SIGMA COMPUTER #7

H. Kolsky

SUBJECT: The Floating Point Instruction Set.

BY: L. G. Allen

DATE: March 25, 1958

The floating point instruction set has been rewritten to incorporate the latest changes. New instructions have been added and changes made in others.

The twenty-nine (29) attached sheets describe the floating point instruction as understood by the writer at this time.

Normalize, as used in the instruction set is defined as follows:

1. If UM = 0 and overflow results, shift the mantissa right 1 and add 1 to the exponent.
2. If UM = 0 and mantissa does not overflow, shift it left until 1 appears in the high order bit and subtract the amount of shift from the exponent.
3. If UM = 1 do not normalize. If overflow occurs, it is lost.

L. G. Allen

L. G. Allen

LGA:bjn

March 18, 1958

59-63 set to 0 on all right

Add 00000

1. The content of the specified operand Address are combined with the contents of A B 0-59 and A B (s) as follows:

   a. Operand bits 0-10 are subtracted from A B bits 0-10 under control of operand bit 11 and A B bit 11.

   b. The larger exponent becomes the exponent of the unnormalized result.

   c. Shift the mantissa with the smaller exponent right by the amount of exponent difference.

   d. The mantissas are added under control of operand bit 60, as modified by SM, and A.B (S), using 49 bits of the adder.

2. Operand bits 61-63 replace I bits 27-29.

3.

   a. If UM = 1 the result replaces A B bits 0-59 and A B (s)
   If UM = 1 and a mantissa overflow occurs, the sum replaces A B bits 0-59 and A B (s) and the lost carry indicator is set. Carry is lost.

   b. If UM = 0 and the result is "0" the Lost Significance indicator is set ~~and 48 is subtracted from the exponent~~. The result replaces A B bits 0-59 and A B (s).
   If UM = 0 and the result is not "0". The result is normalized and replaces A B bits 0-59 and A B (s).

4. Indicators which may be set are 30, 34-41 and 57-60.

   Note: After the instruction is executed, the operand is not available in any addressable machine register.

30 – lost carry
34 – last sig.
35 – prep. shift >48
36 – exp overflow
37 exp 1
38 B
39 C
40 D
41 exp underflow

57 result <0
58 " =0
59 " >0
60 " –

March 18, 1958

Augment 00010

1. The contents of the specified operand address are combined with AB bits 0-59 and AB(S) as follows:

   a. Operand bits 0-10 are subtracted from AB bits 0-10 under control of operand bit 11 and AB bit 11.

   b. The larger exponent becomes the exponent of the sum.

   c. Shift the mantissa with the smaller exponent right by the amount of the exponent difference.

   d. The mantissas are added under control of operand bit 60 as modified by SM and AB (S), using 49 bits of the adder.

2. Operand bits 61-63 replace I bits 27-29.

sign of acc. is never changed

3.

   a. If the sum of the mantissas is 0, the lost significance indicator is set, ~~and 48 is subtracted from the exponent~~.

   b. If the sign of the sum differs from AB (S), ~~the result~~ AB 12-59 is set to 0 ~~and 48 is subtracted from the exponent.~~

   c. If UM=1 and result ≠ 0, the normalized result replaces AB bit 0 to 59 and AB (S). If UM=1 and carry occurs, the Lost Carry indicator is set and result replaces AB bits 0-59 and AB (S). Carry is lost.

   d. If UM=0 and result ≠0, the normalized result replaces AB bits 0-59 and AB (S).

4. Indicators which may be set are 30 34-41 and 57-60.

In both cases where the result is 0, the result is entered in AB bits, 0-59 and AB (S).
The Lost Significance Indicator is set only when the algebraic sum is 0, and not when the sign of the sum differs from AB (S).

keep AB exponent unchanged ?

