Interpretive Floating Point Program for study of "noisy mode".

Define Set of pseudo instructions

O = opposite sign N = normalized
S = same sign U = unnormalized

add	AON	AOU	ASN	ASU
mpy	MON	MOU	MSN	MSO
div	DON	DOU	DSN	DSU
load	LON	LOU	LSN	LSU
store	SON	SOU	SSN	SSU

Add: $(2^a A) + (2^m M) = (2^r R)$

0. Modify sign of M by ∘ modifier
1. Compare a & m, put smaller mantissa in acc, larger in SR, larger exponent → r
2. if $|a-m| > 35$ (29?) no add required
3. shift right $|a-m|$ places
4. add in fixed point double length (watch sign of MQ)
5. Test if R ≡ 0, if so set acc. to zero
   (case I: Insert noise)
6. If not ≡ 0 Test for overflow
   if N & if ov: shift R right 1 & increase r by 1
   if U & if ov: branch to overflow error routine.
7. If no ov & if N, Test for leading zeros & shift R left by no. of zeros, reducing exp. by no. of zeros.
   (case II. Insert noise before shift.)
   If no ov & if U, leave as is
8. answer in acc.

Load:

$2^m M \rightarrow 2^a A$

1. Modify sign of M by s
2. if M=0, load as is
3. If U load as is
4. If N, Test for leading zeros etc same as Add #7 & 8 (incl case II noise)

STORE: $2^a A \rightarrow 2^m M$

1. modify sign of A by s
2. if A=0 store zero
3. If U store as is
4. If N, Test for leading zeros etc as in Add #7 & Store, (incl case II noise)

Mpy: $(2^a A) \cdot (2^m M)$

1. Modify sign of M by s
2. Test if A or M ≡ 0, set R to 0
3. add $a+m = r$
4. Mpy A×M in fixed point (double prod.)
   (case I: doesn't noise)
5. If N, Test for leading zeros & shiftRleft as in Add #7 (incl case II noise)
   If U
6. answer Truncated to single length in acc

Div: $(2^a A)/(2^m M)$

1. Modify sign of M by s
2. If M=0 divide check error
3. If A=0 set R to zero
4. $a-m = r$

Div (cont)

5. If $A-M \geq 0$ shift A right until $A < M$ + increase r one for each shift
6. Div A/M in fixed point (case I noise)
7. If N test for leading zeros as in Add #7 incl case II noise.
8. Answ in accumulator.