COMPARISON SETS:
A USEFUL PARTITIONING OF THE SPACE OF FLOATING POINT OPERAND PAIRS

Jan G. Kent
Norwegian Computing Center Forskningsveien 1B
Oslo 3 Norway

Abstract: In this paper the definition of comparison sets and a discussion of their usefulness are presented based on the research work reported in (14). In addition some new results concerning the distribution of floating point (FLP) operand pairs over comparison sets are given.

INTRODUCTION

The comparison sets are defined according to properties of the ideal FLP operation algorithms.

However, comparison sets can be used to predict important aspects of normalization and truncation in most FLP operations.

Consequently these sets simplify the comparison of FLP operations greatly, hence the name comparison set. In (14) they are used to compare the rounding methods used in the FLP instruction algorithms on the PDP10, CDC6000-Cyber 70, CDC3000, Univac 1100, SM3, SM4 and IBM 360-370 computer series.

The actual distribution of FLP operand pairs over these comparison sets for several large calculations are reported.

BASIC CONCEPTS

Precise definitions of FLP numbers, sets and mappings are given in (14 & 16).

Floating point numbers

A FLP number is defined as a triplet containing sign, exponent and fractional part. The sign and exponent are integers, while the fractional part is a particular type of radix polynomial called limited digit radix (LDR) polynomial defined in (14 & 16).

Since an LDR polynomial always has a positive value with positive coefficients the FLP fraction represented is in signed magnitude.

The base in a FLP number or radix polynomial is called b, where b is an integer greater or equal to 2.

The length of the fraction in the FLP numbers is p, where p is an integer greater or equal to 3.

FLP zero is written Ø.

Sets of FLP numbers

The set of all FLP numbers defined as above with base b is $\underline{S}_b$.

The set of all normalized FLP numbers with fraction length p is $\underline{S}_b^p$.

Mappings on FLP numbers and fractions

Given a FLP number F:

σF is the sign of F.

εF is the exponent of F.

ψF is the fraction of F.

$\omega\psi F$ is the value of the fraction of F.

$\zeta_k F$ scales F, by increasing the exponent with k and "multiplying" the fraction with b^{-k}

λ maps any radix polynomial with positive value into a LDR polynomial

Operations on FLP numbers

FLP numbers are added and subtracted according to the following definitions.

Assume $G \in \underline{S}_b$ and $H \in \underline{S}_b$.

In the rest of this paper e will be used to indicate the exponent difference $\varepsilon G - \varepsilon H$: $e = \varepsilon G - \varepsilon H$.

Two predicates are required.

$$G \geq H = \varepsilon G > \varepsilon H \vee (\varepsilon G = \varepsilon H \ \& \ \omega\psi G \geq \omega\psi H)$$

$$G < H = \varepsilon G < \varepsilon H \vee (\varepsilon G = \varepsilon H \ \& \ \omega\psi G < \omega\psi H)$$

The addition of two FLP numbers G and H is defined as follows.

$$G \oplus H = \begin{cases} (\sigma G, \varepsilon G, \lambda(\psi G + \psi H * b^{-e})) \ ; & G \geq H, \sigma G = \sigma H \\ (\sigma G, \varepsilon G, \lambda(\psi G - \psi H * b^{-e})) \ ; & G \geq H, \sigma G \neq \sigma H \\ (\sigma H, \varepsilon H, \lambda(\psi H + \psi G * b^{e})) \ ; & G < H, \sigma G = \sigma H \\ (\sigma H, \varepsilon H, \lambda(\psi H - \psi G * b^{e})) \ ; & G < H \ \sigma G \neq \sigma H \end{cases}$$

From this definition can be seen that FLP add is completely symmetric in G and H ; $G \oplus H = H \oplus G$.

Note

The preshift of H or G indicated above by $\psi H * b^{-e}$ and $\psi G * b^{e}$ required to equalize exponents, can be done before the add operation. This means that equal exponents can be assumed in the add operation:

Assuming $G \geq H$ this can for instance be done as follows: $G \oplus \zeta_e H$.