March 18, 1958

Add to Memory 00100

1. The contents of the effective operand address is combined with the contents of AB and AB (S) as follows:

   a. Operand bits 0-10 are subtracted from AB bits 0-10 under control of operand bit 11 and AB bit 11.

   b. The larger exponent becomes the exponent of the unnormalized result.

   c. Shift the mantissa with the smaller exponent right by the amount of the exponent difference.

   d. The mantissas are added under control of operand bit 60 ~~as modified by SM and~~ AB (S), using 49 bits of the Adder.
   as mod by SM

2. Operand bits 61-63 replace I bits 27-29.

3.

   a. If UM=1, the result replaces bits 0-63 specified by the effective operand address.
   If UM=1 and overflow of adder occurs, the result goes to the effective operand address and the lost carry indicator is set. Carry is lost.

   b. If UM=0, the normalized result replaces bits 0-63 of the effective operand address.
   If UM=0 and result is "0", Lost Significance Indicator is set.
   48 is subtracted from the exponent and the result replaces bits 0-63 of the effective operand address.

4. AB is not changed by this instruction. Note that the flag bits which were brought out in step 1 were stored back in step 3, unchanged.

other indicators ?
Same as add,

Add to Magnitude 10000

1. This operation is the same as add, except as follows:

   a. The mantissa add is performed as if the sign of the accumulator were positive.

   b. The sign of the result is the original accumulator sign A B (S), ~~if no recomplementation was performed.~~ result of addition is neg.

   c. The sign of the result is the inverse of the original accumulator sign, ~~if recomplementation was performed.~~ result of addition is neg.

$$\frac{|A|}{A} \quad \{|A| + (SM)\, M\}$$

$+ \quad + = +$

$+ \quad - = -$

$- \quad + = -$

$- \quad - = +$

March 19, 1958

Add to Memory Magnitude 10100

1. This operation is the same as add to memory except as follows:

   a. The mantissa add is performed as if the sign of the ~~accumulator~~ memory were positive.

   b. The sign of the result is the original ~~accumulator~~ memory sign AB (S), if no recomplementation was performed.

   c. The sign of the result is the inverse of the original ~~accumulator~~ memory sign if recomplementation was performed.

~~Can~~ sign changes in add

~~Cannot~~ Mod

A-B(S) Mod by SM

March 19, 1958

Add Double 01000

1. This operation is the same as Add Single except:

   a. AB bits 12 - 107 is a 96 bit mantissa which is added to the contents of the effective operand address.

   b. The sum has a 96 bit mantissa which, along with its exponent, replaces AB bits 0-107.

2. Indicators which may be set are 30, 34 - 41 and 57 - 60.

shift greater than 48
zero means all 96 bits

March 18, 1958

Interchange Augment 10010

1. This command is the same as augment except that:

   a. M
   ~~A B (S)~~ may be modified by SM

   b. The result is set to 0 if the sign of the sum differs from the operand sign.

changed? modifies M

SM modifies M SM modifies A

Aug Int Aug

Aug to Mem Int Aug to Mem

March 19, 1958

Add Double to Magnitude 11000

1. This operation is the same as Add Double except:

   a. The addition is performed as if the sign of the Accumulator, AB(S) were positive.

   b. The sign of the result is the original accumulator sign in AB (S) inverted if a recomplementation were performed. The resultant sign replaces AB (S).

sign changed in addition

March 19, 1958

Add Mantissa 00011

1. Bits 61 - 63 of the effective operand address replace I bits 27 - 29.

2. Bits 12 - 59 of the effective operand address are added to AB bits 12 - 107 under control of AB (S) and bit 60 of the effective operand address, as modified by SM. AB bits 0-11 is the resultant exponent.

always double

3. If UM = 0, the result is normalized and replaces AB bits 0-107 and AB (S).

4. If UM = 1, the result replaces AB bits 0-107 and AB (S).

   a. If an overflow occurs, the result replaces AB bits 0-107 and AB (S) and the Lost ~~Significance~~ Carry indicator is set.

5. Indicators which may be set by this command are 30, 34 - 41 and 57 - 60.

March 18, 1958

Store 00101

1. The contents of AB bits 0-59 and AB (S) T, U, V, replaces the contents of the effective operand address as follows:

   a. AB bits 0-59, either normalized or unnormalized as specified by UM, replaces effective operand bits 0-59.

   b. AB (S) as modified by SM replaces effective operand bit 60.

   c. AB (S) bits T, U, V, replace ~~bits~~ effective operand 61-63.

