IBM®

Customer Engineering

Manual of Instruction

IBM CONFIDENTIAL

Automatic Floating Decimal Arithmetic

April 1, 1960, by International Business Machines Corporation
Printed in U.S.A.

D. AUTOMATIC FLOATING DECIMAL ARITHMETIC FEATURE

CONTENTS

FOREWORD

This book is intended for use in training Customer Engineers in the operation of the Automatic Floating Decimal Arithmetic Feature used in the IBM 7070 System. Illustrations accompany the text. In most cases, the sections, sub-sections, and items contained within this manual, are written so that the preceding information serves as a building block for subsequent study.

Each program controlled operation is discussed and is accompanied by operation flow charts and sequence charts. Major signals,key timings, and logic locations are also provided.

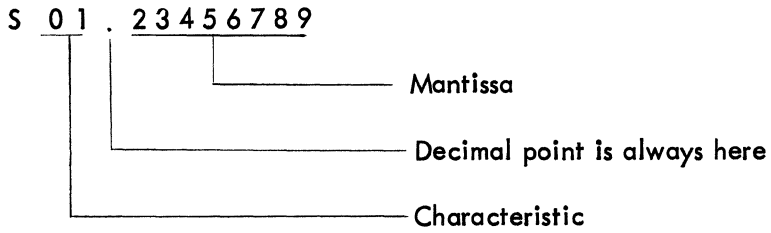
Future engineering changes may cause minor discrepancies in some of the timings given in sequence charts and other charts (signal, timing and location). They should not change the philosophy of the operations unless the changes are of a major nature and constitute revisions to data flow and operation sequence.

ILLUSTRATIONS

1.0.00 FLOATING-DECIMAL OPERATIONS

1.1.00 GENERAL

The floating-decimal feature of the IBM 7070 System allows automatic location of the decimal point during computations. A floating-decimal number is represented by a two-digit characteristic, or exponent, in positions 0-1 of the data word and an eight-digit mantissa in positions 2-9. The characteristic is the power of ten by which the mantissa must be multiplied to give the actual magnitude of the number. The mantissa represents the number of significant digits in the floating-decimal number. The decimal point in all floating-point numbers is to the left of the high-order digit position of the mantissa; i.e., position 2 of the data word. The word format for a floating decimal data word is shown below:



In floating-decimal notation, numbers are classified as normalized or unnormalized. A normalized number has a digit other the zero in the high order digit position of the mantissa. An unnormalized number may have one or more high order zeros in it mantissa. A number is indicated in unnormalized and normalized form as follows:

	S	0	1	2	3	4	5	6	7	8	9
Unnormalized		0	0	3	9	8	7	4	2		
Normalized		3	9	8	7	4	2	0	0		

To represent negative exponents in floating decimal, it is necessary to make use of a modified characteristic. The modified characteristic is a number between 00 and 99. It is obtained by adding 50 to the power of ten required to properly position the decimal point. In this notation, 50 is considered 10^0, 52 is 10^2, 47 is 10^{-3} etc.

The range of numbers thus obtainable is from

$.1 \times 10^{-50}$ to $.99999999 \times 10^{49}$.

The smallest number would appear in floating decimal notation as

Modified Characteristic	Mantissa
00	10000000

and the largest number as

Modified Characteristic	Mantissa
99	99999999

Any floating-decimal number having a mantissa of zero is automatically assigned a characteristic of zero. The floating-decimal representation for zero is, therefore:

Modified Characteristic	Mantissa
00	00000000

The following table shows several examples of mantissas and modified characteristics and their derivation.

Absolute Value	Scientific Notation	Mantissa	Exponent of 10	Modified Characteristic		Floating Decimal No.
3892.41	$.389241 \times 10^{4}$	38924100	4	50 + 4 =	54	5438924100
.0004916	$.4916 \times 10^{-3}$	49160000	-3	50 - 3 =	47	4749160000
-.03494	$-.3494 \times 10^{-1}$	-34940000	-1	50 - 1 =	49	-4934940000
0	0	00000000	0	0 0 =	00	0000000000
Smallest Number	$.1 \times 10^{-50}$	10000000	-50	50 - 50=	00	0010000000
Largest Number	$.99999999 \times 10^{49}$	99999999	49	50 + 49=	99	9999999999

All arithmetic operations may be performed with the floating-point codes. Floating-point instructions appear in the 7070 just as do other instructions. The one exception occurs in field control. Positions 4-5 of the instruction word are not used. There is no field control in floating point.

All floating-point codes may be indexed. The index function is accomplished in exactly the same way as for the fixed point codes.

All of the floating-point codes use all three accumulators. Data left in any accumulator from a previous operation is lost. The operated factor is present in Accumulator 1 either as the result of a floating-reset-add instruction, or as the result of a previous operation.

The mantissas of Accumulators 1 and 2 are considered as one 16-digit number. The number is divided into two eight-digit mantissas each with its own exponent. The exponent of Accumulator 2 is, therefore, 8 less than the exponent of Accumulator 1. Accumulator 2 is assigned an exponent 8 less that Accumulator 1, whether or not Accumulator 2 is zero.

As a result of arithmetic operations it is possible for the machine to attempt to generate a modified characteristic greater than 99, or less than 00. These conditons are called overflow and underflow, respectively. Tests for overflow and underflow are made on the exponent of Accumulator 1 after every arithmetic operation. Only the underflow test is made on the exponent of Accumulator 2.

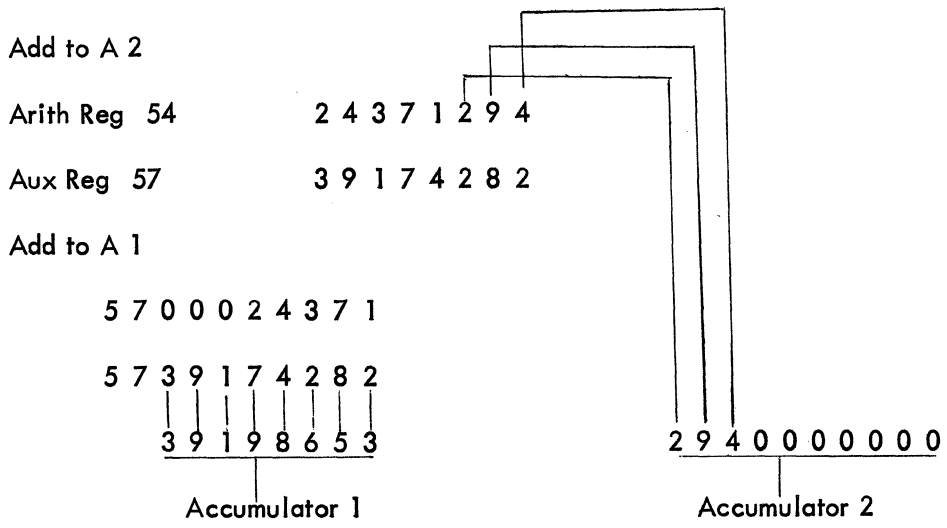
2.0.00 FLOATING ADD-SUBTRACT

2.1.00 INTRODUCTION

To perform a floating add-sub't instruction, it is necessary to align the decimal points of the two factors. This is done normally with pencil and paper by shifting the mantissa of the smaller factor to the right and increasing its exponent by one for each such shift. When the two exponents are equal, the addition or subtraction can then be accomplished.

The process of shifting the smaller mantissa to the right and increasing its exponent accordingly is not actually performed by the 7070. Instead, the smaller factor is placed in the arithmetic register. A number of least significant digits of its mantissa equal to the exponent difference is scanned out, added to zeros and gated into Accumulator 2 starting with position 0. The remainder of the mantissa is then added to the auxiliary register which contains the larger factor and is gated into Accumulator 1. Zeros are inserted in those positions of the arithmetic register left vacant during the scanning out process. This procedure effectively accomplishes the same result as shifting right and adding to the exponent. The process is illustrated below for an exponent difference of 3.

Addition of 2 Floating Point Factors (Exponent Difference of 3)



After the addition of the mantissas, a normalization cycle is initiated. This cycle counts the number of high-order zeros. The accumulators are shifted left together so that the high-order mantissa digit is in position zero of Accumulator 1.

The original exponent is now adjusted and inserted in Accumulator 1. Tests for overflow and underflow are made on this exponent.

The Accumulator 1 exponent is reduced by 8 and inserted in Accumulator 2 positions 0-1. A test for underflow is made on this exponent.

The result in Accumulators 1 and 2 after the normalization and updating cycles just described is:

Floating Ring A seven-position trigger ring called the Floating Ring sequences the various operations performed during a floating-point instruction.

The ring divides the data cycle into 7 subcycles. The ring outputs corresponding to subcycles 1-7 are FP 1-FP 7. Each subcycle controls some special function depending upon the operation code. The number of subcycles actually used during the execution of a given instruction depends upon the operation code and the relative size of the two factors.

The ring outputs for the floating add-sub't operations are described in detail here.

Data Cycle Start The initial access codes signal switches with a Data Control 4 to set the floating ring to position 1. FP 1 then switches with Op-Add-Sub't to give FAS 1. FAS 1 initiates and defines the compare subcycle of the floating-add operation. The floating ring is advanced from FAS 1 by the conditions set by the two factors, i.e., auxiliary register greater than arithmetic register, interchange of factors etc.

Subcycles The seven subcycles are listed and described with reference to data flow as follows:

1. FAS 1 - Compare- Compare Factors-place smaller factor in Arithmetic Reg.

2. FAS 2 - Recomplement Exponent Difference-If Auxiliary Reg is greater than Arithmetic Register.

3. FAS 3 - Add to A 2-Scan out (of Arithmetic Register) number of least significant digits equal to exponent difference - Add digits to A 2.

4. FAS 4 - Add to A 1 - Add Mantissas.

5. FAS 5 - Normalize - Normalize Resulting sum.

6. FAS 6 - Update A 1 Exponent - Adjust original Exponent.

7. FAS 7 - Update A 2 Exponent - Reduce adjusted A 1 Exponent by 8, insert in A 2.

Flow Chart Figure 2.1-1 shows a flow diagram of the floating add-sub't operation. As indicated by the chart, five possible conditions may result from the initial comparison:

1. If there is a difference of 1-7 in the modified characteristics, both add cycles are taken.

2. If the exponent difference is 0, skip cycle 3. Because the smaller factor does not have to be shifted, there will be nothing to add to A 2.

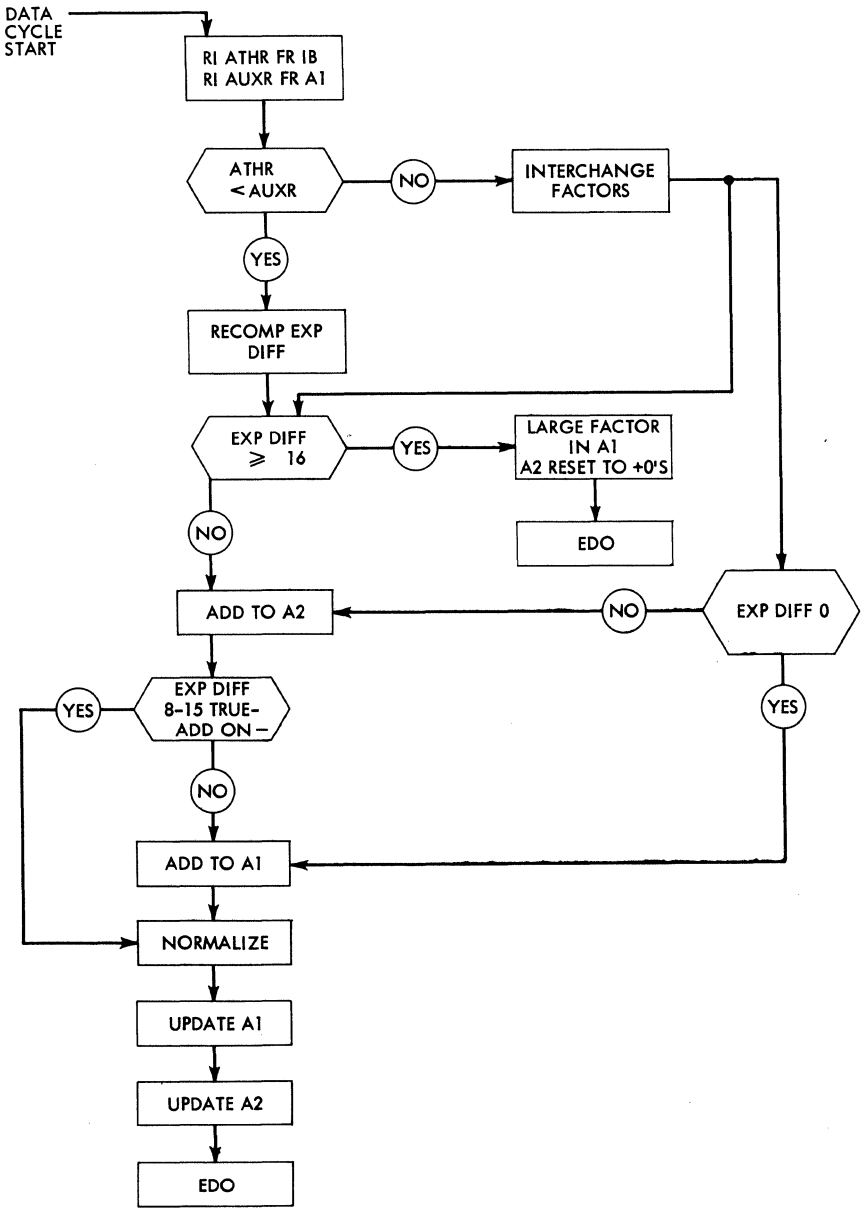


FIGURE 2.1-1 FLOATING ADD-SUB'T FLOW CHART

3. If the exponent difference is 8-15 and the true-add latch is on, skip cycle 4. Cycle 4 is skipped because the entire mantissa has been scanned out. Therefore, there are no digits left to be added to A 1.

4. If the exponent difference is 8-15 and the complement-add latch is on, both cycles 3 and 4 are taken. This is necessary because of the borrow into the least significant stage of Accumulator 1. This condition is discussed in the Add to A 1 subcycle.

5. If the exponent difference is greater than or equal to 16, the larger factor is taken as the result for both true-add and complement-add.

Compare

The compare subcycle determines which of the two starting factors is the smaller and places the smaller factor in the arithmetic register.

Fixed Point OP Compare and OPA 1 are forced on to initiate the comparison. The addressed factor is read into the arithmetic register and the operated Accumulator (A1) is read into the auxiliary register. Accumulator 1 is then subtracted from the arithmetic register. The auxiliary register is read out onto Channel 2, (Adder Entry B) under complement-add control. A carry is inserted in the units position of the exponent to produce a 10's complement number. The compare operation automatically inserts a carry in the units position of the mantissa. The mantissas and the exponents are sensed separately for the carry-no-carry condition.

Two possible conditions may occur as a result of the comparison.

1. If the auxiliary register which contains the A1 factor is the larger of the two factors, a no-carry is sensed at the tens position of the exponent. This indicates that no interchange is required but that the exponent difference is a complement number.

2. If the auxiliary register is smaller than the arithmetic register, a carry is produced which initiates the interchange of the factors. Every comparison cycle therefore results in either an interchange of the factors or a recomplement cycle, not both. The following examples illustrate each case:

Arithmetic register greater than auxiliary register

5 9 7 4 3 6 9 8 2 3 - Athr - Ch 1

5 3 8 4 4 5 7 3 5 2 - Auxr - Ch 2 - Comp Add

1 Carry Insert Units

5 9 7 4 3 6 9 8 2 3

4 6 1 5 5 4 2 6 4 8

C

0 6 8 9 1 2 4 7 1

Carry initiates interchange of factors.

Arithmetic Register less than auxiliary register.

5 3 5 8 6 3 5 9 7 4 - Athr - Ch 1

5 7 4 7 3 9 5 8 2 3 - Auxr - Ch 2 - Comp Add

1 Carry Insert Units

5 3 5 8 6 3 5 9 7 4

4 2 5 2 6 0 4 1 7 7

←N/C

9 6 1 1 2 4 0 1 5 1

No carry-out but exponent difference is a complement number.

At the beginning of the compare subcycle, the contents of Accumulator 1 are read into Accumulator 3. If an interchange is called for, Accumulator 3 may then be read into the arithmetic register and the arithmetic register read into Accumulator 1.

