Exhibit U
Part 1

US007730287B2

(12) **United States Patent**
Hansen et al.

(10) **Patent No.: US 7,730,287 B2**
(45) **Date of Patent: Jun. 1, 2010**

(54) **METHOD AND SOFTWARE FOR GROUP FLOATING-POINT ARITHMETIC OPERATIONS**

(75) Inventors: **Craig Hansen**, Los Altos, CA (US); **John Moussouris**, Palo Alto, CA (US); **Alexia Massalin**, San Jose, CA (US)

(73) Assignee: **Microunity Systems Engineering, Inc.**, Santa Clara, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 80 days.

(21) Appl. No.: **11/878,814**

(22) Filed: **Jul. 27, 2007**

(65) **Prior Publication Data**

US 2008/0091925 A1 Apr. 17, 2008

Related U.S. Application Data

(60) Continuation of application No. 10/436,340, filed on May 13, 2003, now Pat. No. 7,516,308, which is a continuation of application No. 09/534,745, filed on Mar. 24, 2000, now Pat. No. 6,643,765, which is a continuation of application No. 09/382,402, filed on Aug. 24, 1999, now Pat. No. 6,295,599, which is a continuation-in-part of application No. 09/169,963, filed on Oct. 13, 1998, now Pat. No. 6,006,318, which is a continuation of application No. 08/754,827, filed on Nov. 22, 1996, now Pat. No. 5,822,603, which is a division of application No. 08/516,036, filed on Aug. 16, 1995, now Pat. No. 5,742,840, application No. 11/878,814, which is a continuation of application No. 11/511,466, filed on Aug. 29, 2006, now abandoned, which is a continuation of application No. 10/646,787, filed on Aug. 25, 2003, now Pat. No. 7,216,217, which is a continuation of application No. 09/922,319, filed on Aug. 2, 2001, now Pat. No. 6,725,356, which is a continuation of application No. 09/382,402, filed on Aug. 24, 1999, now Pat. No. 6,295,599, and a continuation-in-part of application No. 09/169,963, filed on Oct. 13, 1998, now Pat. No. 6,006,318, which is a continuation of application No. 08/754,827, filed on Nov. 22, 1996, now Pat. No. 5,822,603, which is a division of application No. 08/516,036, filed on Aug. 16, 1995, now Pat. No. 5,742,840.

(60) Provisional application No. 60/097,635, filed on Aug. 24, 1998.

(51) **Int. Cl.**
G06F 9/30 (2006.01)

(52) **U.S. Cl.** **712/222**; 712/221

(58) **Field of Classification Search** 712/222
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,025,772 A 5/1977 Constant

(Continued)

FOREIGN PATENT DOCUMENTS

CA 1 323 451 10/1993

(Continued)

OTHER PUBLICATIONS

Blelloch, G.E. etal., Compiling Collection-oriented languages onto massively parallel computers ,1988, IEEE, pp. 575-585.*

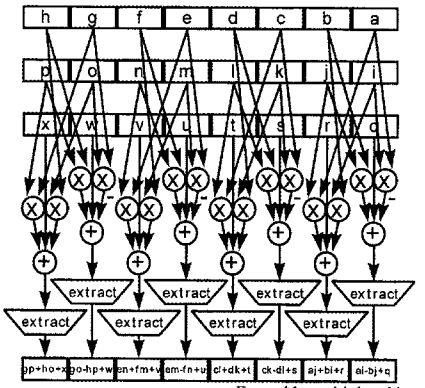
(Continued)

Primary Examiner—Eric Coleman
(74) *Attorney, Agent, or Firm*—McDermott Will & Emery LLP

(57) **ABSTRACT**

Methods and software are presented for processing data in a programmable processor, involving (a) decoding instructions for execution using an execution unit operable to execute instructions by partitioning data stored in registers in a register file into multiple data elements, the instructions selected from an instruction set that includes group arithmetic instructions and group data handling instructions, (b) in response to decoding different group data handling instructions, executing group data handling operations that re-arrange data elements in different ways, and (c) in response to decoding different group arithmetic instructions, executing a plurality of different group floating-point and group integer arithmetic operations that each arithmetically operates on the multiple data elements stored in registers in the register file to produce a catenated result that is returned to a register in the register file, wherein the catenated result comprises a plurality of individual results.

22 Claims, 384 Drawing Sheets



Ensemble multiply add extract immediate complex doublets

U.S. PATENT DOCUMENTS

4,489,393	A		12/1984	Kawahara et al.
4,595,911	A		6/1986	Kregness et al.
4,658,349	A		5/1987	Gafken
4,701,875	A		10/1987	Konishi et al.
4,725,973	A		2/1988	Matsura et al.
4,727,505	A		2/1988	Konishi et al.
4,734,877	A		3/1988	Sakata et al.
4,852,098	A		7/1989	Brechard et al.
4,875,161	A		10/1989	Lahti
4,876,660	A		10/1989	Owen et al.
4,893,267	A		1/1990	Alsup et al.
4,949,294	A		8/1990	Wambergue
4,953,073	A		8/1990	Moussouris et al.
4,956,801	A		9/1990	Priem et al.
4,959,779	A		9/1990	Weber et al.
4,969,118	A		11/1990	Montoye et al.
4,975,868	A		12/1990	Freerksen
5,032,865	A		7/1991	Schlunt
5,043,935	A		8/1991	Taniai et al.
5,081,698	A		1/1992	Kohn
5,113,506	A		5/1992	Moussouris et al.
5,132,898	A		7/1992	Sakamura et al.
5,155,816	A		10/1992	Kohn
5,157,388	A		10/1992	Kohn
5,161,247	A		11/1992	Murakami et al.
5,179,651	A		1/1993	Taaffe et al.
5,201,056	A		4/1993	Daniel et al.
5,208,914	A		5/1993	Wilson et al.
5,231,646	A		7/1993	Health et al.
5,233,690	A		8/1993	Sherlock et al.
5,241,636	A		8/1993	Kohn
5,268,855	A		12/1993	Mason et al.
5,268,995	A		12/1993	Diefendorff et al.
5,280,598	A		1/1994	Osaki et al.
5,325,495	A		6/1994	McLellan
5,327,369	A		7/1994	Ashkenazi
5,347,643	A		9/1994	Kondo et al.
5,390,135	A		2/1995	Lee
5,408,581	A		4/1995	Suzuki et al.
5,410,682	A		4/1995	Sites et al.
5,412,728	A		5/1995	Besnard et al.
5,423,051	A		6/1995	Fuller et al.
5,426,600	A		6/1995	Nakagawa et al.
5,430,660	A		7/1995	Lueker et al.
5,448,509	A		9/1995	Lee
5,467,131	A		11/1995	Bhaskaran
5,471,628	A		11/1995	Phillips et al.
5,487,024	A		1/1996	Girardeau, Jr.
5,500,811	A		3/1996	Corry
5,515,520	A		5/1996	Hatta et al.
5,517,438	A		5/1996	Dao-Trong et al.
5,533,185	A		7/1996	Lentz et al.
5,541,865	A		7/1996	Ashkenazi
5,557,724	A		9/1996	Sampat et al.
5,588,152	A		12/1996	Dapp et al.
5,590,350	A		12/1996	Guttag
5,590,365	A		12/1996	Ide et al.
5,592,405	A		1/1997	Gove et al.
5,600,814	A		2/1997	Gahan et al.
5,630,096	A	*	5/1997	Zuravleff et al. 711/154
5,636,351	A		6/1997	Lee
5,640,543	A		6/1997	Farrell et al.
5,642,306	A		6/1997	Mennemeier et al.
5,666,298	A		9/1997	Peleg et al.
5,669,010	A		9/1997	Duluk, Jr.
5,673,407	A		9/1997	Poland et al.
5,675,526	A		10/1997	Peleg et al.
5,680,338	A		10/1997	Agarwal et al.
5,721,892	A		2/1998	Peleg et al.
5,734,874	A		3/1998	Van Hook et al.
5,737,547	A	*	4/1998	Zuravleff et al. 710/112
5,742,840	A		4/1998	Hansen et al.
5,757,432	A		5/1998	Dulong et al.
5,758,176	A		5/1998	Agarwal et al.
5,768,546	A		6/1998	Kwon
5,778,412	A		7/1998	Gafken
5,794,060	A		8/1998	Hansen et al.
5,794,061	A		8/1998	Hansen et al.
5,802,336	A		9/1998	Peleg et al.
5,809,292	A		9/1998	Wilkinson et al.
5,809,321	A	*	9/1998	Hansen et al. 712/1
5,812,799	A	*	9/1998	Zuravleff et al. 710/52
5,818,739	A		10/1998	Peleg et al.
5,819,101	A		10/1998	Peleg
5,822,603	A	*	10/1998	Hansen et al. 712/1
5,825,677	A		10/1998	Agarwal et al.
5,828,869	A		10/1998	Johnson et al.
5,835,782	A		11/1998	Lin et al.
5,881,275	A		3/1999	Peleg
5,883,824	A		3/1999	Lee
5,886,732	A		3/1999	Humpleman
5,887,183	A		3/1999	Agarwal et al.
5,922,066	A		7/1999	Cho et al.
5,933,160	A	*	8/1999	Kabir et al. 345/561
5,983,257	A		11/1999	Dulong et al.
5,996,057	A		11/1999	Scales, III et al.
6,006,318	A	*	12/1999	Hansen et al. 712/28
6,016,538	A		1/2000	Guttag et al.
6,058,408	A	*	5/2000	Fischer et al. 708/322
6,058,465	A		5/2000	Nguyen
6,092,094	A		7/2000	Ireton
6,381,690	B1		4/2002	Lee
6,401,194	B1		6/2002	Nguyen et al.
6,425,073	B2		7/2002	Roussel et al.
6,453,368	B2		9/2002	Yamamoto
6,516,406	B1		2/2003	Peleg et al.
6,567,908	B1		5/2003	Furuhashi
6,584,482	B1	*	6/2003	Hansen et al. 708/523
6,643,765	B1		11/2003	Hansen et al.
6,725,356	B2		4/2004	Hansen et al.
7,216,217	B2		5/2007	Hansen et al.
7,222,225	B2	*	5/2007	Hansen et al. 712/204
7,464,252	B2		12/2008	Hansen et al.

FOREIGN PATENT DOCUMENTS

CA	1 323 451	C	10/1993
EP	0474246	A2	9/1991
EP	0 468 820	A2	1/1992
EP	0654733	A1	7/1994
EP	0627682	A1	12/1994
EP	0 649 085	A1	4/1995
EP	0 651 321	A	5/1995
EP	0 653 703	A1	5/1995
EP	0 654 733	A	5/1995
EP	0 654 733	A1	5/1995
JP	S60-217435		10/1985
JP	3268024		11/1991
JP	6095843		4/1994
WO	WO 93/01543		1/1993
WO	WO 93/01565		1/1993
WO	WO 93/11500		6/1993
WO	WO 97/07450		2/1997

OTHER PUBLICATIONS

Suzuki, H. etal., Leading-Zero Anticipatory Logic for High-Speed Floating Point Addition, 1996, IEEE. 8 pages.*

Lee, R.B. , Subword Parallelism with Max-2, 1996, IEEE, pp. 51-59.*

Ide, N. etal., 320-MFLOPS CMOS Floating point processing unit for superscalar processors, 1993, IEEE, pp. 352-361.*

MIPS32 4K™ Processor Core Family Software User's Manual, Revision 1.12, Jan. 3, 2001, MIPS Technologies, Inc., Document No. MD00016, pp. i-xii and 1-324.

MIPS64 5Kc™ Processor Core Family Software User's Manual, Revision 2.2, Aug. 11, 2000, MIPS Technologies, Inc., Document No. MD00055, pp. i-xii and 1-580.
Diefendorff et al., "The PowerPC User Instruction Set Architecture," *IEEE Micro*, No. 5, 30-41 (Oct. 1994) [XP000476678].
Greenley et al., "UltraSPARC(™): The Next Generation Superscaler 64-bit SPARC," IEEE, 442-51 (1995) [XP000545452].
Laudon et al., "Interleaving: A Multithreading Technique Targeting Multiprocessors and Workstations," ACM Sigplan Notices, No. 11, 308-18 (Nov. 29, 1994) [XP000491743].
"MC88110 RISC Microprocessor User's Manual, Second Edition," Motorola, Inc. (1990), Sections 1 through 3 (148 pages) [XP002474804].
Sato et al., "Multiple Instruction Streams in a Highly Pipelined Processor," IEEE, 182-89 (1990) [XP010021363].
Yamamoto et al., "Performance Estimation of Multistreamed, Superscaler Processors," IEEE, 195-204 (1994) [XP010096943].
Zhou et al., "MPEG Video Decoding with the UltraSPARC Visual Instruction Set," IEEE, 470-75 (1995) [XP002472254].
P27838EP-D1-PCT Extended European Search Report dated May 2, 2008 in Application No. / Patent No. 07111352.6-1243.
P27838EP-D4-PCT Extended European Search Report dated May 8, 2008 in Application No. / Patent No. 07111349.2-1243.
P27838EP-D5-PCT Extended European Search Report dated Apr. 23, 2008 in Application No. / Patent No. 07111344.3-1243.
P27838EP-D6-PCT Partial European Search Report dated Apr. 1, 2008 in Application No. / Patent No. 07111348.4-1243.
P27838EP-D7-PCT Extended European Search Report dated Mar. 10, 2008 in Application No. / Patent No. 07111473.0/1243.
P27838EP-D8-PCT Extended European Search Report dated May 27, 2008 in Application No. / Patent No. 07111476.3-1243.
P27838EP-D9-PCT Extended European Search Report dated Mar. 26, 2008 in Application No. / Patent No. 07111480.5-1243.
Fuller et al., "The PowerPC 604™ Microprocessor—Multimedia Ready," *Circuits and Systems*, 1995, *Proceedings of the 38th Midwest Symposium on Rio de Janeiro, Brazil* (Aug. 13-16, 1995), New York, NY, IEEE, vol. 2, 1135-38 (Aug. 13, 1995) [XP010165331].
Gwennap, "MIPS R10000 Uses Decoupled Architecture—High-Performance Core Will Drive MIPS High-End for Years," *Microprocessor Report*, vol. 8, No. 14 (Oct. 24, 1994), MicroDesign Resources (1994) [XP002219607].
Murakami et al., "SIMP (Single Instruction Stream / Multiple Instruction Pipelining): A Novel High-Speed Single-Processor Architecture," *Computer Architecture News*, ACM, New York, NY, vol. 17, No. 3, 78-85 (Jun. 1, 1989) [XP000035291].
"UltraSPARC Multimedia Capabilities On-Chip Support for Real-Time Video and Advanced Graphics," Sun Microsystems, Inc., 1-8 (1994) [XP002177546].
P27838EP-D3-PCT Extended European Search Report dated Jul. 3, 2008 in Application No. / Patent No. 07111350.0-1243 / 1873654.
P27838EP-D6-PCT Extended European Search Report dated Jun. 27, 2008 in Application No. / Patent No. 07111348.4-1243 / 1873629.
U.S. Appl. No. 10/418,113 Continuation Application filed Apr. 18, 2003.
U.S. Appl. No. 10/418,113 Preliminary Amendment dated May 27, 2003.
U.S. Appl. No. 10/418,113 First Office Action dated Jun. 10, 2004.
U.S. Appl. No. 10/418,113 Examiner Interview Summary dated Sep. 24, 2004.
U.S. Appl. No. 10/418,113 First Office Action Response filed Nov. 10, 2004.
U.S. Appl. No. 10/418,113 Notice of Allowance and Issues Fee Due dated Mar. 16, 2005.
U.S. Appl. No. 10/418,113 Request for Continued Examination filed Jun. 10, 2005.
U.S. Appl. No. 10/436,340 Continuation Application filed May 13, 2003.
U.S. Appl. No. 10/436,340 First Office Action dated Feb. 9, 2006.
U.S. Appl. No. 10/436,340 First Office Action Response filed Jul. 10, 2006.
U.S. Appl. No. 10/436,340 Notice of Allowance and Issue Fee Due dated Oct. 3, 2006.
U.S. Appl. No. 10/436,340 Response to Notice of Allowance and RCE filed Nov. 14, 2006.
U.S. Appl. No. 10/436,340 Second Office Action dated Feb. 21, 2007.
U.S. Appl. No. 10/436,340 Second Office Action Response filed Aug. 21, 2007.
U.S. Appl. No. 10/436,340 Declaration of Korbin Van Dyke filed Aug. 21, 2007.
U.S. Appl. No. 10/436,340 Third Office Action dated Nov. 21, 2007.
U.S. Appl. No. 10/436,340 Third Office Action Response filed Mar. 21, 2008.
U.S. Appl. No. 10/436,340 Third Office Action Response Supplemental filed Apr. 18, 2008.
7,216,217 (U.S. Appl. No. 10/646,787) Continuation Application filed Aug. 25, 2003.
7,216,217 (U.S. Appl. No. 10/646,787) Preliminary Amendment filed Dec. 23, 2003.
7,216,217 (U.S. Appl. No. 10/646,787) First Office Action dated May 9, 2005.
7,216,217 (U.S. Appl. No. 10/646,787) First Office Action Response filed Nov. 9, 2005.
7,216,217 (U.S. Appl. No. 10/646,787) Second Office Action dated Jan. 12, 2006.
7,216,217 (U.S. Appl. No. 10/646,787) Second Office Action Response filed May 11, 2006.
7,216,217 (U.S. Appl. No. 10/646,787) Notice of Allowance dated Jun. 2, 2006.
7,216,217 (U.S. Appl. No. 10/646,787) Issue Notification Apr. 18, 2007.
U.S. Appl. No. 10/757,836 Continuation Application filed Jan. 16, 2004.
U.S. Appl. No. 10/757,836 Preliminary Amendment filed May 14, 2004.
U.S. Appl. No. 10/757,836 Preliminary Amendment filed Jun. 18, 2004.
U.S. Appl. No. 10/757,836 First Office Action dated Jun. 19, 2006.
U.S. Appl. No. 10/757,836 First Office Action Response filed Sep. 10, 2006.
U.S. Appl. No. 10/757,836 Notice of Allowance and Fees Due dated Nov. 28, 2007.
U.S. Appl. No. 10/757,836 Request for Continued Examination filed Feb. 27, 2008.
U.S. Appl. No. 10/757,836 Second Office Action dated Jun. 6, 2008.
U.S. Appl. No. 10/757,836 Second Office Action Response filed Jun. 26, 2008.
U.S. Appl. No. 10/757,851 Continuation Application filed Jan. 16, 2004.
U.S. Appl. No. 10/757,851 Preliminary Amendment filed May 14, 2004.
U.S. Appl. No. 10/757,851 Preliminary Amendment filed Jun. 18, 2004.
U.S. Appl. No. 10/757,851 First Office Action dated May 23, 2006.
U.S. Appl. No. 10/757,851 First Office Action Response filed Nov. 21, 2006.
U.S. Appl. No. 10/757,851 Second Office Action dated Feb. 21, 2007.
U.S. Appl. No. 10/757,851 Second Office Action Response filed Aug. 21, 2007.
U.S. Appl. No. 10/757,851 Notice of Allowance and Issue Fees Due dated Nov. 16, 2007.
U.S. Appl. No. 10/757,851 Notice of Allowance and Issue Fees Due dated Feb. 12, 2008.
U.S. Appl. No. 10/757,851 Request for Continued Examination filed Feb. 19, 2008.
U.S. Appl. No. 10/757,851 Third Office Action dated May 9, 2008.
U.S. Appl. No. 10/757,851 Third Office Action Response filed Jun. 26, 2008.
U.S. Appl. No. 11/511,466 Continuation Application filed Aug. 29, 2006.
U.S. Appl. No. 11/511,466 Preliminary Amendment filed Aug. 29, 2006.
U.S. Appl. No. 11/878,803 Continuation Application filed Jul. 27, 2007.
U.S. Appl. No. 11/878,804 Continuation Application filed Jul. 27, 2007.

U.S. Appl. No. 11/878,805 Continuation Application filed Jul. 27, 2007.
5,742,840 (U.S. Appl. No. 08/516,036) Transmittal of Application filed Aug. 16, 1995.
5,742,840 (U.S. Appl. No. 08/516,036) First Office Action dated Oct. 15, 1996.
5,742,840 (U.S. Appl. No. 08/516,036) First Office Action Response filed Nov. 14, 1996.
5,742,840 (U.S. Appl. No. 08/516,036) Second Office Action dated Jan. 28, 1997.
5,742,840 (U.S. Appl. No. 08/516,036) Second Office Action Response dated May 28, 1997.
5,742,840 (U.S. Appl. No. 08/516,036) Examiner Interview Summary dated Aug. 4, 1997.
5,742,840 (U.S. Appl. No. 08/516,036) Notice of Allowance and Issue Fee Due dated Aug. 4, 1997.
5,742,840 (U.S. Appl. No. 90/007,583) Reexam Request filed Jun. 13, 2005.
5,742,840 (U.S. Appl. No. 90/007,583) Office Action Granting Reexam dated Jul. 13, 2005.
5,742,840 (U.S. Appl. No. 90/007,583) First Office Action dated Dec. 22, 2006.
5,742,840 (U.S. Appl. No. 90/007,583) First Office Action Response filed Feb. 22, 2007.
5,742,840 (U.S. Appl. No. 90/007,583) Examiner Interview Summary dated Jun. 13, 2007.
5,742,840 (U.S. Appl. No. 90/007,583) Response to Examiner Interview Summary filed Jun. 22, 2007.
5,742,840 (U.S. Appl. No. 90/007,583) Office Action (Final) dated Aug. 9, 2007.
5,742,840 (U.S. Appl. No. 90/007,583) Final Office Action Response filed Oct. 9, 2007.
5,742,840 (U.S. Appl. No. 90/007,583) Dr. V. Thomas Rhyne Declaration dated Oct. 4, 2007.
5,742,840 (U.S. Appl. No. 90/007,583) Examiner Interview Summary dated Jan. 28, 2008.
5,742,840 (U.S. Appl. No. 90/007,583) Response to Examiner Interview Summary filed Feb. 28, 2008.
5,794,060 (U.S. Appl. No. 08/754,826) Divisional Application under C.F.R. 1.60 filed Nov. 22, 1996.
5,794,060 (U.S. Appl. No. 08/754,826) First Office Action dated Oct. 23, 1997.
5,794,060 (U.S. Appl. No. 08/754,826) First Office Action Response filed Feb. 11, 1998.
5,794,060 (U.S. Appl. No. 08/754,826) Examiner Interview Summary dated Apr. 28, 1998.
5,794,060 (U.S. Appl. No. 08/754,826) Notice of Allowance and Issue Fee Due dated Apr. 28, 1998.
5,794,060 (U.S. Appl. No. 08/754,826) Notice of *Ex Parte* Reexamination dated Jun. 17, 2005 (U.S. Appl. No. 90/007,593).
5,794,060 (U.S. Appl. No. 90/007,593) Reexam Request Filed Jun. 17, 2005.
5,794,060 (U.S. Appl. No. 90/007,593) Office Action Granting Reexam dated Jul. 12, 2005.
5,794,060 (U.S. Appl. No. 90/007,593)) First Office Action dated Dec. 22, 2006.
5,794,060 (U.S. Appl. No. 90/007,593) First Office Action Response filed Feb. 22, 2007.
5,794,060 (U.S. Appl. No. 90/007,593) Office Action (Final) dated Sep. 1, 2007.
5,794,060 (U.S. Appl. No. 90/007,593) Office Action Response filed Nov. 1, 2007.
5,794,060 (U.S. Appl. No. 90/007,593) Amendment Entered by Examiner dated Apr. 30, 2008.
5,794,060 (U.S. Appl. No. 90/007,593) Office Action dated May 6, 2008.
5,794,060 (U.S. Appl. No. 90/007,593) Office Action response filed Jul. 7, 2008.
5,794,060 (U.S. Appl. No. 90/007,593) Declaration of Dr. John Moussouris dated Jul. 3, 2008.
5,794,060 (U.S. Appl. No. 90/007,593) Declaration of Craig Hansen dated Jul. 7, 2008.
5,794,061 (U.S. Appl. No. 08/754,829) Divisional Application under C.F.R. 1.60 filed Nov. 22, 1996.
5,794,061 (U.S. Appl. No. 08/754,829) First Office Action dated Sep. 11, 1997.
5,794,061 (U.S. Appl. No. 08/754,829) First Office Action Response filed Feb. 11, 1998.
5,794,061 (U.S. Appl. No. 08/754,829) Notice of Allowance and Issue Fee Due dated Apr. 28, 1998.
5,794,061 (U.S. Appl. No. 08/754,829) Notice of *Ex Parte* Reexamination dated May 26, 2005 (U.S. Appl. No. 90/007,563).
5,794,061 (U.S. Appl. No. 90/007,563) Reexam Request Filed May 26, 2005.
5,794,061 (U.S. Appl. No. 90/007,563) Office Action Granting Reexam dated Jul. 7, 2005.
5,794,061 (U.S. Appl. No. 90/007,563) First Office Action dated Dec. 22, 2006.
5,794,061 (U.S. Appl. No. 90/007,563) First Office Action Response filed Feb. 22, 2007.
5,794,061 (U.S. Appl. No. 90/007,563) Dr. John Moussouris Declaration dated Feb. 22, 2007.
5,794,061 (U.S. Appl. No. 90/007,563) Craig Hansen Declaration dated Feb. 22, 2007.
5,794,061 (U.S. Appl. No. 90/007,563) Examiner Interview Summary dated Jun. 13, 2007.
5,794,061 (U.S. Appl. No. 90/007,563) Response to Examiner Interview Summary dated Jun. 22, 2007.
5,794,061 (U.S. Appl. No. 90/007,563) Office Action (Final) dated Sep. 1, 2007.
5,794,061 (U.S. Appl. No. 90/007,563) Office Action Response filed Nov. 1, 2007.
5,794,061 (U.S. Appl. No. 90/007,563) Dr. V. Thomas Rhyne Declaration dated Oct. 4, 2007.
5,794,061 (U.S. Appl. No. 90/007,563) Dr. John Moussouris Declaration dated Nov. 1, 2007.
5,794,061 (U.S. Appl. No. 90/007,563) Korbin Van Dyke Declaration dated Nov. 1, 2007.
6,584,482 (U.S. Appl. No. 09/377,182) Continuation Application under 37 C.F.R. 1.53(b) filed Aug. 19, 1999.
6,584,482 (U.S. Appl. No. 09/377,182) Preliminary Amendment filed Sep. 20, 1999.
6,584,482 (U.S. Appl. No. 09/377,182) First Office Action dated Aug. 8, 2002.
6,584,482 (U.S. Appl. No. 09/377,182) First Office Action Response (draft) filed Jan. 13, 2003.
6,584,482 (U.S. Appl. No. 09/377,182) Examiner Interview Summary dated Jan. 16, 2003.
6,584,482 (U.S. Appl. No. 09/377,182) First Office Action Response filed Feb. 4, 2003.
6,584,482 (U.S. Appl. No. 09/377,182) Notice of Allowance and Fee(s) Due dated Feb. 28, 2003.
6,584,482 (U.S. Appl. No. 09/377,182) Supplemental Notice of Allowability dated Mar. 5, 2003.
6,584,482 (U.S. Appl. No. 09/377,182) Notice of *Ex Parte* Reexamination dated May 4, 2005 U.S. Appl. No. 90/007,532.
6,584,482 (U.S. Appl. No. 90/007,532) Reexam Request filed May 4, 2005.
6,584,482 (U.S. Appl. No. 90/007,532) Office Action Granting Reexam dated Jun. 13, 2005.
6,584,482 (U.S. Appl. No. 90/007,532) First Office Action dated Jul. 3, 2006.
6,584,482 (U.S. Appl. No. 90/007,532) First Office Action Response filed Sep. 5, 2006.
6,584,482 (U.S. Appl. No. 90/007,532) Second Office Action dated Sep. 18, 2007.
6,584,482 (U.S. Appl. No. 90/007,532) Second Office Action Response filed Nov. 19, 2007.
6,584,482 (U.S. Appl. No. 90/007,532) Final Office Action dated Apr. 1, 2008.
6,584,482 (U.S. Appl. No. 90/007,532) Final Office Action Response filed Jun. 2, 2008.
6,584,482 (U.S. Appl. No. 90/007,532) Notice of Intent to Issue dated Jun. 17, 2008.