2. Indicators which may be set are 37-41 and 57-60.

   Contents of AB are not changed by this instruction.

   Note: AB (0-59) and AB (S) are unchanged by this instruction.

Store Low Order 01101

1. If UM = 1

   a. AB bits 60 - 107 replace bits 12 - 59 of the effective operand address.

   b. AB (S) as modified by SM replace bit 60 of effective operand address.

   c. AB (S) bits T, U, V, replace bits 61-63 of effective operand address.

   d. AB bits 0-10 minus 48 under control of AB bit 11 replace bits 0-11 of effective operand address.

2. If UM = 0

   a. Then (b) and (c) are the same as in 1 above.

   b. Subtract 48 from AB bit 0-10 under control of AB bit 11.

   c. Normalize AB bits 60-107 and the exponent from preceding step.

   d. The normalized result replaces bit 0-59 of the effective operand address.

This command is used to store the low order portion of a double mantissa in AB. AB is not disturbed.

March 19, 1958

Square Root and Store 11110

1. ~~This~~ The square root of AB bits 0-59 replace bits 0-59 of the effective operand address, ~~and AB bits 0-59. AB bits 60-107 are set to zero.~~

2. AB (S) as modified by SM replaces bit 60 of the effective operand address.

3. AB (S) bits T, U, V, replace bits 61-63 of the effective operand address.

4. Indicators which may be set are 17, 33, 37-40, and 57-60.

If the sign AB (S) is minus, the Improper Floating Point Result indicator is set. This command is independent of UM.

~~Note: In determining the square root it is assumed that bits 60-107 are zero.~~

but AB is not disturbed

are not considered Zero in making reg.

Store square root

double shift double.

March 19, 1958

Round and Store 10101

1. A 1 is added to AB at bit 60 with AB bits 12 - 107 considered positive.

2. ~~AB bits 60 - 107 are set to~~ 0. not changed

3. AB (S) as modified by SM replaces bit 60 of effective operand address.

4. If UM = 0, the result is normalized ~~and replaces AB Bit 0-59, AB (S) remains the same.~~ ~~The result also~~ replaces effective operand address bits 0 to 59, and AB (S) bits T, U, V replace bits 61-63 of effective operand address.

5. If UM=1

   a. AB (S) as modified replaces bits 60-63 of effective operand address.

   b. If the result of operation (1) overflows, zeros are entered in bits 12-59 of effective operand address ~~and AB bits 12 to 59~~. Lost Carry Indicator is set. AB bits 0-11 replace bits 0-11 of effective operand.

   c. If no overflow, the result replaces bits 0-59 of the effective operand address, ~~and AB bits 0-59.~~

6. Indicators which may be set are 31, 36 - 41, and 57 - 60.

Note ~~that a Round Store command to location 0 modifies the accumulator only. Also,~~ rounding precedes normalization.

✓ ?

?

do we want to leave acc. as is? yes

go over this.

Store Rounded.

Leave AB unchanged

March 19, 1958

Store with borrow

Borrow and Store 11101

1. ~~AB bits 60 - 107 are set to 0.~~

2. AB (S) as modified by SM replaces AB (S) and bit 60 of the effective operand address.

3. AB (S) bits T U V replace bits 61-63 of the effective operand address.

4. AB bits 0-11 replace bits 0-11 of the effective operand address.

5. If AB bits 12 to 59 are 0, omit step 6 and set zeros in bits 12 to 59 of effective operand address.

6. The absolute value of AB bits 12 to 59 minus 1, subtracted in position 59, replace bits 12 to 59 of the effective operand address.

7. If AB bits 12 to 59 are not 0, set them to 0 and set a 1 in AB bit 12. AB bits 0-10 minus 47 under control of AB bit 11 replaces AB bits 0-11.
If AB bits are 0, do not set 1
Indicators which may be set are 36-41 and 57-60.
These indicators are set for the results in AB. This command is independent of UM. This command is used to provide a coupling between single precision numbers for multiple precision routines.

