Legal Issues to Consider

Ethics

Morals - Ethics

Values

Principles

The Golden Rule

Universality

Transitivity

We, the members of the IEEE, in recognition of the importance of our technologies in affecting the quality of life throughout the world, and in accepting a personal obligation to our profession, its members and the communities we serve, do hereby commit ourselves to the highest ethical and professional conduct and agree:

1. to accept responsibility in making engineering decisions consistent with the safety, health and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;

2. to avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist;

3. to be honest and realistic in stating claims or estimates based on available data;

4. to reject bribery in all its forms;

5. to improve the understanding of technology, its appropriate application, and potential consequences;

6. to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;

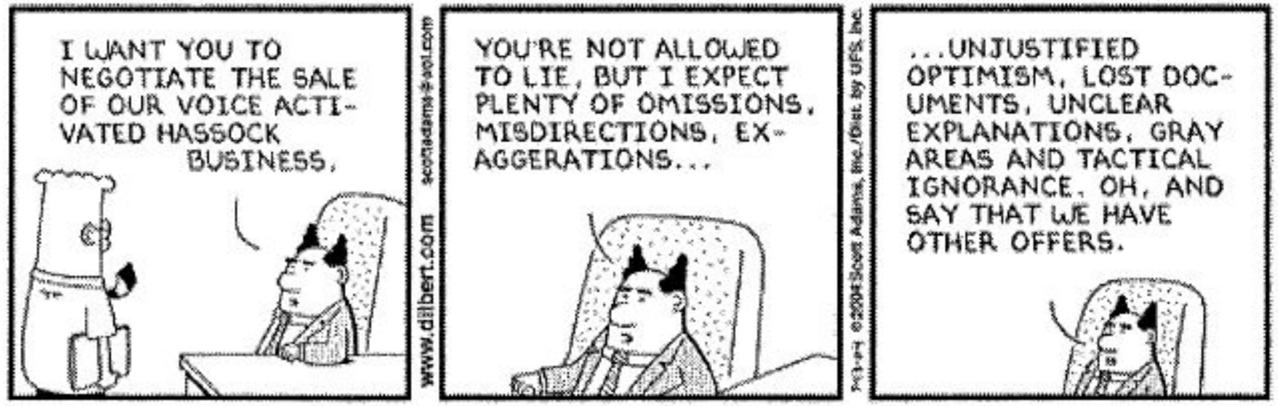
7. to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;

8. to treat fairly all persons regardless of such factors as race, religion, gender, disability, age, or national origin;

9. to avoid injuring others, their property, reputation, or employment by false or malicious action;

10. to assist colleagues and co-workers in their professional development and to support them in following this code of ethics.

DILBERT® by Scott Adams


I WANT YOU TO NEGOTIATE THE SALE OF OUR VOICE ACTIVATED HASSOCK BUSINESS.
YOU'RE NOT ALLOWED TO LIE, BUT I EXPECT PLENTY OF OMISSIONS, MISDIRECTIONS, EXAGGERATIONS...
...UNJUSTIFIED OPTIMISM, LOST DOCUMENTS, UNCLEAR EXPLANATIONS, GRAY AREAS AND TACTICAL IGNORANCE. OH, AND SAY THAT WE HAVE OTHER OFFERS.
www.dilbert.com
scottadams@aol.com
©2004 Scott Adams, Inc./Dist. by UFS, Inc.

Origins:

- Large investments in pipeline
- Tensions between engineering and marketing

Incorporate into the design:

- No single solution to the problem
- Failure mode analysis
- Apply safety standards
- Redundancy for reliability
- Inhibit misuse
- Tight cost/schedule estimates
- Design reviews