6,725,356 (U.S. Appl. No. 09/922,319) Continuation Application filed Aug. 2, 2001.
6,725,356 (U.S. Appl. No. 09/922,319) First Office Action dated Sep. 23, 2002.
6,725,356 (U.S. Appl. No. 09/922,319) First Office Action Response filed Mar. 24, 2003.
6,725,356 (U.S. Appl. No. 09/922,319) Craig Hansen Declaration of Incorporated Subject Matter filed Mar. 24, 2003.
6,725,356 (U.S. Appl. No. 09/922,319) Supplemental Amendment filed May 21, 2003.
6,725,356 (U.S. Appl. No. 09/922,319) Second Supplemental Amendment filed May 29, 2003.
6,725,356 (U.S. Appl. No. 09/922,319) Notice of Allowance and Issue Fee(s) Due dated Jun. 26, 2003.
6,725,356 (U.S. Appl. No. 09/922,319) Comments in Response to Reasons for Allowance filed Jul. 10, 2003.
6,725,356 (U.S. Appl. No. 09/922,319) Notice of *Inter Partes* Reexamination dated Jun. 28, 2005 (U.S. Appl. No. 95/000,100).
6,725,356 (U.S. Appl. No. 95/000,100) Reexam Request Filed Jun. 28, 2005.
6,725,356 (U.S. Appl. No. 95/000,100) Office Action Granting Reexam dated Sep. 8, 2005.
6,725,356 (U.S. Appl. No. 95/000,100) First Office Action dated Sep. 8, 2005.
6,725,356 (U.S. Appl. No. 95/000,100) Supplement to First Office Action dated Sep. 26, 2005.
6,725,356 (U.S. Appl. No. 95/000,100) First Office Action Response filed Dec. 8, 2005.
6,725,356 (U.S. Appl. No. 95/000,100) Dr. John Moussouris Declaration dated Dec. 7, 2005.
6,725,356 (U.S. Appl. No. 95/000,100) Ronald Alepin Declaration dated Dec. 7, 2005.
6,725,356 (U.S. Appl. No. 95/000,100) Korbin Van Dyke Declaration dated Dec. 5, 2005.
6,725,356 (U.S. Appl. No. 95/000,100) Second Office Action dated May 3, 2006.
6,725,356 (U.S. Appl. No. 95/000,100) Second Office Action Response filed Jun. 30, 2006.
6,725,356 (U.S. Appl. No. 95/000,100) Dr. John Moussouris Declaration dated Jun. 30, 2006.
6,725,356 (U.S. Appl. No. 95/000,100) Korbin Van Dyke Declaration dated Jun. 30, 2006.
6,643,765 (U.S. Appl. No. 09/534,745) Continuation Application filed Mar. 24, 2000.
6,643,765 (U.S. Appl. No. 09/534,745) First Office Action dated Sep. 23, 2002.
6,643,765 (U.S. Appl. No. 09/534,745) First Office Action Response filed Mar. 21, 2003.
6,643,765 (U.S. Appl. No. 09/534,745) Craig Hansen Declaration of Incorporated Subject Matter filed Mar. 21, 2003.
6,643,765 (U.S. Appl. No. 09/534,745) Notice of Allowance and Issue Fee(s) Due dated Apr. 17, 2003.
6,643,765 (U.S. Appl. No. 09/534,745) Examiner's Amendment entered Aug. 23, 2003.
6,643,765 (U.S. Appl. No. 95/000,089) Reexam Request Filed Apr. 15, 2005.
6,643,765 (U.S. Appl. No. 95/000,089) Office Action Granting Reexam dated Jun. 1, 2005.
6,643,765 (U.S. Appl. No. 95/000,089) First Office Action dated Jun. 1, 2005.
6,643,765 (U.S. Appl. No. 95/000,089) First Office Action Response filed Oct. 3, 2005.
6,643,765 (U.S. Appl. No. 95/000,089) Dr. John Moussouris Declaration dated Oct. 3, 2005.
6,643,765 (U.S. Appl. No. 95/000,089) Korbin Van Dyke Declaration dated Oct. 3, 2005.
6,643,765 (U.S. Appl. No. 95/000,089) Ronald Alepin Declaration dated Oct. 3, 2005.
6,643,765 (U.S. Appl. No. 95/000,089) Office Action (Final) dated Mar. 2, 2006.
6,643,765 (U.S. Appl. No. 95/000,089) Office Action Response filed May 2, 2006.
6,643,765 (U.S. Appl. No. 95/000,089) Dr. John Moussouris Declaration dated May 2, 2006.
6,643,765 (U.S. Appl. No. 95/000,089) Korbin Van Dyke Declaration dated May 1, 2006.
U.S. Appl. No. 90/007,593 (Reexam of US 5,794,060) Office Action Response dated Dec. 24, 2008.
U.S. Appl. No. 90/007,593 (Reexam of US 5,794,060) Supplemental Office Action Response dated Feb. 12, 2009.
U.S. Appl. No. 95/000,100 (Reexam of US 6,725,356) Office Action mailed Mar. 19, 2009.
U.S. Appl. No. 95/000,089 (Reexam of US 6,643,765) Decision on Petition Filed Under 37 CFR 1.181 mailed Feb. 3, 2009.
U.S. Appl. No. 95/000,089 (Reexam of US 6,643,765) Appeal Brief dated Feb. 5, 2009.
U.S. Appl. No. 95/000,089 (Reexam of US 6,643,765) Decision Dismissing Petition mailed Mar. 6, 2009.
U.S. Appl. No. 10/757,851 RCE and Petition for Withdrawal from Issue dated Dec. 15, 2008.
U.S. Appl. No. 10/757,851 Withdrawal from Issue dated Dec. 16, 2008.
U.S. Appl. No. 10/757,851 Decision Granting Petition Under 37 CFR 1.313 (c)(2) dated Dec. 17, 2008.
U.S. Appl. No. 10/757,851 Office Action and Notice of References Cited dated Mar. 5, 2009.
U.S. Appl. No. 11/878,804 Non-Final Office Action, Notice of References Cited dated Jan. 5, 2009.
U.S. Appl. No. 11/878,805 Non-Final Office Action, Notices of References Cited dated Feb. 2, 2009.
Japanese Patent Application No. 2000-577552 Notice of Reasons of Rejection dated Feb. 10, 2009.
Potmesil, M. et al., The Pixel Machine: A Parallel Image Computer, 1989, ACM, pp. 69-78.
Chart: MicroUnity Media Processor Patent Family (Apr. 2009).
6,643,765 (U.S. Appl. No. 95/000,089) Right of Appeal Notice (37 CFR 1.953) mailed Nov. 5, 2008.
6,643,765 (U.S. Appl. No. 95/000,089) Request for Reconsideration and Response and Objection to the Right of Appeal Notice filed Dec. 5, 2008.
6,643,765 (U.S. Appl. No. 95/000,089) Notice of Appeal From the Primary Examiner to the Board of Appeals filed Dec. 5, 2008.
6,643,765 (U.S. Appl. No. 95/000,089) Petition for Supervisory Authority Under 37 CFR 1.181 filed Dec. 5, 2008.
5,742,840 (U.S. Appl. No. 90/007,583) Office Action mailed Aug. 14, 2008.
5,742,840 (U.S. Appl. No. 90/007,583) Response to Aug. 14, 2008 Office Action and Exhibits 1-5; Declaration of John Moussouris Under 37 CFR 1.131; Declaration of Craig Hansen Under 37 CFR 1.131 filed Oct. 14, 2008.
5,742,840 (U.S. Appl. No. 90/007,583) Supplemental Response filed Oct. 31, 2008.
5,794,060 (U.S. Appl. No. 90/007,593) Office Action mailed Oct. 24, 2008.
5,794,061 (U.S. Appl. No. 90/007,563) Office Action mailed Sep. 30, 2008.
5,794,061 (U.S. Appl. No. 90/007,563) Response to Sep. 30, 2008 Office Action, Declaration of Craig Hansen Under 37 CFR 1.131; Declaration of John Moussouris Under 37 CFR 1.131 filed Dec. 1, 2008.
6,584,482 (U.S. Appl. No. 90/007,532) Ex Parte Reexamination Certificate issued Oct. 28, 2008.
U.S. Appl. No. 10/436,340 Notice of Allowance and Fee(s) Due mailed Nov. 20, 2008.
U.S. Appl. No. 10/757,836 Terminal Disclaimer Approved (Sep. 22, 2008).
U.S. Appl. No. 10/757,836 Notice of Allowance and Fee(s) Due mailed Oct. 7, 2008.
U.S. Appl. No. 10/757,851 Terminal Disclaimer Approved (Sep. 22, 2008).
U.S. Appl. No. 10/757,851 Notice of Allowance and Fee(s) Due mailed Oct. 7, 2008.
U.S. Appl. No. 11/511,466 Office Action and Notice of References Cited mailed Nov. 17, 2008.

MJB08274EP Extended European Search Report dated Nov. 26, 2008 in Application No. / Patent No. 07112545.4-1243 / 1879103.
MJB08275EP Extended European Search Report dated Nov. 11, 2008 in Application No. / Patent No. 07112548.8-1243 / 1879398.
Gwennap, "UltraSPARC Adds Multimedia Instructions," *Microprocessor Report*, vol. 8, No. 6, 1-3 (Dec. 5, 1994), MicroDesign Resources © (1994) (XP000561690).
Shipnes, "Graphics Processing with the 88110 RISC Microprocessor," *IEEE COMPCON SPRING* '92, 169-74 (Feb. 24-28, 1992) (XP000340730).
Zucker, Daniel F. et al., Reuse of High Precision Arithmetic Hardware to Perform Multiple Concurrent Low Precision Calculation, IEEE, Apr. 1994.
MU v. *Dell & Intel* Feb. 25, 2005 Defendant Dell Inc.'s Amended Answer, Affirmative Defenses, and Counterclaims to Plaintiff's First Amended Complaint (*MicroUnity Systems Engineering, Inc.* v. *Dell, Inc. f/k/a/ Dell Computer and Intel Corporation*; C.A. No. 2-04-cv-120; U.S.D.C., E.D. Texas, Marshall Division).
Feb. 25, 2005 Defendant Intel Corporation's Amended Answer, Affirmative Defenses, and Counterclaims to Plaintiff's First Amended Complaint (*MicroUnity Systems Engineering, Inc.* v. *Dell, Inc. f/k/a/ Dell Computer and Intel Corporation*; C.A. No. 2-04-cv-120; U.S.D.C., E.D. Texas, Marshall Division).
Mar. 9, 2005 Joint Claim Construction and Prehearing Statement with Exhibits A through F-2 (*MicroUnity Systems Engineering, Inc.* v. *Dell, Inc. f/k/a/ Dell Computer and Intel Corporation*; C.A. No. 2-04-cv-120; U.S.D.C., E.D. Texas, Marshall Division).
Apr. 11, 2005 MicroUnity Systems Ebgineering, Inc.'s Opening Brief Regarding Claim Construction Pursuant to Patent Local Rule 4-5(a) and Exhibits A-1 (*MicroUnity Systems Engineering, Inc.* v. *Dell, Inc. f/k/a/ Dell Computer and Intel Corporation*; C.A. No. 2-04-cv-120; U.S.D.C., E.D. Texas, Marshall Division).
Apr. 26, 2005 Supplement to Plaintiff MicroUnity Systems Engineering, Inc.'s Opening Brief Regarding Claim Construction (submitting a corrected p. 41) (*MicroUnity Systems Engineering, Inc.* v. *Dell, Inc. f/k/a/ Dell Computer and Intel Corporation*; C.A. No. 2-04-cv-120; U.S.D.C., E.D. Texas, Marshall Division).
May 12, 2005 Dell, Inc. and Intel Corporation's Responsive Brief Regarding Claim Construction Pursuant to Patent Local Rule 4-5(b) (*MicroUnity Systems Engineering, Inc.* v. *Dell, Inc. f/k/a/ Dell Computer and Intel Corporation*; C.A. No. 2-04-cv-120; U.S.D.C., E.D. Texas, Marshall Division).
May 25, 2005 MicroUnity's Reply Brief Regarding Claim Construction Pursuant to Patent Local Rule 4-5(c) (*MicroUnity Systems Engineering, Inc.* v. *Dell, Inc. f/k/a/ Dell Computer and Intel Corporation*; C.A. No. 2-04-cv-120; U.S.D.C., E.D. Texas, Marshall Division).
Jun. 9, 2005 Intel and Dell's Surreply Brief Regarding Claim Construction (*MicroUnity Systems Engineering, Inc.* v. *Dell, Inc. f/k/a/ Dell Computer and Intel Corporation*; C.A. No. 2-04-cv-120; U.S. D.C., E.D. Texas, Marshall Division).
MU v. *SCEA* Mar. 1, 2006 MicroUnity's Answer to SCEA's Counterclaim (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*; 2:05-cv-505; U.S.D.C., E.D. Texas, Marshall Division).
Jan. 25, 2007 First Amended Complaint for Patent Infringement and Exhibits A-J (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*; 2:05-cv-505; U.S.D.C., E.D. Texas, Marshall Division).
Feb. 12, 2007 SCEA's Answer, Affirmative Defenses, And Counterclaims to MicroUnity's First Amended Complaint (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*; 2:05-cv-505; U.S.D.C., E.D. Texas, Marshall Division).
Mar. 2, 2007 MircoUnity's Answer to SCEA's Counterclaim in Response to MicroUnity's First Amended Complaint (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*; 2:05-cv-505; U.S.D.C., E.D. Texas, Marshall Division).
Mar. 5, 2007 SCEA's Invalidity Contentions & Exhibits A-I (Exhibits G-9 and H-9 previously marked Outside Counsel Eyes Only, but now released) (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*; 2:05-cv-505; U.S.D.C., E.D. Texas, Marshall Division).
May 1, 2007 SCEA's Supplemental Invalidity Contentions & Exhibits A-I (Exhibits A-2, E-2, G-9 and H-7 previously marked Outside Counsel Eyes Only, but now released) (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*; 2:05-cv-505; U.S.D.C., E.D. Texas, Marshall Divison).
May 9, 2007 SCEA's Amended Answer, Affirmative Defenses, And Counterclaims to MircoUnity's First Amended Complaint (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*; 2:05-cv-505; U.S.D.C., E.D. Texas, Marshall Division).
May 29, 2007 MicroUnity's Answer to SCEA's Amended Counterclaim in Response to Microunity's First Amended Complaint (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*, 2:05-cv-505; U.S.D.C., E.D. Texas, Marshall Divison).
Jul. 11, 2007 P.R. 4-3 Joint Claim Construction Statement (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*, 2:05-cv-505, U.S.D.C., E.D. Texas, Marshall Division).
Jul. 11, 2007 SCEA's Second Amended Answer, Affirmative Defenses, And Counterclaims to MicroUnity's First Amended Complaint (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*, 2:05-cv-505 U.S.D.C., E.D. Texas, Marshall Divison).
Jul. 25, 2007 MU's Answer to SCEA's Second Amended Counterclaim in Response to MU's First Amended Complaint (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*, 2:05-cv-505 U.S.D.C., E.D. Texas, Marshall Divison).
Aug. 24, 2007 MU's LRP 4-5(a) Opening Brief on Claim Construction, and Exhibits 1-14 (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*, 2:05-cv-505, U.S.D. C., E.D. Texas, Marshall Divison).
Sep. 12, 2007 SCEA's Responsive Brief Regarding Claim Construction Pursuant to P.R. 4-5(b), and Exhibits 1-34 (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*, 2:05-cv-505, U.S.D.C., E.D. Texas, Marshall Divison).
Sep. 13, 2007 Notice of Filing of P.R. 4-5(d) Joint Claim Construction Chart, and Exhibit 1 (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*, 2:05-cv-505, U.S.D. C., E.D. Texas, Marshall Div).
Sep. 17, 2007 LPR 4-5(c) Reply Brief on Claim Construction from MircoUnity Systems Engineering, Inc., and Exhibits 15 and 16 (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*, 2:05-cv-505, U.S.D.C., E.D. Texas, Marshall Divison).
Sep. 20, 2007 Transcript of Claim Construction Hearing Before the Honorable T. John Ward United States District Judge (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*, 2:05-cv-505 U.S.D.C., E.D. Texas, Marshall Divison).
Oct. 15, 2007 SCEA's Motion for Partial Summary Judgment of Invalidity for U.S. Patent Nos. 6,643,765 and 6,725,356, and Proposed Order, and Exhibits A-U (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*, 2:05-cv-505, U.S.D.C., E.D. Texas, Marshall Divison).
Oct. 29, 2007 MU's Response Brief in Opposition to Sony's Motion for Partial Summary Judgment of Invalidity of U.S. Patent Nos. 6,643,765 and 6,725,356, and Proposed Order (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*, 2:05-cv-505, U.S.D.C., E.D. Texas, Marshall Divison).
Oct. 29, 2007 Declaration of Michael Heim in Support of MU's Response Brief in Opposition to Sony's Motion for Partial Summary Judgment of Invalidity of U.S. Patent Nos. 6,643,765 and 6,725,356, and Exhibits A-P (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*, 2:05-cv-505, U.S.D.C., E.D. Texas, Marshall Division).
Nov. 6, 2007 Order Granting Joint Motion to Stay Litigation Pending Settlement (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*, 2:05-cv-505, U.S.D.C., E.D. Texas, Marshall Division).
Dec. 12, 2007 Order of Dismissal with Prejudice and Final Judgment (*MicroUnity Systems Engineering, Inc.* v. *Sony Computer Entertainment America, Inc.*, 2:05-cv-505, U.S.D.C., E.D. Texas, Marshall Division).

MU v. *AMD* Nov. 22, 2006 Complaint Against Advanced Micro Devices, Inc. and Exhibits A-L (*MicroUnity Systems Engineering, Inc.* v. *Advanced Micro Devices, Inc.*; 2:06-cv-486; U.S.D.C., E.D. Texas, Marshall Division).
Feb. 26, 2007 AMD's Answer to MicroUnity's Complaint (*MicroUnity Systems Engineering, Inc.* v. *Advanced Micro Devices, Inc.*; 2:06-cv-486; U.S.D.C., E.D. Texas, Marshall Division).
Aug. 6, 2007 AMD's Invalidity Contentions Under Patent Rule 3-3, and Exhibits A-I (*MicroUnity Systems Engineering, Inc.* v. *Advanced Micro Devices, Inc.*, 2:06-cv-486; U.S.D.C., E.D. Texas, Marshall Division).
Sep. 5, 2007 AMD's Supplemental and Consolidated Invalidity Contentions—LPR 3-3, and Exhibits A-L (*MicroUnity Systems Engineering, Inc.* v. *Advanced Micro Devices, Inc.*, 2:06-cv-486; U.S.D.C., E.D. Texas, Marshall Division).
Sep. 13, 2007 First Amended Complaint, and Exhibits A-K (*MicroUnity Systems Engineering, Inc.* v. *Advanced Micro Devices, Inc.*, 2:06-cv-486; U.S.D.C., E.D. Texas, Marshall Division).
Oct. 1, 2007 AMD's Answer to First Amended Complaint (*MicroUnity Systems Engineering, Inc.* v. *Advanced Micro Devices, Inc.*, 2:06-cv-486; U.S.D.C., E.D. Texas, Marshall Division).
Jan. 8, 2008 Order of Dismissal With Prejudice and Final Judgment (*MicroUnity Systems Engineering, Inc.* v. *Advanced Micro Devices, Inc.*, 2:06-cv-486; U.S.D.C., E.D. Texas, Marshall Division).
The AMD-K6 3D Processor: Revolutionary Multimedia Performance, Ed. H. Kalish and J. Isaac, Abacus (1998).
BIT Data Sheet—Product Summary: B3110/B3120/B2110/B2120 Floating Point Chip Set, Bipolar Integrated Technology, Inc. (Dec. 1986).
BSP, Burroughs Scientific Processor, Burroughs Corporation, 1-29 (Jun. 1977).
BSP and BSP Customer Attributes: Inclosure 5, Burroughs Corporation, 1-3 (Aug. 1, 1977).
BSP Floating Point Arithmetic, Burroughs Corporation, 1-27 (Dec. 1978).
BSP Implementation of Fortran, Burroughs Corporation, E-1-E-19 (Feb. 1978).
Colwell et al., "Architecture and Implementation of a VLIW Supercomputer," *IEEE*, 910-19 (1990).
Colwell et al., "A VLIW Architecture for a Trace Scheduling Compiler," *IEEE Transactions on Computers*, vol. 37, No. 8, 967-79 (Aug. 1988).
Foley, "The Mpact™ Media Processor Redefines the Multimedia PC," *Proceedings of COMPCON Spring '96*, IEEE, 311-18 (1996).
Gajski et al., "Design of Arithmetic Elements for Burroughs Scientific Processor," *Proceedings of the 4th Symposium on Computer Arithmetic*, Santa Monica, CA, 245-56 (1978).
Hansen, "MicroUnity's MediaProcessor Architecture," *IEEE Micro* (Aug. 1996).
Higbie, "Applications of Vector Processing," *Computer Design*, 139-45 (Apr. 1978).
Hwang et al., *Computer Architecture and Parallel Processing*, McGraw Hill Book Co., Singapore (1988) (7 pages).
IEEE Standard for Scalable Coherent Interface (SCI), IEEE Std 1596-1992, © (1993).
"Intel MMX™ Technology Overview," Intel Corp. (Mar. 1996).
Kuck et al., "The Burroughs Scientific Processor (BSP)," *IEEE Transactions on Computers*, vol. C-31, No. 5, 363-76 (May 1982).
Kuck, "The Structure of Computers and Computation: vol. 1," John Wiley & Sons, Inc. (1978).
Lion Extension Architecture (Oct. 12, 1991).
Lowney et al., "The Multiflow Trace Scheduling Compiler," (Oct. 30, 1992).
"MIPS Digital Media Extension" (MDMX), rev. 1.0, C-1 to C-40, 1997.
"Multimedia Extension Unit for the X86 Architecture,"Compaq Computer Corp., Revision 0.8b (Jun. 20, 1995).
"PA-RISC 2.0 Architecture and Instruction Set Reference Manual," Hewlett-Packard Co. (1995).
Rathnam et al., "An Architectural Overview of the Programmable Multimedia Processor, TM-1," *Proceedings of COMPCON Spring '96*, IEEE (1996).
Rubinfeld, et al., "Motion Video Instruction Extensions for Alpha," Semiconductor Eng'g Group (Oct. 18, 1996).
Shanley, Pentium Pro Processor System Architecture, MindShare, Inc., Addison-Wesley Developers Press (1997).
Slater, "MicroUnity Lifts Veil on MediaProcessor," *Microprocessor Report* (Oct. 23, 1995).
TM1000 Preliminary Data Book, TriMedia Product Group (1997).
"TRACE /300 Series: F Board Architecture," *Multiflow Computer* (Dec. 9, 1988).
X86 64-Bit Extension Multimedia Instruction Set Architecture, Intel 64-bit Multimedia ISA Ratification Summit (Apr.-May 1992) [SCEA-1423558-655](*Previously marked Confidential—Counsel Eyes Only, but released for submission*).
Extended European Search Report dated Feb. 18, 2008 re Application No./Patent No. 07111351.8-1243.
L. Kohn et al. "The Visual Instruction Set (VIS) in UltraSPARC" IEEE, 1995. 462-469.
D. Shaver. "A General-Purpose Array Processor for Seismic Processing" (Nov.-Dec. 1984) Jan.-Mar. 1998. 15th Anniversary Issue. 5-26.
R. Lee. "Accelerating Multimedia with Enhanced Microprocessors" IEEE Micro. Apr. 1995. 22-32.
N. Margulis, "i860 Microprocessor Architecture" 1990. 8-10, 171-175, 182-183.
A. Levinthal et al, "Parallel Computers for Graphics Applications" 1987. 193-198.
K. Diefendorff et al. "Organization of the Motorola 88110 Superscalar RISC Microprocessor" IEEE Micro. Apr. 1992. 40-63.
L. Gwennap. "IBM Regains Performance Lead with Power2" Microprocessor Report. Oct. 4, 1993. vol. 7. No. 13. 1,6-10.
L. Gwennap. "IBM Creates Power PC Processors for AS/400" Microprocessor Report. Jul. 31, 1995. 15-16.
Markoff, John, "Intel Settlement Revives a Fading Chip Designer," The New York Times (Oct. 20, 2005).
Intel Press Release, "Intel Announces Record Revenue of $9.96 Billion," Santa Clara, CA, Oct. 18, 2005.
Wang, Yulun., et al. "The 3DP: A processor Architecture for Three-Dimensional Applications." Computer, IEEE Computer Society, vol. 25, No. 1, 1992, pp. 25-36, XP000287832, ISSN: 0018-9162.
Diefendorff, K., et al. "Organization of the Motorola 88110 Superscalar Risc Microprocessor." IEEE Micro, vol. 12, No. 2, Apr. 1, 1992, XP000266192, ISSN: 0272-1732.
Ide, et al., "A 320-MFLOPS CMOS Floating-point Processing Unit for Superscalar Processors," p. 12-21, Mar. 28, 1993, IEEE J. of Solid-State Circuits.
K. Uchiyama et al., The Gmicro/5OO Superscalar Microprocessor with. Branch Buffers, IEEE Micro, Oct. 1993, p. 12-21.
Ruby B. Lee, Realtime MPEG Video Via Software Decompression on a PA-RISC Processor, IEEE (1995).
Karl M. Guttag et al. "The TMS34010: An Embedded Microprocessor", IEEE Jun. 1988, p. 186-190.
M. Awaga et al., "The µVP 64-bit Vector Coprocessor: A New Implementation of High-Performance Numerical Computation", IEEE Micro, vol. 13, No. 5, Oct. 1993, p. 24-36.
Tom Asprey et al., "Performance Features of the PA7100 Microprocessor", IEEE Mlcro (Jun. 1993), p. 22-35.
Gove, Robert J., "The MVP: A Highly-Integrated Video Compression Chip," IEEE Data Compression Conf., Mar. 1994 , pp. 215-224.
Woobin Lee, et al., "Mediastation 5000: Integrating Video and Audio," IEEE Multimedia, 1994, pp. 50-61.
Karl, Guttag et. al "A Single-Chip Multiprocessor for Multimedia: The MVP," IEEE Computer Graphics & Applications, Nov. 1992, p. 53-64.
TMS32OC8O (MVP) Master Processor User's Guide, Texas Instruments, Mar. 1995, p. 1-33.
TMS320C80 (MVP) Parallel Processor User's Guide ["PP"]: Texas Instruments Mar. 1995 p. 1-80.
Shipnes, Julie, "Graphics Processing with the 88110 RISC Microprocessor," IEEE COMPCOM, (Spring,1992) pp. 169-174.
ILLIAC IV: Systems Characteristics and Programming Manual, May 1, 1972, p. 1-78.
N. Abel at al., ILLIAC IV Doc. No. 233, Language Specifications for a Fortran-Like Higher Level Language for Illiav IV, Aug. 28, 1970, p. 1-51.