0110 1010|1011
0110 1010|0000
0110 1001 ← store
0010 1000 ← left in AB

March 18, 1958

RM

Store ~~MR~~ 11111

RM

1. If UM = 1, the contents of ~~MR~~ bits 0-63 replace the contents of the effective operand address.

2. If UM = 0, the contents of MR bits 0-63 are normalized. The result replaces the contents of the effective operand address.

Indicators which may be set are 37 - 41 and 57 - 60.

MR unchanged ?

can eliminate op 7

question of 49 bit quotient

March 19, 1958

Compare 01010

1. Bits 61 - 63 of the effective operand address replace I bits 27 - 29.

2. If UM = 0, the contents of bits 0-59 of operand address and AB bits 0-59 must be normalized before compare is done. Then compare is done as follows:

   a. Operand bit 60 as modified by SM is compared to AB (s).

   b. Operand bit 11 is compared to AB bit 11.

   c. Operand bits 0-10 are compared to AB bits 0-10.

   d. Operand bits 12 - 59 are compared to AB bits 12 - 59.

3. If UM = 1, no normalization is done and the comparison is done the same as above. Note that when UM = 1, there is no assurance that the comparison results are correct except when the mantissa signs are different or both numbers are known to be normalized.

The indicators which may be set by this command are 61 - 63.

AB ~~0~~ unchanged

March 19, 1958

Shift Mantissa

1. AB bits 12 to 107 are shifted by the amount specified by bits 0-10 of the instruction. Bit 11 of the instruction as modified by SM determines whether shift is left or right. If bit 11 is 1, shift right. If bit 11 is 0 shift left.

as modified as modified

If a 1 is shifted left past bit 12 of AB, the Incomplete Field indicator is set. If the amount of shift is greater than 96, AB bits 12 - 107 are set to 0.

what happens after does shift? stop? no. just load

2. Indicators which may be set are 17, 31, and 57 - 60.

incomplete

This command is independent of UM.

March 19, 1958

Add Exponent Immediate 11011

1. Bits 0-10 of the instruction are added to AB bits 0-10 under control of AB bit 11 and instruction bit 11 as modified by SM. Result replaces AB bits 0-11.

2. Indicators which may be set by this instruction are 36-41.

   This command is independent of UM.

March 19, 1958

Add Exponent 01011

1. Bits 61 - 63 of the effective operand address replace I bits 27 - 29.

2. Bits 0-10 of the effective operand address are added to AB bits 0-10 under control of AB bit 11 and operand bit 11 as modified by SM. Results replace AB bits 0-11.

3. Indicators which may be set are 36 - 41.

This command is independent of UM.

March 18, 1958

Divide Double

1. This operation is the same as divide except:

   a. No round is performed.

   b. The unnormalized remainder replaces MR bits 12 - 59. The modified exponent of the dividend minus 48 replaces MR bits 0 - 11,

   c. The sign of the dividend replaces ~~MR sign,~~ and ~~MR~~ RM bit 60.

   d. A 96 bit dividend is used. STUV → RM bits 60 - 63

Indicators which may be set are 30, 32, 33, 36 - 41 and 57 - 60.

Discussion of 48 & 49 bit remainders.

March 18, 1958

Interchange Divide 10111

1. This operation is the same as divide except that:

   a. A B bits 0 - 59 and A B (S) specify the divisor.

   b. Operand bits 0 - 60 become the dividend

   c. A B (S) may be modified by SM before being combined with operand bit 60 to determine the sign of the quotient.

Mem is modified by SM

March 18, 1958

Divide 10000

1. If UM=1, the All Ones Counter and Left Zeros Counter are set to 0.

2. The operand is specified by the effective operand address.

   a. Operand bits 61-63 replace I bits 27-29.

   b. If operand bits 12-59 is 0, set the Zero Divisor indicator (32) and omit the remaining steps.

   c. Normalize the operand. If UM=1 the amount of shift is counted up in the All Ones Counter and Left Zeros Counter.

3. Normalize the accumulator contents (AB bits 12-59).

   a. If UM=1, the amount of the shift is counted down in the Left Zeros Counter. A negative result may occur.

