JULY 1998

WRL
Technical Note TN-55

The Memory Daughter-Card Version 1.5 User's Manual

Marc A. Viredaz

digital **Western Research Laboratory** 250 University Avenue Palo Alto, California 94301 USA

The Western Research Laboratory (WRL) is a computer systems research group that was founded by Digital Equipment Corporation in 1982. Our focus is computer science research relevant to the design and application of high performance scientific computers. We test our ideas by designing, building, and using real systems. The systems we build are research prototypes; they are not intended to become products.

There are two other research laboratories located in Palo Alto, the Network Systems Lab (NSL) and the Systems Research Center (SRC). Another Digital research group is located in Cambridge, Massachusetts (CRL).

Our research is directed towards mainstream high-performance computer systems. Our prototypes are intended to foreshadow the future computing environments used by many Digital customers. The long-term goal of WRL is to aid and accelerate the development of high-performance uni- and multi-processors. The research projects within WRL will address various aspects of high-performance computing.

We believe that significant advances in computer systems do not come from any single technological advance. Technologies, both hardware and software, do not all advance at the same pace. System design is the art of composing systems which use each level of technology in an appropriate balance. A major advance in overall system performance will require reexamination of all aspects of the system.

We do work in the design, fabrication and packaging of hardware; language processing and scaling issues in system software design; and the exploration of new applications areas that are opening up with the advent of higher performance systems. Researchers at WRL cooperate closely and move freely among the various levels of system design. This allows us to explore a wide range of tradeoffs to meet system goals.

We publish the results of our work in a variety of journals, conferences, research reports, and technical notes. This document is a technical note. We use this form for rapid distribution of technical material. Usually this represents research in progress. Research reports are normally accounts of completed research and may include material from earlier technical notes.

Research reports and technical notes may be ordered from us. You may mail your order to:

Technical Report Distribution
DEC Western Research Laboratory, WRL-2
250 University Avenue
Palo Alto, California 94301 USA

Reports and technical notes may also be ordered by electronic mail. Use one of the following addresses:

Digital E-net:	`JOVE::WRL-TECHREPORTS`
Internet:	`WRL-Techreports@decwrl.pa.dec.com`
UUCP:	`decpa!wrl-techreports`

To obtain more details on ordering by electronic mail, send a message to one of these addresses with the word ‘‘`help`’’ in the Subject line; you will receive detailed instructions.

Reports and technical notes may also be accessed via the World Wide Web: `http://www.research.digital.com/wrl/home.html`.

The Memory Daughter-Card Version 1.5
User's Manual

Marc A. Viredaz

July 1998

Abstract

The *memory daughter-card* is memory extension board for the *Itsy pocket computer* developed at Compaq Computer Corporation's *Western Research Laboratory (WRL)*. It provides an additional bank of flash memory and up to three additional banks of DRAM. It is possible to boot from this daughter-card. This document describes the architecture and the low-level programming model of the memory daughter-card.

Revision 1.0

Contents

List of Figures

List of Tables

Disclaimer

1 Introduction

This document describes the architecture and the programmer's model of the *memory daughter-card* for the *Itsy pocket computer* [Vir98]. It should be considered as a guide for low-level software developers.

A good understanding of the Itsy computer [Vir98] is assumed throughout this report.

1.1 Purpose

The design of the memory daughter-card was aimed at achieving several goals:

- Provide additional flash memory and DRAM (memory extension).
- Be usable as a "safety daughter-card," i.e., it should be possible to boot from the daughter-card, in order to recover from a corrupted mother-board flash memory.
- Be usable as a system test and debug platform, i.e., it should be possible to boot from the daughter-card and use sockets for the flash memory, so that it can be programmed with an external programmer.
- Provide a prototyping platform to interface additional hardware to the Itsy computer.

1.2 History

The first *printed-circuit board (PCB)*, referred to as *memory daughter-card version 1.0*, was completed in November 1997. This first prototype had a few minor flaws, all of which could be corrected. The logic design corresponding to a modified (i.e., patched) version 1.0 board is known as *memory daughter-card version 1.1*. This design corresponds only to a set of schematics, no physical PCB having been manufactured.

A second prototype, named *memory daughter-card version 1.5* was complete in July 1998. It corresponds to a version 1.1 system with a few additional features. From the programmer's point-of-view, there are almost no differences between the versions 1.1 and 1.5.

This document describes the memory daughter-card version 1.5. All relevant differences between the versions 1.1 and 1.5 are outlined in foot-notes.

