block Address Translation (BAT) buffers.
- 16 segment registers.
- Hardware Tablewalk updates for TLB misses.

This microprocessor supports two types of address translation:

- *page address translation* - translation through segment registers and TLBs.
- *block address translation* - translation through BAT registers.

The BAT address translation takes priority over the page address translation in the event of dual hit. The BAT mechanism provides a way to map ranges of effective address larger than a single page into contiguous areas of physical memory. Such areas can be used for data that is not subject to normal virtual memory handling (paging), such as a memory-mapped display buffer or an extremely large array of numerical data.

The instruction and data TLBs provide page address translation in parallel with the on-chip cache access. The TLBs contain the most recently used Page Table Entries (PTEs) which are replaced based on an LRU algorithm. The segment registers provide a 24-bit

virtual segment ID which is used to extend the 32-bit effective address space to a 52-bit virtual address space. The segment registers allow the programmer to partition the effective address space into 16, 256-Mbyte blocks which are then mapped anywhere within the terabyte (2^{52}) address space. The 52-bit virtual address is then looked up in the TLB for the required PTE. The physical address is then extracted from the PTE. If the required PTE is not resident in the TLB, the microprocessor performs a hardware table walk to reload the TLB with the corresponding PTE entry from the memory. Hardware tablewalk provides significant improvement in tablewalk performance than a software tablewalk.

12.4. BIU

This microprocessor's BIU implements the PowerPC 60x bus protocol for commonality and ease of migration. The bus is a non-multiplexed bus providing 32-bits of address plus address parity and other transfer attributes, 64-bits of data plus parity, and separate address and data bus control signals. The separate address and data bus controls allow the address and data busses to be operated in a nearly independent fashion allowing advanced bus features such as address pipelining, split transactions, and enveloped snoop push transactions to be implemented by the system. In addition, the data bus size may be configured at power-up to support either 64-bit data bus operation, or 32-bit data bus operation. These features allow for a wide range of system implementations depending on cost and performance goals, from simple systems that do not require these features, to advanced pipelined, fully coherent, multi-cache environments.

Due to the advanced features of the bus protocol and the independent nature of the L1 and L2 caches, the BIU implements several address and data queues. This allows several memory transactions to be outstanding at any given time. The BIU supports 1-level address pipelining so that one more address may be initiated on the bus before the data tenure for the current address has completed or started. The BIU for the microprocessor is generally strictly ordered (data tenures run in the same order as address tenures), however, the BIU also supports limited out-of-order capability in which a snoop push data tenure may be run ahead of an already pipelined read address from this BIU (i.e., enveloping).

Bus snooping is supported to provide hardware cache coherency between the microprocessor's on-chip caches, system memory, and other caching bus masters. Snooping may be selected or not selected on a page basis through the "Global" address attribute, which

is usually reflected from the "M" bit of the MMU address translation tables. On a snoop hit, the microprocessor may "retry" the other bus master (or vice-versa), and then perform a write-back of the modified cache line to system memory so that the original memory request can continue.

Also associated with the bus interface unit are the clock multipliers implemented by the on-chip PLL (Phase Lock Loop). The PLL generates the primary clock for the internal circuitry by multiplying and phase-aligning an internal clock to the external system clock input (SYSCLK). The clock multiplier determines the internal clock frequency with respect to the external bus clock. This microprocessor supports clock multipliers from 2x (core at two times the bus frequency) to 7x, with all integer and half-integer ratios in-between (2x, 2.5x, 3x, 3.5x, etc.). This wide range of clock multipliers allows the microprocessor core to be run at its highest frequency possible with respect to a given bus frequency which may be fixed by other system requirements. This also allows easier upgrades to higher core frequencies as they become available without changing the external system.

13. Performance

The performance evaluation and sensitivity analysis was done using a high-level performance simulator made specifically for this microprocessor. The simulator is referred to as an "ADvanced Alterable Pipeline Timer" (or "ADAPT") [4] and is what guided many of the design decisions and trade-offs. ADAPT is designed in such a way that various parameters (such as the number of instructions fetched per cycle, cache sizes, number of renames, etc.) can be changed easily and simulation runs can be executed extremely fast. This allows for several experiments to be run on entire Spec92, Spec95 and MacSpec suites in a relatively short period of time. (ADAPT running on an RS/6000™ workstation can simulate a full SpecInt95 suite run with 100 million instructions per benchmark in < 15hrs.)

An example finding of the experiments run is that if the number of functional units are kept constant but the number of instructions that can be dispatched and completed in a cycle is allowed to double (from 2 to 4), there is no significant gain in performance. Conversely, an additional fixed point unit would be starved unless the dispatch and completion width is increased. Also increasing the number of I-Buffers, completion buffers and renames does nothing to significantly improve performance.

A non-intuitive result is that for this microprocessor, out-of-order completion only increases performance

by 2-3%. However the complexity involved in allowing for out-of-order completion far out-weighs the increased performance. More importantly, allowing out-of-order completion would have created speed paths that would have made attaining the speed goals difficult.

The results from the ADAPT simulator show that features that would seem to give large improvements in performance do not always achieve as large a gain as intuition would suggest. The number of variables that determine the overall performance in a complex superscalar, pipelined microprocessor like this one are so great that a feature which is thought to vastly improve performance must be modeled in a performance simulator to determine the actual performance gain.