Intellectual Property & Legal Liability

Patents

Trade Secrets

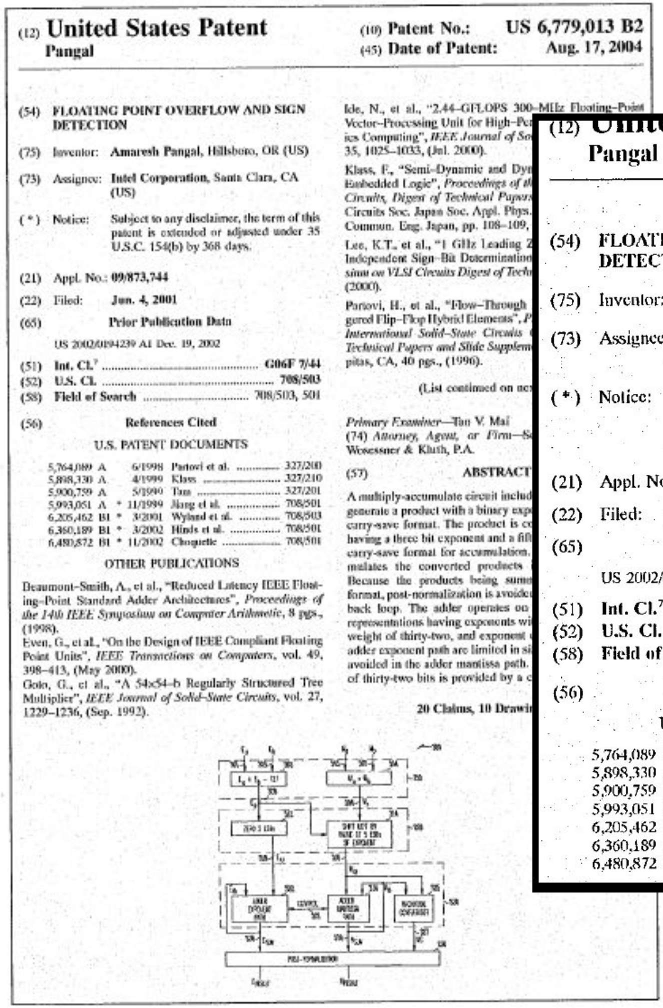
Copyrights

Patents

- Citation of prior art
- Operation/reduce to practice
- Claims

Trade Secrets

Copyrights



(12) **United States Patent**
Pangal

(10) **Patent No.:** **US 6,779,013 B2**
(45) **Date of Patent:** **Aug. 17, 2004**

(54) **FLOATING POINT OVERFLOW AND SIGN DETECTION**

(75) Inventor: **Amaresh Pangal**, Hillsboro, OR (US)

(73) Assignee: **Intel Corporation**, 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 368 days.

(21) Appl. No.: **09/873,744**

(22) Filed: **Jun. 4, 2001**

(65) **Prior Publication Data**

US 2002/0194239 A1 Dec. 19, 2002

(51) **Int. Cl.**^{7} **G06F 7/44**
(52) **U.S. Cl.** **708/503**
(58) **Field of Search** 708/503, 501

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,764,089	A		6/1998	Partovi et al.	327/200
5,898,330	A		4/1999	Klass	327/210
5,900,759	A		5/1999	Tam	327/201
5,993,051	A	*	11/1999	Jiang et al.	708/501
6,205,462	B1	*	3/2001	Wyland et al.	708/503
6,360,189	B1	*	3/2002	Hinds et al.	708/501
6,480,872	B1	*	11/2002	Choquette	708/501

OTHER PUBLICATIONS

Beaumont-Smith, A., et al., "Reduced Latency IEEE Floating-Point Standard Adder Architectures", *Proceedings of the 14th IEEE Symposium on Computer Arithmetic*, 8 pgs., (1998).
Even, G., et al., "On the Design of IEEE Compliant Floating Point Units", *IEEE Transactions on Computers*, vol. 49, 398–413, (May 2000).
Goto, G., et al., "A 54x54–b Regularly Structured Tree Multiplier", *IEEE Journal of Solid-State Circuits*, vol. 27, 1229–1236, (Sep. 1992).

Ide, N., et al., "2.44–GFLOPS 300–MHz Floating–Point Vector–Processing Unit for High–Per
ics Computing", *IEEE Journal of So*
35, 1025–1033, (Jul. 2000).
Klass, F., "Semi–Dynamic and Dyn
Embedded Logic", *Proceedings of th*
Circuits, Digest of Technical Papers
Circuits Soc. Japan Soc. Appl. Phys.
Commun. Eng. Japan, pp. 108–109,
Lee, K.T., et al., "1 GHz Leading Z
Independent Sign–Bit Determination
sium on VLSI Circuits Digest of Tech
(2000).
Partovi, H., et al., "Flow–Through
gered Flip–Flop Hybrid Elements", *P*
International Solid–State Circuits
Technical Papers and Slide Suppleme
pitas, CA, 40 pgs., (1996).