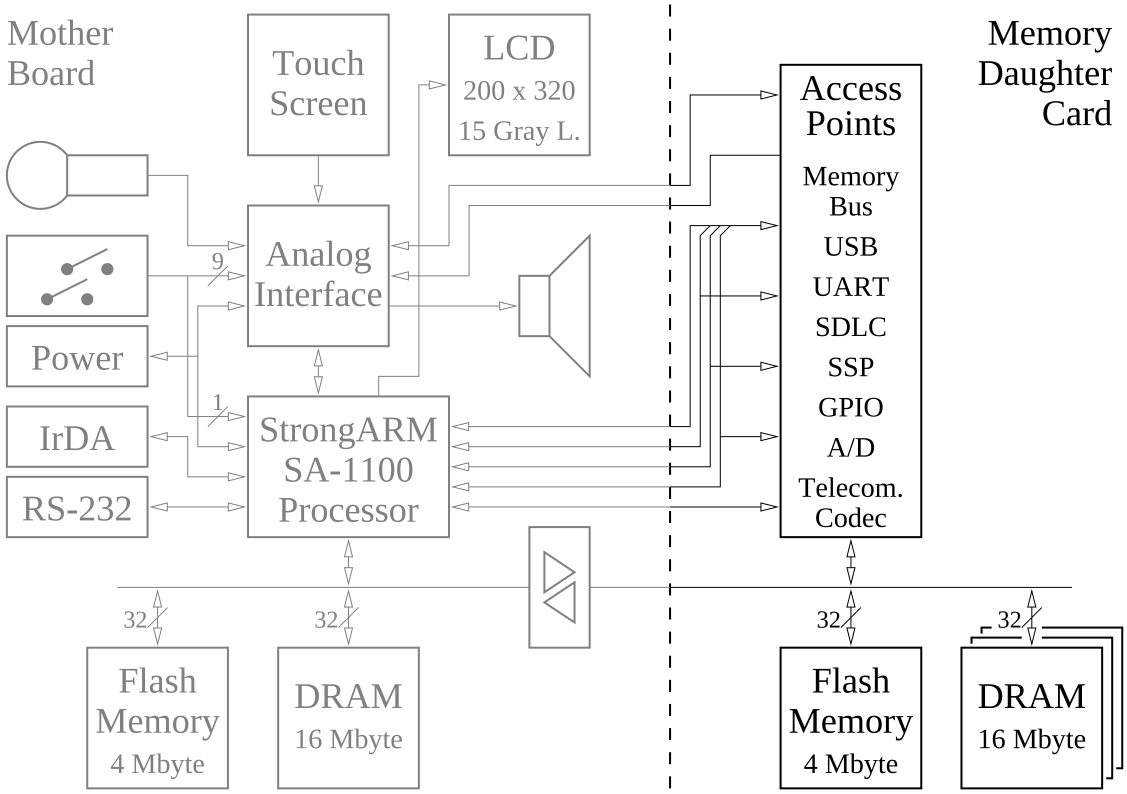

Figure 1: Architecture of the memory daughter-card.

1.3 Notations

In this report, electrical signals are represented as upper-case names in a sans-serif font (e.g., PWR_EN). Active-low signals are denoted by over-lines (e.g., $\overline{\mathsf{RESET\_OUT}}$), while buses and element of buses are specified by subscripts (e.g., $\mathsf{DCD}_{31..0}$, DCA_0). In the schematics, the same signals are represented using the syntax and conventions of the WindowSIL [Tha97] CAD tools. For example, the signal $\overline{\mathsf{DCCS}}_2$ appears as `~dccs[2]`.

2 Architecture

Figure 1 presents the architecture of the memory daughter-card (right part), shown in respect to the Itsy computer (left part). The memory daughter-card versions 1.1 and 1.5 are both compatible with both versions of the Itsy mother-board (i.e., versions 1.1 and 1.5).

2.1 Memory system

The memory daughter-card features a *flash memory* decoded as static-memory bank 2 and the three *dynamic random-access memory (DRAM)* banks 1, 2, and 3. Depending on the application's needs, only some of these banks (or even none) might be present. Since this daughter-card is bootable, the flash memory is also decoded as static-memory bank 0, from which the StrongARM SA-1100 processor [DEC98] boots.

2.1.1 Flash memory

A pair of 16-bit flash-memory circuits implement the 32-bit flash memory. These circuits can be either soldered on the board or put in *zero insertion force (ZIF) sockets*. Many different devices can be accommodated:

AMD Am29LV160B$OssPT$ [AMD97e]
AMD Am29LV800O-$ssPT$/Am29LV800B$OssPT$ [AMD97c, AMD97d]
AMD Am29LV400O-$ssPT$ [AMD97a]
AMD Am29LV200O-$ssPT$ [AMD97b]
Hitachi HN29VO800P-ss/HN29WO800P-ss [Hit97c, Hit97d]
Motorola M29F800A2$OPss$/M29F800A3$OPss$ [Mot97]
Sharp LH28F800SGP-Lss [Sha97]

where "O" specifies the internal sector organization, "ss" specifies the speed, "P" specifies the package, and "T" specifies the temperature range. Any other compatible parts can also be used.

Following the Itsy static-memory identification scheme [Vir98], a non-volatile memory identification structure describes the characteristics of the specific parts used on a given daughter-card and hence allows the software to configure the memory interface correctly.

The *reset/power-down pin* of the flash-memory circuits is asserted (0) during a reset (i.e., hardware, software, or watch-dog reset) and during sleep mode. When the flash memory does not need to be accessed, this signal can also be asserted (0) by setting the signal GPIO_{16} ($\overline{\mathsf{DCFLFOFF}}$) to 0 (see Section 3.1).[1] When the Itsy computer must boot from the daughter-card, this signal should never be set to 0 during sleep mode, since the processor would be unable to read the boot memory upon wake up. This can easily be achieved by setting bit 16 of the *power manager GPIO sleep state register* PGSR of the StrongARM SA-1100 processor [DEC98] to 1.