The exponent difference is read into the shift register and Program Register D as it is formed. This may be the actual difference or the complement of the difference depending upon the outcome of the comparison.

If the exponent difference is zero, the carry from the most significant position of the mantissa initiates the interchange of the two factors.

Recomplement Exponent Difference

The no-carry condition from the tens position specifies that the auxiliary register (A1) is larger than the arithmetic register. The number in the shift register output latches and in Program Register D is therefore a complement number.

The contents of Program Register D is read out to Channel 2 Adder entry B under complement-add control. Zeros are inserted on Channel 1 adder entry A. The re-complemented exponent difference is then read back into the shift register and Program Register D.

If the exponent difference is equal to or greater than 16 (out-of-range indication) at this point, the operation is terminated with the larger factor in Accumulator 1. Accumulator 2 has been reset to plus zeros and carries a modified characteristic of 00.

Add to A 2

During the recomplement subcycle, the field register is set from the shift register output latches. The number of least significant digits of the arithmetic register equal to the exponent difference must be scanned out and added to Accumulator 2.

One of four conditions will arise as a result of the initial comparison of the two factors:

1. Exponent Difference Low (1-7)

2. Exponent Difference High (8-15)

3. Exponent Difference Out of Range (16 or greater)

4. Exponent Difference Zero

An exponent difference low sets the FP Low latch. The FP Low latch conditions a group of translator switches. These switches must transmit an indication of the shift register output latch contents once during subcycle 3, and once during subcycle 4.

Assume an exponent difference of 3 between 2 factors. During subcycle 3, digits 7, 8, and 9 must be scanned out and added to A2 positions 0, 1 and 2. The translator inserts a seven in the tens position of the field register and a 9 in the units position. At the end of subcycle 3, a two must be inserted in the tens position and a 6 in the units position of the field register. This enables the field ring to scan out digits 2-6 and add them to Accumulator 1, during subcycle 4.

Data flow during subcycles 3 and 4 for two factors with an exponent difference of 3 is detailed in Figure 2.1-2.

The exponent difference high latch output also conditions a group of translator switches. During the Add-toA2 subcycle, the arithmetic register is gated onto Channel 2 under complement-add control. If an exponent difference of 8 is indicated by the shift register output latch, the translator sends this indication to the field register units position as a 9. A two is forced into the tens position. If the exponent difference had been 9, the units position of the field register is forced 8 etc. The last 2 positions of Accumulator 2 are always zero. Hence, the exponent difference of 8 places the entire mantissa in Accumulator 2 positions 0-7. An exponent difference of 9 places 7 digits of the mantissa in positions 1-7. As the difference increases to 15, the most significant stage of the arithmetic register mantissa would then appear in Accumulator 2 position 7 with 7 leading zeros.

Operations with an exponent difference greater than or equal to 16 are discussed in Section 2.3.00.

If the exponent difference is 0, the add-to-A2 subcycle is skipped. The operation proceeds directly to subcycle 4, the addition of the mantissas, from subcycle 1.

Add to A1

The Add-to-A1 subcycle adds or subtracts the mantissas of the two factors if the exponent difference is 1-7. When the exponent difference is 8-15 on true-add, there will be no mantissa addition because the arithmetic register will have been completely scanned out. The Add-to-A1 subcycle is skipped, and a normalization cycle is initiated following subcycle 3, Add-to-A2.

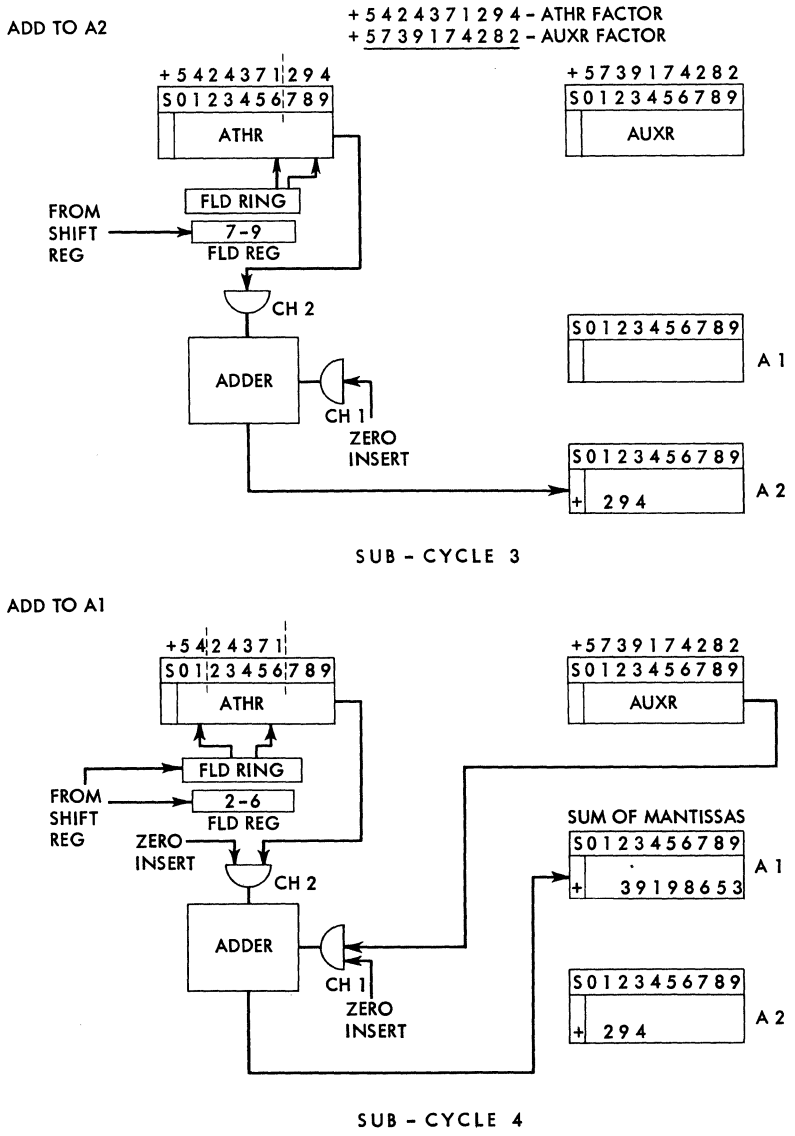
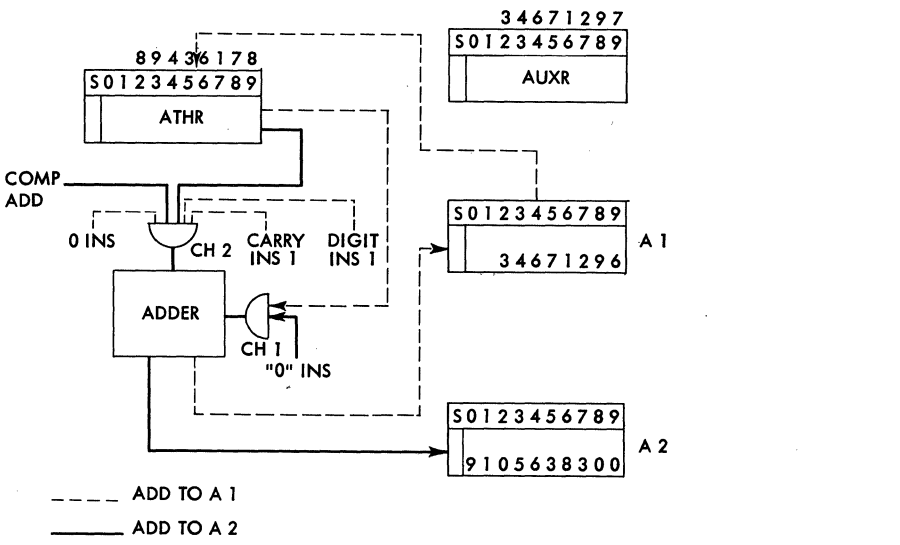


FIGURE 2.1-2 DATA FLOW FOR TWO FACTORS (EXP DIFF 3)

For the complement-add condition, however, with exponent difference 8-15, Add-to-A1 is necessary. The existence of a digit in any position of Accumulator 2 from positions 0-7 causes a borrow from the least significant digit position of Accumulator 1. The borrow is accomplished by inserting a 1 from Channel 2 digit insert under complement-add control. A carry is inserted simulataneously. The 1 digit is thus tens complemented to a 9. Zeros are inserted for the remaining digits on Channel 2. The resulting **nine added** to the least significant digit has the effect of reducing the digit by one. If the digit is other than 0, the resulting carry, added to the 9's (complement zero) on Channel 2 causes the arithmetic register contents to be added to zero. The remaining digits of the arithmetic register, therefore, are not modified by the operation. The end carry from the adder most significant position is ignored.



ACTUAL SUBTRACTION

```
34671297.00000000
         .08943617
34671296.91056383
   A1       A2
```

MACHINE METHOD

```
34671297.   ① LEAST SIGNIFICANT 9 IS COMPLEMENTED 1
99999999    ② REMAINING 9'S ARE COMPLEMENTED 0'S
```

IGNORE END CARRY 1 `34671296.`

FIGURE 2.1-3 BORROW FROM A1 DURING COMP ADD (EXP DIFF 8-15)

Data flow during the operation and illustrative examples indicating pencil and paper subtraction of two factors is shown in Figure 2.1-3. A sample calculation is also shown for the machine method of performing the borrow function.

If the digit is zero, however, the complement operation produces a 9 for the least significant position. The remaining zeros read out on Channel 2 under complement-add control of course become 9's. The second digit position, if non-zero, or the first non-zero position encountered, adds to 9 reducing it by one to satisfy the borrow condition. A carry is generated from this position and succeeding digit positions so that the remaining digits retain their original values. The end carry from the adder is ignored as before.

Normalize

The normalization subcycle counts the number of high-order zeros in Accumulator 1 or Accumulator 2 if Accumulator 1 is all zeros. Fixed point Double Shift and Count is turned on to initiate the operation. Accumulator 1 is transferred to the word buffer register. Simultaneously, the significant digit scanner senses the location of the high-order digit and transmits its indication to the shift register. The shift register is read

out onto Channel 2 with complement-add control. At the same time a 7 digit is inserted on Channel 1 together with a carry insert. The shift register output is thus added to 8. This operation yields the number of left shifts minus 2 required to normalize the number.

For example, assume that Accumulator 1 has 4 high order zeros. The 4 output then transmitted to the shift register is added to 8 giving a 2 output from the adder. A left shift of four is thus indicated.

Accumulators 1 and 2 are shifted all the way to the left, but the correction factor for the exponent is two less than the number of places shifted.

This process is illustrated as follows:

Normalization (Shift Left of 2 places)

```
                0 1 2 3 4 5 6 7 8 9
                0 0 0 0 3 9 7 0 1 4
Before
Normalizing
                          4 Shift Reg to Ch 2
                          7 Insert on Ch 1
                    Add   1 Carry Insert
                    Ignore 1  2 Adder output signals
                    Carry     Shift Left of 4 places.


                0 1 2 3 4 5 6 7 8 9
Resulting       0 0 3 9 7 0 1 4 x x
Normalized                      ┌──┘
Number                          └→From Acc 2, positions 0-1
```

The results of the addition are gated into Auxiliary Register 2 for use in the succeeding cycle. The original exponent must be reduced by 2 to compenstate for the mantissa shift left of 2.

The addition of two mantissas may produce a carry into position 1 of Accumulator 1. In this case the Shift Left and Count result is a 1. The 1 is added to 8 and the resulting 9 stored in Auxiliary Register 2. For this operation, the 9 result indicates a shift left of one place. When Accumulator 1 is adjusted during subcycle 6, the complement-adding of 9 to the original exponent will add 1 to the original exponent.

If Accumulator 1 is all zeros, the significant digit scanner =0 indication causes the program ring to recycle. Accumulator 2 is then transferred to the word buffer register. Simultaneously, the significant digit scanner samples Accumulator 2 for the location of the high order digit. This indication is sent to the shift register as before. The shift register is then read out to Channel 2. The 7 digit and the carry are inserted on Channel 1. The adder operation is the same as for Accumulator 1 except that a 1 digit is inserted at the second digit time to indicate that the left shift exceeded 10. This number is then gated into Auxiliary Register 2.

To adjust the original exponent during subcycle 6, Accumulator 3 is read into the arithmetic register during subcycle 5.

Update A1 Exponent

The original exponent, now in the arithmetic register, must be adjusted and inserted in positions 0-1 of Accumulator 1.

The arithmetic register positions 0-1 are read out to Channel 1. At the same time the the auxiliary register is read out to Channel 2 under complement-add control. Zeros are inserted on Channel 1 to fill the vacant positions. The result of the subtraction is gated into Accumulator 1 (positions 0-1) and Program Register D as it is formed. Accumulator 1 is also shifted right two places at this time to make room for the updated exponent. The digits shifted out are gated into Accumulator 2 positions 0-1. Overflow and underflow tests are then made on the updated exponent.

Update A2 Exponent

The updated A1 exponent is now reduced by eight and inserted in Accumulator 2.

Program Register D is read out to Channel 1. An 8 digit is inserted on Channel 2 with complement-add control. Zeros are inserted for the remainder of the Channel 2 entries. Accumulator 2 is simultaneously shifted right two places to make room for the new exponent.

The exponent is read into Accumulator 2 positions 0-1. An underflow test is made on this exponent.

Sign Control

Sign control is accomplished in the same way as for fixed point instructions. The signs of the addressed factor and the operated accumulator are switched with FP OP + and FP OP - to determine the true-add or complement-add operation.

The conditions for a true-add or complement-add for particular combinations of addressed and operated factor signs and operations is discussed in 5.0.00.

Floating Add-Sub't Absolute

Data flow for the floating add-sub't absolute codes is almost the same as for normal floating add-sub't codes. The one exception occurs in the treatment of the addressed factor sign.

For the floating add absolute code the addressed factor sign is set plus. If an interchange of the factors is called for on the compare subcycle, the OP sign is set plus. In either case the read-in from the arithmetic bus of the addressed sign or the OP sign is blocked.

The floating subtract absolute code sets the addressed factor sign plus or the OP sign plus if interchange is required. The read-in of the addressed factor sign and OP sign from the arithmetic bus is blocked.

2.2.00 FLOATING ADD-SUB'T ± 74

Floating Add-Sub't Absolute ± 77

To perform the Floating Add-Sub't operation, the data cycle is divided into seven subcycles. Depending upon the relative values of the exponents of the two factors, up to seven of these subcycles may be used during the execution of the Floating Add-Sub't instruction.

The floating ring, a seven-position trigger ring, causes the operation to step from one subcycle to the next. An additional control ring, the FP Op End Ring, generates four outputs similar to the program cycle ring outputs at the conclusion of each subcycle except during subcycles 3, 4, and 5. These outputs are:

1. FP Last +

2. FP Test

3. FP Stop

4. FP Stop +

The FP Op End Ring outputs are used so that is is not necessary to generate the normal stop inhibit for the program ring on each subcycle.

The Start FP Op End circuitry (42.80.12) causesthese outputs to be generated by switching the outputs of the floating ring with the program ring outputs at the appropriate time during each subcycle.

The No FP End signal (42.80.11) is switched with the FP test signal from the FP Op End Ring to recycle the program ring. The No FP End Signal is capable of recycling the program ring except when the stop latch (42.80.11) is turned on by an end-data operation, or the FP 7 C1 signal which designates the conclusion of the floating add-sub't instruction. The end-data operation occurs before the end of the instruction when an exponent difference out of range (equal to or greater than 16) is detected during the compare subcycle.

The No FP End signal is switched with FP Test from the FP Op End Ring to recycle the program ring except during subcycles 3, 4, and 5. During the Add to A2, Add to A1, and normalization operations, the program ring is recycled by switching the program cycle test and the FAS 3, FAS 4, and FAS 5 output, respectively.

For the two conditions, Add to A2 and Add to A1, the field ring match signal and arithmetic field end cause the generation of the normal stop inhibit during the execution of the add operation.

During normalization, the program cycle ring is turned on by the Double Shift and Count function.

The data cycle begins in the usual way with the setting of the data latch. This initiates the memory request. The A & P Address Gate then starts the data control ring. The ring reads the data address from Program Register D into the computer address bus latch register at Data Control 1 time. From here it is transferred to the memory address register. The addressed factor is then read out onto the information bus for transfer into the arithmetic register at Data Control 4 time.