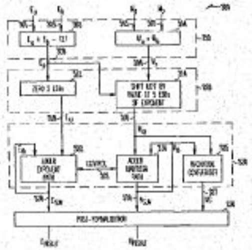
(List continued on ne

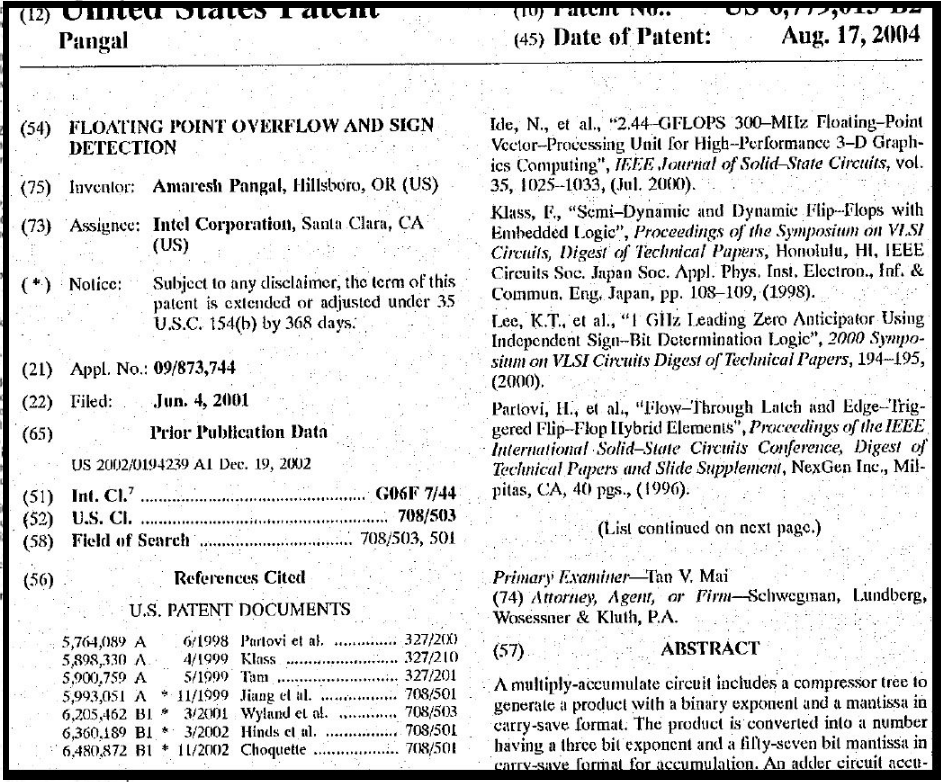
Primary Examiner—Tan V. Mai
(74) *Attorney, Agent, or Firm*—S
Wosessner & Kluth, P.A.

(57) **ABSTRACT**

A multiply-accumulate circuit includ
generate a product with a binary exp
carry-save format. The product is co
having a three bit exponent and a fift
carry-save format for accumulation.
mulates the converted products
Because the products being summ
format, post-normalization is avoide
back loop. The adder operates on
representations having exponents wi
weight of thirty-two, and exponent
adder exponent path are limited in si
avoided in the adder mantissa path.
of thirty-two bits is provided by a c

20 Claims, 10 Drawi





(12) **United States Patent**
Pangal

(10) **Patent No.:** **US 6,779,013 B2**
(45) **Date of Patent:** **Aug. 17, 2004**

(54) **FLOATING POINT OVERFLOW AND SIGN DETECTION**

(75) Inventor: **Amaresh Pangal**, Hillsboro, OR (US)

(73) Assignee: **Intel Corporation**, 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 368 days.

(21) Appl. No.: **09/873,744**

(22) Filed: **Jun. 4, 2001**

(65) **Prior Publication Data**

US 2002/0194239 A1 Dec. 19, 2002

(51) **Int. Cl.**^{7} **G06F 7/44**
(52) **U.S. Cl.** **708/503**
(58) **Field of Search** 708/503, 501

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,764,089	A		6/1998	Partovi et al.	327/200
5,898,330	A		4/1999	Klass	327/210
5,900,759	A		5/1999	Tam	327/201
5,993,051	A	*	11/1999	Jiang et al.	708/501
6,205,462	B1	*	3/2001	Wyland et al.	708/503
6,360,189	B1	*	3/2002	Hinds et al.	708/501
6,480,872	B1	*	11/2002	Choquette	708/501

Ide, N., et al., "2.44–GFLOPS 300–MHz Floating–Point Vector–Processing Unit for High–Performance 3–D Graphics Computing", *IEEE Journal of Solid–State Circuits*, vol. 35, 1025–1033, (Jul. 2000).