4. The rounded and normalized quotient (Accumulator divided by operand) replaces bits 0-59 of the accumulator. AB (S) is set according to the original AB (S) and operand bit 60 as modified by SM.

5. If UM is 1, and the contents of the left zeros counter are greater than 0, Lost Carry Indicator (30) is set.

6. Indicators which may be set by this command are 30, 32, 36-41, and 57-60.

No quotient is produced if the divisor is zero. A zero quotient is produced if the dividend is 0. If the mantissas are not properly scaled in the unnormalized case (UM=1), the amount of misalignment is available as a positive count in the Left Zeros Counter. If there were more leading zeros in the dividend than in the divisor, the number is available as a negative count in the left zeros counter.

Note 1: that the mantissa of the quotient is rounded to 48 bits.

Note 2: Although AB bits 60-107 are not effected by this operation, they all assumed to be zero for determining the quotient.

Note: no. of leading zeros in quotient is same as count in LZC.

Multiply 00110

1. The contents of the effective operand address operates on A B bits 0-59 and A B (s) as follows:

   a. Operand bits 0-10 are added to A B bits 0-10 under control of operand bit 11 and A B bit 11.

   b. Operand bits 12-59 multiply A B bits 12-59. The sign of the product is determined by operand bit 60 as modified by SM and A B (s).

2. Operand bits 61-63 replace I bits 27-29.

3. If UM = 1, the mantissa result truncated to 48 bits and the exponent result replaces A B bits 0-59 and A B (s).

4. If UM = 0, the normalized mantissa product truncated to 48 bits and the exponent result replaces A B bits 0-59 and A B (s).

5. Indicators which may be set are 36-41 and 57-60.

Note: The mantissa of the product is truncated to 48 bits after normalization occured.

March 19, 1958

Multiply Double 01110

1. This operation is the same as Multiply except the full 96 bit mantissa product is developed, which replaces AB bits 12 - 107, either normalized or unnormalized as determined by UM.

March 18, 1958

Cumulative Multiply 01100

1. The content of the right half of MR register is multiplied by the content of the effective operand address.

   a. MR bits ~~0-10~~ 64-74 are added to operand bits 0-10 under control of MR bits ~~11~~ 75 and operand bit 11.

   b. MR bits ~~12-59~~ 36-123 is multiplied by operand bits 12 - 59 under control of MR sign and operand bit 60 as modified by SM. The product has a 96 bit mantissa.

2. The product developed in step 1 is added to AB in the same manner as a double precision add is performed. All conditions apply that apply to Add Double except SM bit has no affect on the add.

3. The final result replaces AB bits 0-107 and AB (S).

4. Indicators which may be set are 30, 34 - 41 and 57 - 60.

   If an exponent overflow or underflow occurs in step 1, the proper indicators will be set and an invalid result will be obtained.

"multiplier" comes from memory

March 18, 1958

Load Double

1. Same as Load Single except that A B bits 0 - 107 and A B (S) are set to zero in step 1.

Load Single 00001

1. AB bits 0-~~59~~ 63 and AB (S) are set to 0.

2. The contents of specified operand address is loaded as follows:

   a. Operand bits 61-63 replace I bits 27-29.

   b. Operand bits 60 as modified by SM replaces AB (S).

   c. Operand bits 0-59 either normalized or unnormalized as specified by UM replaces AB bits 0-59. if zero no reduction of 48 in exponent

3. Indicators which may be set are 37-41 and 57 - 60.

Load With Flag 10001

1. This command is the same as Load except that operand bits 61 - 63 replace AB (S) bits T, U, V as well as replacing I bits 27 - 29.

Load MR 11100

1. The contents of the specified operand address is operated on as follows:

   a. Operand bits 61-63 replace I bits 61-63.

   b. Operand bit 60 as modified by SM replaces MR sign ~~and MR bit 60~~.

   c. If UM=1, operand bits 0-59 replace MR bits ~~0-59~~ 64-123.

   d. If UM=0, operand bits 0-59 are normalized and then replace MR bits ~~0-59~~ 64-123.

2. Indicators which may be set are 37-41 and 57-60.