The initial access codes signal is switched with a Data Control 4 pulse to set the floating ring to Position 1. The output of the Floating Ring, FP 1, is then switched with FP Op Add-Sub't to give Arith Op-FP 1. The Arith Op-FP 1 signal causes the Op Compare Abs and Op A 1 functions to be forced on. This initiates the first subcycle of the data operation.

Objective	Timing	Logic and Location
Op Compare Abs		
Set Floating Ring PSN 1	DC 4	4J-42.80.04
Arith Op-FP 1	FP 1-FP Add-Sub't	4A-42.86.15
Op Compare Abs	Arith Op-FP 1	5A-42.86.12
Op A1	Arith Op-FP 1	3F-42.86.12

The following major objectives are required to perform the floating add-sub't operation:

A. Compare

B. Recomplement Exponent Difference

C. Add to Accumulator 2

D. Add to Accumulator 1

E. Normalize

F. Update Accumulator 1 Exponent

G. Update Accumulator 2 Exponent

H. End Data Operation

Compare (Figures 2.2-1, 2 and 3)

The Compare subcycle determines which of the two factors is the smaller and places the smaller factor in the arithmetic register.

As just indicated, the Arith Op-FP 1 signal causes the Op Compare Abs function to be forced on. As a result of the Compare Abs operation an interchange of the two factors may be specified. The mantissas are sensed at C9 time and the exponents at C11 time for the existence of a carry. If a carry occurs from the tens position of the

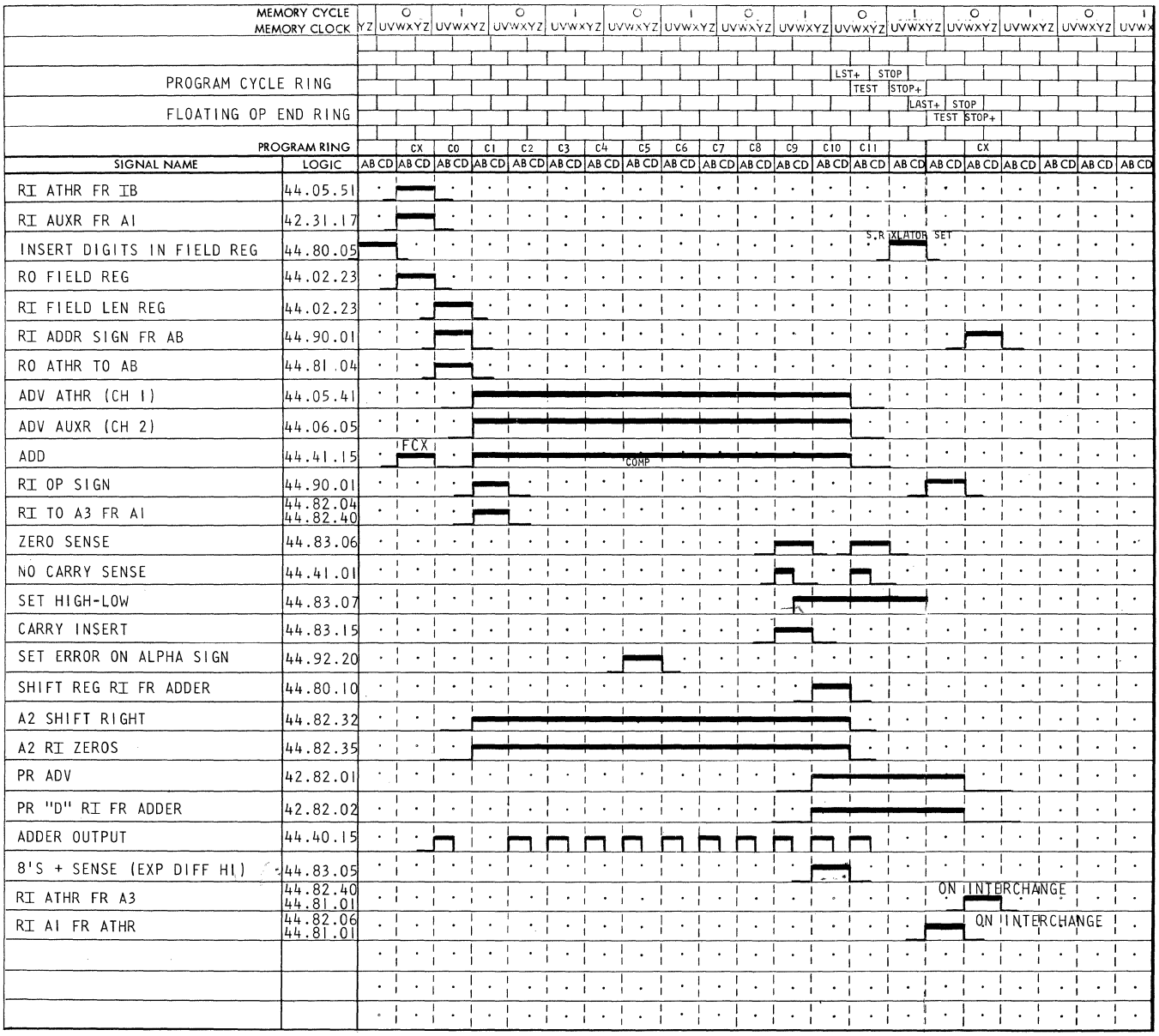


FIGURE 2.2-1 SUBCYCLE 1 - COMPARE (FORCE ON OP COMPARE ABS AND OP A1)

exponent, at C11, an interchange of the factors is indicated. If the exponent difference is 0, the carry from the high-order stage of the mantissa initiates the interchange.

The Compare subcycle also determines the range of the exponent difference by sampling the adder output latches at C10 and C11 time (Logic 44.83.05-44.83.06). One of four latches is turned on during the the operation:

1. Diff 0 Latch - Exp Diff 0

2. Diff Lo Latch - Exp Diff (1-7)

3. Diff Hi Latch - Exp Diff (8-15)

4. Diff Out of Range Latch - (Exp Diff greater than or equal to 16)

PROGRAM RING		CX	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11		
SIGNAL NAME	LOGIC	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD
SET ADDR SIGN LATCH PLUS	44.90.07															
SET OP SIGN PLUS	44.90.03															
ON INTERCHANGE																
BLOCK RI ADDR SIGN	44.90.01															
LATCH FROM AB																
BLOCK RI OP SIGN	44.90.01															
LATCH FROM AB																

FIGURE 2.2-2 FLOATING ADD ABSOLUTE SUBCYCLE 1 (FAAB SAME AS FA EXCEPT AS INDICATED)

PROGRAM RING		CX	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11			CX
SIGNAL NAME	LOGIC	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD	AB CD
SET ADDR SIGN	44.90.07																
LATCH PLUS																	
SET OP SIGN LATCH	44.90.03																
PLUS ON INTERCHANGE																	
BLOCK RI ADDR SIGN	44.90.01																
LATCH FROM AB																	
BLOCK RI OP SIGN	44.90.01																
LATCH FROM AB																	

FIGURE 2.2-3 FLOATING SUBTRACT ABSOLUTE - SUBCYCLE 1 (FSAB SAME AS FS EXCEPT AS INDICATED)

These latches control the subsequent execution of the instruction. The function of these latches is detailed in succeeding sections.

The difference between the floating add-sub't and floating add-sub't absolute codes occurs in the interpretation of the addressed and operated factor signs.

During a normal add-sub't operation, the addressed sign latch is set at C0 time by reading out the arithmetic register to the arithmetic bus. The operated sign latch is set during the transfer of Accumulator 1 to Accumulator 3 at C1 time.

The floating add-sub't absolute code sets the addressed factor sign plus. If an interchange is called for, the op sign is forced plus and a new sign is read into the address sign latches. For both normal and interchange operations, the read-in of addressed and and operated signs from the arithmetic bus is blocked.

The field register is set for full field control at the beginning of the Compare Abs sybcycle by the Arith Op-FP 1 signal and a Data Control 4 pulse (Logics 44.80.05-44.80.06). The tens position is forced to 0 and the units position to 9. This is necessary for the floating point codes because the Compare Abs function would normally be used with the field control.

Objective	Timing	Logic and Location
Athr Par RI IB	DC 4	4E-44.05.51
Auxr RI FR A1	DC 4	3C-42.31.17
Field Reg to Adder (Op FCX)		
RO Field Reg	CX	3B-44.02.23
Comp Add	CX	6E-44.41.15
No-Carry Ins	CX-AP	4C-44.41 02
RI Field Len Reg	C0	4G-44.02.23
RI Addr Sign FR AB	CX-0, FAS 1	4C-44.90.01
RI Op Sign	C0-1, FAS 1	4F-44.90.01
RI A 3 FR A1		
A 1 RO AB	C0-1	4C-44.82.04
A 3 RI FR AB	C0-1	4B-44 82.40
Adv Athr (Entry A)	C1	4A-44 05.41
Adv Auxr (Entry B)	C1-SDS Check	5C-44.06.05
Comp Add	Compare Codes C1	2B-44.41.15
A 2 Sh Rt	C0-1	4A-44.82.32
A 2 RI Zeros	FAS 1-No FAD or FADSN	4A-44.82.35
Set Error on Alpha Sign	C5	4A-4B-44.92.20
Test Zero Sense	C9 + C11	44.83.06
No Carry Sense	C9 + C11	44.41.01
Carry Insert	C9	4J-44.83.15
Set Hi-Low	C9	44 83.07
PR Adv	C10	4A-42.82.01

Objective	Timing	Logic and Location
PR RI FR Adder	C9-10	2A-42.82.02
8's + Sense (Exp Diff Hi)	C10	3G-3H-44.83.05
Sh Reg RI FR Adder	C10	4A-44.80.10
SR Xlator Set	FP Stop +	
Set Recomp	C11-No Carry 1st	44.83.01
Set Interchange	C11-Carry 1st	44.83.01
On Interchange		
RI Athr FR A 3		
A3 AB RO	FP Interchange-FP Stop	4G-44.82.40
Athr RI FR AB	FP Interchange-FP Stop	4G-44.81.01
RI A 1 FR Athr		
A 1 RI FR AB	FP Interchange-Last +	4B-44.82.06
Athr RO AB	FP Interchange-Last +	4B-44.81.01

The floating ring is advanced to subcycles 2, 3 or 4 depending upon the results of the compare operation, If the exponent difference is equal to or greater than 16 (out of range), an end-data operation is initiated.

Objective	Timing	Logic and Location
Set Floating Ring	FP Recomp	4B-42.80.06
PSN 2	FP 1 Adv	
Set Floating Ring	No FP Recomp	4F-42.80.06
PSN 3	Exp Diff Non-zero	
	FP 1 Adv	
Set Floating Ring	FP 1 Adv	5C-5D-42.80.05
PSN 4	No FP Recomp	
	FP Add Sub't	
	Exp Diff Zero	

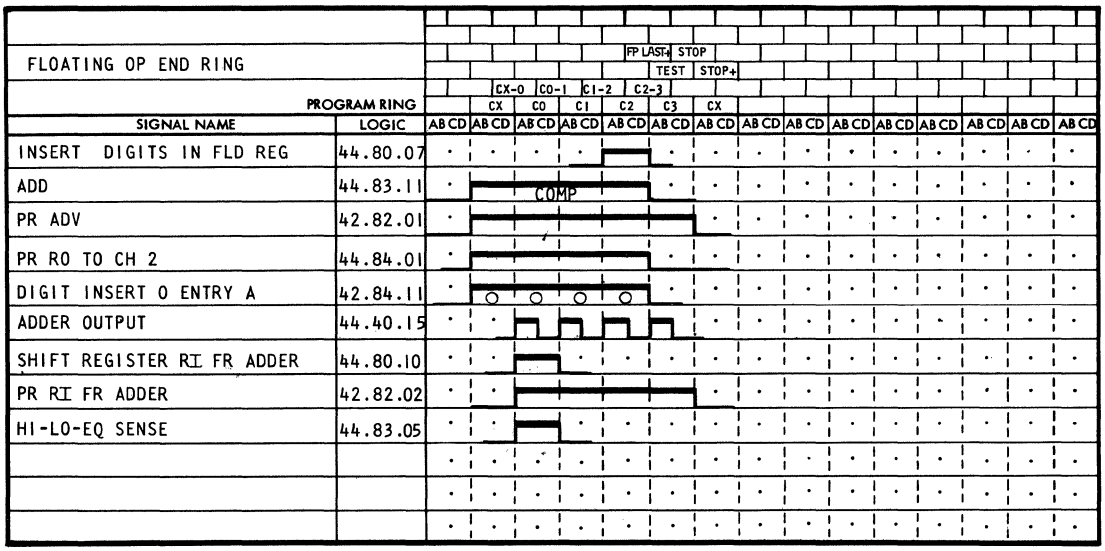


FIGURE 2.2-4 SUBCYCLE 2 - RECOMPLEMENT EXPONENT DIFFERENCE

A forced setting of the field register for the Add to A2 operation is accomplished at PC Test time. If the exponent difference is zero, a full field condition is forced for the Add to A1 subcycle. These conditions are discussed as follows:

Recomplement Exponent Difference (Figure 2.2-4)

The No-Carry 1st condition sets the Recomp latch (Logic 44.83.01). The output of the Recomp latch is switched with FP 1 Adv, the output of trigger 1 of the Floating ring, to give FP 2. FP 2 is switched with FP Op Add-sub't to give FAS 2. The output of the 1/2 time latch of the floating ring, FP 1-2, is in turn switched with FP Stop to advance Program Register D.

FAS 1-2 gates Program Register D, which contains the tens complement of the exponent difference, onto Channel 2 under complement-add control. Zeros are inserted on Ch 1. The recomplemented exponent difference is gated into the shift register and back into PR D starting at C0 time.

Objective	Timing	Logic and Location
PR Adv	FAS 1-2	4B-42.82.01
PR RO to Ch 2	FAS 1-2	4B-44.84.01
Digit Insert 0's Ch 1	FAS 1-2 FP Stop	4D-42.84.11
Comp Add	FP 1-2 FP Add-Sub't	4C-44.83.11
RI Shift Reg FR Adder	FAS 2 C0	4B-44.80.10
PR RI FR Adder	C0	2B-42.82.02
Hi-Lo-Eq Sense	C0	44.83.05
Inserts Digits in Fld. Reg.	FAS 2 C2	4A-44.80.07

When the exponent difference is greater than or equal to 16, the floating ring is advanced to FP 2 and allowed to run five digit times. This allows setting the correct sign into Accumulator 1. Accumulator 2 is also set to positive zero during the same operation.

Objective	Timing	Logic and Location
Set Floating Ring Position 2	FP 1 Adv, Exp Diff Out of Range	42.80.06
ATHR Zero Ins	Exp Diff Out of Range	44.81.02
ATHR RO AB	C0-1	44.81.01
A 2 AB RI	Exp Diff Out of Range	44.82.34
ATHR Sign Ctrl	CX-0	44.90.05

Add to A2 (Figures 2.2-5, 6, and 7)

The field register is set for the Add to A2 operation either during the compare or recomplement subcycles depending upon whether or not a Recomp cycle is necessary. The setting of the field register is accomplished during subcycle 1 or 2 switching the outputs of the shift register output latches with FAS 1 or FAS 2. The FAS 1 or FAS 2 signals together with the exponent difference values determine the digits to be set into the field register. The following table indicates the digits that must be set into the field register for particular exponent difference values.