Klass, F., "Semi–Dynamic and Dynamic Flip–Flops with Embedded Logic", *Proceedings of the Symposium on VLSI Circuits, Digest of Technical Papers*, Honolulu, HI, IEEE Circuits Soc. Japan Soc. Appl. Phys. Inst. Electron., Inf. & Commun. Eng. Japan, pp. 108–109, (1998).

Lee, K.T., et al., "1 GHz Leading Zero Anticipator Using Independent Sign–Bit Determination Logic", *2000 Symposium on VLSI Circuits Digest of Technical Papers*, 194–195, (2000).

Partovi, H., et al., "Flow–Through Latch and Edge–Triggered Flip–Flop Hybrid Elements", *Proceedings of the IEEE International Solid–State Circuits Conference, Digest of Technical Papers and Slide Supplement*, NexGen Inc., Milpitas, CA, 40 pgs., (1996).

(List continued on next page.)

Primary Examiner—Tan V. Mai
(74) *Attorney, Agent, or Firm*—Schwegman, Lundberg, Wosessner & Kluth, P.A.

(57) **ABSTRACT**

A multiply-accumulate circuit includes a compressor tree to generate a product with a binary exponent and a mantissa in carry-save format. The product is converted into a number having a three bit exponent and a fifty-seven bit mantissa in carry-save format for accumulation. An adder circuit accu-

FLOATING POINT OVERFLOW AND SIGN DETECTION

FIELD

The present invention relates generally to floating point operations, and more specifically to floating point multiply accumulators.

BACKGROUND

Fast floating point mathematical operations have become an important feature in modern electronics. Floating point units are useful in applications such as three-dimensional graphics computations and digital signal processing (DSP). Examples of three-dimensional graphics computation include geometry transformations and perspective transformations. These transformations are performed when the motion of objects is determined by calculating physical equations in response to interactive events instead of replaying prerecorded data.

Many DSP operations, such as finite impulse response (FIR) filters, compute $\Sigma(a_i b_i)$, where i=0 to n−1, and a_i and b_i are both single precision floating point numbers. This type of computation typically employs floating point multiply accumulate (FMAC) units which perform many multiplication operations and add the resulting products to give the final result. In these types of applications, fast FMAC units typically execute multiplies and additions in parallel without pipeline bubbles. One example FMAC unit is described in: Nobuhiro et al., "2.44-GFLOPS 300-MHz Floating-Point Vector Processing Unit for High-Performance 3-D Graphics Computing," IEEE Journal of Solid State Circuits, Vol. 35, No. 7, July 2000.

The Institute of Electrical and Electronic Engineers (IEEE) has published an industry standard for floating point operations in the ANSI/IEEE Std 754–1985, *IEEE Standard for Binary Floating-Point Arithmetic*, IEEE, New York, 1985, hereinafter referred to as the "IEEE standard." A typical implementation for a floating point FMAC compliant with the IEEE standard is shown in FIG. 1. FMAC **100** implements a single precision floating point multiply and accumulate instruction "D=(A×B)+C," as an indivisible operation. As can be seen from FIG. 1, fast floating point multipliers and fast floating point adders are both important ingredients to make a fast FMAC.

Multiplicands A and B are received by multiplier **110**, and the product is normalized in post-normalization block **120**. Multiplicands A and B are typically in an IEEE standard floating point format, and post-normalization block **120** typically operates on (normalizes) the output of multiplier **110** to make the product conform to the same format. For example, when multiplicands A and B are IEEE standard single precision floating point numbers, post-normalization block **120** operates on the output from multiplier **110** so that adder **130** receives the product as an IEEE standard single precision floating point number.

Adder **130** adds the normalized product from post-normalization block **120** with the output from multiplexer **140**. Multiplexer **140** can choose between the number C and the previous sum on node **152**. When the previous sum is used, FMAC **100** is performing a multiply-accumulate function. The output of adder **130** is normalized in post-normalization block **150** so that the sum on node **152** is in the standard format discussed above.

Adder **130** and post-normalization block **150** can be "non-pipelined," which means that an accumulation can be

performed in a single clock cycle. When non-pipelined, adder **130** and post-normalization block typically include sufficient logic to limit the frequency at which FMAC **100** can operate, in part because floating point adders typically include circuits for alignment, mantissa addition, rounding, and other complex operations. To increase the frequency of operation, adder **130** and post-normalization block **150** can be "pipelined," which means registers can be included in the data path to store intermediate results. One disadvantage of pipelining is the introduction of pipeline stalls or bubbles, which decrease the effective data rate through FMAC **100**.