The *ready/not-busy pins* of the flash-memory circuits can be monitored using the signals GPIO_{17} ($\mathsf{DCFL0RY}/\overline{\mathsf{BY}}$), for the least significant 16 data bits $\mathsf{D}_{15..0}$, and GPIO_{18} ($\mathsf{DCFL1RY}/\overline{\mathsf{BY}}$), for the most significant 16 data bits $\mathsf{D}_{31..16}$ (see Section 3.1).

The hardware write-protection mechanism, featured by some of the supported parts, is never used, and the corresponding pin is always de-asserted (1). However, it is still possible to protect the flash memory against write accesses, by using the jumper provided for this purpose. The flash memory is protected when this jumper is open and is writable when this jumper is short-circuited.

2.1.2 Dynamic RAM

Three pairs of 64 Mbit (i.e., 2^{12} rows $\times$ 2^{10} columns $\times$ 16 bits) self-refresh DRAM circuits implement the three 32-bit DRAM banks. Many different *fast-page mode* or *enhanced data out (EDO)* devices can be accommodated:

[1] On the memory daughter-card version 1.1, the reset/power-down pins are only asserted (0) during sleep mode or when the signal GPIO_{16} ($\overline{\mathsf{DCFLFOFF}}$) is set to 0.

Signal	Name	Function	Dir.	Def.
GPIO_{16}	$\overline{\mathsf{DCFLFOFF}}$	**D**aughter-**c**ard **fl**ash memory **f**orce **off**	O	1
GPIO_{17}	$\mathsf{DCFL0RY}/\overline{\mathsf{BY}}$	**D**aughter-**c**ard **fl**ash memory **0** **r**ead**y**/**b**us**y**	I	
GPIO_{18}	$\mathsf{DCFL1RY}/\overline{\mathsf{BY}}$	**D**aughter-**c**ard **fl**ash memory **1** **r**ead**y**/**b**us**y**	I	

Table 1: Memory daughter-card general-purpose input/output signals.

Hitachi HM5165160ALTT-*ss* [Hit97a]
Hitachi HM5165165ALTT-*ss* [Hit97b]
Samsung KM416V4100AS-L*ss*/KM416V4100BS-L*ss* [Sam97a, Sam98a]
Samsung KM416V4104AS-L*ss*/KM416V4104BS-L*ss* [Sam97b, Sam98b]
Toshiba TC5165165AFTS-*ss* [Tos96]

where "*ss*" specifies the speed. Any other compatible parts can also be used.

Since all DRAM banks must be accessed at the speed of the slowest one, it is best to use the same devices as on the Itsy mother-board, that is, 50 ns EDO DRAM circuits (KM416V4104AS-L5, KM416V4104BS-L5, or TC5165165AFTS-50).

3 Programmer's model

This section presents additional information on the model that the low-level software has of the memory daughter-card hardware.

3.1 Memory daughter-card general-purpose input/output signals

Table 1 shows the general-purpose input/output signals used on the memory daughter-card. Both input signals can be used as interrupts. After a hardware reset (i.e., power-up or push-button reset), all input/output signals are configured as input. Therefore, a pull-up resistor is used to provide a default value to the signal GPIO_{16} ($\overline{\mathsf{DCFLFOFF}}$), as shown in the last column of Table 1. The function of all signals are:

GPIO_{16}**:** $\overline{\mathsf{DCFLFOFF}}$**: D**aughter-**c**ard **fl**ash memory **f**orce **off**

This output signal is used to control the reset/power-down pin of the flash-memory circuits (see Section 2.1.1). When this signal is set to 0, the reset/power-down pins are asserted (0). When it is set to 1, the reset/power-down pins are only asserted (0) during a reset (i.e., hardware, software, or watch-dog reset) and during sleep mode.[2] A pull-up resistor sets the default value of this signal to 1 when the GPIO_{16} pin is configured as input (e.g., after a hardware reset). When the Itsy computer must boot from the daughter-card, this signal should never be set to 0 during sleep mode, since the processor would be unable to read the boot memory upon wake up. This can be easily achieved by setting bit 16 of the *power manager GPIO sleep state register* PGSR of the StrongARM SA-1100 processor [DEC98] to 1.

[2] On the memory daughter-card version 1.1, the reset/power-down pins are only asserted (0) when this signal is set to 0 and during sleep mode.

GPIO$_{17}$: DCFL0RY/$\overline{\text{BY}}$: Daughter-card flash memory 0 ready/busy

This input signal is connected to the ready/not-busy pin of the flash-memory circuit used for the least significant 16 data bits $D_{15..0}$ (see Section 2.1.1). It is set to 0 when the flash-memory circuit is executing an erase or program operation and to 1 when it is ready for use.

GPIO$_{18}$: DCFL1RY/$\overline{\text{BY}}$: Daughter-card flash memory 1 ready/busy

This input signal is connected to the ready/not-busy pin of the flash-memory circuit used for the most significant 16 data bits $D_{31..16}$ (see Section 2.1.1). It is set to 0 when the flash-memory circuit is executing an erase or program operation and to 1 when it is ready for use.