FAS 3 (Add to A2)

Exp Diff Low (1-7)

	FLD Reg Setting	
Exp Diff	Tens	Units
1	9	9
2	8	9
3	7	9
4	6	9
5	5	9
6	4	9
7	3	9

FAS 3 (Add to A2)

Exp Diff High (8-15)

	FLD Reg Setting	
Exp Diff	Tens	Units
8	2	9
9	2	8
10	2	7
11	2	6
12	2	5
13	2	4
14	2	3
15	2	2

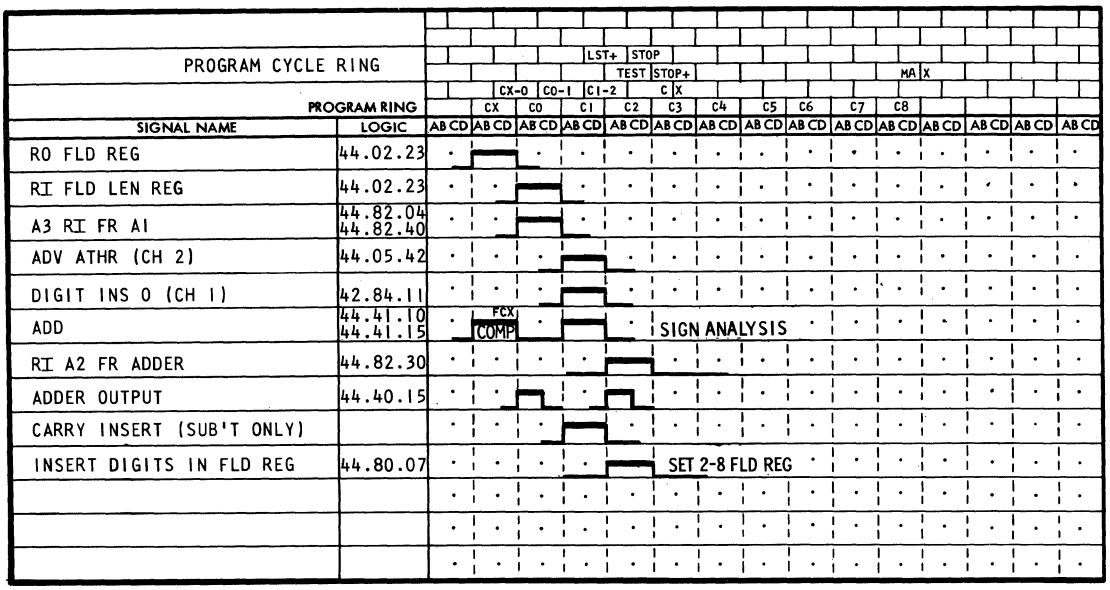


FIGURE 2.2-5 SUBCYCLE 3 - ADD TO A2 EXPONENT DIFFERENCE 1 - FORCE OP ADD-SUB'T

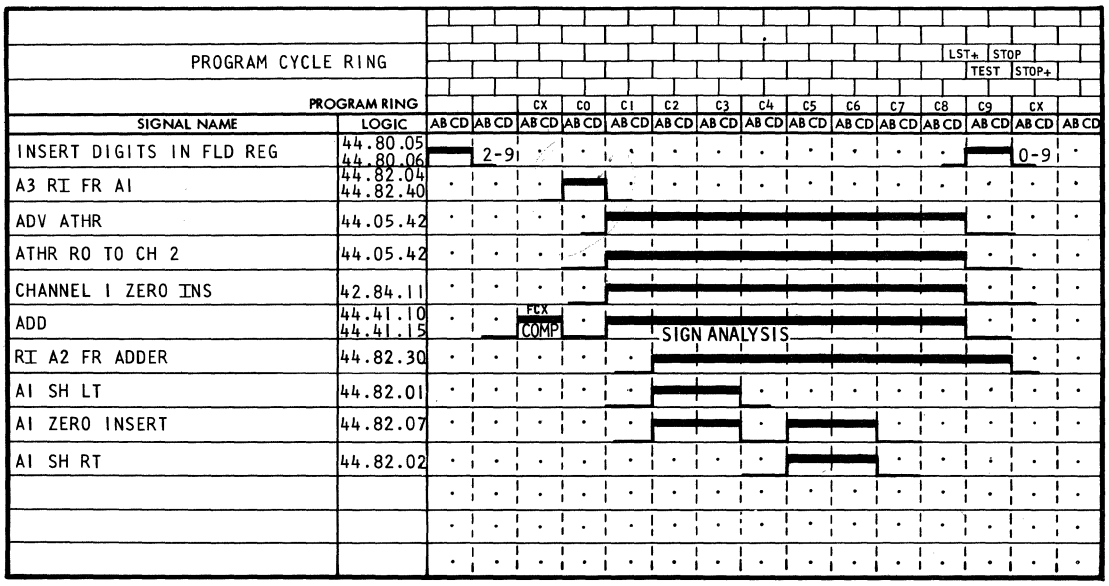


FIGURE 2.2-6 SUBCYCLE 3 - ADD TO A2 EXPONENT DIFFERENCE OF 8 - FORCE OP ADD-SUB'T

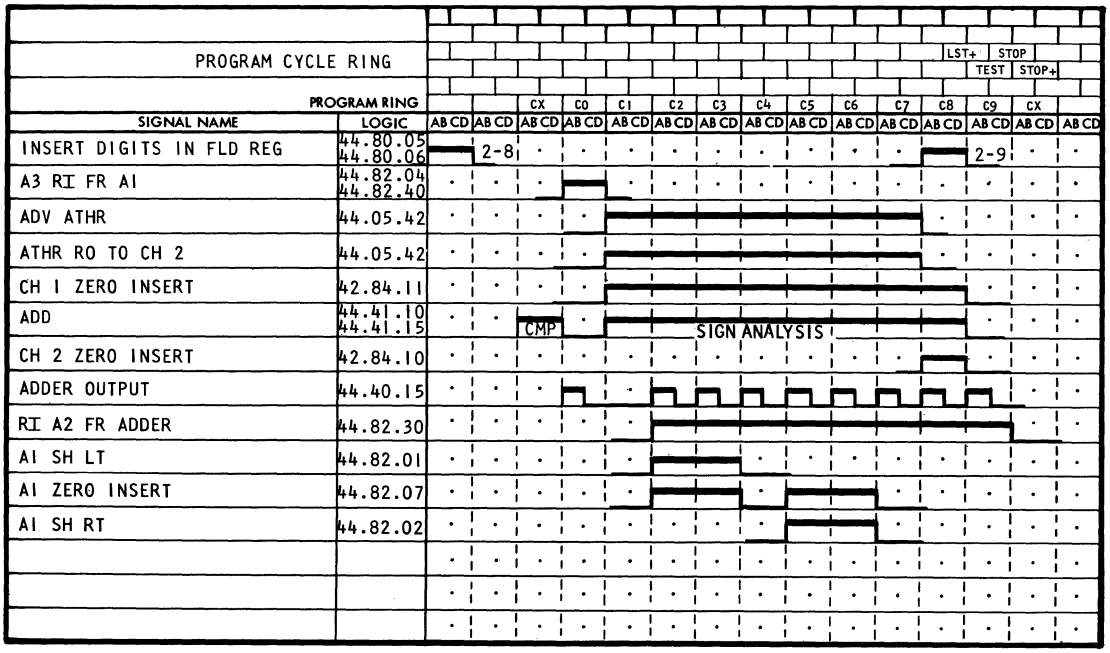


FIGURE 2.2-7 SUBCYCLE 3 - ADD TO A2 EXPONENT DIFFERENCE OF 9 - FORCE ADD-SUB'T

During FAS 1 at Program Cycle Stop + time, or during FAS 2 at C2 time of the program ring, the translator is set for Add to A2. The contents of the shift register output latches, which contain the exponent difference, condition the shift register to Decimal A switches (Logic 44.80.02). Two sets of translator switches are provided. One set is for the exponent difference low condition and the other for exponent difference high. The outputs for the high and low conditions are shown in the following tables.

Exponent Difference Low (44.80.02)

Exp. Diff	S. R. Output Latches	XLATOR OUTPUT FLD REG. Setting (Tens PSN)
1	1B, 0B	9
2	2B, 0B	8
3	3B, 0B	7
4	3B, 1B	6
5	3B, 2B	5
6	6B, 0B	4
7	6B, 1B	3

Exponent Difference High (44.80.02)

Exp. Diff	S. R. Output Latches	XLATOR OUTPUT FLD REG. Setting (Units PSN)
8	6B, 2B	9
9	6B, 3B	8
10	2B, 1B	7
11	1B, 0B	6
12	2B, 0B	5
13	3B, 0B	4
14	3B, 1B	3
15	3B, 2B	2

The outputs of the shift register output latches are first translated to decimal. As indicated by the table, the latch outputs are translated to the tens complement of the exponent difference for the exponent difference low condition. This number is then translated back to two-out-of-five code and inserted in the field register. This yields the correct tens digit for the field register. The units position is always forced to 9 by the exponent difference low signal.

The exponent difference high causes a different translation to take place. For the high condition, the tens position of the field register is always forced to 2. The units position as indicated by the table varies from 9 down to 2 as the exponent difference varies from 8 to 15. This causes the field ring to begin the scanning out operation at the proper units digit.

The operation begins by forcing fixed point add-sub't (Logic 42.86.12). Op A2 is also forced causing A2 to be transferred to the auxiliary register at the beginning of the operation.

Objective	Timing	Logic and Location
Set Add-Sub't	FAS 3	2B-42.86.12
Set Op A1	FAS 3	4B-42.86.12

The field ring cuases the arithmetic register to be scanned out onto Channel 2, adder entry B, as in a normal add operation. A sign analysis determines the true-add or complement-add function in the usual way. A carry is inserted for the subtract operation to give a tens complement. Zeros are inserted for the Channel 1 entries. The results of the addition are read into Accumulator 2 starting with position 0. Accumulator 2 is simultaneously shifted right each digit time.

To save the larger exponent for possible adjustment during a later subcycle, Accumulator 1 is read into Accumulator 3 at CX-0 time.

Objective	Timing	Logic and Location
RO Fld Reg	CX	3B-44.02.23
RI Fld Len Reg	C0 BP	4G-44.02.23
A3 RI FR A1		
A1 AB RO	CX-0	4F-44.82.04
A3 AB RI	CX-0	4F-44.82.40
Adv Athr	C1-Op Add to A	4B-44.05.42
Digit Insert Zeros Ch 1	C0-1	4B-42.84.11
Carry Insert (Sub't only)		
RI A2 FR Adder	C1-2	4D-44.82.30

Objective	Timing	Logic and Location
Stt A2 Sh RT	C1-2	4D-44.82.32
Stop A2 Sh RT	PC Stop	4E-44.82.32
Inserts Digits in Fld Reg	PC Test	44.80.05

Add to A1 (Figures 2.2-8, 9, and 10)

Subcycle 4, Add to A1, adds the mantissas and gates the result into Accumulator 1.

The field register is loaded for the Add to A1 operation either during the previous subcycle or during subcycle 1 if the exponent difference is zero. The table shown below indicates the units position setting of the field register when the exponent difference is low. The tens position of the field register is forced to 2 for this condition (44.80.06).

Exponent Difference Low (44.80.12)

Exp Diff	S. R. Output Latches	XLATOR OUTPUT FLD REG setting (Units PSN)
1	1B, 0B	8
2	2B, 0B	7
3	3B, 0B	6
4	3B, 1B	5
5	3B, 2B	4
6	6B, 0B	3
7	6B, 1B	2

Exponent difference low causes the field ring to scan out the arithmetic register onto Channel 2 just as in a normal add. The ring starts the scan at the appropriate units position digit as specified by the parallel output latches of the shift register. The program cycle ring is started to stop the operation by the coincidence of the field match and shift register A last signals just as in a normal add. The program cycle ring is recycled in the CX position by the FAS 4 and PC Test signals (Logic 42.80.13).

The exponent difference high condition causes Add to A1 to be skipped if the true-add latch is on. During complement-add, Add to A1 is executed because of the borrow from the low order of A1.

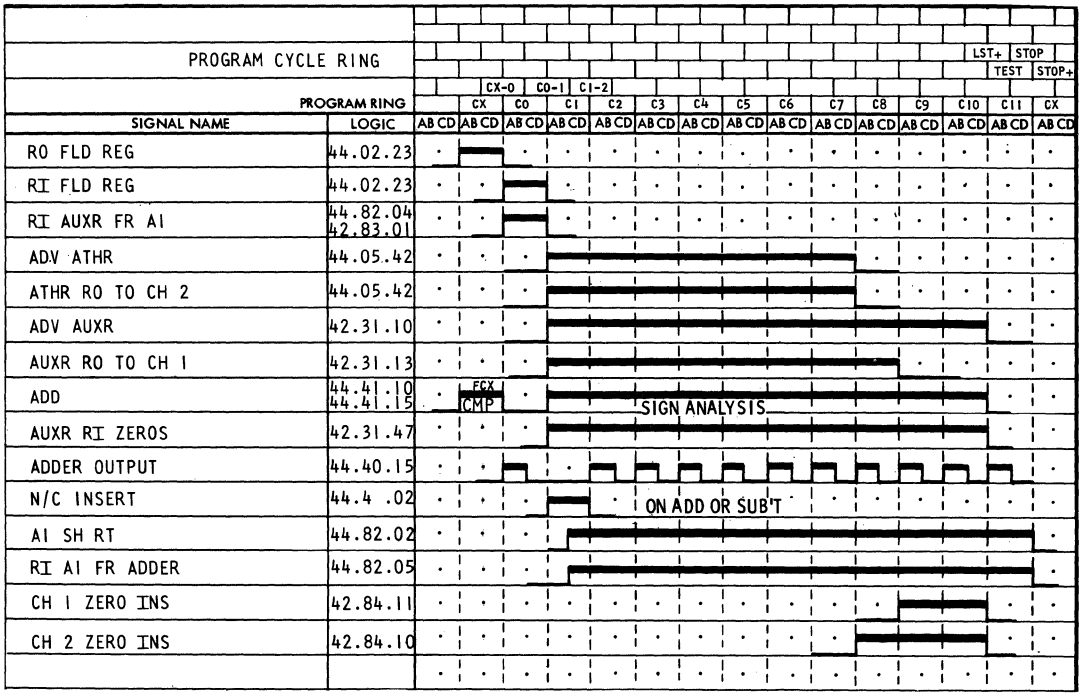


FIGURE 2.2-8 SUBCYCLE 4 - ADD TO A1 EXPONENT DIFFERENCE OF 1 - FORCE OP ADD TO ACC, OP A1

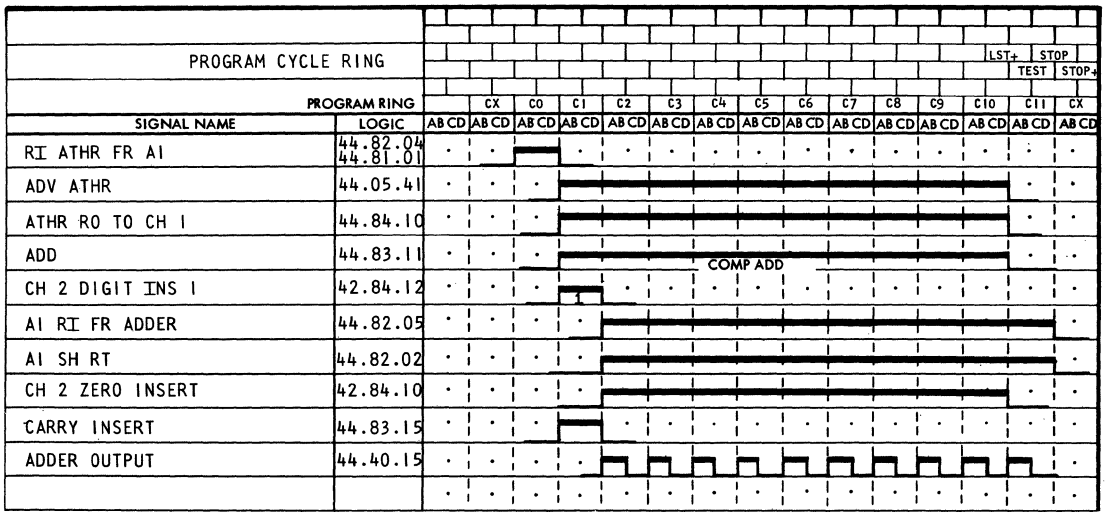


FIGURE 2.2-9 SUBCYCLE 4 - ADD TO A1 EXPONENT DIFFERENCE OF 8 - SKIP THIS CYCLE ON TRUE-ADD

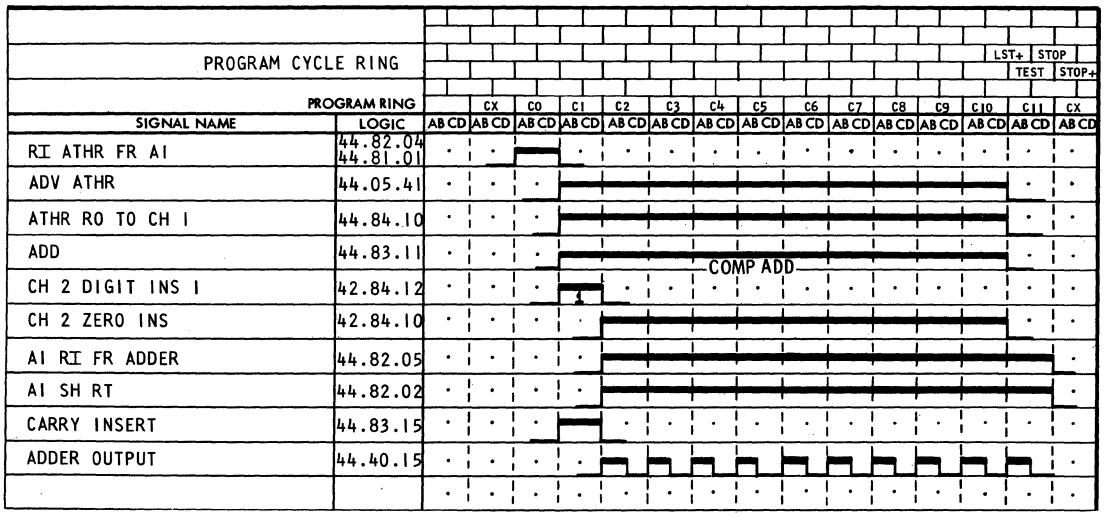


FIGURE 2.2-10 SUBCYCLE 4 - ADD TO A1 EXPONENT DIFFERENCE OF 9 - SKIP THIS CYCLE ON TRUE-ADD

Objective	Timing	Logic and Location
Set Floating Ring to PSN 4	Signs Comp Add FP 3 Adv	4A-42.80.05
Set Floating Ring to PSN 5	Signs True-add Exp Diff High FP 3 Adv	4H-42.80.05

Accumulator 1, which contains the larger factor, is transferred to the arithmetic register at C0-1 time. The arithmetic register is then read out onto Channel 1. The only objective of Add to A1 on complement-add when the exponent difference is high is the reduction of the low order digit by 1 to satisfy the borrow condition. A digit 1 is inserted on Channel 2 under complement-add control. A simultaneous carry insert gives a nine output for the first digit time. Adding a 9 to the low order digit of the larger factor on Channel 1 reduces it by one. The remaining string of zeros on Channel 1 are complemented to 9's. (The carry from the low-order position is propagated through all positions causing the digits of the larger factor to be added to zeros).