The add operation then simplifies to:

$$G \oplus H = \begin{cases} (\sigma G, \varepsilon G, \lambda(\psi G + \psi H)); & \sigma G = \sigma H \\ (\sigma G, \varepsilon G, \lambda(\psi G - \psi H)); & \sigma G \neq \sigma H \end{cases}$$

The subtraction of two FLP numbers G and H is defined in terms of addition:

$$G \ominus H = G \oplus (-1 \cdot \sigma H, \varepsilon H, \psi H)$$

The remarks made above about preshifting, applies to subtraction also.

The multiplication and division of two FLP numbers G and H are straight forward.

$$G \circledast H = (\sigma G \cdot \sigma H, \varepsilon G + \varepsilon H, \lambda(\psi G * \psi H))$$

And assuming $H \neq Ø$

$$G \oslash H = (\sigma G \cdot \sigma H, \varepsilon G - \varepsilon H, \lambda(\psi G / \psi H))$$

The definitions of $\oplus$, $\ominus$, $\circledast$ and $\oslash$ show that these operations map as follows:

$$G \bigcirc H : \underline{S}_b \times \underline{S}_b \rightarrow \underline{S}_b.$$

CAFRY-CANCEL

Normalization, truncation and rounding in a floating point operation depend on the size of the fraction in the intermediate result before normalization. This fraction's value may be greater or equal to 1. It will then be said that the intermediate result has carry. The intermediate result fraction can have a value less than 1 or greater or equal to 1/b. The intermediate result has cancel when its fraction has a value less than 1/b.

Carry predicate:

$F \in \underline{S}_b \ \& \ F \neq Ø; \ \omega\psi F \geq 1$

Cancel predicate:

$F \in S_b \ \& \ F \neq Ø; \ \omega\psi F < 1/b$

COMPARISON SETS

There are three categories of FLP instruction algorithms; add including subtraction, multiply and divide algorithms.

For each of the three categories of FLP instruction algorithms there is a collection of comparison sets partitioning the space of FLP operand pairs $\underline{S}_b^p \times \underline{S}_b^p$. Within each collection the sets are mutually exclusive.

However most algorithms within a category can be analysed with the aid of the same collection of comparison sets.

Comparison sets for addition algorithms

The carry/cancel predicates applied to the intermediate result $G \oplus \zeta_e H$ in the ideal FLP add algorithm are used to partition $\underline{S}_b^p \times \underline{S}_b^p$.

These predicates will be abbreviated as follows: $\omega\psi(G \oplus \zeta_e H)$ is written ω^+. Then the predicates used are:

$\omega^+ \geq 1$ means $G \oplus \zeta_e H$ has carry

$\omega^+ < 1$ means $G \oplus \zeta_e H$ has no carry

$\omega^+ \geq 1/b$ means $G \oplus \zeta_e H$ has no cancel

$\omega^+ < 1/b$ means $G \oplus \zeta_e H$ has cancel .

The union of the comparison sets defined below, covers only those operand pairs where $G \geq H$ since the FLP add instruction algorithms studied are symmetric in G and H.

Set $\underline{S} = \underline{S}_b^p \times \underline{S}_b^p$

Then define

$\underline{S}^> = \{(G,H) \; ; \; (G,H) \in \underline{S} \;\&\; G \geq H\}$

In all the set definitions given below it is assumed that:

$(G,H) \in \underline{S}^>$

The comparison sets for addition are:

$$\underline{S}_o = \{(G,H) \; ; \; G = \emptyset\}$$
$$\underline{S}_1 = \{(G,H) \; ; \; G \neq \emptyset \;\&\; \sigma G = \sigma H \;\&\; \varepsilon G = \varepsilon H\}$$
$$\underline{S}_{2a} = \{(G,H) \; ; \; G \neq \emptyset \;\&\; \sigma G = \sigma H \;\&\; 1 \leq \varepsilon G - \varepsilon H \leq p \;\&\; \omega^+ < 1\}$$
$$\underline{S}_{2b} = \{(G,H) \; ; \; G \neq \emptyset \;\&\; \sigma G = \sigma H \;\&\; \varepsilon G - \varepsilon H \geq p+1\}$$
$$\underline{S}_3 = \{(G,H) \; ; \; G \neq \emptyset \;\&\; \sigma G = \sigma H \;\&\; 1 \leq \varepsilon G - \varepsilon H \;\&\; \omega^+ \geq 1\}$$
$$\underline{S}_4 = \{(G,H) \; ; \; G \neq \emptyset \;\&\; \sigma G \neq \sigma H \;\&\; \varepsilon G = \varepsilon H\}$$
$$\underline{S}_{5a} = \{(G,H) \; ; \; G \neq \emptyset \;\&\; \sigma G \neq \sigma H \;\&\; 1 \leq \varepsilon G - \varepsilon H \leq p \;\&\; \omega^+ \geq 1/b\}$$
$$\underline{S}_{5b} = \{(G,H) \; ; \; G \neq \emptyset \;\&\; \sigma G \neq \sigma H \;\&\; \varepsilon G - \varepsilon H \geq p+1 \;\&\; \omega^+ \geq 1/b\}$$
$$\underline{S}_6 = \{(G,H) \; ; \; G \neq \emptyset \;\&\; \sigma G \neq \sigma H \;\&\; \varepsilon G - \varepsilon H = 1 \;\&\; \omega^+ < 1/b\}$$
$$\underline{S}_7 = \{(G,H) \; ; \; G \neq \emptyset \;\&\; \sigma G \neq \sigma H \;\&\; 2 \leq \varepsilon G - \varepsilon H \;\&\; \omega^+ < 1/b\}$$

The union of these sets covers $\underline{S}^>$.

Theorem 5.4 in (14) shows that when G and H are such that $\varepsilon G - \varepsilon H \geq p$ they cannot be a member of $\underline{S}_3$. It shows in addition that since $\varepsilon G - \varepsilon H \geq 2$ in $\underline{S}_7$ $\omega^+ > 1/b^2$ indicating a cancel of only one position.

Comparison sets for multiplication

The only predicates required to characterize these subsets are the two predicates testing $G \circledast H$, for cancel or noncancel. These predicates will be abbreviated as follows:

$\omega\psi\ (G \circledast H)$ is written ω^*. Then the predicates used are:

$\omega^* \geq 1/b$ means $G \circledast H$ has no cancel

$\omega^* < 1/b$ means $G \circledast H$ has cancel

The comparison sets for multiplication are

$$\underline{S}_8 = \{(G,H) \; ; \; (G,H) \in \underline{S}_b^p \times \underline{S}_b^p \;\&\; \omega^* \geq 1/b\}$$
$$\underline{S}_9 = \{(G,H) \; ; \; (G,H) \in \underline{S}_b^p \times \underline{S}_b^p \;\&\; \omega^* < 1/b\}$$

Lemma 3.7 in 14 shows that the fraction in $G \circledast H$ will have a leading coefficient index of 1 or 2. This means that $G \circledast H$ will never have carry. $\underline{S}_8 \cup \underline{S}_9$ will therefore cover $\underline{S}_b^p \times \underline{S}_b^p$. Leading coefficient index not lower than 2 also means that only single cancel will occur. In other words, assuming $(G,H) \in \underline{S}_9 \Rightarrow \omega^* \geq 1/b^2$.

Comparison sets for division

The only predicates required to characterize these subsets are the two predicates $\omega\psi G \geq \omega\psi H$ and $\omega\psi G < \omega\psi H$.