For the reasons stated above, and for other reasons stated below which will become apparent to those skilled in the art upon reading and understanding the present specification, there is a need in the art for fast floating point multiply and accumulate circuits.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a prior art floating point multiply-accumulate circuit;

FIG. 2 shows an integrated circuit with a floating point multiply-accumulate circuit;

FIG. 3 shows the exponent and mantissa paths of a floating point multiply-accumulate circuit;

FIG. 4 shows a mantissa multiplier circuit;

FIG. 5 shows a floating point conversion unit;

FIG. 6 shows a carry-save negation circuit;

FIG. 7 shows a base 32 floating point number representation;

FIG. 8 shows an exponent path of a floating point adder;

FIG. 9 shows a mantissa path of a floating point adder;

FIG. 10 shows an overflow detection circuit;

FIG. 11 shows a post-normalization circuit; and

FIG. 12 shows a sign detection circuit.

DESCRIPTION OF EMBODIMENTS

In the following detailed description of the embodiments, reference is made to the accompanying drawings which show, by way of illustration, specific embodiments in which the invention may be practiced. In the drawings, like numerals describe substantially similar components throughout the several views. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention. Other embodiments may be utilized and structural, logical, and electrical changes may be made without departing from the scope of the present invention. Moreover, it is to be understood that the various embodiments of the invention, although different, are not necessarily mutually exclusive. For example, a particular feature, structure, or characteristic described in one embodiment may be included within other embodiments. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims, along with the full scope of equivalents to which such claims are entitled.

Floating Point Multiply Accumulator

FIG. 2 shows an integrated circuit with a floating point multiply-accumulate circuit. Integrated circuit **200** includes floating point multiplier **210**, floating point conversion unit **220**, floating point adder **230**, and post-normalization circuit **250**. Each of the elements shown in FIG. 2 is explained in further detail with reference to figures that follow. In this section, a brief overview of the FIG. 2 elements and their

-continued

SI	CI	MC	Sign
1	0	0	0
1	0	1	1
1	1	X	-

Magnitude comparator **325** operates in parallel with adder mantissa path **324**, so MC is available for sign detection circuit **1104** at substantially the same time as Msum. In this manner, the operation of sign detection circuit **1104** does not appreciably increase the delay within the feedback loop.

CONCLUSION

The method and apparatus of the present invention provide a fast multiply-accumulate operation that can be made compliant with any floating point format. Furthermore, the method and apparatus of the present invention can provide precision comparable to the precision available using prior art double precision arithmetic units, in part because the mantissa fields are expanded. In some embodiments, IEEE standard single precision operands are multiplied and the products are summed. The multiplier includes a compressor tree to generate a product with a binary exponent and a mantissa in carry-save format. The product is converted into a number having a three bit exponent and a fifty-seven bit mantissa in carry-save format for accumulation. An adder circuit accumulates the converted products in carry-save format. Because the products being summed are in carry-save format, post-normalization is avoided within the adder feedback loop. In addition, because the adder operates on floating point number representations having exponents with a least significant bit weight of thirty-two, exponent comparisons within the adder exponent path are fast, and variable shifters can be avoided in the adder mantissa path. When the adder is not pipelined, a fast single cycle accumulation is realized with the method and apparatus of the present invention.

It is to be understood that the above description is intended to be illustrative, and not restrictive. Many other embodiments will be apparent to those of skill in the art upon reading and understanding the above description. The scope of the invention should, therefore, be determined with reference to the appended claims, along with the full scope of equivalents to which such claims are entitled.

What is claimed is:

1. A floating point multiply-accumulate circuit comprising:
   - an exponent path including:
     - an exponent summer to sum two input exponents having a first weight to produce a product exponent;
     - an exponent conversion unit coupled to the output of the exponent summer, to convert the product exponent to a second weight; and
     - an exponent accumulation stage to accumulate the converted product exponent and to choose a larger exponent from the converted product exponent and an accumulated exponent; and
   - a mantissa path including:
     - a mantissa multiplier to multiply two input mantissas and produce a product mantissa in carry-save format that includes a sum field and a carry field;
     - a mantissa shifter to shift the sum field and the carry field responsive to the exponent conversion unit in the exponent path; and

a mantissa accumulator to accumulate shifted pro
mantissas in carry-save format, the mantissa a
mulator including an overflow detection cir
responsive to two most significant bits of a sum f
output from the mantissa accumulator.