ILLIAC IV Quarterly Progress Report: Oct. Nov. Dec. 1969; Published Jan. 15, 1970, pp. 1-15.
N.E. Abel et al., Extensions to Fortran for Array Processing (1970) pp. 1-16.
Morris A, Knapp et al.ILLIAC IV Systems Characteristics and Programming Manual (1972) "Bulk Storage Applications in the ILLIAC IV System," p. 1-10.
Rohrbacher, Donald, et al., "Image Processing with the Staran Parallel Computer," IEEE Computer, vol. 10, No. 8, pp. 54-59 (Aug. 1977) (reprinted version pp. 119-124).
Inter Partes Reexamination Communication issued in related Reexamination U.S. Appl. No. 95/000,100, mailed Sep. 8, 2005.
Supplemental Office Action issued in related Reexamination U.S. Appl. No. 95/000,100, mailed Sep. 26, 2005.
Response to Sep. 8, 2005 Reexamination Communication issued in related Reexamination U.S. Appl. No. 95/000,100, filed Dec. 8, 2005.
Inter Partes Reexamination Communication issued in related Reexamination U.S. Appl. No. 95/100,100, mailed May 3, 2006.
Response to May 3, 2006 Reexamination Communication issued in related Reexamination U.S. Appl. No. 95/000,100, filed on Jun. 30, 2006.
Inter Partes Reexamination Communication issued in related Reexamination U.S. Appl. No. 95/000,089, mailed Jun. 1, 2005.
Response to Jun. 1, 2005 Reexamination Communication issued in related Reexamination U.S. Appl. No. 95/000,089, filed Oct. 3, 2005.
Inter Partes Reexamination Communication issued in related Reexamination U.S. Appl. No. 95/000,089, mailed Mar. 2, 2006.
Response to Mar. 2, 2006 Reexamination Communication issued in related Reexamination U.S. Appl. No. 95/000,089, filed May 2, 2006.
Siegel, Howard Jay, "Interconnection Networks for SIMO Machines," IEEE Computer, vol. 12, No. 6, (Jun. 1979) (reprinted version pp. 110-118).
Mike Chastain, et. al., "The Convex C240 Architecture", Conference of Supercomputing, IEEE 1988, p. 321-329.
Gwennap, Linley, "New PA-RISC Processor Decodes MPEG Video: HP's PA-71 00LC Uses New Instructions to Eliminate Decoder Chip," Microprocessor Report, (Jan. 24, 1994) pp. 16-17.
Patrick Knebel et al., "HP's PA7100LC: A Low-Cost Superscalar PARISC Processor," IEEE (1993), pp. 441-447.
Kurpanek et al., "PA7200: A PA-RISC Processor with Integrated High Performance MP Bus Interface," EEEE (1994), pp. 375-382.
Hewlett Packard, PA-RISC 1.1 Architecture and Instruction Set Reference Manual, 3rd ed. Feb. 1994, pp. 1-424.
Margaret Simmons, et. al "A Performance Comparison of Three Supercomputers—Fujitsu VP-2600, NEC SX-3, and Cray Y-MP",. 1991 ACM, p. 150-157.
Smith, J. E., "Dynamic Instruction Scheduling and the Astronautics ZS-1," Computer, vol. 22, No. 7, Jul. 1989, at 21-35 and/or the Astronautics ZS- 1 computers made used, and/or sold in the United States, pp. 159-173.
Nikhil et al., "T: A Multithreaded Massively Parallel Architecture" Computation Structures Group Memo 325-2 (Mar. 5, 1992) , pp. 1-13.
Undy, et al., "A Low-Cost Graphics and Multimedia Workstation Chip Set," IEEE pp. 10-22 (1994).
Feng, Tse-Yun, "Data Manipulating Functions in Parallel Processors and Their Implementations," IEEE Transactions on Computers, vol. C-23, No. 3, Mar. 1974 (reprinted version pp. 89-98.
Lawrie, Duncan H., "Access and Alignment of Data in an Array Processor," IEEE Transactions on Computers, vol. c-24, No. 12, Dec. 1975 pp. 99-109.
Broomell, George, et al., "Classification Categories and Historical Development of Circuit Switching Topologies," Computing Surveys, vol. 15, No. 2, Jun. 1983 pp. 95-133.
Jain, Vijay, K., "Square-Root, Reciprocal, SINE/COSINE, ARCTANGENT Cell for Signal and Image Processing," IEEEICASSP'94 Apr. 1994 , pp. II-521-II-524.
Spaderna et al., "An Integrated Floating Point Vector Processor for DSP and Scientific Computing", 1989 IEEE, ICCD, Oct. 1989 p. 8-13.
Gwennap, Linley, "Digital, MIPS Add Multimedia Extensions," Microdesign Resources Nov. 18, 1996 pp. 24-28.
Toyokura, M., "A Video DSP with a Macroblock-Level-Pipeline and a SIMD Type Vector-Pipeline Architecture for MPEG2 CODEC," ISSCC94, Section 4, Video and Communications Signal Processors, Paper WP 4.5, 1994 pp. 74-75.
Ide, et al., "A 320-MFLOPS CMOS Floating-point Processing Unit for Superscalar Processors," Nobuhiro Ide, et. Al. IEEE Tokyo Section, Denshe Tokyo No. 32, 1993, p. 103-109.
Papadopoulos et al., "T: Integrated Building Blocks for Parallel Computing," ACM (1993) p. 824- and p. 625-63.5.
Ruby B. Lee, "Accelerating Multimedia with Enhanced Microprocessor," IEEE Micro Apr. 1995 p. 22-32.
Ruby B. Lee, "Realtime MPEG Video Via Software Decompression on a PA-RISC Processor," IEEE (1995), pp. 186-190.
K. Diefendorff , M. Allen, The Motorola 88110 Superscalar RISC Microprocessor, IEEE Micro, Apr. 1992, p. 157-162.
Kristen Davidson, Declaration of Kristen Davidson, p. 1 and H. Takahashi et al., A 289 MFLOPS Single Chip Vector Processing Unit, The Institute of Electronics, Information, and Communication Engineers Technical Research Report, May 28, 1992, pp. 17-22.
Kristen Davidson, Declaration of Kristen Davidson, p. 1 and M. Kimura et al., Development of Ginicro 32-bit Family of Microprocessors, Fujitsu Semiconductor Special Part 2, vol. 43, No. 2 Feb. 1992.
Bit Manipulator, IBM Technical Disclosure Bulletin, Nov. 1974, pp. 1575-1576 https://www.delphion.com/tdbs/tdb?order=75C+0016.
"Using a Common Barrel Shifter for Operand Normalization, Operand Alignment and Operand Unpack and Pack in Floating Point," IBM Technical Disclosure Bulletin, Jul. 1986, p. 699-701 https://www.delphion.com/tdbs/tdb?order=86A+6157B.
Motorola MCBB110 Second Generation RISC Microprocessor User's Manual (1991).
Berkerele, Michael J., "Overview of the START (*T) Multithreaded Computer" IEEE Jan. 1993, p. 148-1 56.
Diefendorff, et al., "Organization of the Motorola 88110 Superscalar RISC Microprocessor" IEEE Micro Apr. 1992, p. 39-63.
Barnes, et al., The ILLIAC IV Computer, IEEE Transactions on Computers, vol. C-17, No. 8, Aug. 1968.
Ruby B. Lee et al., Real-Time Software MPEG Video Decoder on Multimedia-Enhanced PA 7 100LC Processors, Hewlett-Packard J. Apr. 1995, p. 60-68.
Ruby B. Lee, "Realtime MPEG Video Via Software Decompression on a PA-RISC Processor," IEEE 1995, p. 186-192.
"The Multimedia Video Processor (MVP): A Chip Architecture for Advanced DSP Applications," Robert J. Gove, IEEE DSP Workshop (1994).
Convex Assembly Language Reference Manual, First Ed., Dec. 1991.
Convex Architecture Reference Manual (C Series), Sixth Edition, Convex Computer Corporation (Apr. 1992).
Manferdelli, et al., "Signal Processing Aspects of the S-1 Multiprocessor Project," submitted to SPIE Annual International Technical Symposium, Sm Diego, Society of Photo Optical Instrumentation Engineers, Jul. 30, 1980, p. 1-8.
Paul Michael Farmwald, Ph.D. "On the Design of High-Performance Digital Arithmetic Units," Thesis, Aug. 1981, p. 1-95.
GsAs Supercomputer Vendors Hit Hard,, Electronic News, Jan. 31, 1994, 1991, pp. 32.
Convex Adds GaAs System, Electronic News, Jun. 20, 1994.
Kevin Wadleigh et al., High-Performance FFT Algorithms for the Convex C4/XA Supercomputer, Journal of Super Computing, vol. 9, pp. 163-178 (1995).
Peter Michielse, Programming the Convex Exemplar Series SPP System, Parallel Scientific Computing, First Intl Workshop, PARA '94, Jun. 20-23, 1994, pp. 375-382.
Ryne, Robert D., "Advanced Computers and Simulation," Los Alamos National Laboratory IEEE 1 993, p. 3229-3233.
Singh et al., "A Programmable HIPPI Interface for a Graphics Supercomputer," ACM (1993) p. 124-132.
Bell, Gordon, "Ultracomputers: A Terflop Before its Time," Comm.'s of the ACM Aug. 1992 pp. 27-47.
Geist, G. A., "Cluster Computing: The Wave of the Future?" Oak Ridge National Laboratory, 84OR2 1400 May 30, 1994, p. 236-246.

Vetter et al., "Network Supercomputing," IEEE Network May 1992, p. 38-44.
Renwick, John K. "Building a Practical HIPPI LAN," IEEE 1992, p. 355-360.
Tenbrink, et al., "HIPPI: The First Standard for High-Performance Networking," Los Alamos Science 1994 p. 1-4.
Arnould at al., "The Design of Nectar: A Network Backplane for Heterogeneous Multicomputers," ACM 1989 p. 1-12.
Watkins, John, et al., "A Memory Controller with an Integrated Graphics Processor," IEEE 1993 p. 324-336.
Control Data 6400/6500/ 6600 Computer Systems, Instant SMM Maintenance Manual, Nov. 1, 1969.
Control Data 6400/6500/ 6600 Computer Systems, SCOPE Reference Manual, Sep. 1966.
Control Data 6400/6500/ 6600 Computer Systems COMPASS Reference Manual, 1969.
Tolmie, Don, "Gigabit LAN Issue: HIPPI, Fibre Channel, or ATM?" Los Alamos National Laboratory Rep. No. LA-UR 94-3994 (1994).
ILLIAC IV: Systems Characteristics and Programming Manual, May 1, 1972.
1979 Annual Report: The S-1 Project vol. 1 Architecture 1979.
1979 Annual Report:The S-1 Project vol. 2 Hardware 1979.
S-1 Uniprocessor Architecture, Apr. 21, 1983 (UCID 19782) See also S-1 Uniprocessor Architecture (SMA-4), Steven Cornell.
Broughton, et al., The S-1 Project: Top-End Computer Systems for National Security Applications, Oct. 24, 1985.
Convex Data Sheet C4/XA High Performance Programming Environment, Convex Computer Corporation, 1994.
Bowers et al., "Development of a Low-Cost, High Performance, Multiuser Business Server System," Hewlett-Packard J. Apr. 1995 p. 79-84.
Mick Bass et al., The PA 7100LC Microprocessor: A Case Study of Design Decisions in a Competitive Environment Hewlett-Packard J. Apr. 1995, p. 12-18.
Mick Bass, et. al. "Design Methodologies for the PA 7100LC Microprocessor", Hewlett Packard Journal Apr. 1995 p. 23-35.
Wang, Chin-Liang, "Bit-Level Systolic Array for Fast Exponentiation in GF(2Am)," IEEE Transactions on Computers, vol. 43, No. 7, Jul. 1994 p. 838-841.
Markstein, P.W., "Computation of Elementary Functions on the IBM RISC System/6000 Processor," IBM J. Res. Develop., vol. 34, No. 1, Jan. 1990 p. 111-119.
Donovan, Walt, et al., "Pixel Processing in a Memory Controller," IEEE Computer Graphics and Applications, Jan. 1995 p. 51-61.
Ware at al., 64 Bit Monolithic Floating Point Proceseors, IEEE Journal Of Solid-state Circuits, vol. Sc-17, No. 5, Oct. 1982, pp. 898-907.
Hwang, "Advanced Computer Architecture: Parallelism, Scalability, Programmability" (1 993) at 475, p. 898-907.
Hwang & Degroot, "Parallel Processing for Supercomputers & Artificial Intelligence," 1993.
Nienhaus, Harry A., "A Fast Square Rooter Combining Algorithmic and Table Lookup Techniques," IEEE Proceedings Southeastcon, 1989 pp. 1103-1105.
Eisig, David, et al., "The Design of a 64-Bit Integer Multiplier/Divider Unit," IEEE 1993 pp. 171-178.
Margulis, Neal, "i860 Microprocessor Architecture," Intel Corporation 1990.
Intel Corporation, 3860 XP Microprocessor Data Book (May 1991).
Hewlett-Packard, "HP 9000 Series 700 Workstations Technical Reference Manual Model 712 (System)" Jan. 1, 1 994.
Ruby Lee, et al., Pathlength Reduction Features in the PA-RISC Architecture Feb. 24-28, 1992 p. 129-135.
Kevin Wadleigh et al., High Performance FFT Algorithms for the Convex C4/XA Supercomputer, Poster, Conference on Supercomputing, Washington, D.C., Nov. 1994.
Fields, Scott, "Hunting for Wasted Computing Power: New Software for Computing Networks Puts Idle PC's to Work," Univ. of Wisconsin- Madison 1993 p. 1-8.
Litzkow et al., "Condor—A Hunter of Idle Workstations," IEEE (1 988) p. 104-111.
Gregory Wilson, The History of the Development of Parallel Computing http://ei.cs.vt.edu/-history/Parallel.html, p. 1-38, Oct. 1994.
Marsha Jovanovic and Kimberly Claffy, Computational Science: Advances Through Collaboration "Network Behavior" San Diego Supercomputer Center 1993 Science Report, p. 1-11 [http://www.sdsc.edu/Publications/SR93/network_behavior.html], 1993.
National Science Foundation (NSF) [www.itrd.gov/pubs/blue94/section.4.2.html] 1994.
Intel Corporation, "Paragon User's Guide" (Oct. 1993).
Turcotte, Louis H., "A Survey of Software Environments for Exploiting Networked Computing Resources" Engineering Research Center for Computational Field Simulation Jun. 11, 1993, p. 1-150.
Patterson, Barbara, "Motorola Announces First High Performance Single Board Computer Using Superscaler Chip" Motorola Computer Group, p. 1-3 [http://badabada.org/misc/mvme197_announce.txt], Aug. 20, 1992.
Culler, David E., et al., "Analysis Of Multithreaded Microprocessors Under Multiprogramming", Report. No. UCBICSD 921687, May 1992 p. 1-17.
James Laudon et al, "Architectural and Implementation Tradeoffs in the Design of Multiple-Context Processors", CSL-TR-92-523, May 1992 p. 1-24.
Ide, et al., "A 320-MFLOPS CMOS Floating-point Processing Unit for Superscalar Processors," 28 IEEE Custom Integrated Circuits Conference, 1992, p. 30.2.1-30.2.4.
High Speed DRAMs, Special Report, IEEE Spectrum, vol. 29, No. 10, Oct. 1992.
Moyer, Steven A.., "Access Ordering Algorithms for a Multicopy Memory," IPC-TR-92-0 1 3, Dec. 18, 1992.
Moyer, Steven A., "Access Ordering and Effective Memory Bandwidth," Ph.D. dissertation, University of Virginia, Apr. 5, 1993.
"Hardware Support for Dynamic Access Ordering: Performance of Some Design Options", Sally McKee, Computer Science Report No. CS-93-08, Aug. 9, 1993.
McGee et al., "Design of a Processor Bus Interface ASIC for the Stream Memory Controller" p. 462-465, Apr. 1994.
McKee et al., "Experimental Implementation of Dynamic Access Ordering ," Aug. 1, 1993, p. 1-10.
McKee et al., lncreasing Memory Bandwidth for Vector Computations, Technical Report CS-93-34 Aug. 1, 1993, p. 1-18.
Sally A. McKee et al., "Access Order and Memory-Conscious Cache Utilization" Computer Science Report No. CS-94- 10, Mar. 1, 1994, p. 1-17.
McKee, et. al., "Bounds on Memory Bandwidth in Streamed Computations," Computer Science Report CS-95-32, Mar. 1, 1995.
McKee, Sally A., "Maximizing Memory Bandwidth for Streamed Computation," A Dissertation Presented to the Faculty of the School of Engineering and Applied Science at the University of Virginia, May 1995.
A Systematic Approach to Optimizing and Verifying Synthesized High-Speed ASICs, Trevor Landon, et. Al. , Computer Science Report No. CS-95-51, Dec. 11, 1995.
"Control Data 6400/6500/ 6600 Computer Systems Reference Manuals" 1969 available at http:/led-thelen.org/comp-hist/CDC-6600-R-M.html ("CDC 6600 Manuals").
"Where now for Media processors?", Nick Flaherty, Electronics Times, Aug. 24, 1998.
George H. Barnes et al., The ILLIAC IV Computer' , 'IEEE Trans., C-17 vol. 8, pp. 746-757, Aug. 1968.
J.E. Thornton, Design of a Computer—The Control Data 6600 (1970).
Gerry Kane, PA-RISC 2.0 Architecture, Chp. 6 Instruction Set Overview, Prentice Hall isbn 0-13-182734-0, p. 6-1-6-26, 1995.
Cosoroaba, A.B., "Synchronous DRAM products revolutionize memory system design," Fujitsu Microelectronics, Southcod95 May 709 1995.
Intel 450KX/GX PCIset, Inetel Corporation, 1996.
Baland, Granito, Marcotte, Messina, Smith, "The IBM System1360 Model 91 : Storage System" IBM System Journal, Jan. 1967, pp. 54-68.
File History of U.S. Appl. No. 08/340,740, filed Nov. 16, 1994.
File history of U.S. Appl. No. 07/663,314, filed Mar. 1, 1991.
S.S. Reddi et. al. "A Conceptual Framework for Computer Architecture" Computing Surveys,. vol. 8, No. 2, Jun. 1976.

Yulun Wang, et al, The 3DP: A processor Architecture for Three-Dimensional Applications, Jan. 1992, p. 25-36.
"IEEE Draft Standard for High-Bandwidth Memory Interface Based on SCI Signaling Technology (RamLink)", 1995, pp. 1-104, IEEE.
Gerry Kane and Joe Heinrich, "MIPS RISC Architecture" 1992, Publisher: Prentice-Hall Inc., A Simon & Shuster Company, Upper Saddle River New Jersey.
Cathy May et al. "The Power PC Architecture: A Specification For A New Family of Risc Processors" Second Edition May 1994, pp. 1-518, Morgan Kaufmann Publishers, Inc. San Franciso CA, IBM International Business Machines, Inc.
"IEEE Standard for Scalable Coherent Interface (SCI)" , Published by the Institute of Electrical and Electronics Engineers, Inc. Aug. 2, 2003, pp. 1-248.
Don Tolmie and Don Flanagan, "HIPPI: It's Not Just for Supercomputers Anymore" Data Communications published May 8, 1995.
IEEE Draft Standard for "High-Bandwidth Memory Interface Based on SCI Signaling Technology (RamLink)", IEEE Standards Department, Draft 1.25 IEEE P1596.4-199X May 1995.
Joe Heinrich, "MIPS R4000 Microprocessor User's Manual Second Edition"1994 MIPS Technologies, Inc. pp. 1-754.
Litigation proceedings in the matter of *Microunity Systems Engineering, Inc.* v. *Dell, Inc.* et al., Corrected Preliminary Invalidity Contentions and Exhibits, filed Jan. 12, 2005, Civil Action No. 2:04-CV-120(TJW), U.S. District Court for the Eastern District of Texas Marshall Division.
Ang, StarT Next Generation: Integrating Global Caches and Dataflow Architecture, Proceedings of the ISCA 1992.
Saturn Architecture Specification, published Apr. 29, 1993.
C4/XA Architecture Overview, Convex Technical Marketing presentation dated Nov. 11, 1993 and Feb. 4, 1994.
Convex 3400 Supercomputer System Overview, published Jul. 24, 1991.
Gilol, Parallel Programming Models and Their Interdependence with Parallel Architectures, IEEE Proceedings published Sep. 1993.
PCT International Search Report and Written Opinion dated Mar. 11, 2005, corresponding to PCT/US04/22126.
Supplementary European Search Report dated Mar. 18, 2006, corresponding to Application No. 96928129.4.
IEEE Draft Standard for "Scalable Coherent Interface-Low-Voltage Differential Signal Specifications and Packet Encoding", IEEE Standards Department, P1596.3/D0.15 (Mar. 1992) (5.0006DOC018530-563).
IEEE Draft Standard for "High-Bandwidth Memory Interface Based on SCI Signaling Technology (RamLink)," IEEE Standards Department, Draft 1.25 IEEE P1596.4-199X (May 1955) (50006DOC018413-529).
Gerry Kane et al., "MIPS RISC Architecture," Prentice Hall (1995) (50006DOC018576-848).
IBM, "The PowerPC Architecture: A Specification For A New Family of RISC Processors," 2nd Ed., Morgan Kaufmann Publishers, Inc., (1994) (50006DOC019229-767).
Hewlett-Packard Co., "PA-RISC 1.1 Architecture and Instruction Set," Manual Part No. 09740-90039, (1990) (5006DOC018849-19228).
MIPS Computer Systems, Inc., "MIPS R4000 User's Manual," Mfg. Part No. M8-00040, (1990) (50006DOC017026-621).
IB60™ Microprocessor Architecture, Neal Margulis, Foreword by Les Kohn, (1990).
Gove, "The MVP: A Highly-Integrated Video Compression Chip," IEEE Data Compression Conference, pp. 215-224 (Mar. 1994) (51056DOC000891-990).
Gove, "The Multimedia Video Processor (MVP): A Chip Architecture for Advanced DSP Applications," IEEE DSP Workshop, pp. 27-30 (Oct. 2-5, 1994) (51056DOC015452-455).
Guttag et al., "A Single-Chip Multiprocessor for Multimedia: The MVP," IEEE Computer Graphics & Applications, pp. 53-64 (Nov. 1992) (51056DOC000913-924).
Lee et al., "MediaStation 5000: Intergrating Video and Audio," IEEE Multimedia pp. 50-61 (Summer 1994) (51056DOC000901-912).
TMS320C80 (MVP) Parallel Processor User's Guide, Texas Instruments (Mar. 1995) (51056DOC003744-4437).
TMS320C80 (MVP) Master Processor User's Guide, Texas Instruments (Mar. 1995) (51056DOC000925-957).
Bass et al., "The PA 7100LC Microprocessor: A Case Study of IC Design Decisions in a Competitive Environment," Hewlett-Packard Journal, vol. 46, No. 2, pp. 12-22 (Apr. 1995) (51056DOC059283-289).
Bowers et al., "Development of a Low-Cost, High Performance, Multiuser Business Server System," Hewlett-Packard Journal, vol. 46, No. 2, p. 79 (Apr. 1995) (51056DOC059277-282).
Gwennap, "New PA-RISC Processor Decodes MPEG Video: Hewlett-Packard's PA-7100LC Uses New Instructions to Eliminate Decoder Chip," Microprocessor Report, pp. 16-17 (Jan. 24, 1994) (51056DOC002140-141).
Gwennap, "Digital MIPS Add Multimedia Extensions," Microdesign Resources, pp. 24-28 (Nov. 18, 1996) (51056DOC003454-459).
Kurpanek et al., "PA7200: A PA-RISC Processor with Integrated High Performance MP Bus Interface," IEEE COMPCON '94, pp. 375-382 (Feb. 28-Mar. 4, 1994) (51056DOC002149-156).
Lee et al., "Pathlength Reduction Features in the PA-RISC Architecture," IEEE COMPCON, pp. 129-135 (Feb. 24-28, 1992) (51056DOC068161-167).
Lee et al., "Real-Time Software MPEG Video Decoder on Multimedia-Enhanced PA 7100LC Processors," Hewlett-Packard Journal, vol. 46, No. 2, pp. 60-68 (Apr. 1995) (51056DOC013549-557).
Leo, "Realtime MPEG Video via Software Decompression on a PA-RISC Processor," IEEE, pp. 186-192 (1995) (51056DOC007345-351).
Martin, "An Integrated Graphics Accelerator for a Low-Cost Multimedia Workstation," Hewlett-Packard Journal, vol. 46, No. 2, pp. 43-50 (Apr. 1995) (51056DOC072083-090).
Undy et al., "A Low-Cost Graphics and Multimedia Workstation Chip Set," IEEE Micro, pp. 10-22 (Apr. 1994) (51056DOC002578-590).
HP 9000 Series 700 Workstations Technical Reference Manual: Model 712, Hewlett-Packard (Jan. 1994) (51056DOC068048-141).
PA-RISC 1.1 Architecture and Instruction Set Reference Manual, Third Edition, Hewlett-Packard (Feb. 1994) (51056DOC002157-176).
Ang, "StarT Next Generation: Integrating Global Caches and Dataflow Architecture," Proceedings of the ISCA 1992 Dataflow Workshop (1992) (51056DOC071743-776).
Beckerle, "Overview of the StarT (*T) Multithreaded Computer," IEEE COMPCON '93, pp. 148-156 (Feb. 22-26, 1993) (51056DOC002511-519).
Diefendorff et al., "The Motorola 88110 Superscalar RISC Microprocessor," IEEE pp. 157-162 (1992) (51056DOC008746-751).
Gipper, "Designing Systems for Flexibility, Functionality, and Performance with the 88110 Symmetric Superscalar Microprocessor," IEEE (1992) (51056DOC008758-763).
Nikhil et al., "T: A Multithreaded Massively Parallel Architecture," Computation Structures Group Memo 325-2, Laboratory for Computer Science, Massachusetts Institute of Technology (Mar. 5, 1992) (51056DOC002464-476).
Papadopoulos et al., "*T: Integrated Building Blocks for Parallel Computing," ACM, pp. 624-635 (1993) (51056DOC007278-289).
Patterson, "Motorola Announces First High Performance Single Board Computer Using Superscalar Chip," Motorola Computer Group (Sep. 1992) (51056DOC069260-262).
M. Phillip, "Performance Issues for 88110 RISC Microprocessor," IEEE, 1992 (51056DOC008752-757).
M. Smotherman et al., "Instruction Scheduling for the Motorola 88110," IEEE, 1993 (51056DOC008784-789).
R. Mueller, "The MC88110 Instruction Sequencer," Northcon, 1992 (51056DOC009735-738).
J. Arends, "88110: Memory System and Bus Interface," Northcon, 1992 (51056DOC009739-742).
K. Pepe, "The MC88110's High Performance Load/Store Unit," Northcon, 1992 (51056DOC009743-747).
J. Maguire, "MC8810: Datpath," Northcon, 1992 (51056DOC010059-063).