The comparison sets for division are

$$\underline{S}_{10} = \{(G,H); \; (G,H) \in \underline{S}_b^p \times \underline{S}_b^p \;\&\; \omega\psi G \geq \omega\psi H\}$$
$$\underline{S}_{11} = \{(G,H); \; (G,H) \in \underline{S}_b^p \times \underline{S}_b^p \;\&\; \omega\psi G < \omega\psi H\}$$

Lemma 3.9 in (14) shows that the predicates $\omega\psi G \geq \omega\psi H$ and $\omega\psi G < \omega\psi H$ picks the same operand pairs as the predicates testing for carry and nocarry in $G \oslash H$. It is clear that $\underline{S}_{10} \cup \underline{S}_{11} = \underline{S}_b^p \times \underline{S}_b^p$ and that only carry and nocarry can occur in $G \oslash H$.

THE USE OF COMPARISON SETS

FLP algorithms that differ from the ideal algorithms will in most cases have an intermediate result before normalization different from $G \oplus \zeta_e H$. For each algorithm investigated in (14) it is shown that carry and cancel either occurs when it occurs in $G \oplus \zeta_e H$ or that the final result is independent of whether carry and cancel is detected in $G \oplus \zeta_e H$ or in the algorithm's intermediate result.

The FLP algorithms investigated in (14) were the FLP instruction algorithms for add/subtract, multiply and divide on the following computer series: CDC6000-Cyber 70, CDC3000, Univac 1100, SM3, SM4 and IBM 360-370.

All of the algorithms investigated were generalized to accept any positive integer base greater or equal to 2, and any fraction length greater or equal to 3.

Any FLP operand pair can be placed uniquely in a comparison set. This makes it possible to predict the most important aspects of the normalization and truncation required when applying any of the FLP operations above to this FLP operand pair. The aspects of normalization and truncation that can be predicted are those that determine the rounding and accuracy of FLP operations. The predictions can be made without taking into account any of the peculiar aspects of each algorithm.

The actual distribution of FLP operand pairs over these comparison sets has been investigated for several large calculations. This is of interest since the major rounding errors in a FLP operation only occur for operands from a few of the comparison sets.

When the operand pairs that actually occur in large calculations seldom are from these critical comparison sets the rounding errors incurred cannot affect the calculations noticeably. When however the operand pairs largely come from the critical comparison sets the rounding errors incurred may cause significant errors in the results of the calculation.

Finally a FLP operation that behaves badly for operand pairs from a critical comparison set can be found acceptable if operand pairs from this critical

comparison set are very infrequent in the types of calculation where the FLP operation will be used.
The investigations are based on complete instruction and operand traces of the following 5 programs:

A) Bairstow's method for polynomial roots; No 30 (19).

B) Crout's method for linear equations; No 43 (19).

C) Håvie's method for definite integrals; No 257 (19).

D) Aitken's method for polynomial interpolation.

E) Secant method for simultaneous linear equations.

The traces of these and many other programs were made by Dr. Å. Lunde for his thesis work (17).
Most of the programs were run several times with different input in the trace. In addition traces of several versions in different programming languages were made for programs A, B and C. There were insignificant differences between the distributions obtained with different versions of the same program.

The trace of the 5 programs give 750 000 instructions of which 46 000 were FLP addition and subtraction, 43 000 were FLP multiplication and 11 000 FLP division.

The results of this investigation was compared with D. Sweeney's analysis of FLP additions (18) and found to correlate well with his results.

The new comparison set

$$S_{oa} = \{(G,H);\ H = \emptyset\}$$

was introduced to eliminate from S_{2b} and S_{5b} the operand pairs where the large exponentdifference was caused by one operand being zero.
Thus S_o contains FLP operand pairs where both are zero and S_{oa} FLP operand pairs where one of the operands are zero.

The tables below give for each problem the percentage of FLP operand pairs in each comparison set.