3.2 Non-volatile memory identification structure

Following the Itsy *static-memory identification scheme*, the flash memory must implement a *non-volatile memory identification structure* [Vir98]. The *class identification* value is $\mathsf{CID} = 0$, the read-only bit is $\mathsf{R} = 0$, the daughter-card bit is $\mathsf{D} = 1$, and the width bit is $\mathsf{W} = 0$. The values of the SIZE field and of the different MSC fields depend on the specific parts used on a given daughter-card (see Section 2.1.1). The fields EN0, RY/$\overline{\text{BY}}$0, EN1, and RY/$\overline{\text{BY}}$1, defining the general-purpose input/output signals, have the values $\mathsf{EN0} = 16 = 10_{16}$, $\mathsf{RY}/\overline{\mathsf{BY}}\mathsf{0} = 17 = 11_{16}$, $\mathsf{EN1} = 16 = 10_{16}$, and $\mathsf{RY}/\overline{\mathsf{BY}}\mathsf{1} = 18 = 12_{16}$.

References

[AMD97a] Advanced Micro Devices. *Am29LV400T/Am29LV400B: 4 Megabit (524,288 × 8-Bit/ 262,144 × 16-Bit) CMOS 3.0 Volt-Only Sector Architecture Flash Memory*, May 1997. Publication no. 20514, rev. B, amendment +1, preliminary.

[AMD97b] Advanced Micro Devices. *Am29LV200T/Am29LV200B: 2 Megabit (262,144 × 8-Bit/ 131,072 × 16-Bit) CMOS 3.0 Volt-Only Sector Architecture Flash Memory*, August 1997. Publication no. 20513, rev. C, amendment +1, preliminary.

[AMD97c] Advanced Micro Devices. *Am29LV800T/Am29LV800B: 8 Megabit (1,048,576 × 8-Bit/ 524,288 × 16-Bit) CMOS 3.0 Volt-Only, Sectored Flash Memory*, November 1997. Publication no. 20478, rev. D, amendment 0, preliminary.

[AMD97d] Advanced Micro Devices. *Am29LV800B: 8 Megabit (1 M × 8-Bit/512 K × 16-Bit) CMOS 3.0 Volt-Only Boot Sector Flash Memory*, December 1997. Publication no. 21490, rev. E, amendment 0, preliminary.

[AMD97e] Advanced Micro Devices. *Am29LV160B: 16 Megabit (2 M × 8-Bit/1 M × 16-Bit) CMOS 3.0 Volt-Only Boot Sector Flash Memory*, December 1997. Publication no. 21358, rev. F, amendment 0, preliminary.

[DEC98] Digital Equipment Corporation, Maynard, MA (USA). *DIGITAL Semiconductor SA-1100 Microprocessor: Technical Reference Manual*, March 1998. Document no. EC-R5MTC-TE.

[Hit97a] Hitachi. *HM5164160A Series, HM5165160A Series: 4194304-Word × 16-Bit Dynamic RAM*, January 1997. Document no. ADE-203-596 (Z), rev. 0.1, preliminary.

[Hit97b] Hitachi. *HM5164165A Series, HM5165165A Series: 4194304-Word × 16-Bit Dynamic RAM*, April 1997. Document no. ADE-203-453 (Z), rev. 0.4, preliminary.

[Hit97c] Hitachi. *HN29VT800 Series, HN29VB800 Series: 1048576-Word × 8-Bit / 524288-Word × 16-Bit CMOS Flash Memory*, April 1997. Document no. ADE-203-781A (Z), rev. 1.0.

[Hit97d] Hitachi. *HN29WT800 Series, HN29WB800 Series: 1048576-Word × 8-Bit / 524288-Word × 16-Bit CMOS Flash Memory*, May 1997. Document no. ADE-203-537A (Z), rev. 1.0.

[Mot97] Motorola. *M29F800A3, M29F800A2: 8 M CMOS Flash Memory*, October 1997. Document no. M29F800A3/D, rev. 4.

[Sam97a] Samsung Electronics. *KM416V4000A, KM416V4100A CMOS DRAM: 4 M × 16 Bit CMOS Dynamic RAM with Fast Page Mode*, 1997.

[Sam97b] Samsung Electronics. *KM416V4004A, KM416V4104A CMOS DRAM: 4 M × 16 Bit CMOS Dynamic RAM with Extended Data Out*, 1997.

[Sam98a] Samsung Electronics. *KM416V4000B, KM416V4100B CMOS DRAM: 4 M × 16 Bit CMOS Dynamic RAM with Fast Page Mode*, 1998. Preliminary spec.

[Sam98b] Samsung Electronics. *KM416V4004B, KM416V4104B CMOS DRAM: 4 M × 16 Bit CMOS Dynamic RAM with Extended Data Out*, 1998. Preliminary spec.

[Sha97] Sharp. *LH28F800SG: 8 Mbit (512 Kbit × 16) SmartVoltage Flash Memory*, February 1997. Preliminary.

[Tha97] C. Thacker. *WindowSIL/WC User's Guide.* SRC, Digital Equipment Corporation, Palo Alto, CA (USA), February 1997.

[Tos96] Toshiba. *Toshiba MOS Digital Integrated Circuit TC5165165AJS/AFTS-40 TC5165165AJS/AFTS-50 TC5165165AJS/AFTS-60 Silicon Gate CMOS: 4,194,304 Words × 16 Bit EDO (Hyper Page) Dynamic RAM*, October 1996. Tentative data.

[Vir98] Marc A. Viredaz. *The Itsy Pocket Computer Version 1.5: User's Manual.* Technical note TN-54, WRL, Compaq, Palo Alto, CA (USA), July 1998. Revision 1.0.