Objective	Timing	Logic and Location
Op A1	FAS 4	3E-42.86.12
Op Add to Acc	FAS 4	3G-42.86.12
RO Fld Reg	CX	3B-44.02.23
RI Fld Len Reg	C0	4C-44.02.23
RI Auxr FR A1		
A1 RO AB	CX-0	44.82.04
Auxr RI AB	CX-0	42.83.01
Adv Athr	C1	
Athr RO to Ch 2	C1-Op Add to A	4B-44.05.42
Adv Auxr	SDS Check	4D-42.31.10
Auxr RO to Ch 1	Op Add to A	4D-42.31.13
Auxr RI Zeros	C1	
A1 Sh RT	C1-2	5A-44.82.02
RI A1 FR Adder	C1-2	4D-44.82.05
Ch 1 Zero Insert	C8-9	5F-42.84.11
Ch 2 Zero Insert	C1-2 Exp Diff Hi	5D-42.84.10

Objective	Timing	Logic and Location
No-Carry Insert (T-A)	C1-True-Add	3D-44.41.02
Exp Diff High		
Digit Insert Ch 2	C1-Exp Diff Hi	3G-42.84.12
Comp Add	C0-1 Exp Diff Hi	5F-44.83.11
Ch 2 0 Ins	C1-2 Exp Diff Hi	5D-42.84.10
Athr RI AB	CX-0	4H-44.81.01
A1 AB RO	CX-0	4E-44.82.04
Carry Insert	FAS 4 Exp Diff Hi C1	5E-44.83.15
Athr RO to Ch 1	C0-1	4E-44.84.10

Normalize (Figures 2.2-11, 12)

The normalization subcycle counts the number of high-order zeros in Accumulator 1 or Accumulator 2 if Accumulator 1 is all zeros. This cycle also records the number of such shifts and stores the count in the auxiliary register for use in the updating of the exponent during the succeeding cycle.

Fixed point Double Shift and Count is forced on by FP 5 (Logic 42.86.13). Accumulators 1 and 2 are coupled and shifted left. The shifts continue until a non-zero digit appears in the high order position of A1 (position 0).

A1 Non Zero The operation begins by first checking Accumulator 1 for the location of the high-order digit. The significant digit scanner is set with the location of the high-order digit during the transfer of Accumulator 1 to the word buffer register at C0-1 time. Simultaneously Accumulator 2 is read into Accumulator 1.

The significant digit scanner sets the shift register with the position of the high-order digit of Accumulator 1. The shift register reads out this number to its parallel output latches in 9's complement form. A seven is inserted on Channel 1 at C2 time. The shift register output latches are read out to the Channel 2 mix at C2 time under complement-add control. A carry insert gives a tens complement number.

During normalization, Accumulators 1 and 2 are shifted until a non-zero digit appears appears in position 0 of A1. This is actually two positions more than is required for normalization: The number in the Auxiliary Register 2 specified the amount by which the exponent must be reduced to compensate for the cycle.

A1 Zero If Accumulator 1 is all zeros, the significant digit scanner equal zero indication at C0 time causes the program ring to recycle. The program ring recycles in the CX position. Accumulator 2 is then transferred to the word buffer register and the shift is executed exactly as in the A1 non-zero case.

The adjustment of the SDS indication in the shift register output latches is the same as for the previous case with one exception. On the second digit time, a one is inserted on Channel 1 at C3 time. This indicates that the left shift equalled or exceeded ten.

As before, the word buffer register is shifted left until a non-zero digit is present in position 0. Accumulator 1, which contained all zeros, is transferred to Accumulator 2 by PC Last +. The word buffer is read into Accumulator 1 one digit time later by a PC Stop pulse.

Objective	Timing	Logic and Location
RI WBR FR A1		
A1 AB RO	PC Last +	3E-44.11.01
WBR RI AB	C0-1	44.82.60
SDS Sense	C0-1	2A-44.06.09
RI A 1 FR A2		
A2 AB RO	CX-0	3A-44.21.00
A1 AB RI	C0-1	4E-44.11.00
Stop Sh Reg RO	CX	4G-44.80.11
Sh Reg Par RO	C1-2	4H-44.80.10
SR RO to Ch 2	C2	4E-44.80.10
Ch 1 Digit Ins 7	C2	2B-42.84.13
Carry Ins	C2	4E-44.83.15
Comp Add	C1-2	4B-44.83.11
ZI Lt Sh	C0-1	5H-44.50.23
Auxr 2 RI FR Adder	C2-3	4B-42.83.03
Auxr 2 Sh RT	C2-3	4A-42.83.04
Ch 2 Zero Ins	C2-3	5H-42.84.10
Ch 1 Zero Ins	C3-4	5D-42.84.11

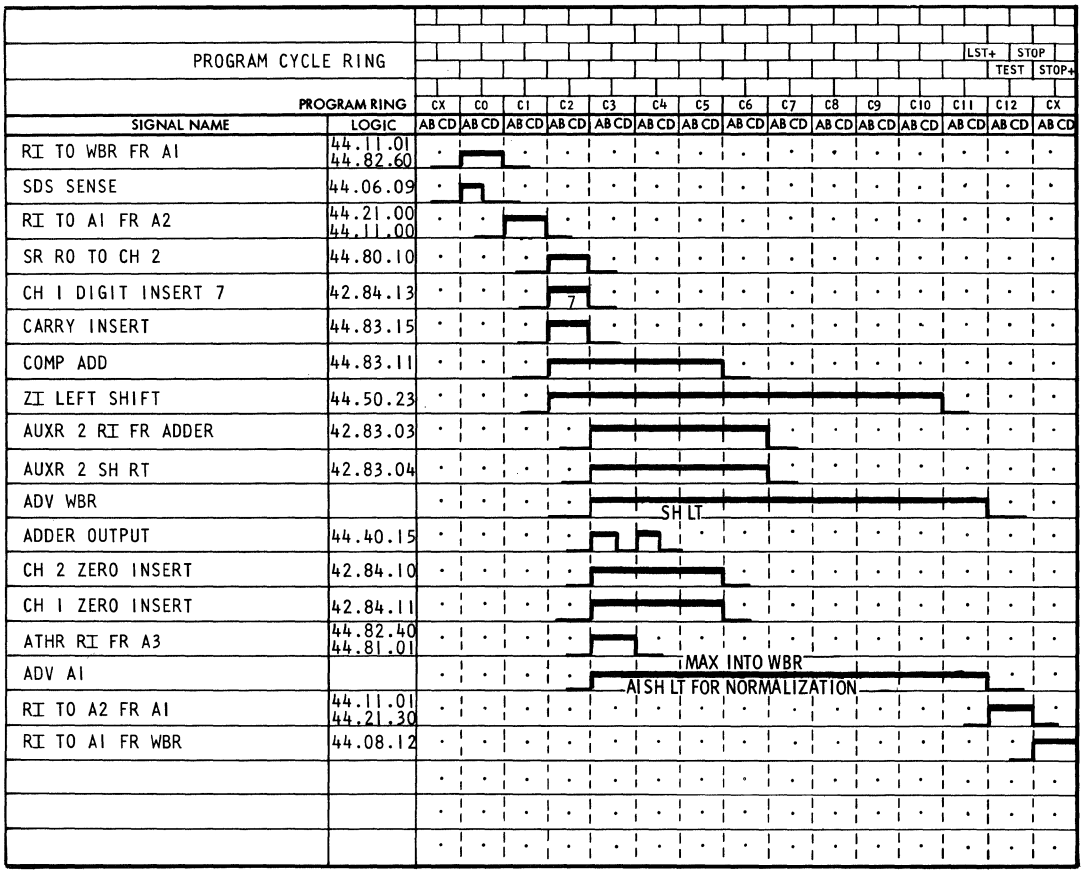


FIGURE 2.2-11 SUBCYCLE 5 - NORMALIZE 1ST SIGNIFICANT DIGIT IN A1 - FORCE DOUBLE SHIFT AND COUNT

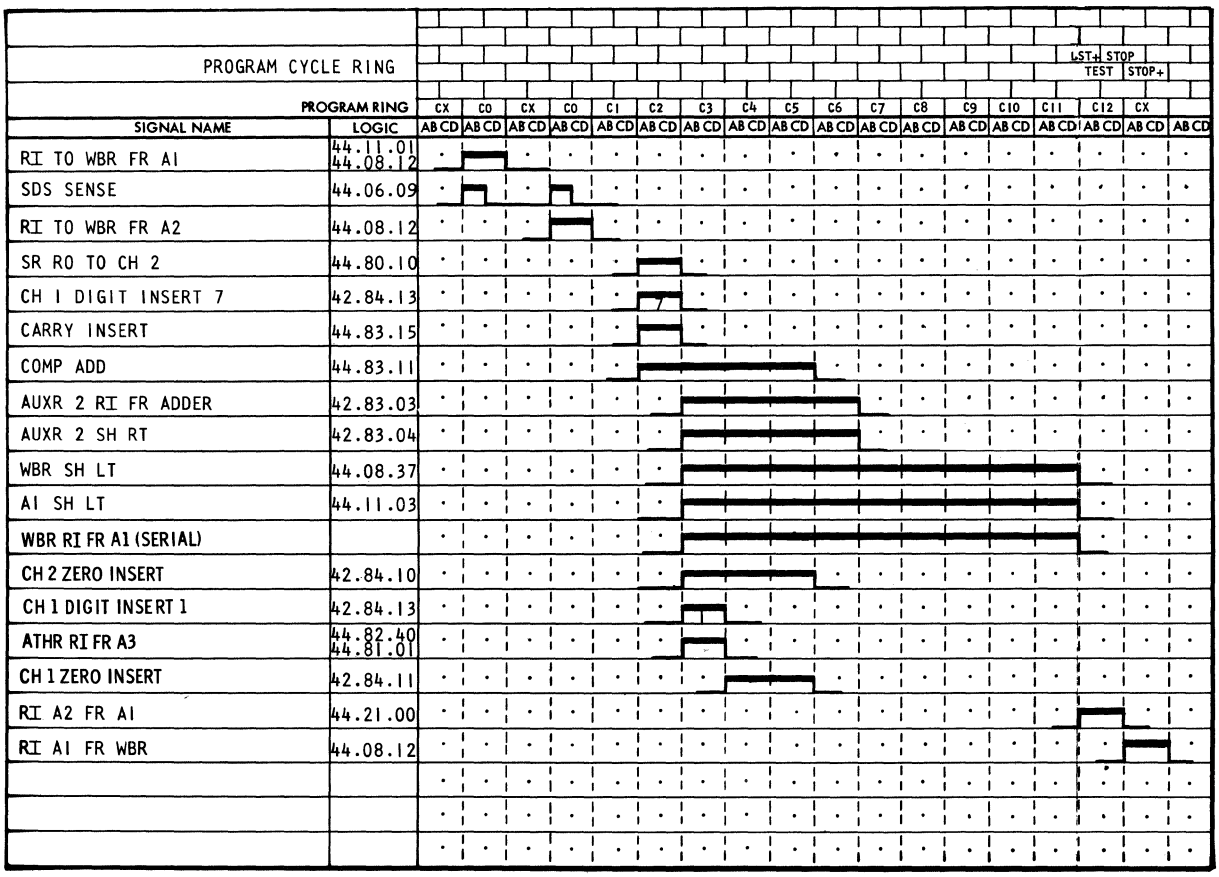


FIGURE 2.2-12 SUBCYCLE 5 - NORMALIZE 1ST SIGNIFICANT DIGIT IN A2

Objective	Timing	Logic and Location
Athr RI FR A3		
A3 AB RO	C2-3	4H-44.82.40
Athr RI AB	C2-3	4J-44.81.01

1st Significant Digit in A2

Objective	Timing	Logic and Location
RI A2 FR A1		
A1 AB RO	PC Last +	3E-44.11.01
A2 AB RI	PC Last +	3B-44.21.30
RI A1 FR WBR		
WBR AB RO	PC Stop +	
A1 AB RI	PC Stop +	
Ch 1 Dig Ins 1	C3	2H-42.84.13
Ch 2 Dig Ins 7	C2	2B-42.84.13

Update A1 Exponent (Figure 2.2-13)

At the beginning of subcycle 6, the exponent of the larger factor is in the arithmetic register positions 0-1. This number must be reduced by an amount equal to the number in Auxiliary Register 2 to compensate for the normalization operation.

The arithmetic register positions 0-1 are read out onto Channel 1 by FP Stop of the previous cycle. Simultaneously, Auxiliary Register 2 is read out onto Channel 2 under complement-add control. The results of the addition are read into Accumulator 1 positions 0-1 and Program Register D at CX-0 time. Accumulator 2 is shifted right two places one digit time later. The digits shifted out of Accumulator 1 are read into Accumulator 2 positions 0-1.

Overflow and underflow tests are made on the updated exponent. The adder output latches are sensed at C1-2 time (44.92.01) for either condition. If the underflow condition exists, the adder output latches will contain 9 at this time. Auxiliary Register 2 contains, at most, two digits prior to the updating A1 exponent subcycle. The third digit to be entered into the adder will be a 9 (complemented 0). The adder output latches are sampled at the end of the third digit time for the existence of the nine digit. During all cases except when an exponent less than 00 is generated, a carry is produced which adds to the nine producing a zero. An attempt to produce an exponent less than 00 fails to generate the carry and the 9 digit appears in the adder output latches indicating an underflow.