Abel et al., "Extensions to FORTRAN for Array Processing," ILLIAC IV Document No. 235, Department of Computer Science, University of Illinois at Urbana-Champaign (Sep. 1, 1970) (51056DOC001630-646).
Barnes et al., "The ILLIAC IV Computer," IEEE Transactions on Computers, vol. C-17, No. 8, pp. 746-757 (Aug. 1968) (51056DOC012650-661).
Knapp et al., "Bulk Storage Applications in the ILLIAC IV System," ILLIAC IV Document No. 250, Center for Advanced Computation, University of Illinois at Urbana-Champaign (Aug. 3, 1971) (51056DOC001647-656).
Awaga et al., "The μVP 64-bit Vector Coprocessor: A New Impiementation og High-Performance Numerical Computation," IEEE Micro, vol. 13, No. 5, pp. 24-36 (Oct. 1993) (51056DOC011921-934).
Takahashi et al., "A 289 MFLOPS Single Chip Vector Processing Unit," The Institute of Electronics, Information, and Communication Engineers Technical Research Report, pp. 17-22 (May 28, 1992) (51056DOC009798-812).
Uchiyama et al., "The Gmicro/500 Superscalar Microprocessor with Branch Buffers," IEEE Micro (Oct. 1993) (51056DOC00185-194).
Broughton et al., "The S-1 Project: Top-End Computer Systems for National Security Applications," (Oct. 24, 1985) (51056DOC057368-607).
Farmwald et al., "Signal Processing Aspects of the S-1 Multiprocessor Project," SPIE vol. 241, Real-Time Signal Processing (1980) (51056DOC072280-291).
Farmwaid, "High Bandwidth Evaluation of Elementary Functions," IEEE Proceedings, 5th Symposium on Computer Arithmetic (1981) (51056DOC071029-034).
Gilbert, "An Investigation of the Partitioning of Algorithms Across an MIMD Computing System," (Feb. 1980) (51056DOC072244-279).
Widdoes, "The S-1 Project: Developing High-Performance Digital Computers," IEEE Computer Society COMPCON Spring 1980 (Dec. 11, 1979) (51056DOC071574-585).
Cornell, S-1 Uniprocessor Architecture SMA-4 (51056DOC056505-895), 1979.
The S-1 Project, Jan. 1985, S-1 Technical Staff (51056DOC057368-607).
S-1 Architecture and Assembler SMA-4 Manual, Dec. 19, 1979 (Preliminary Version) (51056DOC057608-918).
Michielse, "Performing the Convex Exemplar Series SPP System," Proceedings of Parallel Scientific Computing, First Intl Workshop, PARA '94, pp. 375-382 (Jun. 20-23, 1994) (51056DOC020754-758).
Wadleigh et al., "High Performance FFT Algorithms for the Convex C4/XA Supercomputer," Poster, Conference on Supercomputing, Washington, D.C. (Nov. 1994) (51056DOC068618).
C4 Technical Overview (Sep. 23, 1993) (51056DOC017111-157).
Saturn Assembly Level Performance Tuning Guide (Jan. 1, 1994) (51056DOC017369-376).
Saturn Differences from C Series (Feb. 6, 1994) (51056DOC017150-157).
"Convex Adds GaAs System," Electronic News (Jun. 20, 1994) (51056DOC019388-390).
Convex Architecture Reference Manual, Sixth Edition (1992) (51056DOC016599-993).
Convex Assembly Language Reference Manual, First Edition (Dec. 1991) (51056DOC015996-6598).
Convex Data Sheet C4/XA Systems, Convex Computer Corporation (51056DOC059235-236), 1994.
Saturn Overview (Nov. 12, 1993) (51056DOC017111-157).
Convex Notebook containing various "Machine Descriptions" (51056DOC016994-7510), 1994.
"Convex C4/XA Offer 1 GFLOPS from GaAs Uniprocessor," Computergram International, Jun. 15, 1994 (51056DOC019383).
Excerpt from Convex C4600 Assembly Language Manual, 1995 (51056DOC061441-443).
Excerpt from "Advanced Computer Architectures—A Design Space Approach," Chapter 14.8, "The Convex C4/XA System" (51056DOC061453-459), Jan. 17, 2005.
Convex C4600 Assembly Language Manual, First Edition, May 1995 (51056DOC064728-5299).
Alvarez et al., "A 450MHz PowerPC Microprocessor with Enhanced Instruction Set and Copper Interconnect," ISSCC (Feb. 1999) (51056DOC071393-394).
Complaint for Patent Infringement, *Microunity Systems Engineering, Inc.* v. *Sony Corporation of America*, Civil Action No. 2:05 CV 505, Filed Nov. 5, 2005.
Convex Data Sheet C4/XA System, Convex Computer Corporation (51056DOC059235-236), 1994.
Margulis, "i860 Microprocessor Architecture," Intel Corporation (1990) (51056DOC066610-7265 and 5156DOC069971-70626).
Paragon User's Guide, Intel Corporation (Oct. 1993) (51056DOC068802-9097).
Hwang, "Computer Architecture and Parallel Processing," McGraw Hill (1984) (51056DOC070166-1028).
Data General AViiON AV500, 550, 4500 and 5500 Servers.
Desposition of Larry Mennemeier on Sep. 22, 2005 and Exhibit 501; *MicroUnity Systems Engineering, Inc.* v. *Dell, Inc. f/k/a/ Dell Computer and Intel Corporation*; C.A. No. 2-04CV-120; In the United States District Court of the Eastern District of Texas, Marshall Division.
Deposition of Leslie Kohn on Sep. 22, 2005; *MicroUnity Systems Engineering, Inc.* v. *Dell, Inc. f/k/a/ Dell Computer and Intel Corporation*, C.A. No. 2-04CV-120; In the United States District Court of the Eastern District of Texas, Marshall Divison.
Sony Computer Entertainment America, Inc.'s Answer, Affirmative Defenses, and Counterclaim to Microunity Systems Engineering, Inc,'s Original Complaint, *Microunity Systems Engineering, Inc.* v. *Sony Corporation of America*, Civil Action No. 2:05 CV 505, Filed Feb. 14, 2006.
Tyler et el., "AltiVec™; Bringing Vector Technology to the PowerPC™ Processor Family," IEEE (Feb. 1999) (51056DOC071035-042).
AltiVec™ Technology Programming Environments Manual (1998) (51056DOC071043-392).
Atkins, "Performance and the i860 Microprocessor," IEEE Micro, pp. 24-27, 72-78 (Oct. 1991) (5156DOC070655-666).
Grimes et al., "A New Processor with 3-D Graphics Capabilities," NCGA '89 Conference Proceedings vol. 1, pp. 275-284 (Apr. 17-20, 1989) (5156DOC070711-717).
Grimes et al., "The Intel i860 64-Bit Processor: A General-Purpose CPU with 3D Graphics Capabilities," IEEE Computer Graphics & Applications, pp. 85-94 (Jul. 1989) (5156DOC070701-710).
Kohn et al., "A 1,000,000 Transistor Microprocessor," 1989 IEEE International Solid-States Circuits Conference Digest of Technical Papers, pp. 54-55, 290 (Feb. 15, 1989) (51056DOC072091-094).
Kohn et al., "A New Microprocessor with Vector Processing Capabilities," Electro/89 Conference Record, pp. 1-6 (Apr. 11-13, 1989) (5156DOC070672-678).
Kohn et al., "Introducing the Intel i860 64-Bit Microprocessor," IEEE Micro, pp. 15-30 (Aug. 1989) (5156DOC070627-642).
Kohn et al., "The i860 64-Bit Supercomputing Microprocessor," AMC, pp. 450-456 (1989) (51056DOC000330-336).
Mittal et al., "MMX Technology Architecture Overview," Intel Technology Journal Q3 '97, pp. 1-12 (1997) (5156DOC070689-700).
Patel et al., "Architectural Features of the i860—Microprocessor RISC Core and On-Chip Caches," IEEE, pp. 385-390 (1989) (5156DOC070679-684).
Rhodehamel, "The Bus Interface and Paging Units of the i860 Microprocessor," IEEE, pp. 380-384 (1989) (5156DOC070643-647).
Perry, "Intel's Secret is Out," IEEE Spectrum, pp. 22-28 (Apr. 1989) (5156DOC070648-654).
Sit et al., "An 80 MFLOPS Floating-Point Engine in the Intel i860 Processor," IEEE, pp. 374-379 (1989) (51056DOC072095-101).
i860 XP Microprocessor Data Book, Intel Corporation (May 1991) (51056DOC067265-427).
N15 Micro Architecture Specification, dated Apr. 29, 1991 (50781DOC000001-982).
N15 External Architecture Specification, dated Oct. 17, 1990 (51056DOC017511-551).
N15 External Architecture Specification, dated Dec. 14, 1990 (50781DOC001442-509).
N15 Product Requirements Document, dated Dec. 21, 1990 (50781DOC001420-441).

N15 Product Implementation Plan, dated Dec. 21, 1990 (50781DOC001794-851).
N12 Performance Analysis document version 2.0, dated Sep. 21, 1990 (51056DOC072992-73027).
Hansen, "Architecture of a Broadband Mediaprocessor," IEEE COMPCON 96 (Feb. 25-29, 1996) (MU0013276-283 and 51057DOC001825-831).
Moussouris et al., "Architecture of a Broadband MediaProcessor," Microprocessor Forum (1995) (MU0048611-630).
Arnould et al., "The Design of Nectar: A Network Backplane for Heterogeneous Multicomputers," ACM (1989) (51056DOC020947-958).
Bell, "Ultracomputers: A Teraflop Before Its Time," Communications of the ACM, (Aug. 1992) pp. 27-47 (51056DOC020903-923).
Broomell et al., "Classification Categories and Historical Development of Circuit Switching Topologies," Computing Surveys, vol. 15, No. 2, pp. 95-133 (Jun. 1983) (51056DOC003002-040).
Culler et al., "Analysis of Multithreaded Microprocessors Under Multiprogramming," Report No. UCB/CSD 92/687 (May 1992) (51056DOC069283-300).
Donovan et al., "Pixel Processing in a Memory Controller," IEEE Computer Graphics and Applications, pp. 51-61 (Jan. 1995) (51056DOC059635-645).
Fields, "Hunting for Wasted Computing Power: New Software for Computing Networks Puts Idle PC's to Work," Univ. of Wisconsin-Madison, http://www.cs.wisc.edu/condor/doc/Wiscidea.html (1993) (51056DOC068704-711).
Geist, "Cluster Computing: The Wave of the Future?," Oak Ridge National Laboratory, 84OR21400 (May 30, 1994) (51056DOC020924-929).
Ghafoor, "Systolic Architecture for Finite Field Exponentiation," IEEE Proceedings, vol. 136 (Nov. 1989) (51056DOC071700-705).
Giloi, "Parallel Programming Models and their Interdependence with Parallel Architectures," IEEE Proceedings (Sep. 1993) (51056DOC071792-801).
Hwang et al., "Parallel Processing for Supercomputers and Artifical Intelligence," (1993) (51056DOC059663-673).
Hwang, "Advanced Computer Architecture: Parallelism, Scalability, Programmability," (1993) (51056DOC059656-662).
Iwaki, "Architecture of a High Speed Reed-Solomon Decoder," IEEE Consumer Electronics (Jan. 1994) (51056DOC071687-694).
Jain et al., "Square-Root, Reciprocal, SINE/COSINE, ARCTANGENT Cell for Signal and Image Processing," IEEE ICASSP '94, pp. II-521-II-524 (Apr. 1994) (51056DOC003070-073).
Laudon et al., "Architectural and Implementation Tradeoffs in the Design of Multiple-Context Processors," Technical Report: CSL-TR-92-523 (May 1992) (51056DOC069301-327).
Lawrie, "Access and Alignment of Data in an Array Processor," IEEE Transactions on Computers, vol. C-24, No. 12, pp. 99-109 (Dec. 1975) (51056DOC002932-942).
Le-Ngoc, "A Gate-Array-Based Programmable Reed-Solomon Codec: Structure-Implementation-Applications," IEEE Military Communications (1990) (51056DOC071695-699).
Litzkow et al., "Condor—A Hunter of Idle Workstations," IEEE (1988) (51056DOC068712-719).
Markstein, "Computation of Elementary Functions on the IBM RISC System/6000 Processor," IBM J. Res. Develop., vol. 34, No. 1, pp. 111-119 (Jan. 1990) (51056DOC059620-628).
Nienhaus, "A Fast Square Rooter Combining Algorithmic and Table Lookup Techniques," IEEE Proceedings Southeastcon, pp. 1103-1105 (1989) (51056DOC061469-471).
Renwick, "Building a Practical HIPPI LAN," IEEE, pp. 355-360 (1992) (51056DOC020937-942).
Rohrbacher et al., "Image Processing With the Staran Parallel Computer," IEEE Computer, vol. 10, No. 8, pp. 54-59 (Aug. 1977) (reprinted version pp. 119-124) (51056DOC002943-948).
Ryne, "Advanced Computers and Simulation," IEEE, pp. 3229-3233 (1993) (51056DOC020883-887).
Siegel, "Interconnection Networks for SIMD Machines," IEEE Computer, vol. 12, No. 6 (Jun. 1979) (reprinted version pp. 110 118) (51056DOC002949-957).
Singh et al., "A Programmable HIPPI Interface for a Graphics Supercomputer," ACM (1993) (51056DOC020888-896).
Smith, "Cache Memories," Computing Surveys, vol. 14, No. 3 (Sep. 1982) (51056DOC071586-643).
Tenbrink et al,, "HIPPI: The First Standard for High-Performance Networking," Los Alamos Science (1994) (51056DOC020943-946).
Tolmie, "Gigabit LAN Issues: HIPPI, Fibre Channel, or ATM," Los Alamos National Laboratory Report No. LA-UR 94-3994 (1994) (51056DOC046599-609).
Tolmie, "HIPPI: It's Not Just for Supercomputers Anymore," Data Communications (May 8, 1995) (51056DOC071802-809).
Toyokura et al., "A Video DSP with a Macroblock-Level-Pipeline and a SIMD Type Vector-Pipelined Architecture for MPEG2 CODEC," ISSCC94, Section 4, Video and Communications Signal Processors, Paper WP 4.5, pp. 74-75 (1994) (51056DOC003659-660).
Tullsen et al., "Simultaneous Multithreading: Maximizing On-Chip Parallelism," Proceedings of the 22nd Annual International Symposium on Computer Architecture (Jun. 1995) (51056DOC071434-443).
Turcotte, "A Survey of Software Environments for Exploiting Networked Computing Resources," Engineering Research Center for Computational Field Simulation (Jun. 11, 1993) (51056DOC069098-256).
Vetter et al., "Network Supercomputing: Connecting Cray Supercomputers with a HIPPI Network Provides Impressively High Execution Rates," IEEE Network (May 1992) (51056DOC020930-936).
Wang, "Bit-Level Systolic Array for Fast Exponentiation in GF(2m)," IEEE Transactions on Computers, vol. 43, No. 7, pp. 838-841 (Jul. 1994) (51056DOC059407-410).
Ware et al., "64 Bit Monolithic Floating Point Processors," IEEE Journal of Solid-State Circuits, vol. Sc-17, No. 5 (Oct. 1962) (51056DOC059646-655).
"Bit Manipulator," IBM Technical Disclosure Bulletin, pp. 1575-1576 (Nov. 1974) (51056DOC010205-206).
Finney et al., "Using a Common Barrel Shifter for Operand Normalization, Operand Alignment and Operand Unpack and Pack in Floating Point," IBM Technical Disclosure Bulletin, pp, 699-701 (Jul. 1986) (51056DOC010207-209).
Jovanovic et al., "Computational Science: Advances Through Collaboration," San Diego Supercomputer Center Science Report (1993) (51056DOC068769-779).
High Performance Computing and Communications: Toward a National Information Infrastructure, National Science Foundation (NSF) (1994) (51056DOC068791-801).
National Coordination Office for High Performance Computing and Communications, "High Performance Computing and Communications: Foundation for America's Information Future" (1996) (51056DOC072102-243).
Wilson, "The History of the Development of Parallel Computing," http://el.cs.vt.edu/~history/Parallel.html (51056DOC068720-757) Oct. 1994.
IEEE Standard 754 (ANSI/IEEE Std. 754-1985) (51056DOC019304-323).
Original Complaint for Patent Infringement, *MicroUnity Systems Engineering, Inc.* v. *Dell, Inc. f/k/a/ Dell Computer and Intel Corporation*; C.A. No. 2-04CV-120; In the United States District Court of the Eastern District of Texas, Marshall Division filed Mar. 26, 2004.
Amended Complaint for Patent Infringement, *MicroUnity Systems Engineering, Inc.* v. *Deli, Inc. f/k/a/ Dell Computer and Intel Corporation*; C.A. No. 2-04CV-120; In the United States District Court of the Eastern District of Texas, Marshall Division filed Apr. 20, 2004.
Expert Witness Report of Richard A. Killworth, Esq., *MicroUnify Systems Engineering, Inc.* v. *Dell, Inc. f/k/a/ Dell Computer and Intel Corporation*; C.A. No. 2-04CV-120; In the United States District Court of the Eastern District of Texas, Marshall Division filed Sep. 12, 2005.

Declaration and Expert Witness Report of Ray Mercer Regarding Written Description and Enablement Issues, *MicroUnity Systems Engineering, Inc.* v. *Dell, Inc. f/k/a/ Dell Computer and Intel Corporation*; C.A. No. 2-04CV-120; In the United States District Court of the Eastern District of Texas, Marshall Division filed Sep. 12, 2005.
Corrected Expert Report of Dr. Stephen B. Wicker Regarding Invalidity of U.S. Patent Nos. 5,742,840; 5,794,060; 5,764,061; 5,809,321; 6,584,482; 6,643,765; 6,725,356 and Exhibits A-I; *MicroUnity Systems Engineering, Inc.* v. *Dell, Inc. f/k/a/ Dell Computer and Intel Corporation*; C.A. No. 2-04CV-120; In the United States District Court of the Eastern District of Texas, Marshall Division filed Oct. 6, 2005.
Defendants Intel and Dell's Invalidity Contentions with Exhibits A-G; *MicroUnity Systems Engineering, Inc.* v. *Dell, Inc. f/k/a/ Dell Computer and Intel Corporation*; C.A. No. 2-04CV-120; In the United States District Court of the Eastern District of Texas, Marshall Division filed Sep. 19, 2005.
Defendants Dell Inc. and Intel Corporation's Identification of Prior Art Pursuant to 35 USC §282; *MicroUnity System Engineering, Inc.* v. *Dell, Inc. f/k/a/ Dell Computer and Intel Corporation*; C.A. No. 2-04CV-120; In the United States District Court of the Eastern District of Texas, Marshall Division filed Oct. 7, 2005.
Request for *Inter Partes* Reexamination Under 35 USC §§ 311-318 of U.S. Patent No. 6,725,356 flied on Jun. 28. 2005.
Deposition of Larry Mennemeler on Sep. 22, 2005 and Exhibit 501; *MicroUnity Systems Engineering, Inc.* v. *Dell, Inc. f/k/a/ Dell Computer and Intel Corporation*; C.A. No. 2-04CV-120; In the United States District Court of the Eastern District of Texas Marshall Division.
Intel Article, "*Intel Announces Record Revenue of 9.96 Billion*", Oct. 18, 2005.
The New York Times Article, "*Intel Posts 5% Profit Increase on Demand for Notebook Chips*", Oct. 19, 2005.
USA Today Article, "*Intel's Revenue Grew 18% In Robust Quarter for Tech*", Oct. 19, 2005.
The Wall Street Journal Article, "*Intel Says Chip Demand May Slow*", Oct. 19, 2005.
The New York Times Article, "*Intel Settlement Revives A Fading Chip Designer*", Oct. 20, 2005.
U.S. Appl. No. 10/757,939 Non-final Rejection mailed Dec. 7, 2009.
U.S. Appl. No. 90/007,583 Amendment filed Sep. 24, 2009.
U.S. Appl. No. 90/007,593 (Reexam of US 5,794,060) Non-final office action mailed Oct. 14, 2009.
U.S. Appl. No. 90/007,593 (Reexam of US 5,794,060) Response after Non-final office action filed Dec. 14, 2009.
U.S. Appl. No. 95/000,100 (Reexam of US 6,725,356) Notice of Intent to Issue Reexamination Certificate mailed Oct. 26, 2009.
U.S. Appl. No. 95/000,089 (Reexam of US 6,643,765) Examiner's Answer to Appeal Brief Sep. 29, 2009.
U.S. Appl. No. 95/000,089 (Reexam of US 6,843,765) Rebuttal Brief -owner Oct. 29, 2009.
U.S. Appl. No. 10/757,925 Response after Non-final action Sep. 21, 2009.
U.S. Appl. No. 10/757,925 Final Rejection mailed Dec. 23, 2009.
U.S. Appl. No. 11/878,804 Final Rejection mailed Oct. 15, 2009.
U.S. Appl. No. 11/878,804 Request for Continued Examination and Response a to Oct. 15, 2009 Office Action Nov. 13, 2009.
U.S. Appl. No. 11/878,805 Notice of Allowance mailed Nov. 17, 2009.
U.S. Appl. No. 10/757,851 Notice of Allowance mailed Dec. 4, 2009.