WRL Research Reports

‘‘Titan System Manual.’’ **Michael J. K. Nielsen.** WRL Research Report 86/1, September 1986.

‘‘Global Register Allocation at Link Time.’’ **David W. Wall.** WRL Research Report 86/3, October 1986.

‘‘Optimal Finned Heat Sinks.’’ **William R. Hamburgen.** WRL Research Report 86/4, October 1986.

‘‘The Mahler Experience: Using an Intermediate Language as the Machine Description.’’ **David W. Wall and Michael L. Powell.** WRL Research Report 87/1, August 1987.

‘‘The Packet Filter: An Efficient Mechanism for User-level Network Code.’’ **Jeffrey C. Mogul, Richard F. Rashid, Michael J. Accetta.** WRL Research Report 87/2, November 1987.

‘‘Fragmentation Considered Harmful.’’ **Christopher A. Kent, Jeffrey C. Mogul.** WRL Research Report 87/3, December 1987.

‘‘Cache Coherence in Distributed Systems.’’ **Christopher A. Kent.** WRL Research Report 87/4, December 1987.

‘‘Register Windows vs. Register Allocation.’’ **David W. Wall.** WRL Research Report 87/5, December 1987.

‘‘Editing Graphical Objects Using Procedural Representations.’’ **Paul J. Asente.** WRL Research Report 87/6, November 1987.

‘‘The USENET Cookbook: an Experiment in Electronic Publication.’’ **Brian K. Reid.** WRL Research Report 87/7, December 1987.

‘‘MultiTitan: Four Architecture Papers.’’ **Norman P. Jouppi, Jeremy Dion, David Boggs, Michael J. K. Nielsen.** WRL Research Report 87/8, April 1988.

‘‘Fast Printed Circuit Board Routing.’’ **Jeremy Dion.** WRL Research Report 88/1, March 1988.

‘‘Compacting Garbage Collection with Ambiguous Roots.’’ **Joel F. Bartlett.** WRL Research Report 88/2, February 1988.

‘‘The Experimental Literature of The Internet: An Annotated Bibliography.’’ **Jeffrey C. Mogul.** WRL Research Report 88/3, August 1988.

‘‘Measured Capacity of an Ethernet: Myths and Reality.’’ **David R. Boggs, Jeffrey C. Mogul, Christopher A. Kent.** WRL Research Report 88/4, September 1988.

‘‘Visa Protocols for Controlling Inter-Organizational Datagram Flow: Extended Description.’’ **Deborah Estrin, Jeffrey C. Mogul, Gene Tsudik, Kamaljit Anand.** WRL Research Report 88/5, December 1988.

‘‘SCHEME->C A Portable Scheme-to-C Compiler.’’ **Joel F. Bartlett.** WRL Research Report 89/1, January 1989.

‘‘Optimal Group Distribution in Carry-Skip Adders.’’ **Silvio Turrini.** WRL Research Report 89/2, February 1989.

‘‘Precise Robotic Paste Dot Dispensing.’’ **William R. Hamburgen.** WRL Research Report 89/3, February 1989.

‘‘Simple and Flexible Datagram Access Controls for Unix-based Gateways.’’ **Jeffrey C. Mogul.** WRL Research Report 89/4, March 1989.

‘‘Spritely NFS: Implementation and Performance of Cache-Consistency Protocols.’’ **V. Srinivasan and Jeffrey C. Mogul.** WRL Research Report 89/5, May 1989.

‘‘Available Instruction-Level Parallelism for Superscalar and Superpipelined Machines.’’ **Norman P. Jouppi and David W. Wall.** WRL Research Report 89/7, July 1989.

‘‘A Unified Vector/Scalar Floating-Point Architecture.’’ **Norman P. Jouppi, Jonathan Bertoni, and David W. Wall.** WRL Research Report 89/8, July 1989.

‘‘Architectural and Organizational Tradeoffs in the Design of the MultiTitan CPU.’’ **Norman P. Jouppi.** WRL Research Report 89/9, July 1989.

‘‘Integration and Packaging Plateaus of Processor Performance.’’ **Norman P. Jouppi.** WRL Research Report 89/10, July 1989.

‘‘A 20-MIPS Sustained 32-bit CMOS Microprocessor with High Ratio of Sustained to Peak Performance.’’ **Norman P. Jouppi and Jeffrey Y. F. Tang.** WRL Research Report 89/11, July 1989.

‘‘The Distribution of Instruction-Level and Machine Parallelism and Its Effect on Performance.’’ **Norman P. Jouppi.** WRL Research Report 89/13, July 1989.

‘‘Long Address Traces from RISC Machines: Generation and Analysis.’’ **Anita Borg, R.E.Kessler, Georgia Lazana, and David W. Wall.** WRL Research Report 89/14, September 1989.

‘‘Link-Time Code Modification.’’ **David W. Wall.** WRL Research Report 89/17, September 1989.

‘‘Noise Issues in the ECL Circuit Family.’’ **Jeffrey Y.F. Tang and J. Leon Yang.** WRL Research Report 90/1, January 1990.

‘‘Efficient Generation of Test Patterns Using Boolean Satisfiablilty.’’ **Tracy Larrabee.** WRL Research Report 90/2, February 1990.

‘‘Two Papers on Test Pattern Generation.’’ **Tracy Larrabee.** WRL Research Report 90/3, March 1990.