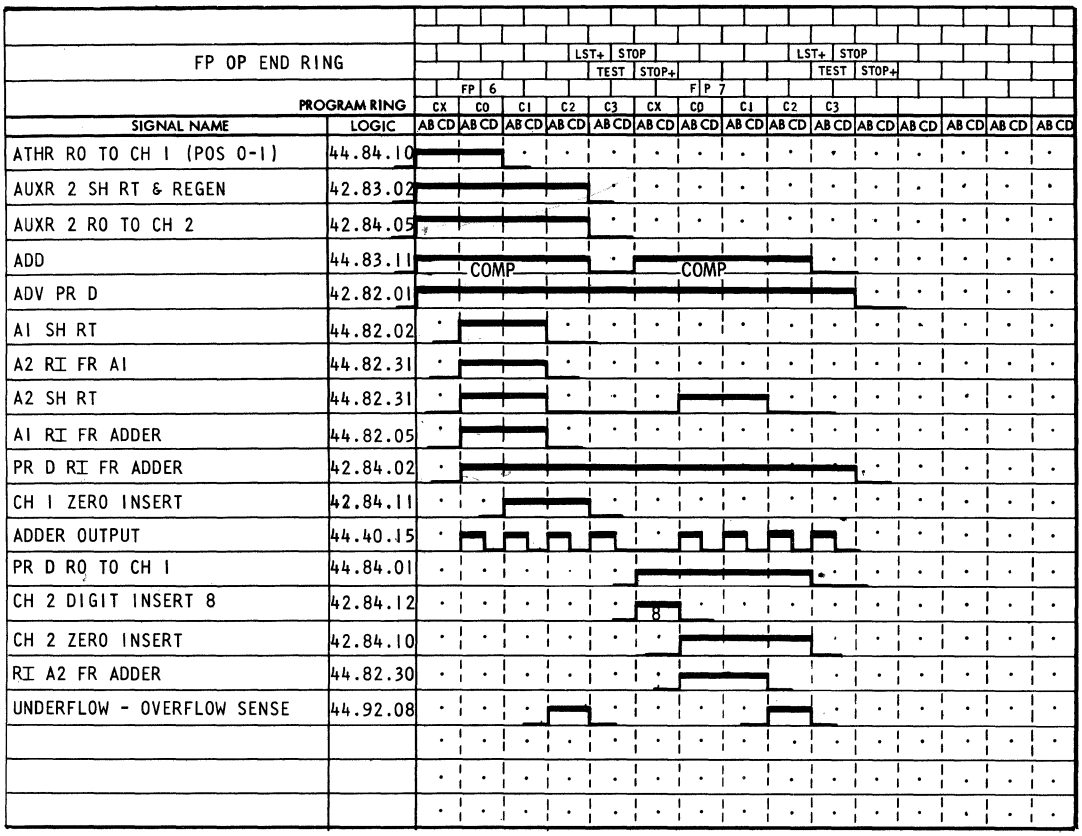


FIGURE 2.2-13 SUBCYCLE 6 - UPDATE A1 EXPONENT
SUBCYCLE 7 - UPDATE A2 EXPONENT

The overflow condition causes a 1 digit to appear in the adder latches at C1-2 time corresponding to the third digit time.

Objective	Timing	Logic and Location
Athr to Ch 1 (Pos 0-1)	FP Stop	4D-44 84.10
Adv PR D	FP Stop	4B-42.82.01
Aux 2 Sh RT & Regen	FP Stop	4B-42.83.02
Auxr 2 RO to Ch 2	FP Stop	4B-42.84.05
Comp Add	FP Stop	4C-44.83.11
A1 RI FR Adder	CX-0	4B-44.82.05
Stt A2 RI FR A1	CX-0	4B-44.82.31
A1 Sh RT	CX-0	4B-44.82.02
PR D RI FR Adder	CX-0	4B-42.82.02
A2 Sh RT	C0-1	4B-44.82.31

Objective	Timing	Logic and Location
Ch 1 Zero Ins	C0-1	5A-42.84.11
Stop Athr RO to Ch 1	C0-1	4J-44.84.10
Underflow-Overflow Sense	C1-2	4B-44.92.01
Stop A2 RI FR A1	C1-2	4G-44.82.31
Stop Aux 2 Regen RI	C2-3	4F-42.83.02

Update A2 Exponent (Figure 2.2-13)

The updated Accumulator 1 exponent in Program Register D is reduced by 8 and inserted in Accumulator 2 positions 0-1 during subcycle 7. Program Register D is read out onto Channel 1 by the output of the half-time latch of the floating ring, FP 6-7. An eight is inserted at CX time on Channel 2 under complement-add control. Zeros are inserted for the remaining Channel 2 entries. The resulting "2" (tens complement of 8) added to the PR D entry and the remaining 9 (complement 0) entries cause the original count in the auxiliary register to be reduced by eight.

The updated Accumulator 2 exponent is then read into Accumulator 2 at CX-0 time. A2 is shifted right by the Arith Op -FP 7 signal to make room for the exponent.

Objective	Timing	Logic and Location
PR D RO to Ch 1	FP 6-7	4G-44.84.01
Ch 2 Digit Ins 8	CX	2A-42.84.12
Comp Add	FP Stop	4E-44.83.11
Ch 2 Zero Ins	CX-0	4B-42 84.10
A2 Sh RT	Arith Op-FP 7	4C-44.82.30
A2 RI FR Adder	CX-0	4C-44.82.30
PR D RI FR Adder	FP6-CX-0	2C-42.82.02
Ch 2 Zero Ins	CX-0	4B-42.84.10
Stop PRD RO to Ch 1	FP Last +	4J-44.84.01
Underflow Sense	Arith Op-FP 7	4B-44.92.08

End Data Operation

The normal Stt End Data Operation is generated at C2 of the last subcycle (FP 7).

An end-data operation may, however, be initiated earlier than the last subcycle for the case of exponent difference equal to or greater than 16, as determined during compare.

Objective	Timing	Logic and Location
Stt End Data Op	FP 7 - C2	3F-42.80.25
End Data Op	Exp Diff Equal to or Greater than 16	

3.0.00 FLOATING RESET ADD + 75

The floating reset add instruction requires three subcycles of the basic data cycle. The operations to be preformed include reading in and storing the exponent, normalizing the mantissa, and adjusting the original exponent value.

FRA, the output of the Op Matrix is switched with FP 1 from the floating ring to initiate the first operation. The floating ring is set to position 1 as usual by a Data Control 4 pulse.

Objective	Timing	Logic and Location
Set Floating Ring to Position 1	DC 4	4J-42.80.04

The following objectives are required to perform the floating-reset-add operation.

A. Read In and Store Exponent

B. Normalize

C. Update Exponent of Accumulator 1

D. End Data Operation

Read In and Store Exponent (Figure 3.0-1)

The read-in operation reads the addressed factor into the arithmetic register and transfers it to Accumulator 1. Accumulator 1 is alternately shifted left, then right with zeros being entered, at the low order and high order, respectively. This operation assures that the positions normally occupied by the exponent are zero prior to attempting the normalization subcycle.

The process is shown as follows (the digit positions 2-9 are used to represent the mantissa):

	S	0	1	2	3	4	5	6	7	8	9
ZI LT Sh	2	3	4	5	6	7	8	9	0	0
ZI RT Sh	0	0	2	3	4	5	6	7	8	9

The original exponent is stored by transferring it from the arithmetic register to Accumulator 3 at C1 time.

The floating ring is advanced to FP 5 to set up the normalization operation.

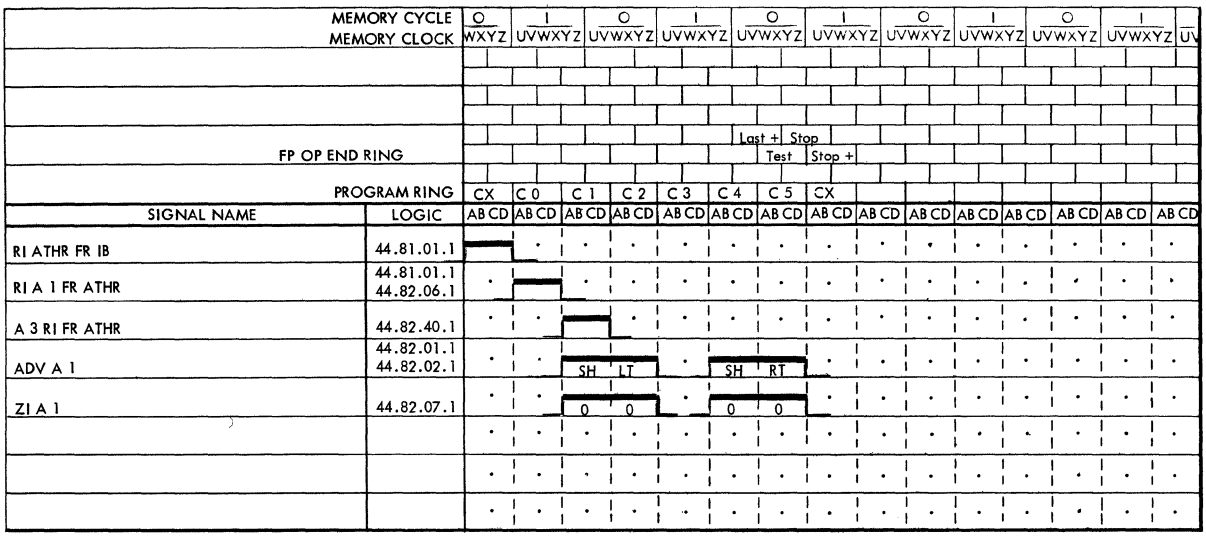


FIGURE 3.0-1 FLOATING RESET ADD - SUBCYCLE 1 - FP 1 READ IN AND STORE EXPONENT

Objective	Timing	Logic and Location
Set FP Ring Psn 5	FRA-FP 1 Adv	4F-42.80.05
RI Athr FR Mem	DC-4	6J-44.81.01.1
RI A1 FR Athr		
Athr RO AB	CX-0-FRA	4C-44.81.01.1
A1 RI AB	CX-0-FRA	4F-44.82.06.1
RI A3 FR Athr		
Athr RO AB	FRA-C0-1	
A3 RI AB	FRA-Co-1	4C-44.82.40.1
Sh A1 Sh Lt	FRA 1-C0-1	4A-44.82.01.1
Stop A1 Sh Lt	FRA 1-C2-3	4E-44.82.03.1
Stt A1 Sh RT	FRA 1-C2-3	5C-44.82.02.1
Stop A1 Sh RT	FRA 1-FP Stop	5F-44.82.02.1
ZI A1	FRA 1	4D-4G-44.82.07.1

Normalize (Figure 3.0-2)

The program ring is recycled in the CX position at the conclusion of the first subcycle. Fixed point double-shift-and-count is turned on by FP 5 (Logic 42.86.13). Because the normalization operation is the same as for floating add, FAS 5 is forced on.

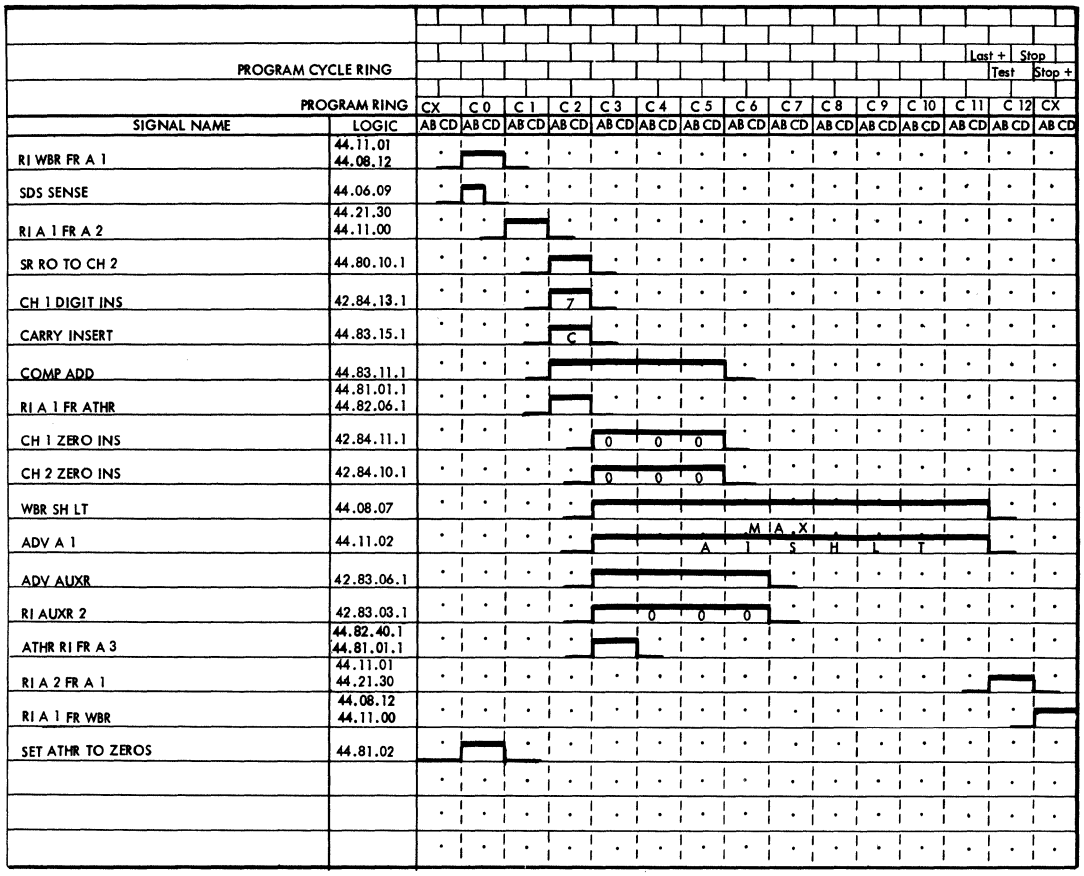


FIGURE 3.0-2 FLOATING RESET ADD - SUBCYCLE 2 - FP 5 NORMALIZE

Objective	Timing	Logic and Location
FAS 5	FRA-FP 5	4B-42.86.09.1

The normalization operation proceeds as usual with Accumulator 1 transferred to the word buffer register at C0. The significant digit scanner is turned on at C0 and transmits the location of the high-order digit to the shift register. The shift register reads out to its output latches in nines complement form. The resulting indication in the shift register is then read onto Channel 2 gated by complement add and carry insert. The Channel 1 digit insert injects a 7 at C2 time. The result of the addition is read into auxiliary register 2 low order to be used in the succeeding operation.

Objective	Timing	Logic and Location
RI WBR FR A1		
A1 AB RO	CX-0	2D-44.11.01
WBR AB RI	CX-0	3A-44.08.12
SDS Sense	C0-1	2A-44.06.09

Objective	Timing	Logic and Location
ZI Athr	FRA 5	4C-44.81.02
RI A1 FR A2		
A2 AB RO	C1-2	3F-44.21.30
A1 AB RI	C1-2	4E-44.11.00
RI A1 FR Athr		
Athr AB RO	FRA 5-C1-2	4D-44.81.01.1
A1 AB RI	FRA 5-C1-2	4C-44.82.06.1
SR RO to Ch 2	C2	4E-44.80.10.1
Ch 1 Digit Ins 7	C2	2B-42.84.13.1
Carry Ins	C2	4E-44.83.15.1
Comp Add	FP5-6 FP Stop	5D-44.83.11.1
Ch 1 Zero Ins	C3-4	5D-42.84.11
Ch 2 Zero Ins	C1-2	4E-42.84.10.1
RI Auxr	C2-3	4B-42.83.03
WBR Sh Lt	C2-3	44.08.07
Adv A1 (A1 Lt Sh)	C3-4	5G-44.11.02
Athr RI FR A3		
Athr AB RI	C2-3	4H-44.82.40
A3 AB RO	C2-3	4J-44.81.01
RI A2 FR A1		
A1 AB RO	PC Last +	3E-44.11.01
A2 AB RI	PC Last +	3B-44.21.30
RI A3 FR Athr		
Athr AB RO	PC Stop	44.08.12
A3 AB RI	PC Stop	2F-44.11.00

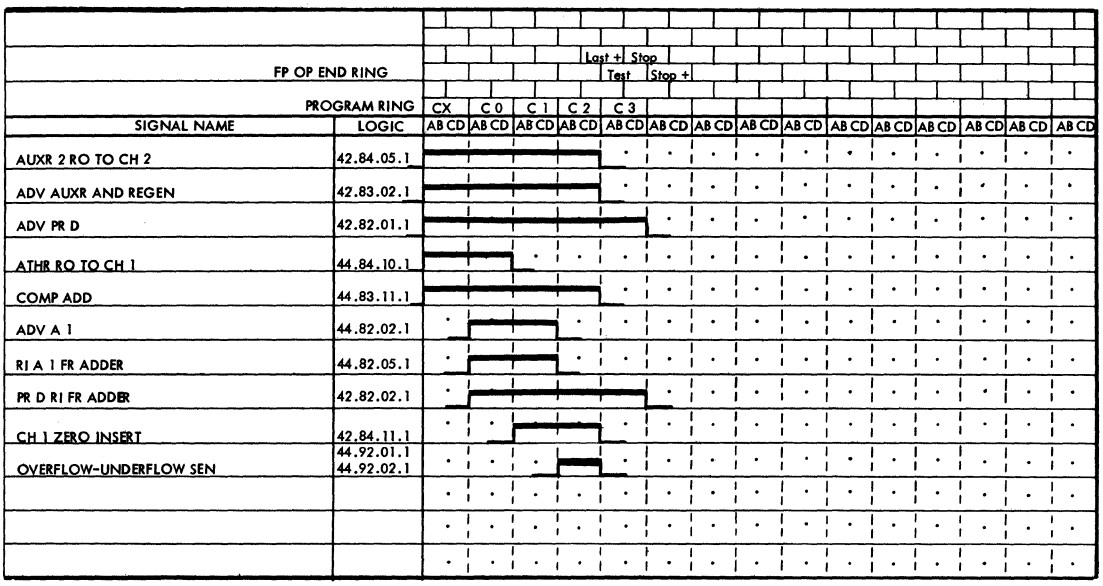


FIGURE 3.0-3 FLOATING RESET ADD - SUBCYCLE 3 - FP 6 UPDATE ACCUMULATOR 1 EXPONENT

Update A1 Exponent (Figure 3.0-3)

This subcycle causes the Auxiliary Register 2 contents to be subtracted from the original exponent now in the arithmetic register positions 0-1. Auxiliary Register 2 low-order position contains the digit specifying the amount by which the original exponent must be reduced to compensate for the normalization cycle.