* cited by examiner

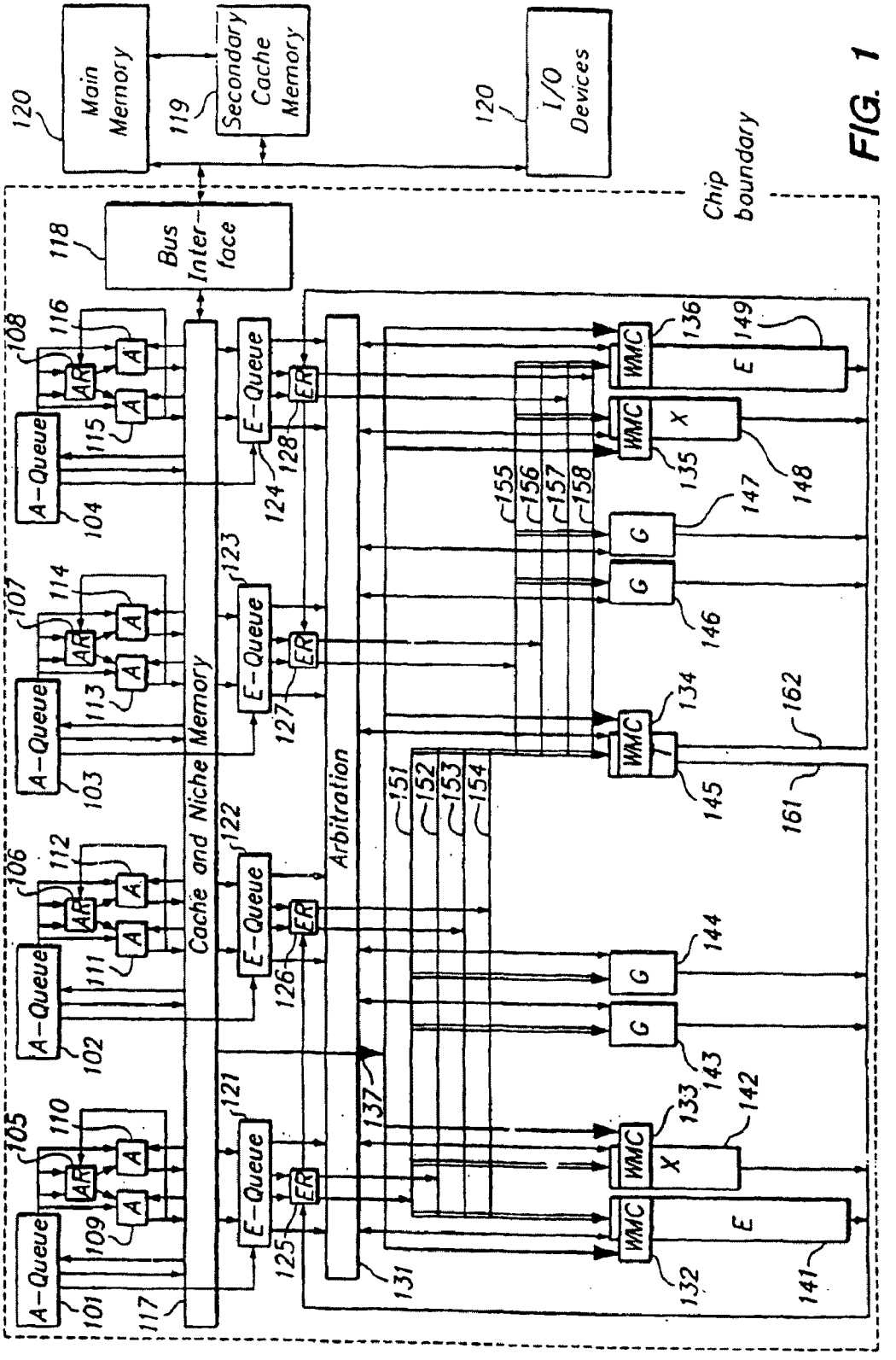

Main Memory
120
Secondary Cache Memory
119
I/O Devices
120
Chip boundary
Bus Interface
118
Cache and Niche Memory
117
Arbitration
131
A-Queue
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
AR
A
E-Queue
121
122
123
124
125
126
127
128
ER
WMC
132
133
134
135
136
137
141
142
143
144
145
146
147
148
149
151
152
153
154
155
156
157
158
161
162
E
X
G
I

FIG. 1

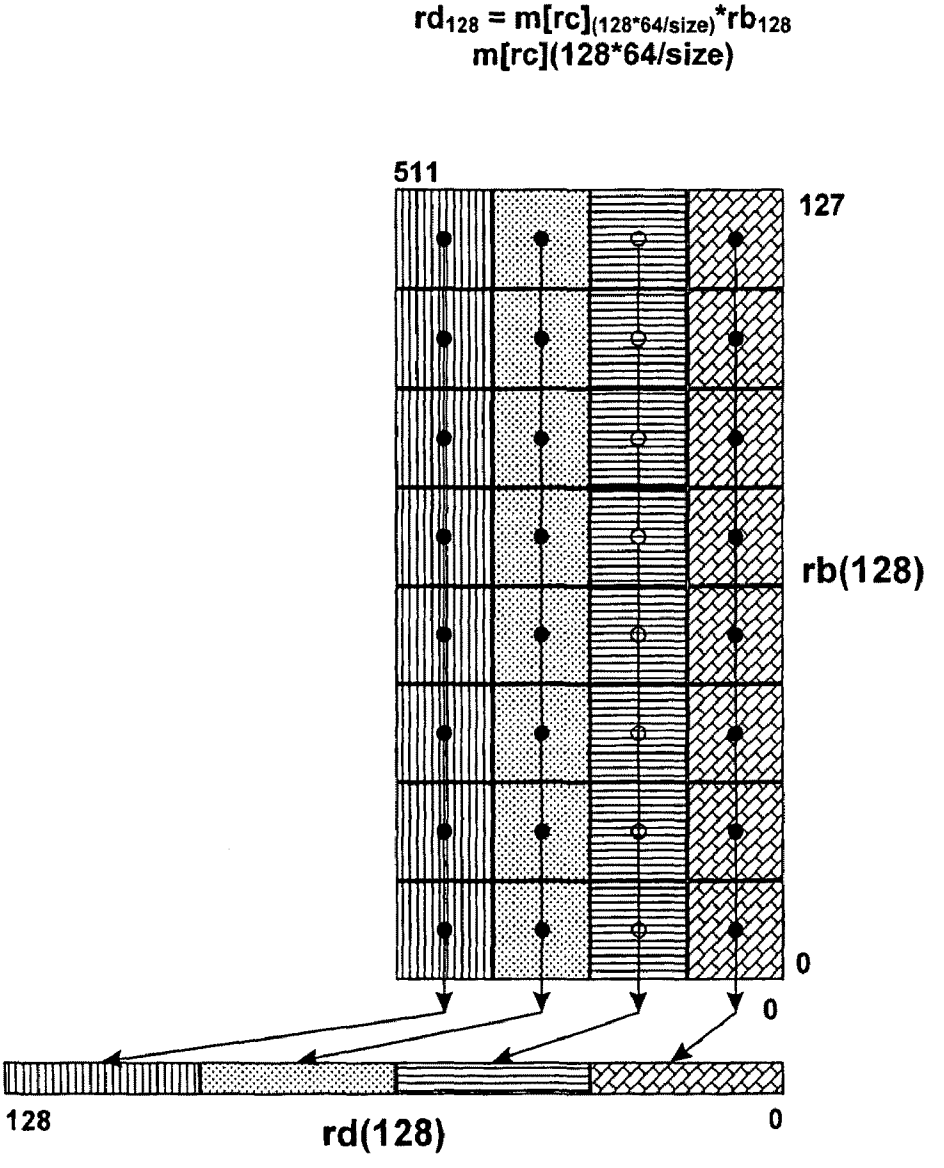

rd128 = m[rc](128*64/size)*rb128
m[rc](128*64/size)
511
127
rb(128)
0
0
128
rd(128)
0

FIG. 2

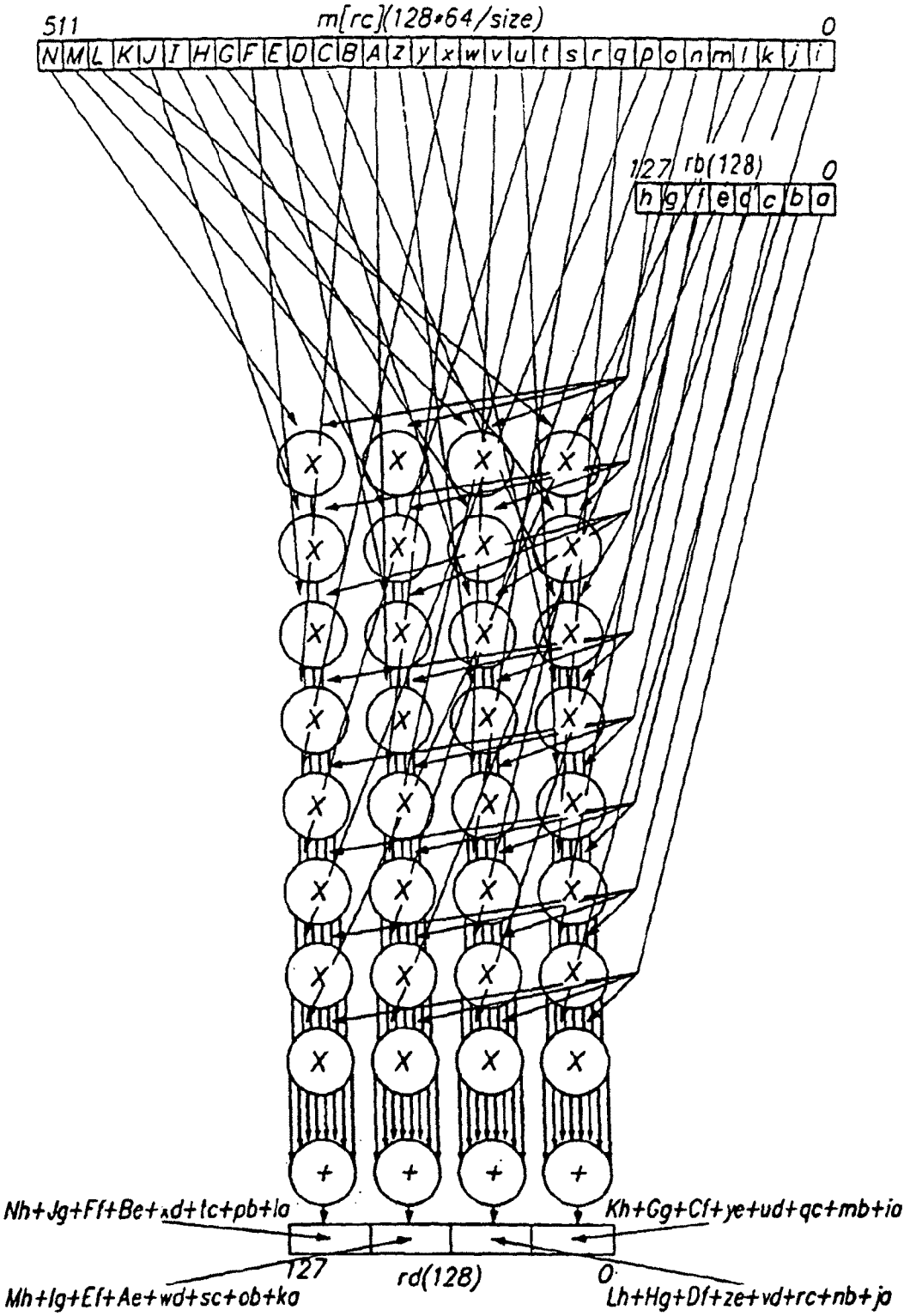

511
m[rc](128*64/size)
0
N M L K J I H G F E D C B A z y x w v u t s r q p o n m l k j i
127
rb(128)
0
h g f e d c b a
Nh+Jg+Ff+Be+xd+tc+pb+la
Kh+Gg+Cf+ye+ud+qc+mb+ia
127
rd(128)
0
Mh+Ig+Ef+Ae+wd+sc+ob+ka
Lh+Hg+Df+ze+vd+rc+nb+ja

FIG. 3

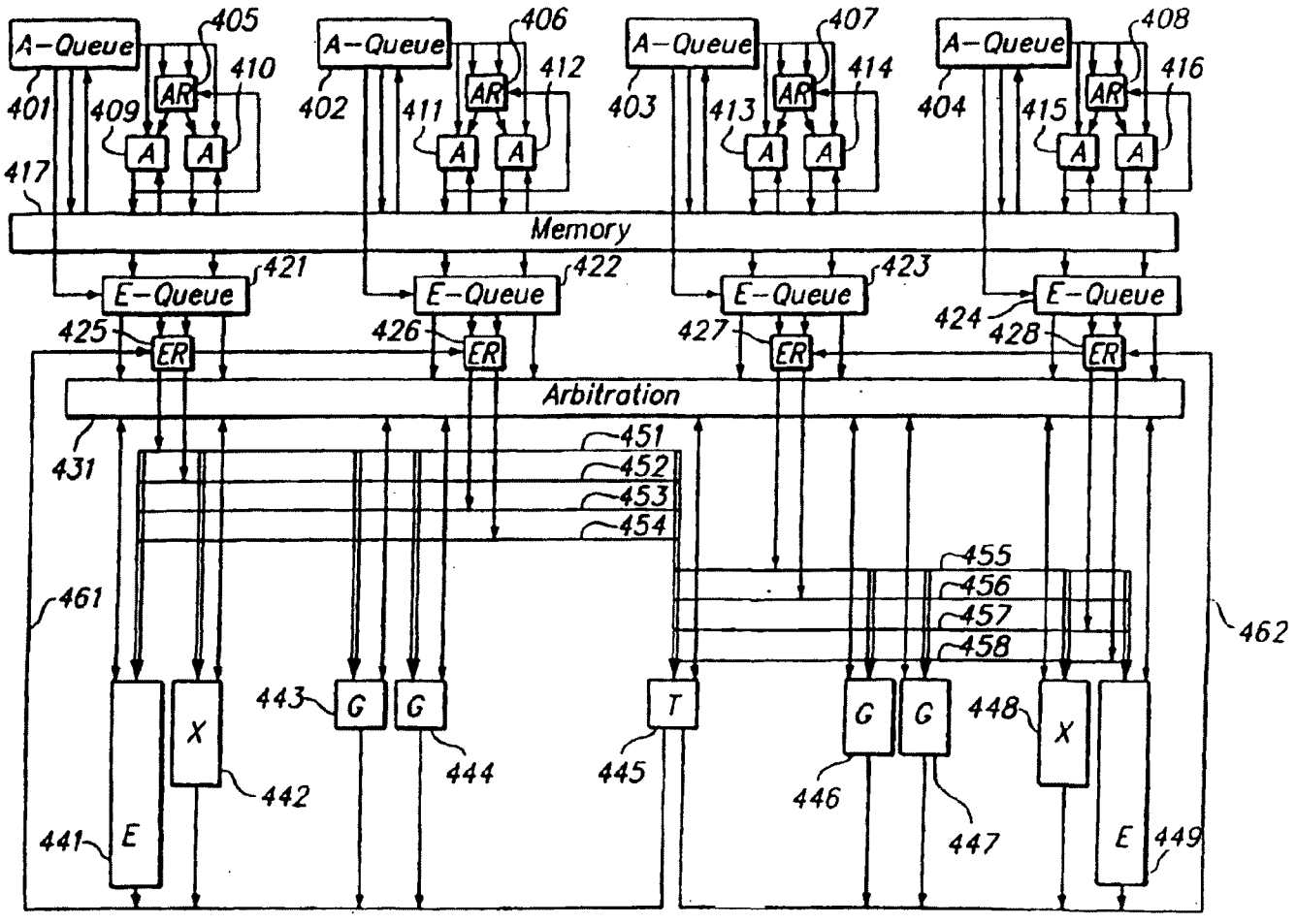

A-Queue
405
406
407
408
401
402
403
404
AR
410
412
414
416
409
411
413
415
A
417
Memory
E-Queue
421
422
423
425
426
427
424
428
ER
Arbitration
431
451
452
453
454
455
456
457
458
461
462
443
G
444
T
445
446
447
448
X
442
441
E
449

FIG. 4

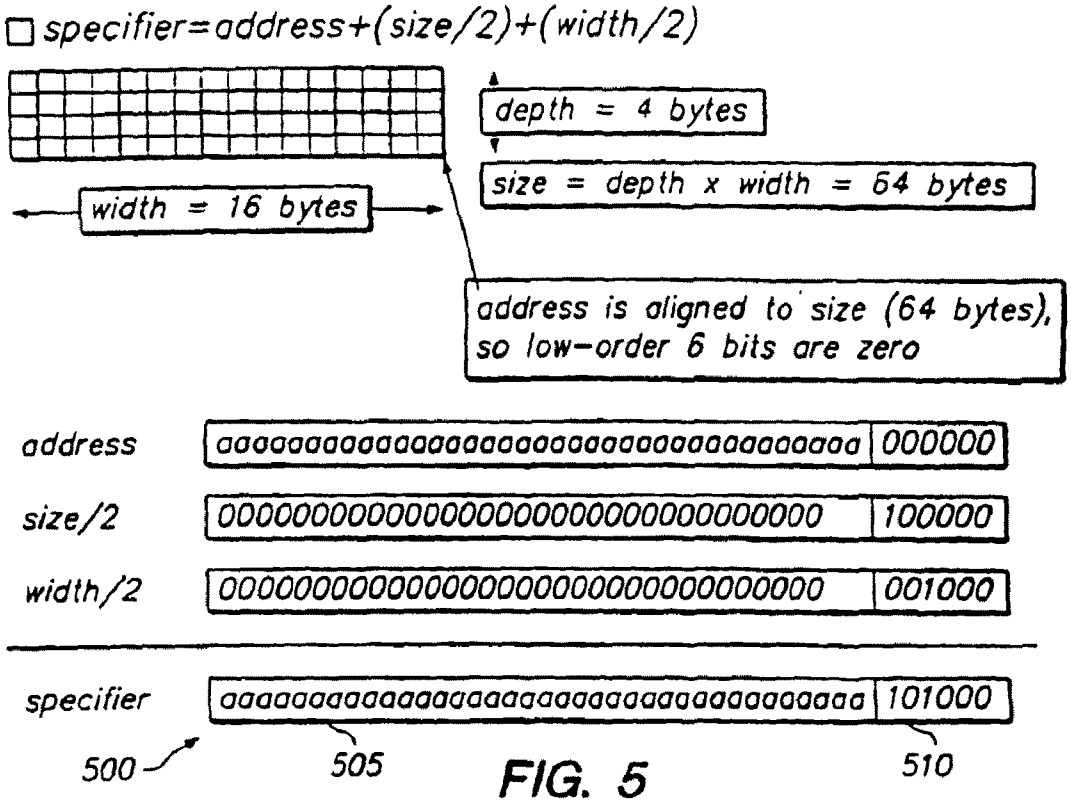

specifier=address+(size/2)+(width/2)
depth = 4 bytes
size = depth x width = 64 bytes
width = 16 bytes
address is aligned to size (64 bytes), so low-order 6 bits are zero
address
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa 000000
size/2
00000000000000000000000000000000 100000
width/2
00000000000000000000000000000000 001000
specifier
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa 101000
500
505
510

FIG. 5

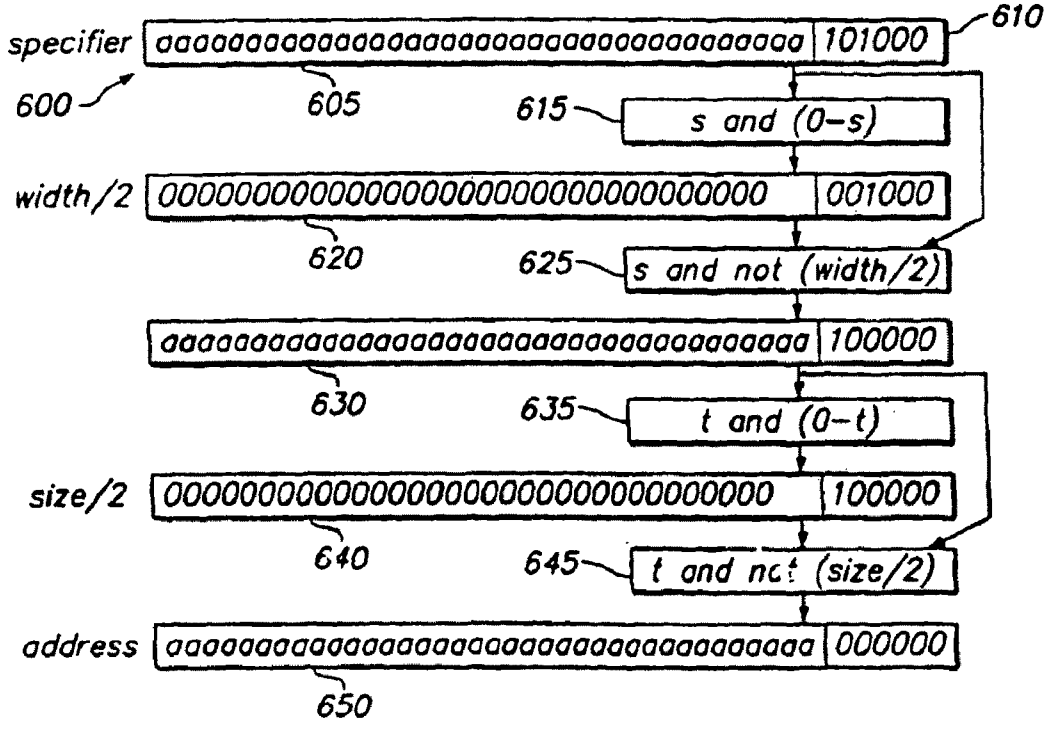

specifier
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa 101000
610
600
605
615
s and (0-s)
width/2
00000000000000000000000000000000 001000
620
625
s and not (width/2)
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa 100000
630
635
t and (0-t)
size/2
00000000000000000000000000000000 100000
640
645
t and not (size/2)
address
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa 000000
650

FIG. 6

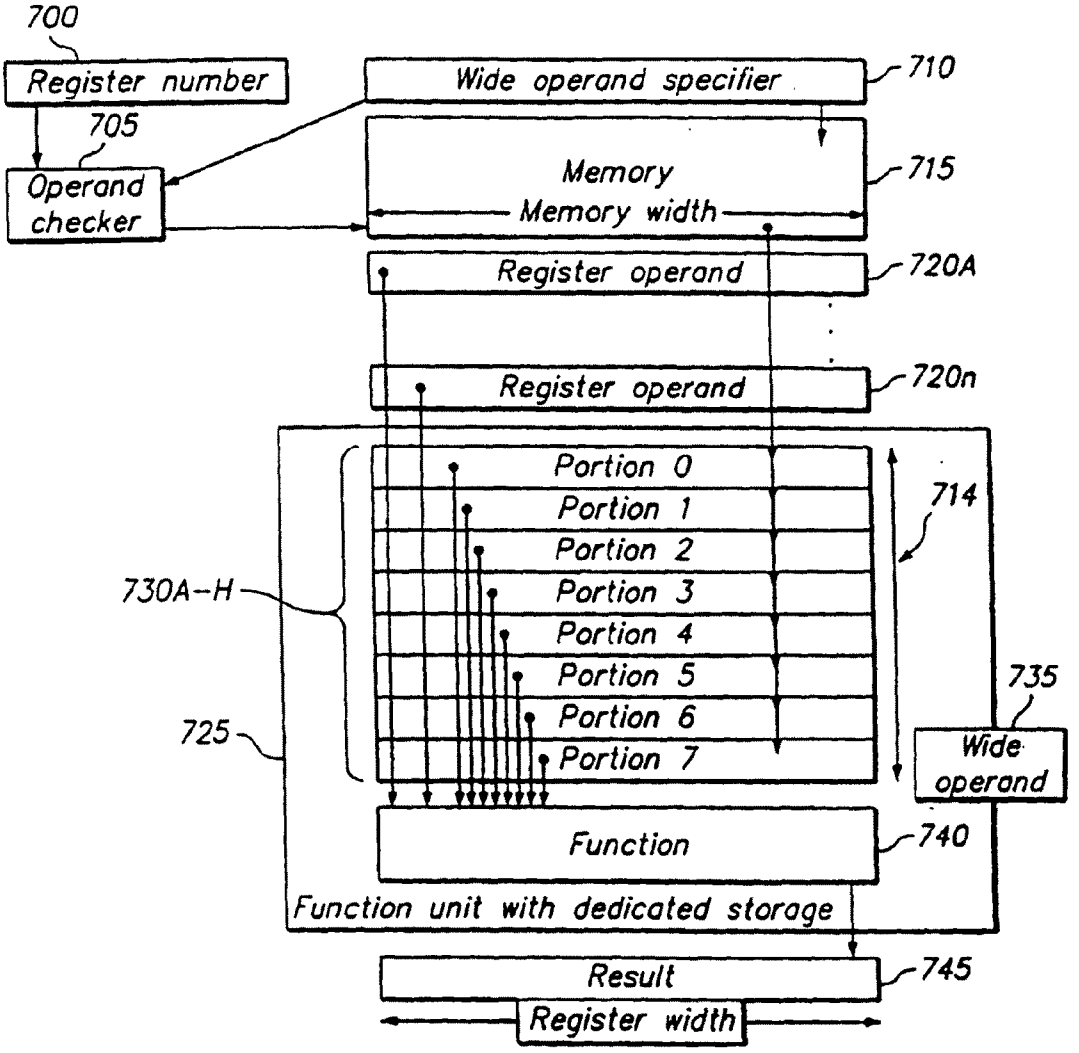

700
Register number
705
Operand checker
Wide operand specifier
710
Memory
Memory width
715
Register operand
720A
Register operand
720n
Portion 0
Portion 1
Portion 2
Portion 3
Portion 4
Portion 5
Portion 6
Portion 7
714
730A-H
725
735
Wide operand
Function
740
Function unit with dedicated storage
Result
745
Register width

FIG. 7

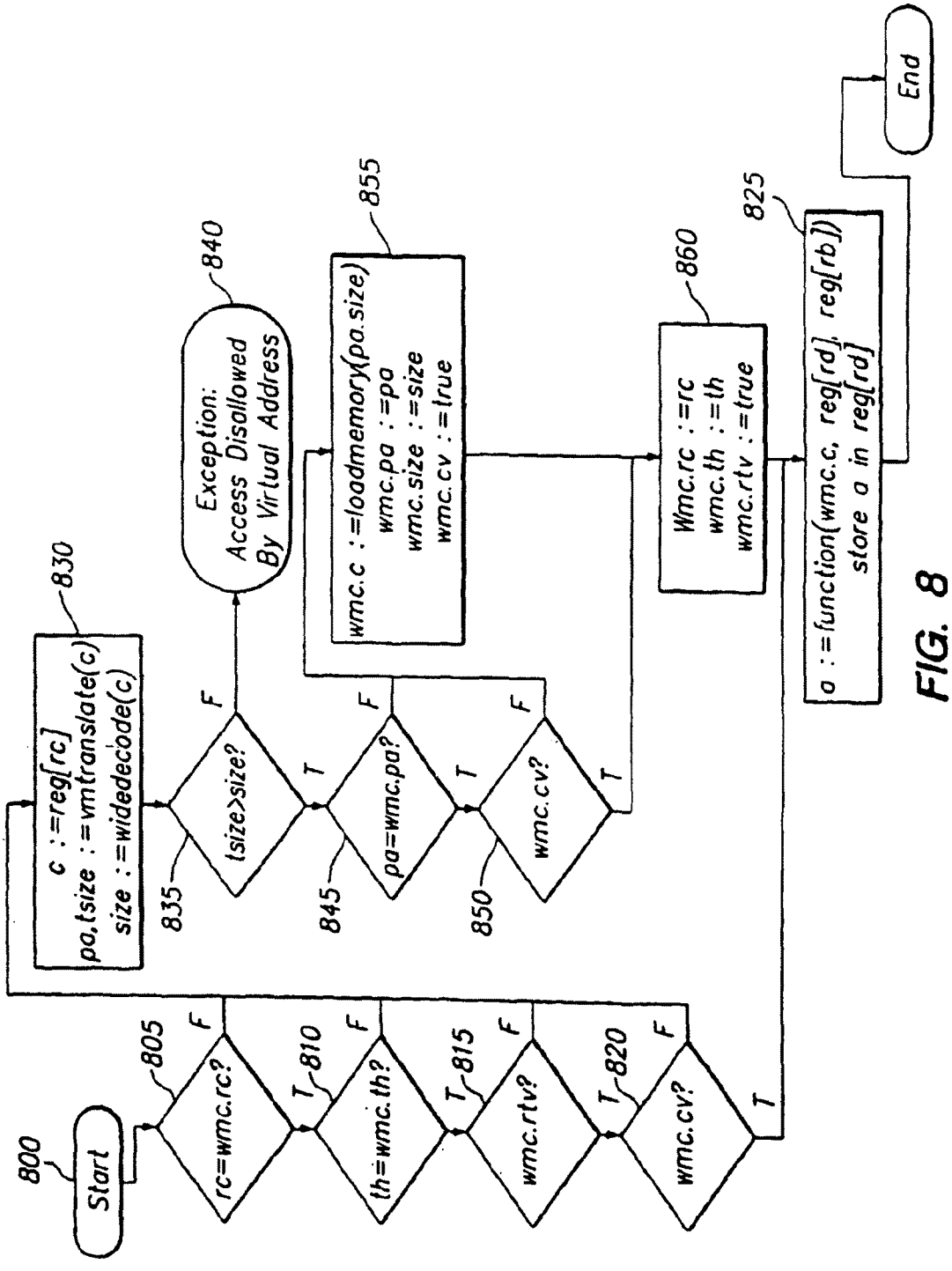

800
Start
805
rc=wmc.rc?
F
T
810
th=wmc.th?
F
T
815
wmc.rtv?
F
T
820
wmc.cv?
F
T
825
o := function(wmc.c, reg[rd], reg[rb])
store o in reg[rd]
End
830
c := reg[rc]
pa,tsize := vmtranslate(c)
size := widedecode(c)
835
tsize>size?
F
T
840
Exception:
Access Disallowed
By Virtual Address
845
pa=wmc.pa?
F
T
850
wmc.cv?
F
T
855
wmc.c := loadmemory(pa.size)
wmc.pa := pa
wmc.size := size
wmc.cv := true
860
Wmc.rc := rc
wmc.th := th
wmc.rtv := true