‘‘Virtual Memory vs. The File System.’’ **Michael N. Nelson.** WRL Research Report 90/4, March 1990.

‘‘Efficient Use of Workstations for Passive Monitoring of Local Area Networks.’’ **Jeffrey C. Mogul.** WRL Research Report 90/5, July 1990.

‘‘A One-Dimensional Thermal Model for the VAX 9000 Multi Chip Units.’’ **John S. Fitch.** WRL Research Report 90/6, July 1990.

‘‘1990 DECWRL/Livermore Magic Release.’’ **Robert N. Mayo, Michael H. Arnold, Walter S. Scott, Don Stark, Gordon T. Hamachi.** WRL Research Report 90/7, September 1990.

‘‘Pool Boiling Enhancement Techniques for Water at Low Pressure.’’ **Wade R. McGillis, John S. Fitch, William R. Hamburgen, Van P. Carey.** WRL Research Report 90/9, December 1990.

‘‘Writing Fast X Servers for Dumb Color Frame Buffers.’’ **Joel McCormack.** WRL Research Report 91/1, February 1991.

‘‘A Simulation Based Study of TLB Performance.’’ **J. Bradley Chen, Anita Borg, Norman P. Jouppi.** WRL Research Report 91/2, November 1991.

‘‘Analysis of Power Supply Networks in VLSI Circuits.’’ **Don Stark.** WRL Research Report 91/3, April 1991.

‘‘TurboChannel T1 Adapter.’’ **David Boggs.** WRL Research Report 91/4, April 1991.

‘‘Procedure Merging with Instruction Caches.’’ **Scott McFarling.** WRL Research Report 91/5, March 1991.

‘‘Don’t Fidget with Widgets, Draw!.’’ **Joel Bartlett.** WRL Research Report 91/6, May 1991.

‘‘Pool Boiling on Small Heat Dissipating Elements in Water at Subatmospheric Pressure.’’ **Wade R. McGillis, John S. Fitch, William R. Hamburgen, Van P. Carey.** WRL Research Report 91/7, June 1991.

‘‘Incremental, Generational Mostly-Copying Garbage Collection in Uncooperative Environments.’’ **G. May Yip.** WRL Research Report 91/8, June 1991.

‘‘Interleaved Fin Thermal Connectors for Multichip Modules.’’ **William R. Hamburgen.** WRL Research Report 91/9, August 1991.

‘‘Experience with a Software-defined Machine Architecture.’’ **David W. Wall.** WRL Research Report 91/10, August 1991.

‘‘Network Locality at the Scale of Processes.’’ **Jeffrey C. Mogul.** WRL Research Report 91/11, November 1991.

‘‘Cache Write Policies and Performance.’’ **Norman P. Jouppi.** WRL Research Report 91/12, December 1991.

‘‘Packaging a 150 W Bipolar ECL Microprocessor.’’ **William R. Hamburgen, John S. Fitch.** WRL Research Report 92/1, March 1992.

‘‘Observing TCP Dynamics in Real Networks.’’ **Jeffrey C. Mogul.** WRL Research Report 92/2, April 1992.

‘‘Systems for Late Code Modification.’’ **David W. Wall.** WRL Research Report 92/3, May 1992.

‘‘Piecewise Linear Models for Switch-Level Simulation.’’ **Russell Kao.** WRL Research Report 92/5, September 1992.

‘‘A Practical System for Intermodule Code Optimization at Link-Time.’’ **Amitabh Srivastava and David W. Wall.** WRL Research Report 92/6, December 1992.

‘‘A Smart Frame Buffer.’’ **Joel McCormack & Bob McNamara.** WRL Research Report 93/1, January 1993.

‘‘Recovery in Spritely NFS.’’ **Jeffrey C. Mogul.** WRL Research Report 93/2, June 1993.

‘‘Tradeoffs in Two-Level On-Chip Caching.’’ **Norman P. Jouppi & Steven J.E. Wilton.** WRL Research Report 93/3, October 1993.

‘‘Unreachable Procedures in Object-oriented Programing.’’ **Amitabh Srivastava.** WRL Research Report 93/4, August 1993.

‘‘An Enhanced Access and Cycle Time Model for On-Chip Caches.’’ **Steven J.E. Wilton and Norman P. Jouppi.** WRL Research Report 93/5, July 1994.

‘‘Limits of Instruction-Level Parallelism.’’ **David W. Wall.** WRL Research Report 93/6, November 1993.

‘‘Fluoroelastomer Pressure Pad Design for Microelectronic Applications.’’ **Alberto Makino, William R. Hamburgen, John S. Fitch.** WRL Research Report 93/7, November 1993.

‘‘A 300MHz 115W 32b Bipolar ECL Microprocessor.’’ **Norman P. Jouppi, Patrick Boyle, Jeremy Dion, Mary Jo Doherty, Alan Eustace, Ramsey Haddad, Robert Mayo, Suresh Menon, Louis Monier, Don Stark, Silvio Turrini, Leon Yang, John Fitch, William Hamburgen, Russell Kao, and Richard Swan.** WRL Research Report 93/8, December 1993.

‘‘Link-Time Optimization of Address Calculation on a 64-bit Architecture.’’ **Amitabh Srivastava, David W. Wall.** WRL Research Report 94/1, February 1994.