Addition, subtraction

Problem	A	B	C	D	E	Total
Percentage of FLP add/ subtract instructions	10.6	5.2	15.1	5.6	5.3	6.1
S_o	0.4	1.0	0	1.6	2.2	1.4
S_{oa}	23.8	27.7	1.7	12.1	0	10.5
S_1	4.6	5.2	50.9	12.2	2.3	14.4
S_{2a}	22.6	47.9	38.9	26.7	14.7	27.6
S_{2b}	0.5	0.9	0	0.6	0	0.5
S_3	6.8	5.5	3.3	9.2	3.7	7.2
S_4	10.1	3.0	0.9	8.6	57.6	15.2
S_{5a}	15.6	4.7	2.0	15.4	6.0	11.5
S_{5b}	0.1	0	0	0.3	0	0.2
S_6	10.8	2.7	0.6	9.3	12.0	8.3
S_7	4.7	1.4	1.7	4.0	1.5	3.2
Σ	100	100	100	100	100	100

Multiplication

Problem	A	B	C	D	E	Total
Percentage of FLP multiply instructions	10.4	2.2	3.2	6.9	3.0	5.7
S_8	38.6	49.3	38.9	51.1	39.7	49.1
S_9	61.4	50.7	61.1	48.9	60.3	50.9
Σ	100	100	100	100	100	100

Division

Problem	A	B	C	D	E	Total
Percentage of FLP divide instructions	2.0	4.1	5.9	1.4	1.2	1.5
S_{10}	55.2	69.5	68.0	54.4	58.2	57.7
S_{11}	44.8	30.5	32.0	45.6	41.8	42.3
Σ	100	100	100	100	100	100

CONCLUSION

Important aspects of normalization and truncation in FLP instructions in most of todays computers can be predicted with the aid of comparison sets. This warrants the belief, that comparison sets can be used to investigate and compare many present and future FLP operations. This will make comparison sets an increasingly important tool for the study of FLP operations.

Information on the statistical distribution of FLP operand pairs over comparison sets is of intrinsic value especially when tied to normalization and truncation properties in FLP operations.

The distribution of FLP operand pairs over comparison sets were found to be very problem dependent. It does not seem possible to find a general distribution that will be useful when studying spesific calculations.

It is hoped that comparison sets will be used by those who design, investigate or use FLP arithmetic, and that it brings the theory of FLP arithmetic a step forward.

REFERENCES

1. Yohe, J.M.: Foundations of Floating Point Computer Arithmetic; The University of Wisconsin, MRC Technical Summary. Report # 1302, January 1973

2. Knuth, D.: The Art of Computer Programming; Vol.2. Section 4.2.1 Algorithm A

3. Knuth, D.: The Art of Computer Programming; Vol.1 and 2; Addison Wesley, 1968 and 1969 respectively

4. CDC3600 Computer System. Reference Manual; Pub.60021330OG

5. Yohe, J.M.: Machine Language Programming for the CDC3600; The University of Wisconsin, MRC Technical Summary Report # 721, August 1967

6. CDC3300 Computer System Reference Manual; Pub.No.60157000-01

7. Univac 1107 General Manual, Central Computer; UP-2463 Rev.2

8. Univac 1108 Processor Storage Manual; UP-4053 Rev.1

9. IBM System/370, Principles of Operation; GA22-7000-2

10. PDP10 Reference Handbook, PDP10 Handbook Series; 1969

11. CDC6000 Series Computer Systems, Reference Manual; Pub No.60100000-M

12. Thornton, J.E. Design of a Computer, The Control Data 6600; Scott, Foresman and Company, Illinois, 1970

13. Listings of microprograms for the IBM System 360/25

14. Kent, J.G.: Theoretical Definition, Analysis and Comparison of Floating Point Instructions, Norwegian Computing Center, Publication no. 425, September 1973.

15. Kent, J.G.: Procedures for the Description and Simulation of Floating Point Instructions, Norwegian Computing Center, Publication no. 426, September 1973

16. Kent, J.G.: Highlights of a Theoretical Study of Floating Point Instructions, Digest of Papers, IEEE CompCon Fall 74 Conference, September 1974

17. Lunde, Å.: Evaluation of Instruction Set Processor Architecture by Program Tracing, Ph.d.thesis, Dept of Computer Science, Carnegie-Mellon University, Pittsburgh, July 1974

18. Sweeney, D.W.: An Analysis of Floating Point Addition; IBM Systems Journal, Vol 4, No. 1, 1966

19. Collected Algorithms, Communications of the ACM