FIG. 8

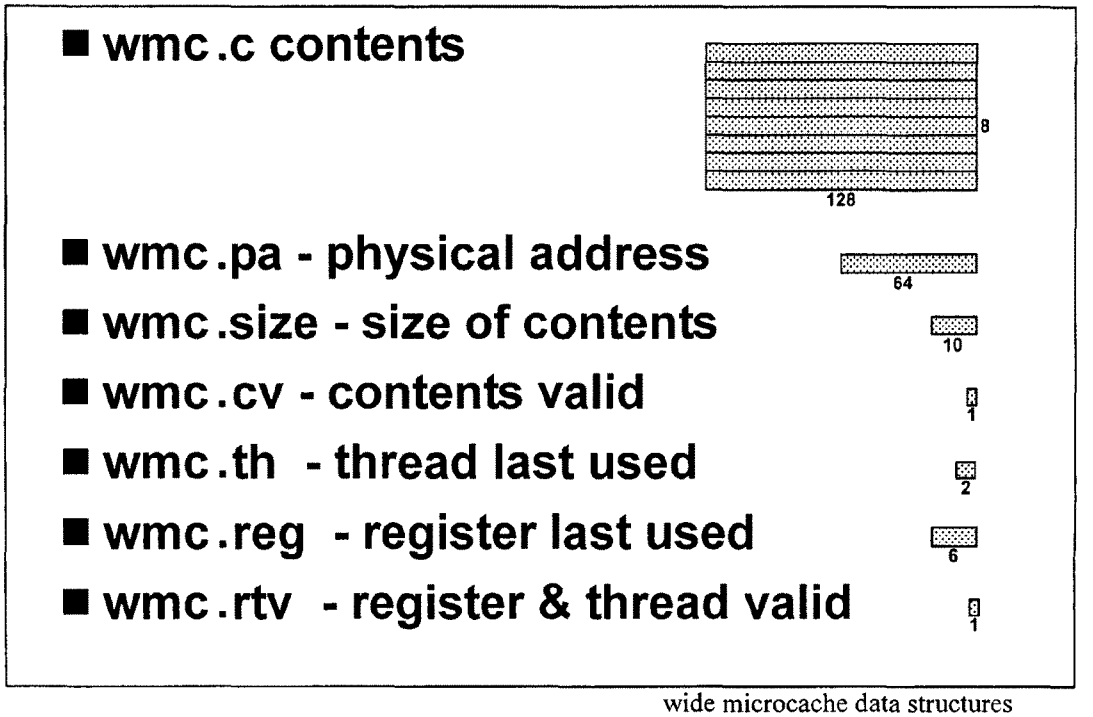

wmc.c contents
8
128
wmc.pa - physical address
64
wmc.size - size of contents
10
wmc.cv - contents valid
1
wmc.th - thread last used
2
wmc.reg - register last used
6
wmc.rtv - register & thread valid
1
wide microcache data structures

FIG. 9

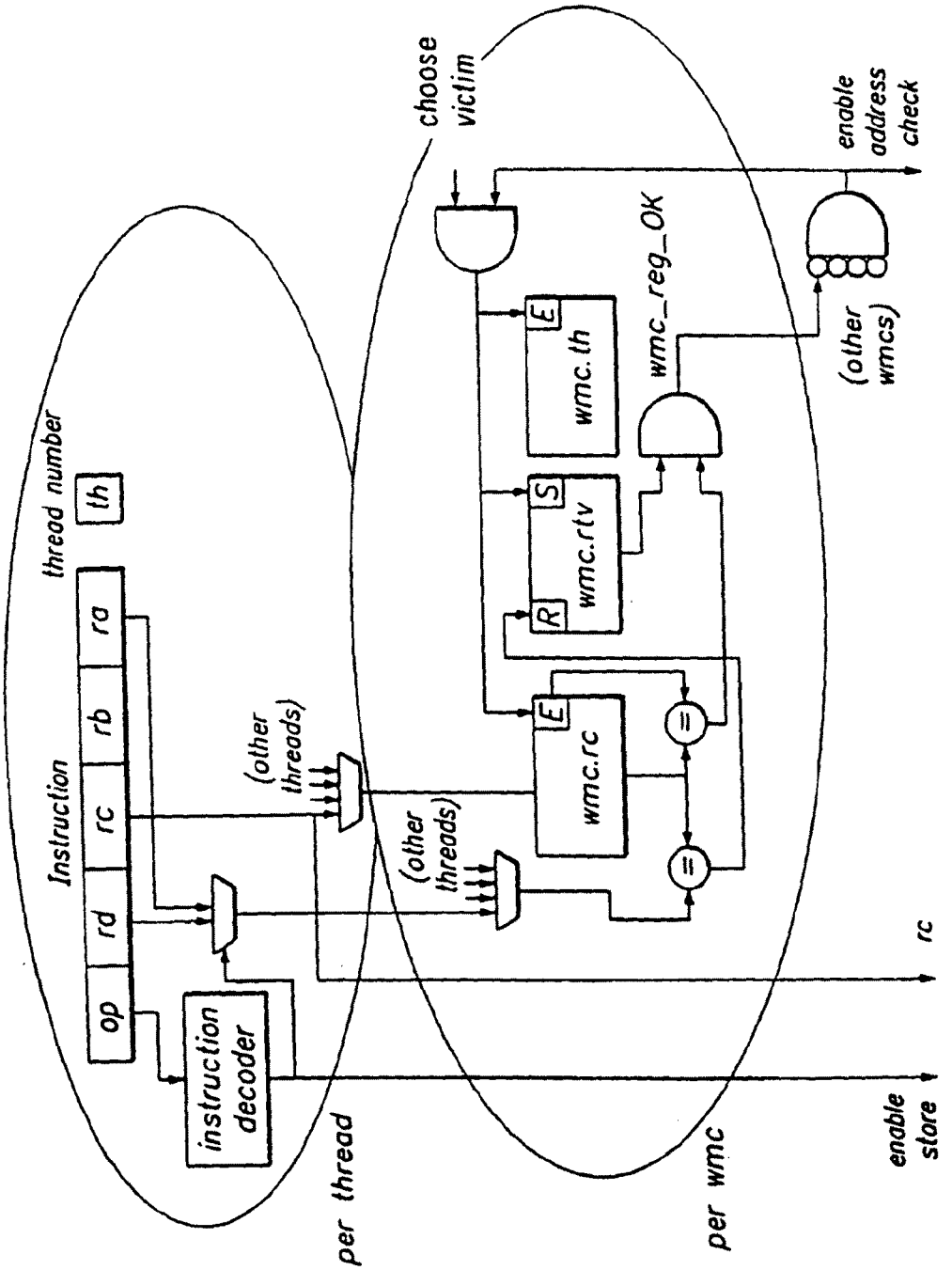

thread number
th
Instruction
op
rd
rc
rb
ra
instruction decoder
(other threads)
(other threads)
choose victim
E
wmc.th
S
R
wmc.rtv
E
wmc.rc
=
=
wmc_reg_OK
enable address check
(other wmcs)
rc
enable store
per thread
per wmc

FIG. 10

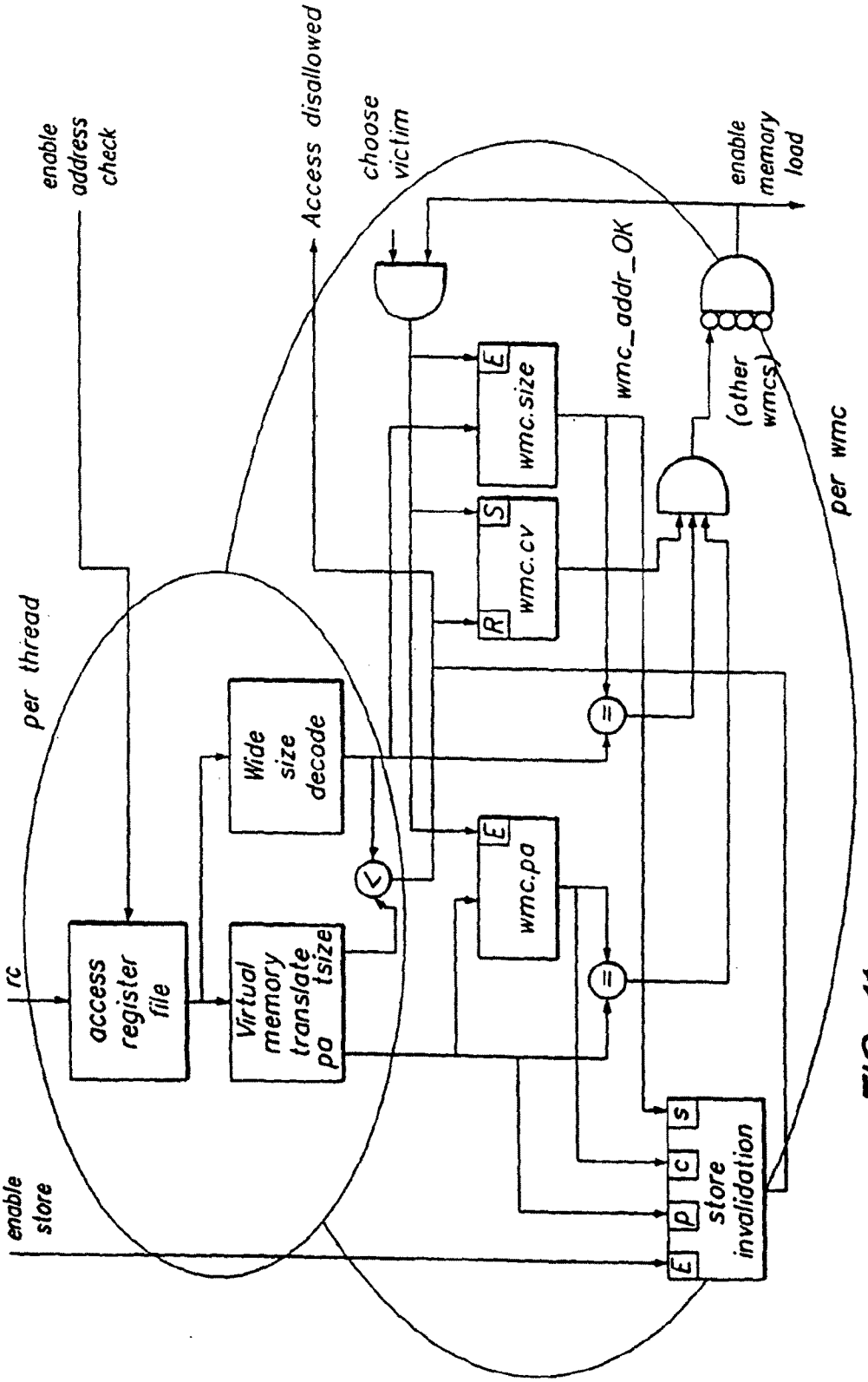

enable address check
Access disallowed
choose victim
enable memory load
wmc_addr_OK
per thread
per wmc
(other wmcs)
wmc.size
wmc.cv
wmc.pa
Wide size decode
access register file
Virtual memory translate pa tsize
store invalidation
rc
enable store

FIG. 11

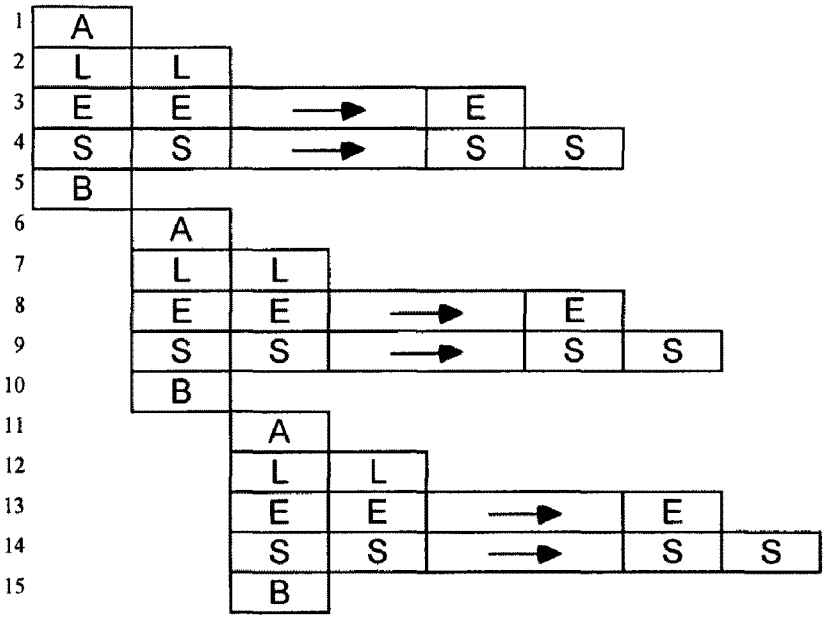

1 A
2 L L
3 E E E
4 S S S S
5 B
6 A
7 L L
8 E E E
9 S S S S
10 B
11 A
12 L L
13 E E E
14 S S S S
15 B

FIG. 12

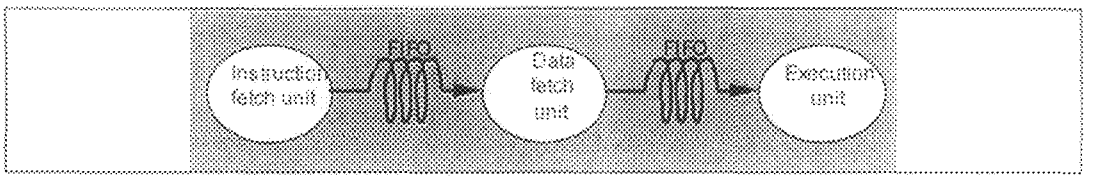
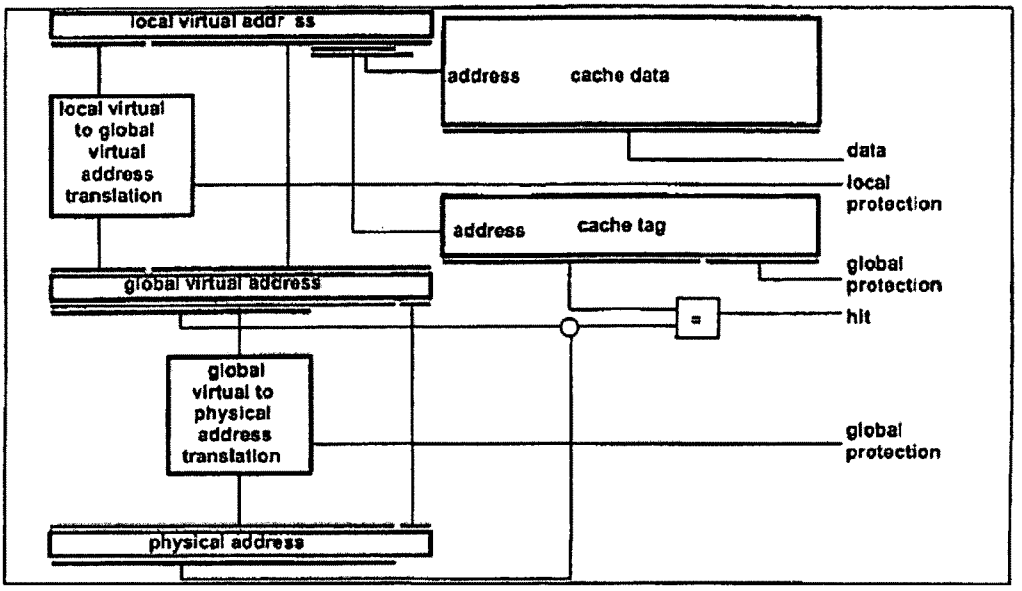

Instruction fetch unit
FIFO
Data fetch unit
FIFO
Execution unit

FIG. 13



memory management organization

FIG. 14

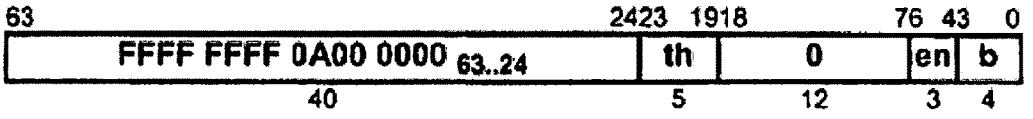


FIG. 15

```
def data,flags ← AccessPhysicalLTB(pa,op,wdata) as
    th ← pa23..19
    en ← pa6..4
    if (en < (1 || 0LE)) and (th < T) and (pa18..6=0) then
        case op of
            R:
                data ← 064 || LTBArray[th][en]
            W:
                LocalTB[th][en] ← wdata63..0
        endcase
    else
        data ← 0
    endif
enddef
```

FIG. 16

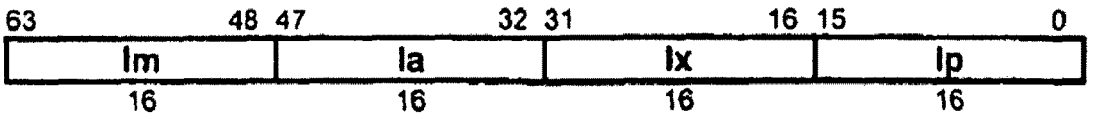


FIG. 17

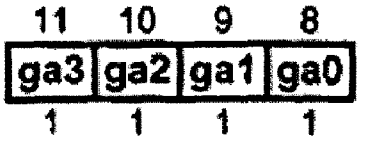


FIG. 18

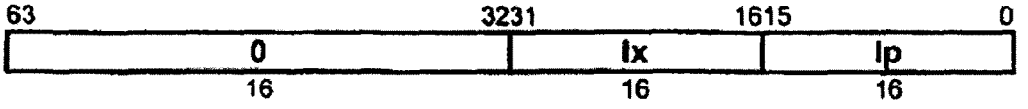


FIG. 19

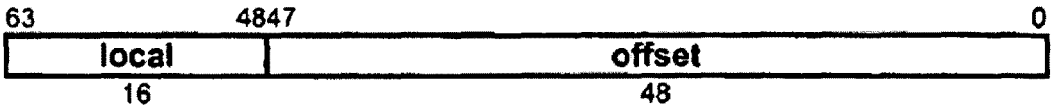


FIG. 20

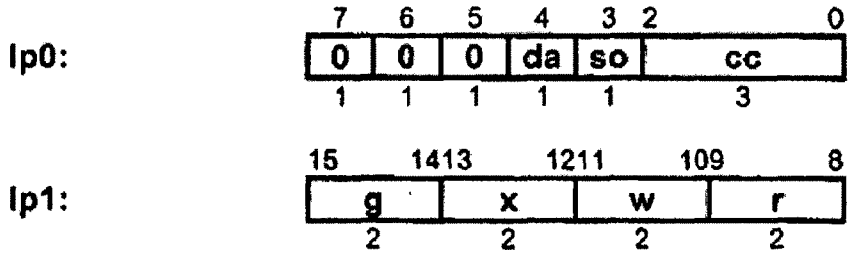

lp0:
7 6 5 4 3 2 0
0 0 0 da so cc
1 1 1 1 1 3
lp1:
15 1413 1211 109 8
g x w r
2 2 2 2

FIG. 21

```
def ga,LocalProtect ← LocalTranslation(th,ba,la,pl) as
    if LB & (ba63..48 ⏲ la63..48) then
        raise AccessDisallowedByVirtualAddress
    endif
    me ← NONE
    for i ← 0 to (1 || 0^LE)-1
        if (la63..48 & ~LocalTB[th][i]63..48) = LocalTB[th][i]47..32 then
            me ← i
        endif
    endfor
    if me = NONE then
        if ~ControlRegisterpl+8 then
            raise LocalTBMiss
        endif
        ga ← la
        LocalProtect ← 0
    else
        ga ← (va63..48 ^ LocalTB[th][me]31..16) || va47..0
        LocalProtect ← LocalTB[th][me]15..0
    endif
enddef
```

FIG. 22

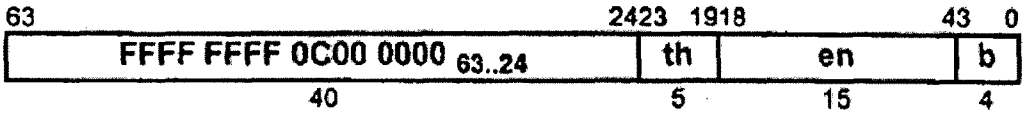


FIG. 23

```
def data,flags ← AccessPhysicalGTB(pa,op,wdata) as
        th ← pa23..19+GT || 0GT
        en ← pa18..4
        if (en < (1 || 0G)) and (th < T) and (pa18+GT..19 = 0) then
                case op of
                        R:
                                data ← GTBArray[th5..GT][en]
                        W:
                                GTBArray[th5..GT][en] ← wdata
                endcase
        else
                data ← 0
        endif
enddef
```

FIG. 24

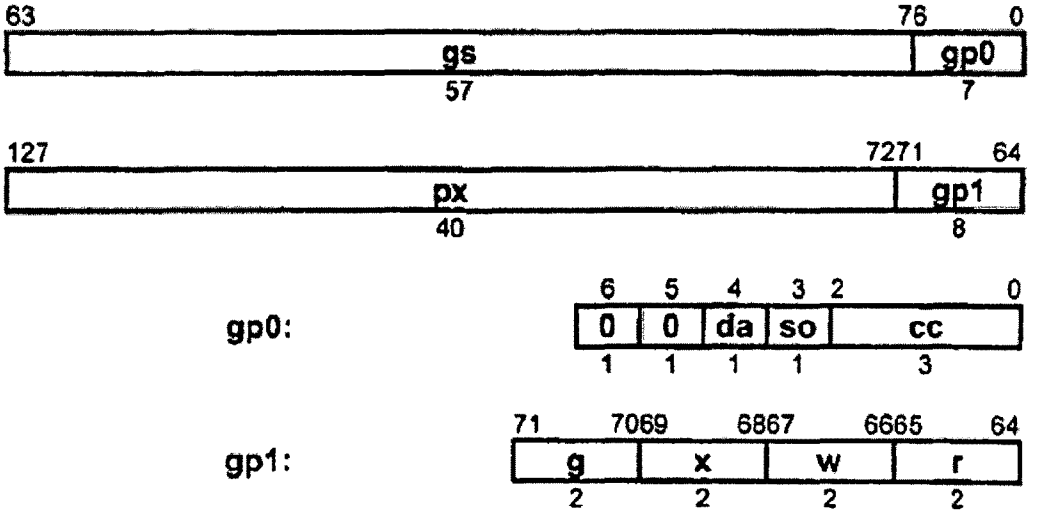

63
76
0
gs
gp0
57
7
127
7271
64
px
gp1
40
8
gp0:
6
5
4
3
2
0
0
0
da
so
cc
1
1
1
1
3
gp1:
71
7069
6867
6665
64
g
x
w
r
2
2
2
2

FIG. 25

```
def pa,GlobalProtect ← GlobalAddressTranslation(th,ga,pl,lda) as
    me ← NONE
    for i ← 0 to (1 || 0^G) −1
        if GlobalTB[th5..GT][i] ≠ 0 then
            size ← (GlobalTB[th5..GT][i]63..7 and (0^64-GlobalTB(th5..GT][i]63..7)) || 0^8
            if ((ga63..8||0^8) ^ (GlobalTB[th5..GT][i]63..8||0^8)) and (0^64-size)) = 0 then
                me ← GlobalTB[th5..GT][i]
            endif
        endif
    endfor
    if me = NONE then
        if lda then
            PerformAccessDetail(AccessDetailRequiredByLocalTB)
        endif
        raise GlobalTBMiss
    else
        pa ← (ga63..8 ^ GlobalTB[th5..GT][me]127..72) || ga7..0
        GlobalProtect ← GlobalTB[th5..GT][me]71..64 || 0^1 || GlobalTB[th5..GT][me]6..0
    endif
enddef
```

FIG. 26

```
def GTBUpdateWrite(th,fill,data) as
    me ← NONE
    for i ← 0 to (1 || 0^G) -1
        size ← (GlobalTB[th5..GT][i]63..7 and (0^64-GlobalTB(th5..GT][i]63..7)) || 0^8
        if ((data63..8||0^8) ^ (GlobalTB[th5..GT][i]63..8||0^8)) and (0^64-size) = 0 then
            me ← i
        endif
    endfor
    if me = NONE then
        if fill then
            GlobalTB[th5..GT][GTBLast[th5..GT]] ← data
            GTBLast[th5..GT] ← (GTBLast[th5..GT] + 1)G-1..0
            if GTBLast[th5..GT] = 0 then
                GTBLast[th5..GT] ← GTBFirst[th5..GT]
                GTBBump[th5..GT] ← 1
            endif
        endif
    else
        GlobalTB[th5..GT][me] ← data
    endif
enddef
```

FIG. 27

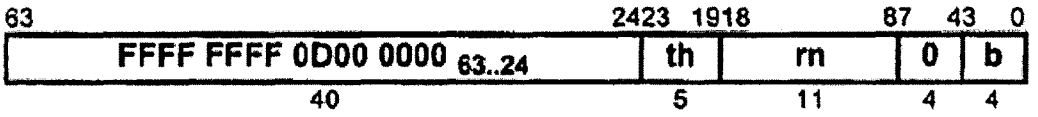


FIG. 28

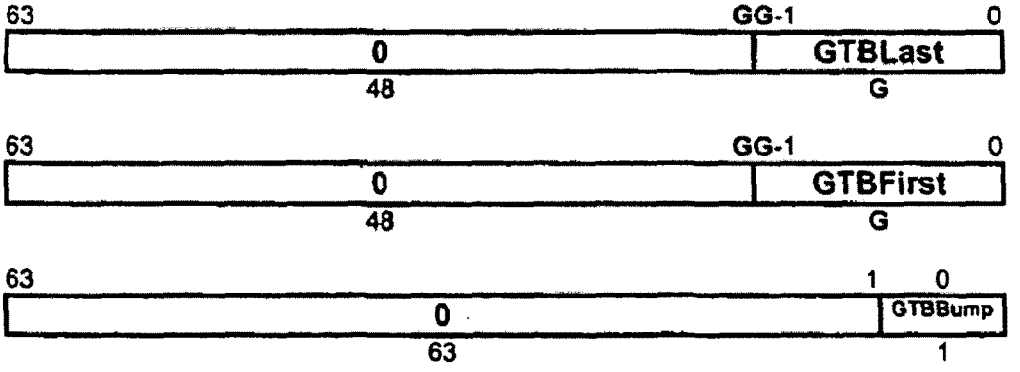

63
GG-1
0
0
GTBLast
48
G
63
GG-1
0
0
GTBFirst
48
G
63
1
0
0
GTBBump
63
1

FIG. 29

```
def data,flags ← AccessPhysicalGTBRegisters(pa,op,wdata) as
    th ← pa23..19+GT || 0GT
    rn ← pa18..8
    if (rn < 5) and (th < T) and (pa18+GT..19 = 0)  and (pa7..4 =
        case rn || op of
            0 || R, 1 || R:
                data ← 0
            0 || W, 1 || W:
                GTBUpdateWrite(th,rn0,wdata)
            2 || R:
                data ← 064-G || GTBLast[th5..GT]
            2 || W:
                GTBLast[th5..GT] ← wdataG-1..0
            3 || R:
                data ← 064-G || GTBFirst[th5..GT]
            3 || W:
                GTBFirst[th5..GT] ← wdataG-1..0
            3 || R:
                data ← 063 || GTBBump[th5..GT]
            3 || W:
                GTBBump[th5..GT] ← wdata0
        endcase
    else
        data ← 0
    endif
enddef
```