‘‘ATOM: A System for Building Customized Program Analysis Tools.’’ **Amitabh Srivastava, Alan Eustace.** WRL Research Report 94/2, March 1994.

‘‘Complexity/Performance Tradeoffs with Non-Blocking Loads.’’ **Keith I. Farkas, Norman P. Jouppi.** WRL Research Report 94/3, March 1994.

‘‘A Better Update Policy.’’ **Jeffrey C. Mogul.** WRL Research Report 94/4, April 1994.

‘‘Boolean Matching for Full-Custom ECL Gates.’’ **Robert N. Mayo, Herve Touati.** WRL Research Report 94/5, April 1994.

‘‘Software Methods for System Address Tracing: Implementation and Validation.’’ **J. Bradley Chen, David W. Wall, and Anita Borg.** WRL Research Report 94/6, September 1994.

‘‘Performance Implications of Multiple Pointer Sizes.’’ **Jeffrey C. Mogul, Joel F. Bartlett, Robert N. Mayo, and Amitabh Srivastava.** WRL Research Report 94/7, December 1994.

‘‘How Useful Are Non-blocking Loads, Stream Buffers, and Speculative Execution in Multiple Issue Processors?.’’ **Keith I. Farkas, Norman P. Jouppi, and Paul Chow.** WRL Research Report 94/8, December 1994.

‘‘Drip: A Schematic Drawing Interpreter.’’ **Ramsey W. Haddad.** WRL Research Report 95/1, March 1995.

‘‘Recursive Layout Generation.’’ **Louis M. Monier, Jeremy Dion.** WRL Research Report 95/2, March 1995.

‘‘Contour: A Tile-based Gridless Router.’’ **Jeremy Dion, Louis M. Monier.** WRL Research Report 95/3, March 1995.

‘‘The Case for Persistent-Connection HTTP.’’ **Jeffrey C. Mogul.** WRL Research Report 95/4, May 1995.

‘‘Network Behavior of a Busy Web Server and its Clients.’’ **Jeffrey C. Mogul.** WRL Research Report 95/5, October 1995.

‘‘The Predictability of Branches in Libraries.’’ **Brad Calder, Dirk Grunwald, and Amitabh** Srivastava. WRL Research Report 95/6, October 1995.

‘‘Shared Memory Consistency Models: A Tutorial.’’ **Sarita V. Adve, Kourosh Gharachorloo.** WRL Research Report 95/7, September 1995.

‘‘Eliminating Receive Livelock in an Interrupt-driven Kernel.’’ **Jeffrey C. Mogul and K. K. Ramakrishnan.** WRL Research Report 95/8, December 1995.

‘‘Memory Consistency Models for Shared-Memory Multiprocessors.’’ **Kourosh Gharachorloo.** WRL Research Report 95/9, December 1995.

‘‘Register File Design Considerations in Dynamically Scheduled Processors.’’ **Keith I. Farkas, Norman P. Jouppi, Paul Chow.** WRL Research Report 95/10, November 1995.

‘‘Optimization in Permutation Spaces.’’ **Silvio** Turrini. WRL Research Report 96/1, November 1996.

‘‘Shasta: A Low Overhead, Software-Only Approach for Supporting Fine-Grain Shared Memory.’’ **Daniel J. Scales, Kourosh Gharachorloo, and Chandramohan A. Thekkath.** WRL Research Report 96/2, November 1996.

‘‘Efficient Procedure Mapping using Cache Line Coloring.’’ **Amir H. Hashemi, David R. Kaeli, and Brad Calder.** WRL Research Report 96/3, October 1996.

‘‘Optimizations and Placement with the Genetic Workbench.’’ **Silvio Turrini.** WRL Research Report 96/4, November 1996.

‘‘Memory-system Design Considerations for Dynamically-scheduled Processors.’’ **Keith I. Farkas, Paul Chow, Norman P. Jouppi, and Zvonko Vranesic.** WRL Research Report 97/1, February 1997.

‘‘Performance of the Shasta Distributed Shared Memory Protocol.’’ **Daniel J. Scales and Kourosh Gharachorloo.** WRL Research Report 97/2, February 1997.

‘‘Fine-Grain Software Distributed Shared Memory on SMP Clusters.’’ **Daniel J. Scales, Kourosh Gharachorloo, and Anshu Aggarwal.** WRL Research Report 97/3, February 1997.

‘‘Potential benefits of delta encoding and data compression for HTTP.’’ **Jeffrey C. Mogul, Fred Douglis, Anja Feldmann, and Balachander Krishnamurthy.** WRL Research Report 97/4, July 1997.

WRL Technical Notes

‘‘TCP/IP PrintServer: Print Server Protocol.’’ **Brian K. Reid and Christopher A. Kent.** WRL Technical Note TN-4, September 1988.

‘‘TCP/IP PrintServer: Server Architecture and Implementation.’’ **Christopher A. Kent.** WRL Technical Note TN-7, November 1988.

‘‘Smart Code, Stupid Memory: A Fast X Server for a Dumb Color Frame Buffer.’’ **Joel McCormack.** WRL Technical Note TN-9, September 1989.

‘‘Why Aren’t Operating Systems Getting Faster As Fast As Hardware?.’’ **John Ousterhout.** WRL Technical Note TN-11, October 1989.

‘‘Mostly-Copying Garbage Collection Picks Up Generations and C++.’’ **Joel F. Bartlett.** WRL Technical Note TN-12, October 1989.