FAS 6 is forced on to operate the registers as in the normal floating-add operation:

Objective	Timing	Logic and Location
FAS 6	FRA-FP 6	4G-42.86.09.1

The operation reads out the arithmetic register positions 0-1 to Channel 1 starting at CX time. Simultaneously Auxiliary Register 2 is read out onto Channel 2. The entries are gated by complement add and carry insert. Auxiliary Register 2 is regenerated as it is advanced. The result of the complement add operation is read into A1 starting at C0. A1 is shifted right as each digit is entered into the high order.

Program Register D is also advanced at CX time. The result of the operation also enters Program Register D at C0 time. Thus, at the end of the updating operation, the updated exponent is contained in A1 and Program Register D.

The underflow-overflow test is made on the exponent of Accumulator 1 at C2 time of FP 6. This operation is described in 2.2.00.

Objective	Timing	Logic and Location
RO Auxr (Entry B)	FP Stop	4B-42.84.05.1
Adv Auxr and Regen	FP Stop	4B-42.83.02.1
Athr RO to Ch 1 (Pos 0-1)	FP Stop	4D-44.84.10.1
Comp Add	FP 5-6 FP Stop	5D-44.83.11.1
Stt A1 Sh RT	CX-0	4B-44.82.02.1
Stop A1 Sh RT	C1-2	4G-44.82.02.1
RI A1 FR Adder	CX-0	4B-44.82.05.1
Adv PR D	FP Stop	4B-42.82.01.1
PR D RI FR Adder	Arith Op-FP 6-CX-0	2C-42.82.02.1
Ch 1 Zero Insert	C0-1	5A-42.84.11
Overflow-Underflow Sense	C2	44.92.01-44.92.02

End-Data Operation

The end-data operation occurs at C2 of FP 6.

Objective	Timing	Logic and Location
Stt End Data Op	C2-FP 6 FRA	3G-42.80.25.1

4.0.00 FLOATING ADD DOUBLE +76
FLOATING ADD DOUBLE SUPP NORM -76

The Floating Add Double and Add Double Suppress Normalization instructions are executed in the same way with one exception. Floating Add Double Suppress Normalization does not normalize the result of the addition in Accumulators 1 and 2.

The two instructions require six subcycles. The operation is similar to floating add with two exceptions:

1. Accumulator 2 is not reset to all zeros because it is a factor in the operation.

2. The initial comparison of the two factors is not for the purpose of placing the smaller factor in the arithmetic register. Instead, the comparison determines that the addressed factor exponent is equal to or greater than the exponent of the factor in Accumulator 1. If it is not, an error signal causes the machine to halt.

The add-double instruction differs little from the normal floating add. The operation effectively causes a serial shift right of Accumulators 1 and 2. The number of positions shifted depends upon the exponent difference.

In practice, the right shift is not accomplished. Instead Accumulator 1 is transferred to the ~~arithmetic register~~ AUX REG. The factor in A1 is then scanned into the high order of Accumulator 2 which simultaneously shifts right each digit time. The digits shifted out of the low order of A2 are lost. Thus, the number of digits lost is equal to the exponent difference. The digits remaining in A1 are then added to the factor from core storage as in the normal floating add. The operation is illustrated for two factors with an exponent difference of 6 as follows:

ATHR (Core Storage Factor)

S	0	1	2	3	4	5	6	7	8	9
+	5	7	6	4	4	3	2	9	1	7

A1									A2								
S	0	1	2	3	4	5	6	7	8	9	S	0	1	2	3	4	5	6	7	8	9
+	5	1	3,	8	6	4	9	1	3	6	+	4	3	7	1	4	9	0	3	5	6

Add to A2 (Shift A1 and A2 Right 6 places)

A1									A2								
S	0	1	2	3	4	5	6	7	8	9	S	0	1	2	3	4	5	6	7	8	9
								3	8					6	4	9	1	3	6	7	1

Add to A1

	S	0	1		2	3	4	5	6	7	8	9
Athr	+	5	7		6	4	4	3	2	9	1	7
A1											3	8

A2 6 4 9 1 3 6 7 1

Final Result A1 A2

	+	5	7		6	4	4	3	2	9	5	5	
A2											4	9	
											6	4	9 1 3 6 7 1

During a complement-add operation when the exponent difference is 0-7, it is necessary to complement the contents of Accumulator 2 to obtain the correct result. A separate machine cycle is used to accomplish this function.

FAD - Complement Add - Exp Diff of 3

ATHR (Core Storage Factor)

+	5	4		7	4	3	8	2	1	2	4

A1 A2

	S	0	1	2	3	4	5	6	7	8	9		S	0	1	2	3	4	5	6	7	8	9
	-	5	1	3	2	1	9	7	4	1	3			4	3	5	1	9	6	1	2	4	4

Add to A2

A1 A2

	S	0	1	2	3	4	5	6	7	8	9		S	0	1	2	3	4	5	6	7	8	9
	-					3	2	1	9	7								4	1	3	5	1	9	6	1

Add to A1

	S	0	1	2	3	4	5	6	7	8	9		S	0	1	2	3	4	5	6	7	8	9
Athr	+			7	4	3	8	2	1	2	4														
A1	-					3	2	1	9	7		A2						4	1	3	5	1	9	6	1

Result:

A1 A2

	+			7	4	3	4	9	9	2	6							4	1	3	5	1	9	6	1

Complement A2

A2 (10's Complement)

S 0 1 2 3 4 5 6 7 8 9

5 8 6 4 8 0 3 9

Final Result:

A1		A2	
S 0 1	2 3 4 5 6 7 8 9	S 0 1	2 3 4 5 6 7 8 9
+ 5 4	7 4 3 4 9 9 2 6	+ 4 6	5 8 6 4 8 0 3 9

The floating ring is set to position 1 by a Data Control 4 pulse. The Op Compare Abs function is forced on similar to the other floating-point arithmetic operations.

Objective	Timing	Logic and Location
Op Compare Abs		
Set Floating Ring PSN 1	DC 3A	4J-42.80.04.1
Arith Op - FP 1	FP 1- FP Add-Sub't	4A-42.86.15.1
Op Compare ABS	Arith Op-FP 1	5A-42.86.12.1
Op A1	Arith Op-FP 1	3F-42.86.12.1

The following functions must be accomplished during the data cycle to perform the add-double and add-double-suppress normalization operations:

A. Compare Exponents

B. Add to A2

C. Add to A1

D. Complement A2 (On Complement Add if Exponent difference is 0-7)

E. Normalize (FAD Only)

F. Update A1 Exponent

G. Update A2 Exponent

H. End Data Operation

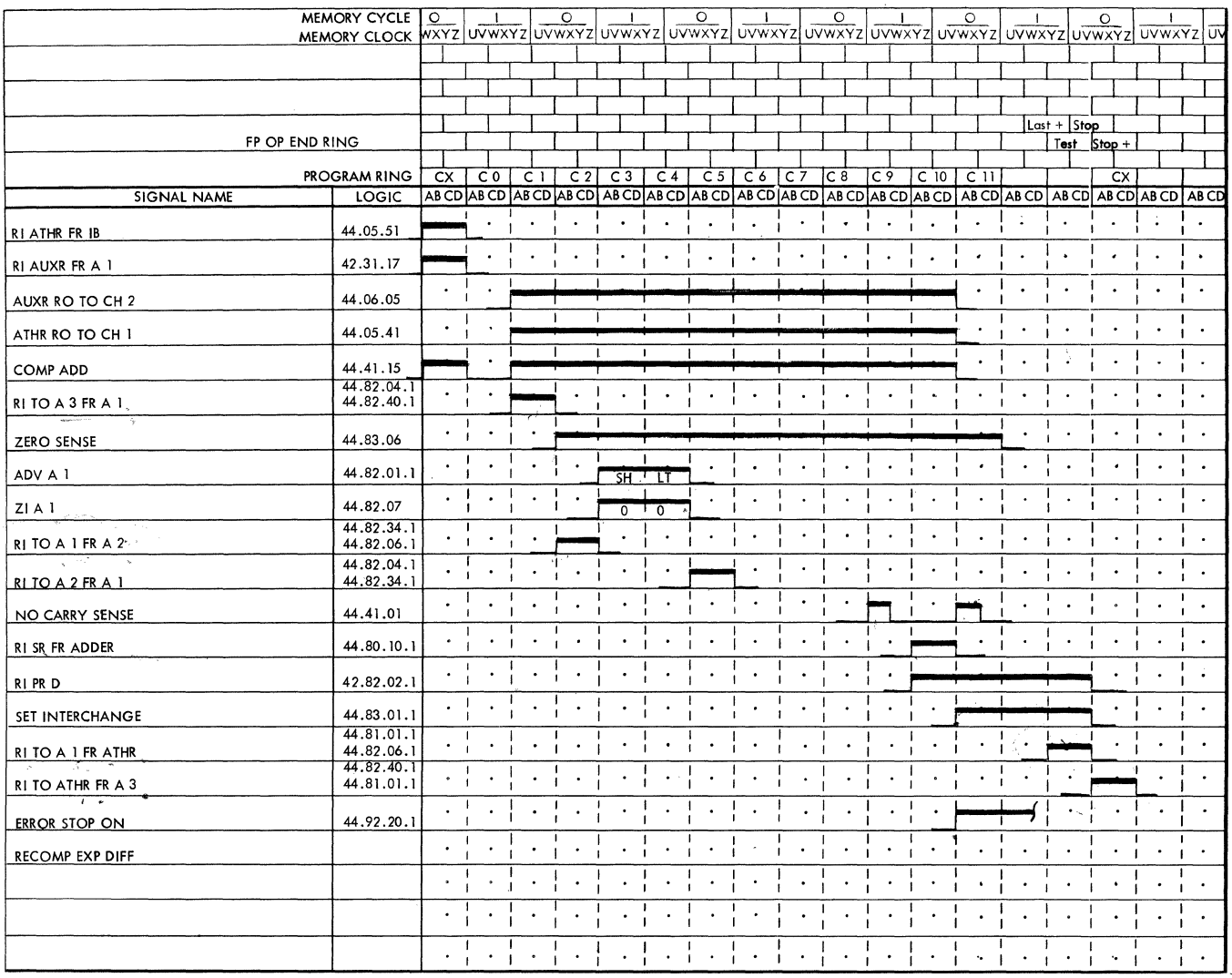


FIGURE 4.0-1 FLOATING ADD DOUBLE SUBCYCLE 1 - COMPARE

Compare Exponents (Figure 4.0-1)

The compare function senses the value of the exponent difference. The exponent of the addressed factor in the arithmetic register must be greater than or equal to the exponent of the factor in Accumulator 1.

Because there is no interchange of factors possible with add double, the result of the mantissa comparison is ignored. The exponent difference is read into the shift register and Program Register D as in a normal add. The information in the shift register output latches is used to set up the field register for the Add to Accumulator 2 subcycle in the usual manner (see 2.2.00).

The operation follows the same sequence as Floating Add except that the recomplement subcycle of floating add is not required.

The exponent of Accumulator 2 is set to zero during the compare operation. This is accomplished by transferring Accumulator 2 to Accumulator 1 at C2 time. A1 is then shifted left for two digit times. Zeros are read in at the low order. The register is then transferred back to A2 at C5 time. This operation sets up A2 for the Add to A2 subcycle.

Objective	Timing	Logic and Location
RI Athr IB	DC 4	4E-44.05.51
RI Auxr FR A1	DC 4	3C-42.31.17
Fld Reg to Adder (Op FCX)		
RO Fld Reg	CX	3B-44.02.23
No-Carry Ins	CX-AP	4C-44.41.02
Comp Add	CX	6E-44.41.15
RI Fld Len Reg	C0	4G-44.02.23
RO Auxr (Entry B)	C1-SDS Check	5C-44.06.05
Adv Athr (Entry A)	C1	4A-44.05.41
Comp Add	C1	2B-44.41.15
RI A3 FR A1		
A1 AB RO	C0-1	4C-44.82.04
A3 AB RI	C0-1	4B-44.82.40
RI A1 FR A2		
A2 AB RO	C1-2	4H-44.82.34.1
A1 AB RI	FP1-FAD	4E-44.82.06.1
Zero Sense	C9 + C11	44.83.06.1
Adv A1 (Sh Lt)	C2-3 - Any FAD	4B-44.82.01.1
ZI A1	C2-3	
RI A2 FR A1		
A1 AB RO	C4-5	4D-44.82.04.1
A2 AB RI	C4-5	4F-44.82.34.1

Objective	Timing	Logic and Location
Error Stop	FP 1 - Any FAD FP Recomp	4C-44.92.20.1
No-Carry Sense	C9 + C11	44.41.01
RI PR D	C9-10	2A-42.82.02.1
RI SR FR Adder	C10	4A-44.80.10.1
Set Interchange	C11 - Carry 1st	44.83.01
RI A1 FR Athr		
Athr AB RO	FP Last +	4B-44.81.01.1
A1 AB RI	FP Last +	4B-44.82.06.1
RI Athr FR A3		
A3 AB RO	FP Stop	4G-44.82.40.1
Athr AB RI	FP Stop	4G-44.81.01.1

Add to A2 (Figure 4.0-2)

This subcycle, as in normal floating add, causes the contents of the arithmetic register to be scanned into Accumulator 2. Accumulator 1 is transferred to the arithmetic register during the compare operation. The field register is loaded as outlined in the description of floating add (Section 2.2.00).

The arithmetic register is scanned into the adder on Channel 2 under true-add control. Zeros are fed for the channel 1 entries. The result is read into the high order of A2 with a right shift each digit time.

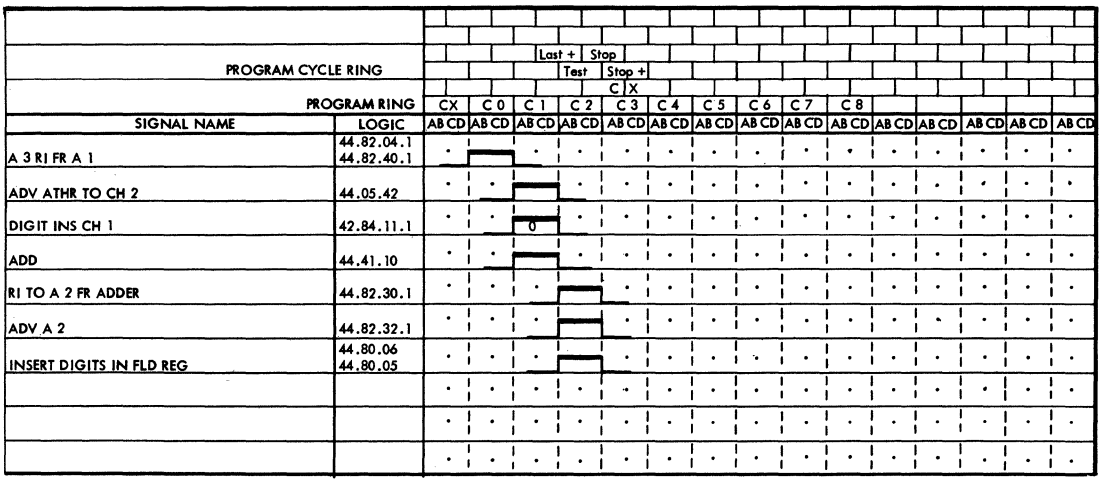


FIGURE 4.0-2 FLOATING ADD DOUBLE SUBCYCLE 2 - ADD TO A2 - EXP DIFF OF 1

Objective	Timing	Logic and Location
Insert Digits in Fld Reg	PC Test	44.80.05
RO Fld Len Reg	CX	3B-44.02.23
RI Fld Reg	C0 BP	4G-44.02.23
A3 RI FR A1		
A1 AB RO	CX-0	4F-44.82.04.1
A3 AB RI	CX-0	4F-44.82.40.1
Adv Athr (Entry B)	C1	4B-44.05.42
Digit Ins (Entry A)	C0-1	4B-42.84.11.1
Add	C1	44.41.10
Adv A2	C1-2	4D-44.82.32.1
RI A2 FR Adder	C1-2	4D-44.82.30.1

Add to A1 (Figure 4.0-3)

This subcycle functions for double add just as it does for normal floating add. The true-add or complement-add operation depends upon a sign analysis. On complement-add a no-carry insert is generated to give a nines complement so that the units position of Accumulator 1 is one less than usual. This function gives the borrow required during the operation.