Verifying that ADAPT actually correlates to the functional models is done by running large code traces on both the simulator and the models and insuring that they both give close to the same IPC. Since ADAPT is not an execution based simulator, it cannot account for performance differences that are dependent on things like data inputs or results. When precise performance modeling can not be done, ADAPT is made to be conservative. Correlations indeed show that the performance numbers given by ADAPT are conservative compared to the actual functional models.

14. Power Management

This microprocessor has dynamic power management built into the hardware which turns off clocks to latches that are not in use. This is done without any degradation to performance since the clocks can be restarted as soon as the latch is needed.

As with many of the other PowerPC microprocessors, this microprocessor includes three low power modes: doze, nap and sleep. In doze mode, most of the functional units are turned off but cache coherency is maintained by keeping the BIU and snooping enabled. In nap mode, the microprocessor is disabled but the phase locked loop (PLL) stays on and the time base/decrementer can still be enabled. In sleep mode, (which saves the most power), the PLL may be disabled and the system clock to the processor may be removed.

A unique feature introduced in this microprocessor is the Thermal Assist Unit (TAU). This is an on-chip thermometer which can signal an interrupt when the temperature exceeds a software specified level. This gives the programmer the ability to take power saving steps when the temperature exceeds a certain threshold.

One power saving feature that software can evoke is I-Cache Throttling. I-Cache Throttling "slows down" the rate at which the I-Cache provides instructions to the dispatcher. This helps to reduce the average power con-

sumption by increasing the amount of time the functional units sit idle. Once the part temperature has gone below a user-specified threshold, I-Cache throttling can be made to automatically disable.

15. Verification

Most of this microprocessor functionality was verified using two methods:

- writing and generating tests which specifically targeted a particular unit or function of the chip.
- running randomly generated tests on the functional models for billions of cycles.

16. Technology

Table 3 lists the technology information of the microprocessor [5]. The die photo is given in Figure 5.

17. Conclusion

This third generation PowerPC microprocessor, a member of the G3 family of PowerPC processors, offers both the high performance required for high end desktop systems, as well as the low typical power dissipation needed for portable systems.

18. Acknowledgment

Thanks to everyone involved in the design and manufacturing effort of this microprocessor.

Technology	n-well CMOS p-/p+ epi sub
Isolation	shallow trench
Min Feature Size	0.25 μ m
Gate Oxide	5nm
L_{eff}	0.18 μ m
Metal Layers	5 + local interconnect
Die Size	7.6x8.8mm
Transistors	6.35 million
6T Bitcell Size	3.5x5.25 μ m

Table 3: Technology Features

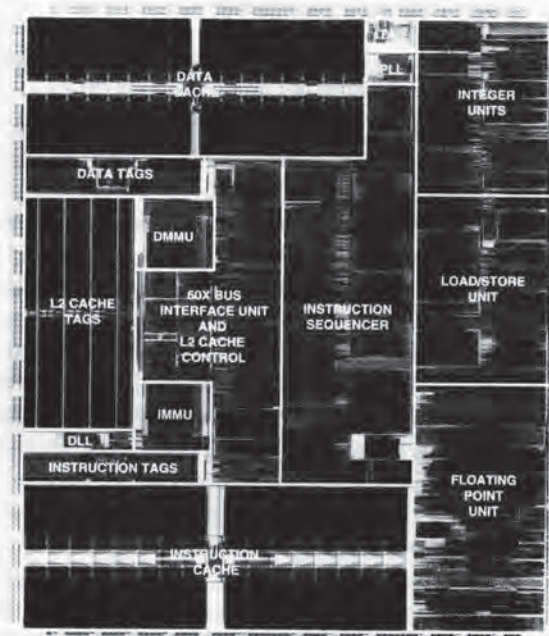


Figure 5: Die Photo of the microprocessor.

19. References

- [1] Slaton, J., S. Litch, et al, "The PowerPCTM 603e Microprocessor: An Enhanced, Low-Power, Superscalar Microprocessor" *Proceedings of the International Conference on Computer Design* 1995, October 1995.
- [2] IBM Corporation, *The PowerPC ArchitectureTM; A specification for a new family of RISC processors*; Morgan Kaufmann Inc., 1993.
- [3] Lee, J.K.F., A.J. Smith, "Branch Prediction Strategies and Branch Target Buffer Design," *Computer*, Jan 1984, pp. 6-22.
- [4] Burgess, B., Litch, S., "ADvanced Alterable Pipeline Timer (ADAPT): A Tool to Design a High Performance PowerPCTM Microprocessor" 1997 *IEEE International Performance, Computing, and Communications Conference*, February 1997.
- [5] Reed, P. et al, "A 250MHz 5W RISC Microprocessor with On-Chip Cache Controller", *Proceedings of the International Solid-State Circuits Conference* 1997, February 1997.

PowerPC, PowerPC Architecture, PowerPC 60x, RS/6000 and IBM are trademarks of International Business Machines Corporation in the United States or other countries or both. **SPEC92, SPEC95 and SPECint95** are trademarks of the Standard Performance Evaluation Council. **MacOS and MacSpec** are trademarks of Apple Computer Corporation. Other company, product and service names may be trademarks or service marks of others.