‘‘Characterization of Organic Illumination Systems.’’ **Bill Hamburgen, Jeff Mogul, Brian Reid, Alan Eustace, Richard Swan, Mary Jo Doherty, and Joel Bartlett.** WRL Technical Note TN-13, April 1989.

‘‘Improving Direct-Mapped Cache Performance by the Addition of a Small Fully-Associative Cache and Prefetch Buffers.’’ **Norman P. Jouppi.** WRL Technical Note TN-14, March 1990.

‘‘Limits of Instruction-Level Parallelism.’’ **David W. Wall.** WRL Technical Note TN-15, December 1990.

‘‘The Effect of Context Switches on Cache Performance.’’ **Jeffrey C. Mogul and Anita Borg.** WRL Technical Note TN-16, December 1990.

‘‘MTOOL: A Method For Detecting Memory Bottlenecks.’’ **Aaron Goldberg and John** Hennessy. WRL Technical Note TN-17, December 1990.

‘‘Predicting Program Behavior Using Real or Estimated Profiles.’’ **David W. Wall.** WRL Technical Note TN-18, December 1990.

‘‘Cache Replacement with Dynamic Exclusion.’’ **Scott McFarling.** WRL Technical Note TN-22, November 1991.

‘‘Boiling Binary Mixtures at Subatmospheric Pressures.’’ **Wade R. McGillis, John S. Fitch, William R. Hamburgen, Van P. Carey.** WRL Technical Note TN-23, January 1992.

‘‘A Comparison of Acoustic and Infrared Inspection Techniques for Die Attach.’’ **John S. Fitch.** WRL Technical Note TN-24, January 1992.

‘‘TurboChannel Versatec Adapter.’’ **David Boggs.** WRL Technical Note TN-26, January 1992.

‘‘A Recovery Protocol For Spritely NFS.’’ **Jeffrey C. Mogul.** WRL Technical Note TN-27, April 1992.

‘‘Electrical Evaluation Of The BIPS-0 Package.’’ **Patrick D. Boyle.** WRL Technical Note TN-29, July 1992.

‘‘Transparent Controls for Interactive Graphics.’’ **Joel F. Bartlett.** WRL Technical Note TN-30, July 1992.

‘‘Design Tools for BIPS-0.’’ **Jeremy Dion & Louis Monier.** WRL Technical Note TN-32, December 1992.

‘‘Link-Time Optimization of Address Calculation on a 64-Bit Architecture.’’ **Amitabh Srivastava and David W. Wall.** WRL Technical Note TN-35, June 1993.

‘‘Combining Branch Predictors.’’ **Scott McFarling.** WRL Technical Note TN-36, June 1993.

‘‘Boolean Matching for Full-Custom ECL Gates.’’ **Robert N. Mayo and Herve Touati.** WRL Technical Note TN-37, June 1993.

‘‘Piecewise Linear Models for Rsim.’’ **Russell Kao, Mark Horowitz.** WRL Technical Note TN-40, December 1993.

‘‘Speculative Execution and Instruction-Level Parallelism.’’ **David W. Wall.** WRL Technical Note TN-42, March 1994.

‘‘Ramonamap - An Example of Graphical Groupware.’’ **Joel F. Bartlett.** WRL Technical Note TN-43, December 1994.

‘‘ATOM: A Flexible Interface for Building High Performance Program Analysis Tools.’’ **Alan Eustace and Amitabh Srivastava.** WRL Technical Note TN-44, July 1994.

‘‘Circuit and Process Directions for Low-Voltage Swing Submicron BiCMOS.’’ **Norman P. Jouppi, Suresh Menon, and Stefanos Sidiropoulos.** WRL Technical Note TN-45, March 1994.

‘‘Experience with a Wireless World Wide Web Client.’’ **Joel F. Bartlett.** WRL Technical Note TN-46, March 1995.

‘‘I/O Component Characterization for I/O Cache Designs.’’ **Kathy J. Richardson.** WRL Technical Note TN-47, April 1995.

‘‘Attribute caches.’’ **Kathy J. Richardson, Michael J. Flynn.** WRL Technical Note TN-48, April 1995.

‘‘Operating Systems Support for Busy Internet Servers.’’ **Jeffrey C. Mogul.** WRL Technical Note TN-49, May 1995.

‘‘The Predictability of Libraries.’’ **Brad Calder, Dirk Grunwald, Amitabh Srivastava.** WRL Technical Note TN-50, July 1995.

‘‘Simultaneous Multithreading: A Platform for Next-generation Processors.’’ **Susan J. Eggers, Joel Emer, Henry M. Levy, Jack L. Lo, Rebecca Stamm and Dean M. Tullsen.** WRL Technical Note TN-52, March 1997.

‘‘Reducing Compulsory and Capacity Misses.’’ **Norman P. Jouppi.** WRL Technical Note TN-53, August 1990.

‘‘The Itsy Pocket Computer Version 1.5: User’s Manual.’’ **Marc A. Viredaz.** WRL Technical Note TN-54, July 1998.

‘‘The Memory Daughter-Card Version 1.5: User’s Manual.’’ **Marc A. Viredaz.** WRL Technical Note TN-55, July 1998.

WRL Research Reports and Technical Notes are available on the World Wide Web, from `http://www.research.digital.com/wrl/techreports/index.html`.