FIG. 30

G.BOOLEAN	Group Boolean

Equivalencies

G.AAA	Group three-way and
G.AAA.1	Group add add add bits
G.AAS.1	Group add add subtract bits
G.ADD.1	Group add bits
G.AND	Group and
G.ANDN	Group and not
G.COPY	Group copy
G.NAAA	Group three-way nand
G.NAND	Group nand
G.NOOO	Group three-way nor
G.NOR	Group nor
G.NOT	Group not
G.NXXX	Group three-way exclusive-nor
G.OOO	Group three-way or
G.OR	Group or
G.ORN	Group or not
G.SAA.1	Group subtract add add bits
G.SAS.1	Group subtract add subtract bits
G.SET	Group set
G.SET.AND.E.1	Group set and equal zero bits
G.SET.AND.NE.1	Group set and not equal zero bits
G.SET.E.1	Group set equal bits
G.SET.G.1	Group set greater signed bits
G.SET.G.U.1	Group set greater unsigned bits
G.SET.G.Z.1	Group set greater zero signed bits
G.SET.GE.1	Group set greater equal signed bits
G.SET.GE.Z.1	Group set greater equal zero signed bits
G.SET.L.1	Group set less signed bits
G.SET.L.Z.1	Group set less zero signed bits
G.SET.LE.1	Group set less equal signed bits
G.SET.LE.U.1	Group set less equal unsigned bits
G.SET.LE.Z.1	Group set less equal zero signed bits
G.SET.NE.1	Group set not equal bits
G.SET.GE.U.1	Group set greater equal unsigned bits
G.SET.L.U.1	Group set less unsigned bits

FIG. 31A

G.SSA.1	Group subtract subtract add bits
G.SSS.1	Group subtract subtract subtract bits
G.SUB.1	Group subtract bits
G.XNOR	Group exclusive-nor
G.XOR	Group exclusive-or
G.XXX	Group three-way exclusive-or
G.ZERO	Group zero

G.AAA rd@rc,rb	←	G.BOOLEAN rd@rc,rb,0b10000000
G.AAA.1 rd@rc,rb	→	G.XXX rd@rc,rb
G.AAS.1 rd@rc,rb	→	G.XXX rd@rc,rb
G.ADD.1 rd=rc,rb	→	G.XOR rd=rc,rb
G.AND rd=rc,rb	←	G.BOOLEAN rd@rc,rb,0b10001000
G.ANDN rd=rc,rb	←	G.BOOLEAN rd@rc,rb,0b01000100
G.BOOLEAN rd@rb,rc,i	→	G.BOOLEAN rd@rc,rb,$i_7 i_5 i_6 i_4 i_3 i_1 i_2 i_0$
G.COPY rd=rc	←	G.BOOLEAN rd@rc,rc,0b10001000
G.NAAA. rd@rc,rb	←	G.BOOLEAN rd@rc,rb,0b01111111
G.NAND rd=rc,rb	←	G.BOOLEAN rd@rc,rb,0b01110111
G.NOOO rd@rc,rb	←	G.BOOLEAN rd@rc,rb,0b00000001
G.NOR rd=rc,rb	←	G.BOOLEAN rd@rc,rb,0b00010001
G.NOT rd=rc	←	G.BOOLEAN rd@rc,rc,0b00010001
G.NXXX rd@rc,rb	←	G.BOOLEAN rd@rc,rb,0b01101001
G.OOO rd@rc,rb	←	G.BOOLEAN rd@rc,rb,0b11111110
G.OR rd=rc,rb	←	G.BOOLEAN rd@rc,rb,0b11101110
G.ORN rd=rc,rb	←	G.BOOLEAN rd@rc,rb,0b11011101
G.SAA.1 rd@rc,rb	→	G.XXX rd@rc,rb
G.SAS.1 rd@rc,rb	→	G.XXX rd@rc,rb
G.SET rd	←	G.BOOLEAN rd@rd,rd,0b10000001
G.SET.AND.E.1 rd=rb,rc	→	G.NAND rd=rc,rb
G.SET.AND.NE.1 rd=rb,rc	→	G.AND rd=rc,rb
G.SET.E.1 rd=rb,rc	→	G.XNOR rd=rc,rb
G.SET.G.1 rd=rb,rc	→	G.ANDN rd=rc,rb
G.SET.G.U.1 rd=rb,rc	→	G.ANDN rd=rb,rc
G.SET.G.Z.1 rd=rc	→	G.ZERO rd
G.SET.GE.1 rd=rb,rc	→	G.ORN rd=rc,rb
G.SET.GE.Z.1 rd=rc	→	G.NOT rd=rc

FIG. 31A *continued*

G.SET.L.1 rd=rb,rc	→	G.ANDN rd=rb,rc
G.SET.L.Z.1 rd=rc	→	G.COPY rd=rc
G.SET.LE.1 rd=rb,rc	→	G.ORN rd=rb,rc
G.SET.LE.U.1 rd=rb,rc	→	G.ORN rd=rc,rb
G.SET.LE.Z.1 rd=rc	→	G.SET rd
G.SET.NE.1 rd=rb,rc	→	G.XOR rd=rc,rb
G.SET.GE.U.1 rd=rb,rc	→	G.ORN rd=rb,rc
G.SET.L.U.1 rd=rb,rc	→	G.ANDN rd=rc,rb
G.SSA.1 rd@rc,rb	→	G.XXX rd@rc,rb
G.SSS.1 rd@rc,rb	→	G.XXX rd@rc,rb
G.SUB.1 rd=rc,rb	→	G.XOR rd=rc,rb
G.XNOR rd=rc,rb	←	G.BOOLEAN rd@rc,rb,0b10011001
G.XOR rd=rc,rb	←	G.BOOLEAN rd@rc,rb,0b01100110
G.XXX rd@rc,rb	←	G.BOOLEAN rd@rc,rb,0b10010110
G.ZERO rd	←	G.BOOLEAN rd@rd,rd,0b00000000

Selection

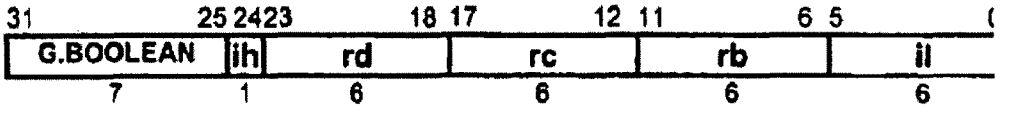
operation	function (binary)	function (decimal)
d	11110000	240
c	11001100	204
b	10101010	176
d&c&b	10000000	128
(d&c)|b	11101010	234
d|c|b	11111110	254
d?c:b	11001010	202
d^c^b	10010110	150
~d^c^b	01101001	105
0	00000000	0

FIG. 31A *continued*

Format

G.BOOLEAN rd@trc,trb,f

rd=gbooleani(rd,rc,rb,f)




```
if f6=f5 then
        if f2=f1 then
                if f2 then
                        rc ← max(trc,trb)
                        rb ← min(trc,trb)
                else
                        rc ← min(trc,trb)
                        rb ← max(trc,trb)
                endif
                ih ← 0
                il ← 0 || f6 || f7 || f4 || f3 || f0
        else
                if f2 then
                        rc ← trb
                        rb ← trc
                else
                        rc ← trc
                        rb ← trb
                endif
                ih ← 0
                il ← 1 || f6 || f7 || f4 || f3 || f0
        endif
else
        ih ← 1
        if f6 then
                rc ← trb
                rb ← trc
                il ← f1 || f2 || f7 || f4 || f3 || f0
        else
                rc ← trc
                rb ← trb
                il ← f2 || f1 || f7 || f4 || f3 || f0
        endif
endif
```

FIG. 31B

Definition

```
def GroupBoolean (ih,rd,rc,rb,il)
        d ← RegRead(rd, 128)
        c ← RegRead(rc, 128)
        b ← RegRead(rb, 128)
        if ih=0 then
                if il5=0 then
                        f ← il3 || il4 || il4 || il2 || il1 || (rc>rb)2 || il0
                else
                        f ← il3 || il4 || il4 || il2 || il1 || 0 || 1 || il0
                endif
        else
                f ← il3 || 0 || 1 || il2 || il1 || il5 || il4 || il0
        endif
        for i ← 0 to 127 by size
                ai ← f(di||ci||bi)
        endfor
        RegWrite(rd, 128, a)
enddef
```

Exceptions

none

FIG.31C

Operation codes

G.MUX	Group multiplex

Redundancies

G.MUX ra=rd,rc,rc	⇔	*G.COPY ra=rc*
G.MUX ra=ra,rc,rb	⇔	G.BOOLEAN ra@rc,rb,0x11001010
G.MUX ra=rd,ra,rb	⇔	G.BOOLEAN ra@rd,rb,0x11100010
G.MUX ra=rd,rc,ra	⇔	G.BOOLEAN ra@rd,rc,0x11011000
G.MUX ra=rd,rd,rb	⇔	G.OR ra=rd,rb
G.MUX ra=rd,rc,rd	⇔	G.AND ra=rd,rc

Format

G.MUX ra=rd,rc,rb

ra=gmux(rd,rc,rb)

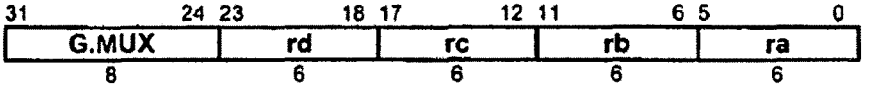


FIG. 31D

Definition

```
def GroupTernary(op,size,rd,rc,rb,ra) as
    d ← RegRead(rd, 128)
    c ← RegRead(rc, 128)
    b ← RegRead(rb, 128)
    case op of
        G.MUX:
            a ← (c and d) or (b and not d)
    endcase
    RegWrite(ra, 128, a)
enddef
```

Exceptions

none

FIG. 31E

Operation codes

G.ADD. 8	Group add bytes
G.ADD. 16	Group add doublets
G.ADD. 32	Group add quadlets
G.ADD. 64	Group add octlets
G.ADD.128	Group add hexlet
G.ADD.L. 8	Group add limit signed bytes
G.ADD.L. 16	Group add limit signed doublets
G.ADD.L. 32	Group add limit signed quadlets
G.ADD.L. 64	Group add limit signed octlets
G.ADD.L.128	Group add limit signed hexlet
G.ADD.L.U. 8	Group add limit unsigned bytes
G.ADD.L.U. 16	Group add limit unsigned doublets
G.ADD.L.U. 32	Group add limit unsigned quadlets
G.ADD.L.U. 64	Group add limit unsigned octlets
G.ADD.L.U.128	Group add limit unsigned hexlet
G.ADD. 8.O	Group add signed bytes check overflow
G.ADD. 16.O	Group add signed doublets check overflow
G.ADD. 32.O	Group add signed quadlets check overflow
G.ADD. 64.O	Group add signed octlets check overflow
G.ADD.128.O	Group add signed hexlet check overflow
G.ADD.U. 8.O	Group add unsigned bytes check overflow
G.ADD.U. 16.O	Group add unsigned doublets check overflow
G.ADD.U. 32.O	Group add unsigned quadlets check overflow
G.ADD.U. 64.O	Group add unsigned octlets check overflow
G.ADD.U.128.O	Group add unsigned hexlet check overflow

Redundancies

G.ADD.size rd=rc,rc	⇔	G.SHL.I.size rd=rc,1
G.ADD.size.O rd=rc,rc	⇔	G.SHL.I.size.O rd=rc,1
G.ADD.U.size.O rd=rc,rc	⇔	G.SHL.I.U.size.O rd=rc,1

FIG. 32A

Format

G.op.size rd=rc,rb

rd=gopsize(rc,rb)

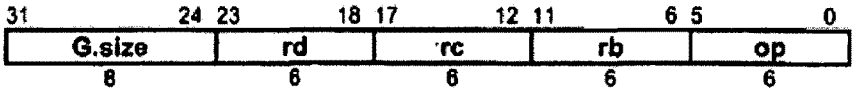


FIG 32B

Definition

```
def Group(op,size,rd,rc,rb)
    c ← RegRead(rc, 128)
    b ← RegRead(rb, 128)
    case op of
        G.ADD:
            for i ← 0 to 128-size by size
                a[i+size-1..i] ← c[i+size-1..i] + b[i+size-1..i]
            endfor
        G.ADD.L:
            for i ← 0 to 128-size by size
                t ← (c[i+size-1] || c[i+size-1..i]) + (b[i+size-1] || b[i+size-1..i])
                a[i+size-1..i] ← (t[size] ≠ t[size-1]) ? (t[size] || t[size-1]^(size-1)) : t[size-1..0]
            endfor
        G.ADD.L.U:
            for i ← 0 to 128-size by size
                t ← (0^1 || c[i+size-1..i]) + (0^1 || b[i+size-1..i])
                a[i+size-1..i] ← (t[size] ≠ 0) ? (1^size) : t[size-1..0]
            endfor
        G.ADD.O:
            for i ← 0 to 128-size by size
                t ← (c[i+size-1] || c[i+size-1..i]) + (b[i+size-1] || b[i+size-1..i])
                if t[size] ≠ t[size-1] then
                    raise FixedPointArithmetic
                endif
                a[i+size-1..i] ← t[size-1..0]
            endfor
        G.ADD.U.O:
            for i ← 0 to 128-size by size
                t ← (0^1 || c[i+size-1..i]) + (0^1 || b[i+size-1..i])
                if t[size] ≠ 0 then
                    raise FixedPointArithmetic
                endif
                a[i+size-1..i] ← t[size-1..0]
            endfor
    endcase
    RegWrite(rd, 128, a)
enddef
```

Exceptions

Fixed-point arithmetic

FIG. 32C

Operation codes

G.SET.AND.E.8	Group set and equal zero bytes
G.SET.AND.E.16	Group set and equal zero doublets
G.SET.AND.E.32	Group set and equal zero quadlets
G.SET.AND.E.64	Group set and equal zero octlets
G.SET.AND.E.128	Group set and equal zero hexlet
G.SET.AND.NE.8	Group set and not equal zero bytes
G.SET.AND.NE.16	Group set and not equal zero doublets
G.SET.AND.NE.32	Group set and not equal zero quadlets
G.SET.AND.NE.64	Group set and not equal zero octlets
G.SET.AND.NE.128	Group set and not equal zero hexlet
G.SET.E.8	Group set equal bytes
G.SET.E.16	Group set equal doublets
G.SET.E.32	Group set equal quadlets
G.SET.E.64	Group set equal octlets
G.SET.E.128	Group set equal hexlet
G.SET.GE.8	Group set greater equal signed bytes
G.SET.GE.16	Group set greater equal signed doublets
G.SET.GE.32	Group set greater equal signed quadlets
G.SET.GE.64	Group set greater equal signed octlets
G.SET.GE.128	Group set greater equal signed hexlet
G.SET.GE.U.8	Group set greater equal unsigned bytes
G.SET.GE.U.16	Group set greater equal unsigned doublets
G.SET.GE.U.32	Group set greater equal unsigned quadlets
G.SET.GE.U.64	Group set greater equal unsigned octlets
G.SET.GE.U.128	Group set greater equal unsigned hexlet
G.SET.L.8	Group set signed less bytes
G.SET.L.16	Group set signed less doublets
G.SET.L.32	Group set signed less quadlets
G.SET.L.64	Group set signed less octlets
G.SET.L.128	Group set signed less hexlet
G.SET.L.U.8	Group set less unsigned bytes
G.SET.L.U.16	Group set less unsigned doublets
G.SET.L.U.32	Group set less unsigned quadlets
G.SET.L.U.64	Group set less unsigned octlets
G.SET.L.U.128	Group set less unsigned hexlet
G.SET.NE.8	Group set not equal bytes
G.SET.NE.16	Group set not equal doublets

FIG 33A

G.SET.NE.32	Group set not equal quadlets
G.SET.NE.64	Group set not equal octlets
G.SET.NE.128	Group set not equal hexlet
G.SUB.8	Group subtract bytes
G.SUB.8.O	Group subtract signed bytes check overflow
G.SUB.16	Group subtract doublets
G.SUB.16.O	Group subtract signed doublets check overflow
G.SUB.32	Group subtract quadlets
G.SUB.32.O	Group subtract signed quadlets check overflow
G.SUB.64	Group subtract octlets
G.SUB.64.O	Group subtract signed octlets check overflow
G.SUB.128	Group subtract hexlet
G.SUB.128.O	Group subtract signed hexlet check overflow
G.SUB.L.8	Group subtract limit signed bytes
G.SUB.L.16	Group subtract limit signed doublets
G.SUB.L.32	Group subtract limit signed quadlets
G.SUB.L.64	Group subtract limit signed octlets
G.SUB.L.128	Group subtract limit signed hexlet
G.SUB.L.U.8	Group subtract limit unsigned bytes
G.SUB.L.U.16	Group subtract limit unsigned doublets
G.SUB.L.U.32	Group subtract limit unsigned quadlets
G.SUB.L.U.64	Group subtract limit unsigned octlets
G.SUB.L.U.128	Group subtract limit unsigned hexlet
G.SUB.U.8.O	Group subtract unsigned bytes check overflow
G.SUB.U.16.O	Group subtract unsigned doublets check overflow
G.SUB.U.32.O	Group subtract unsigned quadlets check overflow
G.SUB.U.64.O	Group subtract unsigned octlets check overflow
G.SUB.U.128.O	Group subtract unsigned hexlet check overflow

FIG 33A *continued*

Equivalencies

G.SET.E.Z.8	Group set equal zero bytes
G.SET.E.Z.16	Group set equal zero doublets
G.SET.E.Z.32	Group set equal zero quadlets
G.SET.E.Z.64	Group set equal zero octlets
G.SET.E.Z.128	Group set equal zero hexlet
G.SET.G.Z.8	Group set greater zero signed bytes
G.SET.G.Z.16	Group set greater zero signed doublets
G.SET.G.Z.32	Group set greater zero signed quadlets
G.SET.G.Z.64	Group set greater zero signed octlets
G.SET.G.Z.128	Group set greater zero signed hexlet
G.SET.GE.Z.8	Group set greater equal zero signed bytes
G.SET.GE.Z.16	Group set greater equal zero signed doublets
G.SET.GE.Z.32	Group set greater equal zero signed quadlets
G.SET.GE.Z.64	Group set greater equal zero signed octlets
G.SET.GE.Z.128	Group set greater equal zero signed hexlet
G.SET.L.Z.8	Group set less zero signed bytes
G.SET.L.Z.16	Group set less zero signed doublets
G.SET.L.Z.32	Group set less zero signed quadlets
G.SET.L.Z.64	Group set less zero signed octlets
G.SET.L.Z.128	Group set less zero signed hexlet
G.SET.LE.Z.8	Group set less equal zero signed bytes
G.SET.LE.Z.16	Group set less equal zero signed doublets
G.SET.LE.Z.32	Group set less equal zero signed quadlets
G.SET.LE.Z.64	Group set less equal zero signed octlets
G.SET.LE.Z.128	Group set less equal zero signed hexlet
G.SET.NE.Z.8	Group set not equal zero bytes
G.SET.NE.Z.16	Group set not equal zero doublets
G.SET.NE.Z.32	Group set not equal zero quadlets
G.SET.NE.Z.64	Group set not equal zero octlets
G.SET.NE.Z.128	Group set not equal zero hexlet

FIG. 33A *continued*

G.SET.LE.8	Group set less equal signed bytes
G.SET.LE.16	Group set less equal signed doublets
G.SET.LE.32	Group set less equal signed quadlets
G.SET.LE.64	Group set less equal signed octlets
G.SET.LE.128	Group set less equal signed hexlet
G.SET.LE.U.8	Group set less equal unsigned bytes
G.SET.LE.U.16	Group set less equal unsigned doublets
G.SET.LE.U.32	Group set less equal unsigned quadlets
G.SET.LE.U.64	Group set less equal unsigned octlets
G.SET.LE.U.128	Group set less equal unsigned hexlet
G.SET.G.8	Group set signed greater bytes
G.SET.G.16	Group set signed greater doublets
G.SET.G.32	Group set signed greater quadlets
G.SET.G.64	Group set signed greater octlets
G.SET.G.128	Group set signed greater hexlet
G.SET.G.U.8	Group set greater unsigned bytes
G.SET.G.U.16	Group set greater unsigned doublets
G.SET.G.U.32	Group set greater unsigned quadlets
G.SET.G.U.64	Group set greater unsigned octlets
G.SET.G.U.128	Group set greater unsigned hexlet

G.SET.E.Z.size rd=rc	←	G.SET.AND.E.size rd=rc,rc
G.SET.G.Z.size rd=rc	⇐	G.SET.L.U.size rd=rc,rc
G.SET.GE.Z.size rd=rc	⇐	G.SET.GE.size rd=rc,rc
G.SET.L.Z.size rd=rc	⇐	G.SET.L.size rd=rc,rc
G.SET.LE.Z.size rd=rc	⇐	G.SET.GE.U.size rd=rc,rc
G.SET.NE.Z.size rd=rc	←	G.SET.AND.NE.size rd=rc,rc
G.SET.G.size rd=rb,rc	→	G.SET.L.size rd=rc,rb
G.SET.G.U.size rd=rb,rc	→	G.SET.L.U.size rd=rc,rb
G.SET.LE.size rd=rb,rc	→	G.SET.GE.size rd=rc,rb
G.SET.LE.U.size rd=rb,rc	→	G.SET.GE.U.size rd=rc,rb

FIG. 33A *continued*

Redundancies

G.SET.E.size rd=rc,rc	⇔	*G.SET rd*
G.SET.NE.size rd=rc,rc	⇔	*G.ZERO rd*
G.SUB.size rd=rc,rc	⇔	*G.ZERO rd*
G.SUB.L.size rd=rc,rc	⇔	*G.ZERO rd*
G.SUB.L.U.size rd=rc,rc	⇔	*G.ZERO rd*
G.SUB.size.O rd=rc,rc	⇔	*G.ZERO rd*
G.SUB.U.size.O rd=rc,rc	⇔	*G.ZERO rd*

Selection

class	operation	cond	operand	size	check
arithmetic	SUB			8 16 32 64 128	
			NONE U	8 16 32 64 128	O
	SUB.L		NONE U	8 16 32 64 128	
boolean	SET.AND SET	E NE		8 16 32 64 128	
	SET	L GE *G LE*	NONE U	8 16 32 64 128	
	SET	*G GE L LE*	*Z*	8 16 32 64 128	

Format

G.op.size rd=rb,rc

rd=gopsize(rb,rc)

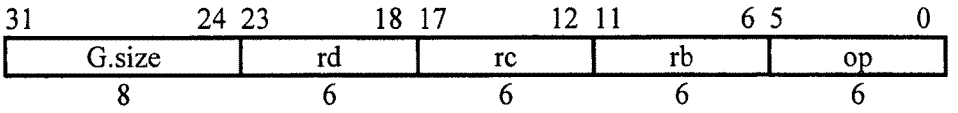


FIG. 33B

Definition

```
def GroupReversed(op,size,rd,rc,rb)
    c ← RegRead(rc, 128)
    b ← RegRead(rb, 128)
    case op of
        G.SUB:
            for i ← 0 to 128-size by size
                a[i+size-1..i] ← b[i+size-1..i] - c[i+size-1..i]
            endfor
        G.SUB.L:
            for i ← 0 to 128-size by size
                t ← (b[i+size-1] || b[i+size-1..i]) - (c[i+size-1] || c[i+size-1..i])
                a[i+size-1..i] ← (t[size] ≠ t[size-1]) ? (t[size] || t[size-1]^size) : t[size-1..0]
            endfor
        G.SUB.LU:
            for i ← 0 to 128-size by size
                t ← (0^1 || b[i+size-1..i]) - (0^1 || c[i+size-1..i])
                a[i+size-1..i] ← (t[size] ≠ 0) ? 0^size : t[size-1..0]
            endfor
        G.SUB.O:
            for i ← 0 to 128-size by size
                t ← (b[i+size-1] || b[i+size-1..i]) - (c[i+size-1] || c[i+size-1..i])
                if (t[size] ≠ t[size-1]) then
                    raise FixedPointArithmetic
                endif
                a[i+size-1..i] ← t[size-1..0]
            endfor
        G.SUB.U.O:
            for i ← 0 to 128-size by size
                t ← (0^1 || b[i+size-1..i]) - (0^1 || c[i+size-1..i])
                if (t[size] ≠ 0) then
                    raise FixedPointArithmetic
                endif
                a[i+size-1..i] ← t[size-1..0]
            endfor
        G.SET.E:
            for i ← 0 to 128-size by size
                a[i+size-1..i] ← (b[i+size-1..i] = c[i+size-1..i])^size
            endfor
        G.SET.NE:
            for i ← 0 to 128-size by size
                a[i+size-1..i] ← (b[i+size-1..i] ≠ c[i+size-1..i])^size
            endfor
        G.SET.AND.E:
            for i ← 0 to 128-size by size
                a[i+size-1..i] ← ((b[i+size-1..i] and c[i+size-1..i]) = 0)^size
            endfor
```

FIG. 33C

```
        G.SET.AND.NE:
            for i ← 0 to 128-size by size
                a[i+size-1..i] ← ((b[i+size-1..i] and c[i+size-1..i]) ≠ 0)^size
            endfor
        G.SET.L:
            for i ← 0 to 128-size by size
                a[i+size-1..i] ← ((rc = rb) ? (b[i+size-1..i] < 0) : (b[i+size-1..i] < c[i+size-1..i]))^size
            endfor
        G.SET.GE:
            for i ← 0 to 128-size by size
                a[i+size-1..i] ← ((rc = rb) ? (b[i+size-1..i] ≥ 0) : (b[i+size-1..i] ≥ c[i+size-1..i]))^size
            endfor
        G.SET.L.U:
            for i ← 0 to 128-size by size
                a[i+size-1..i] ← ((rc = rb) ? (b[i+size-1..i] > 0) :
                    ((0 || b[i+size-1..i]) < (0 || c[i+size-1..i])))^size
            endfor
        G.SET.GE.U:
            for i ← 0 to 128-size by size
                a[i+size-1..i] ← ((rc = rb) ? (b[i+size-1..i] ≤ 0) :
                    ((0 || b[i+size-1..i]) ≥ (0 || c[i+size-1..i])))^size
            endfor
    endcase
    RegWrite(rd, 128, a)
enddef
```