2. The floating point multiply-accumulate circuit of cl
1 wherein the overflow detection circuit comprises
exclusive-or gate.

3. The floating point multiply-accumulate circuit of cl
1 wherein the exponent conversion unit is configured to
the least significant five bits of the product exponent.

4. The floating point multiply-accumulate circuit of cl
1 wherein the mantissa shifter is configured to shift the
field and carry field of the product mantissa by a numbe
bit positions equal to a value of the least significant five
of the product exponent.

5. The floating point multiply-accumulate circuit of cl
1, wherein the mantissa accumulator comprises four-to-
compressors.

6. The floating point multiply-accumulate circuit of cl
1 further comprising a post-normalization stage to prod
a normalized floating point resultant.

7. The floating point multiply-accumulate circuit of cl
6 wherein the post-normalization stage includes a sign
detection circuit.

8. The floating point multiply-accumulate
7 further comprising a magnitude compara
with the mantissa accumulator, wherein the
circuit is responsive to the magnitude comp

9. The floating point multiply-accumulate
6 wherein the post-normalization stage is co
turned off until accumulation is complete.

10. The floating point multiply-accum
claim 1 wherein the exponent conversion un
to convert the product exponent to have a leas
weight equal to thirty-two.

11. A method performed within a progra
computer, comprising:
multiplying two floating point mantissas
two floating point exponents to form a
converting the product to have a different l
bit weight exponent field;
accumulating the converted product in car
detecting overflow as a function of two m
bits of a sum field of an accumulated p
post-normalizing the accumulated product

12. The method of claim 11 wherein acc
product comprises adding a first plurality of p
last product, the method further comprisin
post-normalization until the last product is a

13. The method of claim 11 wherein co
prises:
shifting a mantissa of the product by an a
the value of the least significant five bi
nent of the product; and
zeroing the least significant five bits of an e
product.

14. The method of claim 11 wherein accu
prises:
comparing an exponent of a first converted
exponent of a second converted product
conditionally shifting right by a fixed amou
of the converted product having a small
selecting the larger exponent as a resultant
producing a resultant mantissa from a mant
converted product and a mantissa of th
verted product.

What is claimed is:

1. A floating point multiply-accumulate circuit comprising:
   - an exponent path including:
     - an exponent summer to sum two input exponents having a first weight to produce a product exponent;
     - an exponent conversion unit coupled to the output of the exponent summer, to convert the product exponent to a second weight; and
     - an exponent accumulation stage to accumulate the converted product exponent and to choose a larger exponent from the converted product exponent and an accumulated exponent; and
   - a mantissa path including:
     - a mantissa multiplier to multiply two input mantissas and produce a product mantissa in carry-save format that includes a sum field and a carry field;
     - a mantissa shifter to shift the sum field and the carry field responsive to the exponent conversion unit in the exponent path; and

a mantissa accumulator to accumulate shifted product mantissas in carry-save format, the mantissa accumulator including an overflow detection circuit responsive to two most significant bits of a sum field output from the mantissa accumulator.

2. The floating point multiply-accumulate circuit of claim 1 wherein the overflow detection circuit comprises an exclusive-or gate.

3. The floating point multiply-accumulate circuit of claim 1 wherein the exponent conversion unit is configured to zero the least significant five bits of the product exponent.

4. The floating point multiply-accumulate circuit of claim 1 wherein the mantissa shifter is configured to shift the sum field and carry field of the product mantissa by a number of bit positions equal to a value of the least significant five bits of the product exponent.

5. The floating point multiply-accumulate circuit of claim 1, wherein the mantissa accumulator comprises four-to-two compressors.

6. The floating point multiply-accumulate circuit of claim 1 further comprising a post-normalization stage to produce a normalized floating point resultant.

7. The floating point multiply-accumulate circuit of claim 6 wherein the post-normalization stage includes a sign detection circuit.

8. The floating point multiply-accumulate circuit of claim 7 further comprising a magnitude comparator in parallel with the mantissa accumulator, wherein the sign detection circuit is responsive to the magnitude comparator.

9. The floating point multiply-accumulate circuit of claim 6 wherein the post-normalization stage is configured to be turned off until accumulation is complete.

10. The floating point multiply-accumulate circuit of claim 1 wherein the exponent conversion unit is configured to convert the product exponent to have a least significant bit weight equal to thirty-two.

Liability / Negligence

Product Liability
(Product Design)

⇩

Negligence vs. Strict liability

Negligence

Did not follow reasonable standards

Strict Liability

Dangerous or defective at the time of manufacture