Exceptions

Fixed-point arithmetic

FIG. 33C *continued*

E.DIV.64	Ensemble divide signed octlets
E.DIV.U.64	Ensemble divide unsigned octlets
E.MUL.8	Ensemble multiply signed bytes
E.MUL.16	Ensemble multiply signed doublets
E.MUL.32	Ensemble multiply signed quadlets
E.MUL.64	Ensemble multiply signed octlets
E.MUL.SUM.8	Ensemble multiply sum signed bytes
E.MUL.SUM.16	Ensemble multiply sum signed doublets
E.MUL.SUM.32	Ensemble multiply sum signed quadlets
E.MUL.SUM.64	Ensemble multiply sum signed octlets
E.MUL.C.8	Ensemble complex multiply bytes
E.MUL.C.16	Ensemble complex multiply doublets
E.MUL.C.32	Ensemble complex multiply quadlets
E.MUL.M.8	Ensemble multiply mixed-signed bytes
E.MUL.M.16	Ensemble multiply mixed-signed doublets
E.MUL.M.32	Ensemble multiply mixed-signed quadlets
E.MUL.M.64	Ensemble multiply mixed-signed octlets
E.MUL.P.8	Ensemble multiply polynomial bytes
E.MUL.P.16	Ensemble multiply polynomial doublets
E.MUL.P.32	Ensemble multiply polynomial quadlets
E.MUL.P.64	Ensemble multiply polynomial octlets
E.MUL.SUM.C.8	Ensemble multiply sum complex bytes
E.MUL.SUM.C.16	Ensemble multiply sum complex doublets
E.MUL.SUM.C.32	Ensemble multiply sum complex quadlets
E.MUL.SUM.M.8	Ensemble multiply sum mixed-signed bytes
E.MUL.SUM.M.16	Ensemble multiply sum mixed-signed doublets
E.MUL.SUM.M.32	Ensemble multiply sum mixed-signed quadlets
E.MUL.SUM.M.64	Ensemble multiply sum mixed-signed octlets
E.MUL.SUM.U.8	Ensemble multiply sum unsigned bytes
E.MUL.SUM.U.16	Ensemble multiply sum unsigned doublets
E.MUL.SUM.U.32	Ensemble multiply sum unsigned quadlets
E.MUL.SUM.U.64	Ensemble multiply sum unsigned octlets
E.MUL.U.8	Ensemble multiply unsigned bytes
E.MUL.U.16	Ensemble multiply unsigned doublets
E.MUL.U.32	Ensemble multiply unsigned quadlets
E.MUL.U.64	Ensemble multiply unsigned octlets

FIG. 34A

Format

E.op.size rd=rc,rb

rd=eopsize(rc,rb)

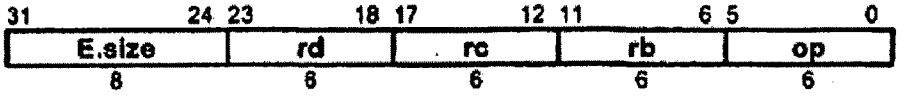


FIG. 34B

Definition

```
def mul(size,h,vs,v,i,ws,w,j) as
        mul ← ((vs&v_{size-1+i})^{h-size} || v_{size-1+i,..i}) * ((ws&w_{size-1+j})^{h-size} || w_{size-1+j,..j})
enddef

def c ← PolyMultiply(size,a,b) as
        p[0] ← 0^{2*size}
        for k ← 0 to size-1
                p[k+1] ← p[k] ^ a_k ? (0^{size-k} || b || 0^k) : 0^{2*size}
        endfor
        c ← p[size]
enddef

def Ensemble(op,size,rd,rc,rb)
        c ← RegRead(rc, 128)
        b ← RegRead(rb, 128)
        case op of
                E.MUL:, E.MUL.C:, EMUL.SUM, E.MUL.SUM.C, E.CON, E.CON.C, E.DIV:
                        cs ← bs ← 1
                E.MUL.M:, EMUL.SUM.M, E.CON.M:
                        cs ← 0
                        bs ← 1
                E.MUL.U:, EMUL.SUM.U, E.CON.U, E.DIV.U, E.MUL.P:
                        cs ← bs ← 0
        endcase
        case op of
                E.MUL, E.MUL.U, E.MUL.M:
                        for i ← 0 to 64-size by size
                                d_{2*(i+size)-1..2*i} ← mul(size,2*size,cs,c,i,bs,b,i)
                        endfor
                E.MUL.P:
                        for i ← 0 to 64-size by size
                                d_{2*(i+size)-1..2*i} ← PolyMultiply(size,c_{size-1+i,..i},b_{size-1+i,..i})
                        endfor
                E.MUL.C:
                        for i ← 0 to 64-size by size
                                if (i and size) = 0 then
                                        p ← mul(size,2*size,1,c,i,1,b,i) - mul(size,2*size,1,c,i+size,1,b,i+size)
                                else
                                        p ← mul(size,2*size,1,c,i,1,b,i+size) + mul(size,2*size,1,c,i,1,b,i+size)
                                endif
                                d_{2*(i+size)-1..2*i} ← p
                        endfor
                E.MUL.SUM, E.MUL.SUM.U, E.MUL.SUM.M:
                        p[0] ← 0^{128}
                        for i ← 0 to 128-size by size
                                p[i+size] ← p[i] + mul(size,128,cs,c,i,bs,b,i)
                        endfor
                        a ← p[126]
                E.MUL.SUM.C:
                        p[0] ← 0^{64}
                        p[size] ← 0^{64}
                        for i ← 0 to 128-size by size
                                if (i and size) = 0 then
                                        p[i+2*size] ← p[i] + mul(size,64,1,c,i,1,b,i)
                                                          - mul(size,64,1,c,i+size,1,b,i+size)
                                else
                                        p[i+2*size] ← p[i] + mul(size,64,1,c,i,1,b,i+size)
                                                          + mul(size,64,1,c,i+size,1,b,i)
                                endif
                        endfor
                        a ← p[128+size] || p[128]
```

FIG. 34C

```
        E.CON, E.CON.U, E.CON.M:
            p[0] ← 0^128
            for j ← 0 to 64-size by size
                for i ← 0 to 64-size by size
                    p[j+size]2*(i+size)-1..2*i ← p[j]2*(i+size)-1..2*i +
                        mul(size,2*size,cs,c,i+64-j,bs,b,j)
                endfor
            endfor
            a ← p[64]
        E.CON.C:
            p[0] ← 0^128
            for j ← 0 to 64-size by size
                for i ← 0 to 64-size by size
                    if ((-i) and j and size) = 0 then
                        p[j+size]2*(i+size)-1..2*i ← p[j]2*(i+size)-1..2*i +
                            mul(size,2*size,1,c,i+64-j,1,b,j)
                    else
                        p[j+size]2*(i+size)-1..2*i ← p[j]2*(i+size)-1..2*i +
                            mul(size,2*size,1,c,i+64-j+2*size,1,b,j)
                    endif
                endfor
            endfor
            a ← p[64]
        E.DIV:
            if (b = 0) or ( (c = (1||0^63)) and (b = 1^64) ) then
                a ← undefined
            else
                q ← c / b
                r ← c - q*b
                a ← r63..0 || q63..0
            endif
        E.DIV.U:
            if b = 0 then
                a ← undefined
            else
                q ← (0 || c) / (0 || b)
                r ← c - (0 || q)*(0 || b)
                a ← r63..0 || q63..0
            endif
    endcase
    RegWrite(rd, 128, a)
enddef
```

Exceptions

none

FIG 34C *continued*

Operation codes

G.COM.AND.E. 8	Group compare and equal zero bytes
G.COM.AND.E. 16	Group compare and equal zero doublets
G.COM.AND.E. 32	Group compare and equal zero quadlets
G.COM.AND.E. 64	Group compare and equal zero octlets
G.COM.AND.E.128	Group compare and equal zero hexlet
G.COM.AND.NE. 8	Group compare and not equal zero bytes
G.COM.AND.NE. 16	Group compare and not equal zero doublets
G.COM.AND.NE. 32	Group compare and not equal zero quadlets
G.COM.AND.NE. 64	Group compare and not equal zero octlets
G.COM.AND.NE.128	Group compare and not equal zero hexlet
G.COM.E. 8	Group compare equal bytes
G.COM.E. 16	Group compare equal doublets
G.COM.E. 32	Group compare equal quadlets
G.COM.E. 64	Group compare equal octlets
G.COM.E.128	Group compare equal hexlet
G.COM.GE. 8	Group compare greater equal signed bytes
G.COM.GE. 16	Group compare greater equal signed doublets
G.COM.GE. 32	Group compare greater equal signed quadlets
G.COM.GE. 64	Group compare greater equal signed octlets
G.COM.GE.128	Group compare greater equal signed hexlet
G.COM.GE.U. 8	Group compare greater equal unsigned bytes
G.COM.GE.U. 16	Group compare greater equal unsigned doublets
G.COM.GE.U. 32	Group compare greater equal unsigned quadlets
G.COM.GE.U. 64	Group compare greater equal unsigned octlets
G.COM.GE.U.128	Group compare greater equal unsigned hexlet
G.COM.L. 8	Group compare signed less bytes
G.COM.L. 16	Group compare signed less doublets
G.COM.L. 32	Group compare signed less quadlets
G.COM.L. 64	Group compare signed less octlets
G.COM.L.128	Group compare signed less hexlet
G.COM.L.U. 8	Group compare less unsigned bytes
G.COM.L.U. 16	Group compare less unsigned doublets
G.COM.L.U. 32	Group compare less unsigned quadlets
G.COM.L.U. 64	Group compare less unsigned octlets
G.COM.L.U.128	Group compare less unsigned hexlet
G.COM.NE. 8	Group compare not equal bytes
G.COM.NE. 16	Group compare not equal doublets
G.COM.NE. 32	Group compare not equal quadlets
G.COM.NE. 64	Group compare not equal octlets
G.COM.NE.128	Group compare not equal hexlet

FIG. 35A

Equivalencies

G.COM.E.Z. 8	Group compare equal zero signed bytes
G.COM.E.Z. 16	Group compare equal zero signed doublets
G.COM.E.Z. 32	Group compare equal zero signed quadlets
G.COM.E.Z. 64	Group compare equal zero signed octlets
G.COM.E.Z.128	Group compare equal zero signed hexlet
G.COM.G. 8	Group compare signed greater bytes
G.COM.G. 16	Group compare signed greater doublets
G.COM.G. 32	Group compare signed greater quadlets
G.COM.G. 64	Group compare signed greater octlets
G.COM.G.128	Group compare signed greater hexlet
G.COM.G.U. 8	Group compare greater unsigned bytes
G.COM.G.U. 16	Group compare greater unsigned doublets
G.COM.G.U. 32	Group compare greater unsigned quadlets
G.COM.G.U. 64	Group compare greater unsigned octlets
G.COM.G.U.128	Group compare greater unsigned hexlet
G.COM.G.Z. 8	Group compare greater zero signed bytes
G.COM.G.Z. 16	Group compare greater zero signed doublets
G.COM.G.Z. 32	Group compare greater zero signed quadlets
G.COM.G.Z. 64	Group compare greater zero signed octlets
G.COM.G.Z.128	Group compare greater zero signed hexlet
G.COM.GE.Z. 8	Group compare greater equal zero signed bytes
G.COM.GE.Z. 16	Group compare greater equal zero signed doublets
G.COM.GE.Z. 32	Group compare greater equal zero signed quadlets
G.COM.GE.Z. 64	Group compare greater equal zero signed octlets
G.COM.GE.Z.128	Group compare greater equal zero signed hexlet
G.COM.L.Z. 8	Group compare less zero signed bytes
G.COM.L.Z. 16	Group compare less zero signed doublets
G.COM.L.Z. 32	Group compare less zero signed quadlets
G.COM.L.Z. 64	Group compare less zero signed octlets
G.COM.L.Z.128	Group compare less zero signed hexlet
G.COM.LE. 8	Group compare less equal signed bytes
G.COM.LE. 16	Group compare less equal signed doublets
G.COM.LE. 32	Group compare less equal signed quadlets
G.COM.LE. 64	Group compare less equal signed octlets
G.COM.LE.128	Group compare less equal signed hexlet
G.COM.LE.U. 8	Group compare less equal unsigned bytes
G.COM.LE.U. 16	Group compare less equal unsigned doublets
G.COM.LE.U. 32	Group compare less equal unsigned quadlets
G.COM.LE.U. 64	Group compare less equal unsigned octlets
G.COM.LE.U.128	Group compare less equal unsigned hexlet

FIG. 35A *continued*

G.COM.LE.Z. 8	Group compare less equal zero signed bytes
G.COM.LE.Z. 16	Group compare less equal zero signed doublets
G.COM.LE.Z. 32	Group compare less equal zero signed quadlets
G.COM.LE.Z. 64	Group compare less equal zero signed octlets
G.COM.LE.Z.128	Group compare less equal zero signed hexlet
G.COM.NE.Z. 8	Group compare not equal zero signed bytes
G.COM.NE.Z. 16	Group compare not equal zero signed doublets
G.COM.NE.Z. 32	Group compare not equal zero signed quadlets
G.COM.NE.Z. 64	Group compare not equal zero signed octlets
G.COM.NE.Z.128	Group compare not equal zero signed hexlet
G.FIX	Group fixed point arithmetic exception
G.NOP	Group no operation

G.COM.E.Z.size rc	←	G.COM.AND.E.size rc,rc
G.COM.G.size rd,rc	→	G.COM.L.size rc,rd
G.COM.G.U.size rd,rc	→	G.COM.L.U.size rc,rd
G.COM.G.Z.size rc	⇐	G.COM.L.U.size rc,rc
G.COM.GE.Z.size rc	⇐	G.COM.GE.size rc,rc
G.COM.L.Z.size rc	⇐	G.COM.L.size rc,rc
G.COM.LE.size rd,rc	→	G.COM.GE.size rc,rd
G.COM.LE.U.size rd,rc	→	G.COM.GE.U.size rc,rd
G.COM.LE.Z.size rc	⇐	G.COM.GE.U.size rc,rc
G.COM.NE.Z.size rc	←	G.COM.AND.NE.size rc,rc
G.FIX	←	G.COM.E.128 r0,r0
G.NOP	←	G.COM.NE.128 r0,r0

FIG. 35A *continued*

Redundancies

G.COM.E.size rd,rd	⇔	*G.FIX*
G.COM.NE.size rd,rd	⇔	*G.NOP*

Selection

class	operation	cond	type	size
boolean	COM.AN D COM	E NE		8 16 32 64 128
arithmetic	COM	L GE *G LE*	NONE U	□ 16 32 64 128
	COM	*L GE G LE E NE*	*Z*	8 16 32 64 128

Format

G.COM.op.size rd,rc
G.COM.opz.size rcd

gcomopsize(rd,rc)

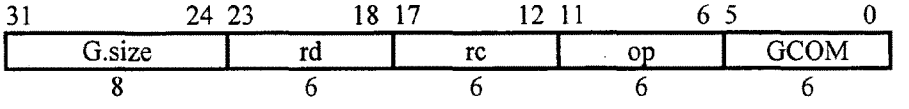


FIG. 35B

Definition

```
def GroupCompare(op,size,rd,rc)
    d ← RegRead(rd, 128)
    c ← RegRead(rc, 128)
    case op of
        G.COM.E:
            for i ← 0 to 128-size by size
                a[i+size-1..i] ← (d[i+size-1..i] = c[i+size-1..i])^size
            endfor
        G.COM.NE:
            for i ← 0 to 128-size by size
                a[i+size-1..i] ← (d[i+size-1..i] ≠ c[i+size-1..i])^size
            endfor
        G.COM.AND.E:
            for i ← 0 to 128-size by size
                a[i+size-1..i] ← ((c[i+size-1..i] and d[i+size-1..i]) = 0)^size
            endfor
        G.COM.AND.NE:
            for i ← 0 to 128-size by size
                a[i+size-1..i] ← ((c[i+size-1..i] and d[i+size-1..i]) ≠ 0)^size
            endfor
        G.COM.L:
            for i ← 0 to 128-size by size
                a[i+size-1..i] ← ((rd = rc) ? (c[i+size-1..i] < 0) : (d[i+size-1..i] < c[i+size-1..i]))^size
            endfor
        G.COM.GE:
            for i ← 0 to 128-size by size
                a[i+size-1..i] ← ((rd = rc) ? (c[i+size-1..i] ≥ 0) : (d[i+size-1..i] ≥ c[i+size-1..i]))^size
            endfor
        G.COM.L.U:
            for i ← 0 to 128-size by size
                a[i+size-1..i] ← ((rd = rc) ? (c[i+size-1..i] > 0) :
                    ((0 || d[i+size-1..i]) < (0 || c[i+size-1..i])))^size
            endfor
        G.COM.GE.U:
            for i ← 0 to 128-size by size
                a[i+size-1..i] ← ((rd = rc) ? (c[i+size-1..i] ≤ 0) :
                    ((0 || d[i+size-1..i]) ≥ (0 || c[i+size-1..i])))^size
            endfor
    endcase
    if (a ≠ 0) then
        raise FixedPointArithmetic
    endif
enddef
```

Exceptions

Fixed-point arithmetic

FIG. 35C

E.LOG.MOST.8	Ensemble log of most significant bit signed bytes
E.LOG.MOST.16	Ensemble log of most significant bit signed doublets
E.LOG.MOST.32	Ensemble log of most significant bit signed quadlets
E.LOG.MOST.64	Ensemble log of most significant bit signed octlets
E.LOG.MOST.128	Ensemble log of most significant bit signed hexlet
E.LOG.MOST.U.8	Ensemble log of most significant bit unsigned bytes
E.LOG.MOST.U.16	Ensemble log of most significant bit unsigned doublets
E.LOG.MOST.U.32	Ensemble log of most significant bit unsigned quadlets
E.LOG.MOST.U.64	Ensemble log of most significant bit unsigned octlets
E.LOG.MOST.U.128	Ensemble log of most significant bit unsigned hexlet
E.SUM.8	Ensemble sum signed bytes
E.SUM.16	Ensemble sum signed doublets
E.SUM.32	Ensemble sum signed quadlets
E.SUM.64	Ensemble sum signed octlets
E.SUM.U.1	Ensemble sum unsigned bits
E.SUM.U.8	Ensemble sum unsigned bytes
E.SUM.U.16	Ensemble sum unsigned doublets
E.SUM.U.32	Ensemble sum unsigned quadlets
E.SUM.U.64	Ensemble sum unsigned octlets

Selection

class	op	size
sum	SUM	8 16 32 64
	SUM.U	1 8 16 32 64 ..
log most significant bit	LOG.MOST LOG.MOST.U	8 16 32 64 128

FIG. 36A

Format

E.op.size rd=rc

rd=eopsize(rc)

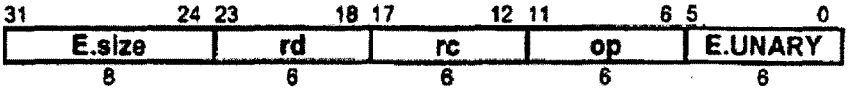


FIG. 36B

Definition

```
def EnsembleUnary(op,size,rd,rc)
    c ← RegRead(rc, 128)
    case op of
        E.LOG.MOST:
            for i ← 0 to 128-size by size
                if (c_{i+size-1..i} =0) then
                    a_{i+size-1..i} ← -1
                else
                    for j ← 0 to size-1
                        if c_{size-1+i..j+i} = (c^{size-1-j}_{size-1+i} || not c_{size-1+i}) then
                            a_{i+size-1..i} ← j
                        endif
                    endfor
                endif
            endfor
        E.LOG.MOSTU:
            for i ← 0 to 128-size by size
                if (c_{i+size-1..i} =0) then
                    a_{i+size-1..i} ← -1
                else
                    for j ← 0 to size-1
                        if c_{size-1+i..j+i} = (0^{size-1-j} || 1) then
                            a_{i+size-1..i} ← j
                        endif
                    endfor
                endif
            endfor
        E.SUM:
            p[0] ← 0^128
            for i ← 0 to 128-size by size
                p[i+size] ← p[i] + (c^{128-size}_{size-1+i} || c_{size-1+i..i})
            endfor
            a ← p[128]
        E.SUMU:
            p[0] ← 0^128
            for i ← 0 to 128-size by size
                p[i+size] ← p[i] + (0^{128-size} || c_{size-1+i..i})
            endfor
            a ← p[128]
    endcase
    RegWrite(rd, 128, a)
enddef
```

Exceptions

none

FIG. 36C

Floating-point function Definitions

```
def eb ← ebits(prec) as
    case pref of
        16:
            eb ← 5
        32:
            eb ← 8
        64:
            eb ← 11
        128:
            eb ← 15
    endcase
enddef

def eb ← ebias(prec) as
    eb ← 0 || 1^(ebits(prec)-1)
enddef

def fb ← fbits(prec) as
    fb ← prec – 1 – eb
enddef

def a ← F(prec, ai) as
    a.s ← ai_(prec-1)
    ae ← ai_(prec-2..fbits(prec))
    af ← ai_(fbits(prec)-1..0)
    if ae = 1^(ebits(prec)) then
        If af = 0 then
            a.t ← INFINITY
        elseif af_(fbits(prec)-1) then
            a.t ← SNaN
            a.e ← -fbits(prec)
            a.f ← 1 || af_(fbits(prec)-2..0)
        else
            a.t ← QNaN
            a.e ← -fbits(prec)
            a.f ← af
        endif
```

FIG. 37

```
        elseif ae = 0 then
            if af = 0 then
                a.t ← ZERO
            else
                a.t ← NORM
                a.e ← 1-ebias(prec)-fbits(prec)
                a.f ← 0 || af
            endif
        else
            a.t ← NORM
            a.e ← ae-ebias(prec)-fbits(prec)
            a.f ← 1 || af
        endif
enddef

def a ← DEFAULTQNAN as
        a.s ← 0
        a.t ← QNAN
        a.e ← -1
        a.f ← 1
enddef

def a ← DEFAULTSNAN as
        a.s ← 0
        a.t ← SNAN
        a.e ← -1
        a.f ← 1
enddef

def fadd(a,b) as faddr(a,b,N) enddef

def c ← faddr(a,b,round) as
        if a.t=NORM and b.t=NORM then
            // d,e are a,b with exponent aligned and fraction adjusted
            if a.e > b.e then
                d ← a
                e.t ← b.t
                e.s ← b.s
                e.e ← a.e
                e.f ← b.f || 0^(a.e-b.e)
            else if a.e < b.e then
                d.t ← a.t
                d.s ← a.s
                d.e ← b.e
                d.f ← a.f || 0^(b.e-a.e)
                e ← b
            endif
            c.t ← d.t
            c.e ← d.e
            if d.s = e.s then
                c.s ← d.s
                c.f ← d.f + e.f
            elseif d.f > e.f then
                c.s ← d.s
                c.f ← d.f – e.f
```

FIG. 37 *continued*

```
            elseif d.f < e.f then
                c.s ← e.s
                c.f ← e.f − d.f
            else
                c.s ← r=F
                c.t ← ZERO
            endif
        // priority is given to b operand for NaN propagation
        elseif (b.t=SNAN) or (b.t=QNAN) then
            c ← b
        elseif (a.t=SNAN) or (a.t=QNAN) then
            c ← a
        elseif a.t=ZERO and b.t=ZERO then
            c.t ← ZERO
            c.s ← (a.s and b.s) or (round=F and (a.s or b.s))
        // NULL values are like zero, but do not combine with ZERO to alter sign
        elseif a.t=ZERO or a.t=NULL then
            c ← b
        elseif b.t=ZERO or b.t=NULL then
            c ← a
        elseif a.t=INFINITY and b.t=INFINITY then
            if a.s ≠ b.s then
                c ← DEFAULTSNAN // Invalid
            else
                c ← a
            endif
        elseif a.t=INFINITY then
            c ← a
        elseif b.t=INFINITY then
            c ← b
        else
            assert FALSE // should have covered al the cases above
        endif
enddef

def b ← fneg(a) as
        b.s ← ~a.s
        b.t ← a.t
        b.e ← a.e
        b.f ← a.f
enddef

def fsub(a,b) as fsubr(a,b,N) enddef

def fsubr(a,b,round) as faddr(a,fneg(b),round) enddef

def frsub(a,b) as frsubr(a,b,N) enddef

def frsubr(a,b,round) as faddr(fneg(a),b,round) enddef

def c ← fcom(a,b) as
        if (a.t=SNAN) or (a.t=QNAN) or (b.t=SNAN) or (b.t=QNAN) then
            c ← U
        elseif a.t=INFINITY and b.t=INFINITY then
            if a.s ≠ b.s then
                c ← (a.s=0) ? G: L
```

FIG. 37 *continued*

```
            else
                c ← E
            endif
        elseif a.t=INFINITY then
            c ← (a.s=0) ? G: L
        elseif b.t=INFINITY then
            c ← (b.s=0) ? G: L
        elseif a.t=NORM and b.t=NORM then
            if a.s ≠ b.s then
                c ← (a.s=0) ? G: L
            else
                if a.e > b.e then
                    af ← a.f
                    bf ← b.f || 0^(a.e-b.e)
                else
                    af ← a.f || 0^(b.e-a.e)
                    bf ← b.f
                endif
                if af = bf then
                    c ← E
                else
                    c ← ((a.s=0) ^ (af > bf)) ? G : L
                endif
            endif
        elseif a.t=NORM then
            c ← (a.s=0) ? G: L
        elseif b.t=NORM then
            c ← (b.s=0) ? G: L
        elseif a.t=ZERO and b.t=ZERO then
            c ← E
        else
            assert FALSE // should have covered all the cases above
        endif
    enddef

    def c ← fmul(a,b) as
        if a.t=NORM and b.t=NORM then
            c.s ← a.s ^ b.s
            c.t ← NORM
            c.e ← a.e + b.e
            c.f ← a.f * b.f
        // priority is given to b operand for NaN propagation
        elseif (b.t=SNAN) or (b.t=QNAN) then
            c.s ← a.s ^ b.s
            c.t ← b.t
            c.e ← b.e
            c.f ← b.f
        elseif (a.t=SNAN) or (a.t=QNAN) then
            c.s ← a.s ^ b.s
            c.t ← a.t
            c.e ← a.e
            c.f ← a.f
        elseif a.t=ZERO and b.t=INFINITY then
            c ← DEFAULTSNAN // Invalid
        elseif a.t=INFINITY and b.t=ZERO then
            c ← DEFAULTSNAN // Invalid
```

FIG. 37 *continued*