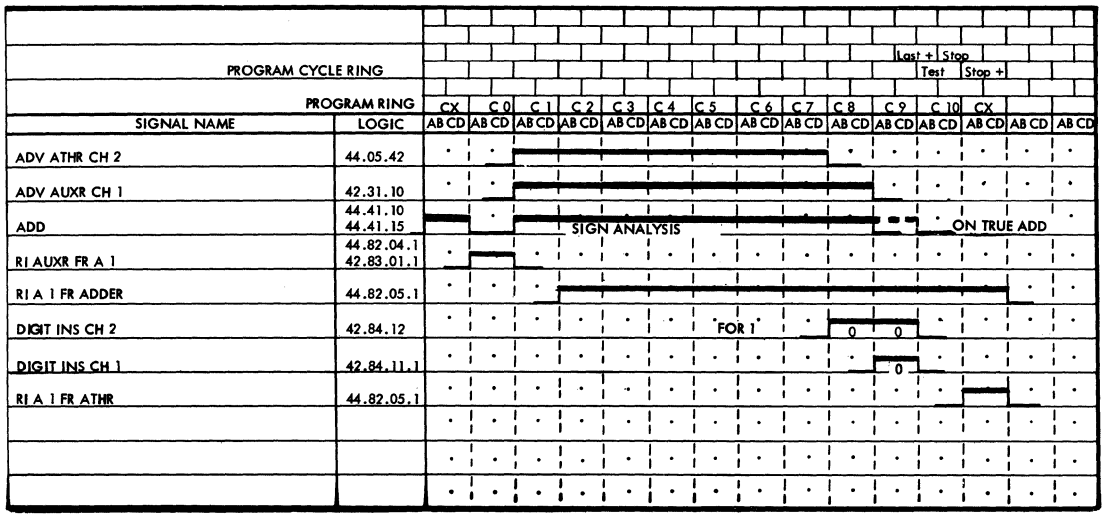


FIGURE 4.0-3 FLOATING ADD DOUBLE SUBCYCLE 3 - ADD TO A1 - EXP DIFF OF 1

Objective	Timing	Logic and Location
RI Auxr FR A1	CX-0 CX-0	4C-44.82.04.1 4B-42.83.01.1
Adv Athr (Entry B)	C1	4B-44.05.42
Adv Auxr (Entry A)	C1	4D-42.31.10
Add (Sign Analysis)	C1	44.41.10-44.41.15
RI A1 FR Adder	C1-2	4D-44.82.05.1
Digit Ins Ch 2 (for 1)	C7-8	42.84.10
Digit Ins Ch 1	C8-9	5F-42.84.11.1
RI A1 FR Athr	C1-2	4D-44.82.05

Complement A2 (Figure 4.0-4)

If the exponent difference equals 0-7 on a complement add operation, the floating ring is set to position 2 after completing the Add to A1 function. The Program Cycle Ring is recycled in the CX position. Accumulator 2 is then run through the adder under complement-add control to obtain the correct result. The correct result is simply the tens complement of the contents of Accumulator 2 for this operation.

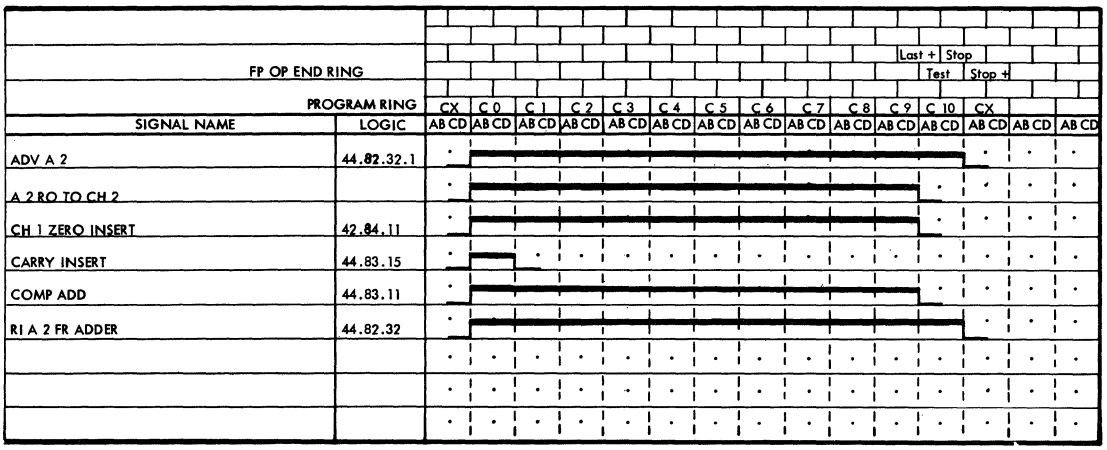


FIGURE 4.0-4 FLOATING ADD DOUBLE SUBCYCLE 4A - COMPLEMENT A2 - PERFORM THIS CYCLE ONLY FOR COMPLEMENT - ADD OPERATION EXP DIFF 0-7

Objective	Timing	Logic and Location
Set Floating Ring to PSN 2	Comp Add Exp Diff 0-7	42.80.06.1
Adv A2	CX-0	44.82.32
A2 RO to Ch 2	CX-0	

Objective	Timing	Logic and Location
Insert Zero Ch 1	CX-0	42.84.11
Carry Insert	CX-0	44.83.15
Comp Add	CX-0	44.83.11
RI to A2 FR Adder	CX-0	44.82.32

If a true-add instead of a complement-add occurs, the floating ring is advanced directly to position 5.

Normalize

In this subcycle, the only difference between FAD and FADSN occurs during FADSN. If the result of the addition has carried into position 1, Accumulator 1 is shifted one place to the left. Otherwise, Accumulator 1 is shifted left two places. This prepares the accumulator to receive the exponent in the succeeding subcycle, Update A1.

Objective	Timing	Logic and Location
Adv A1 (Sh Lt)	FADSN - CX-0 FP 5	4E-44.82.01.1

For FAD, normalization proceeds as it does for normal floating add-sub't.

Update A1 Exponent and Update A2 Exponent

These two subcycles are accomplished in exactly the same manner as for the floating add-sub't codes.

End-Data Operation

The end-data operation is initiated at C2 of FP 7.

The floating-multiply operation forms a sixteen-digit product from an eight-digit multiplicand and an eight-digit multiplier.

The operation adds the exponents of the two factors and performs a fixed-point multiplication on the two mantissas. The resulting product is then normalized and the exponents of Accumulators 1 and 2 updated to conclude the operation.

The floating-multiply instruction utilizes five subcycles of the basic data cycle. FM from the Op matrix is switched with FP 1, the output of the floating ring, giving FM 1.

Similarly, FM is switched with FP 4, FP 5, FP 6 and FP 7, in turn, to define and control the operations occuring within these subcycles. The operation first forces on Op Compare Abs and Op A1 as in floating add-sub't.

Objective	Timing	Logic and Location
Op Compare Abs		
Set Floating Ring PSN 1	DC 3A	4J-42.80.04.1
Arith Op FP 1	FP 1 FM	4A-42.86.15
Op Compare Abs	Arith Op FP 1	5A-42.86.12
Op A1	Arith Op-FP 1	3F-42.86.12

The following objectives are required to execute the floating-multiply instruction.

A. Add Exponents - Store in Program Register D.

B. Multiply Mantissas

C. Normalize

D. Correct Exponent-Insert in A1

E. Correct A1 Exponent-Insert in A2

F. End Data Operation

Compare (Figure 5.0-1)

For the multiply operation the Compare Abs function adds the exponents of the two factors and reads the result into Program Register D. A comparison as such is not actually made of the two factors.

The addressed factor is read into the arithmetic register at CX time. Simultaneously, Accumulator 1 is read into the auxiliary register. At C0 the arithmetic register is parallel-transferred to Accumulator 2. The latter operation performs the same function as the Reset Add to Accumulator 2 of fixed point divide.

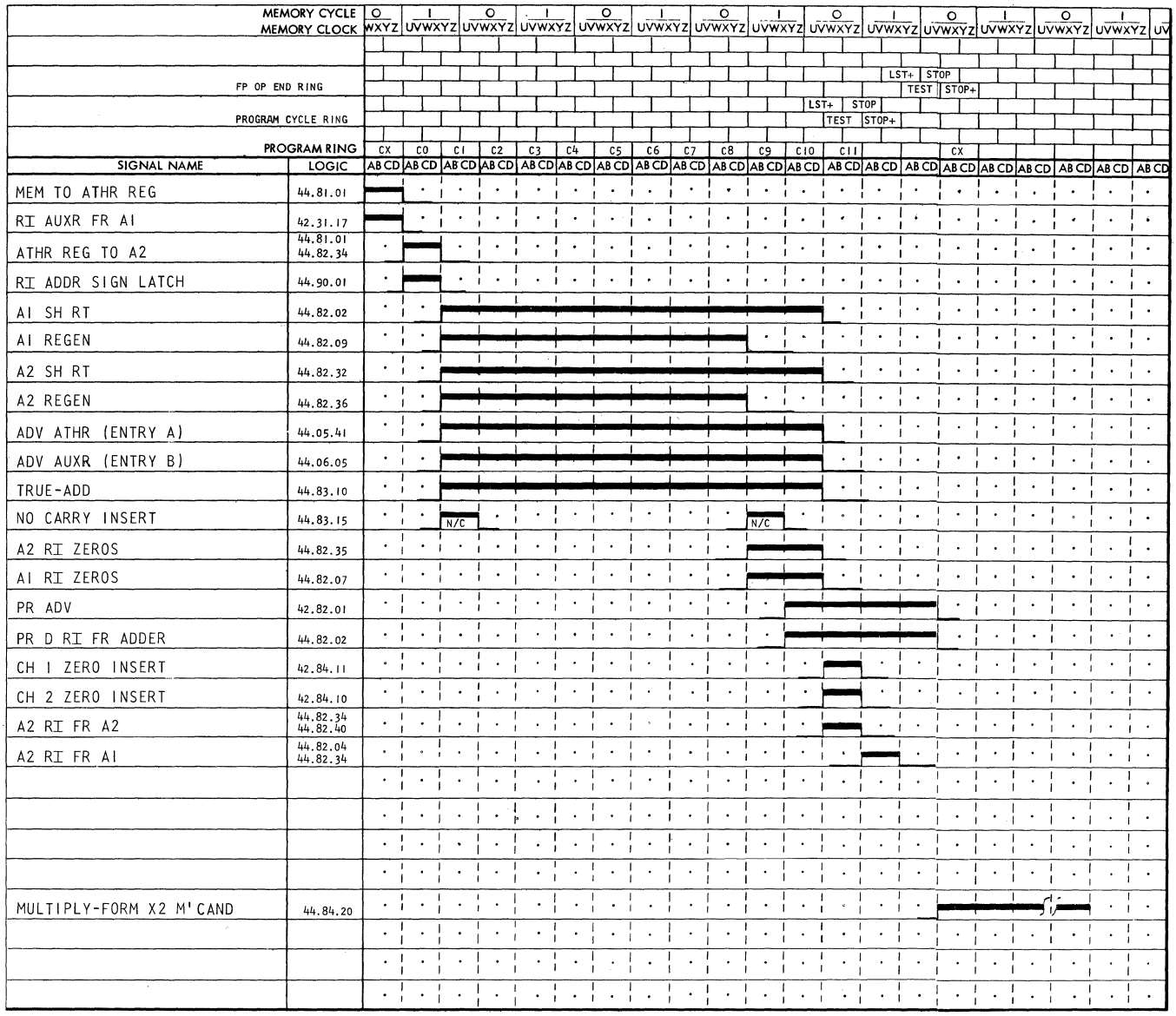


FIGURE 5.0-1 SUBCYCLE 1 - ADD EXPONENTS & STORE IN PR D FORCE FIXED POINT OP COMPARE ABS & OP A1
SUBCYCLE 2 - MULTIPLY FORCE FIXED POINT MULTIPLY

The sign of the addressed factor is read into the addressed sign latches at C0 during the transfer of the arithmetic register to Accumulator 2. The determination of the sign of the result is detailed in 5.2.01.

The field register is set for full field control (0-9) by a Data Control 4 pulse. The arithmetic register is read out to Channel 1 and the auxiliary register to Channel 2. The entries are gated by true-add and no-carry insert. The result of the addition for the first eight digits is ignored. At C9, the units position of the exponent is gated by a no-carry insert. The result of the addition of the exponents is read into Program Register D.

Also at C9 time, both Accumulators 1 and 2 have zeros read into positions 0-1. This operation insures that both factors start with two leading zeros.

The final operations occur at the end of Compare to set up the registers for a fixed-point multiplication. Accumulator 2 parallel-transfers its contents to Accumulator 3 at PC Last +. Accumulator 1 transfers to Accumulator 2 at PC Stop.

Objective	Timing	Logic and Location
Ins Dig in Fld Reg	DC 4	44.80.05.1-44.80.06.1
Mem to ATHR	DC 4	6J-44.81.01.1
RI to AUXR FR A1	DC 4	3C-42.31.17
ATHR to A2		
ATHR RO AB	CX-0	3A-44.81.01.1
A2 RI AB	CX-0	3B-44.82.34.1
RI ADDR Sign Latch	CX-0	4D-44.90.01.1
Adv ATHR (Entry A)	C1	4A-44.05.41
Adv AUXR (Entry B)	C1-SDS Check	5C-44.06.05
A1 Sh RT	C0-1	4C-44.82.02.1
A2 Sh RT	C0-1	4A-44.82.32.1
A1 Regen	C0-1	4B-44.82.09.1
A2 Regen	C0-1	4B-44.82.36.1
No-Carry Insert	C1 + C9	3D-4D-44.83.15.1
True-Add	FM 1 - C0-1	4D-44.83.10.1
A1 RI Zeros	C9 + C10	4A-4B-44.82.07.1
A2 RI Zeros	C8-9	4C-44.82.35.1
Stop A1 Regen	C8-9	4F-44.82.09.1
Stop A2 Regen	C8-9	4F-44.82.36.1
PR Adv	C9-10	4A-42.82.01.1
PR RI FR Adder	C9-10	2A-42.82.02.1
Ch 1 Zero Insert	C10-11	4C-42.84.11.1
Ch 2 Zero Insert	C10-11	4G-42.84.10.1
A3 RI FR A2		
A2 AB RO	PC Last +	3H-44.82.34.1
A3 AB RI	PC Last +	4A-44.82.40.1

Objective	Timing	Logic and Location
A2 RI FR A1		
A1 AB RO	PC Stop	4A-44.82.04.1
A2 AB RI	PC Stop	2B-44.82.34.1

Multiply Mantissas

Fixed point multiply is turned on following the addition of the exponents.

Objective	Timing	Logic and Location
FP Stt Mpy	FP 1-2, FM	4B-44.84.20.1

The multiply operation is turned on beginning at the Form X 2 Multiplicand cycle. The Reset Add to A2 was executed during Compare. The remainder of the operation is described in Section 5.2.01.

Because each factor has two leading zeros, the product of the multiplication has either four or five high-order zeros. How this affects the exponent is indicated in the description of the succeeding subcycle.

At the completion of the fixed point multiply, an end-multiply signal resets the floating ring (Logic 42.80.04).

Normalize (Figure 5.0-2)

During normalization, the significant digit scanner is set with the location of the high-order digit of Accumulator 1 in the usual way by the forced double-shift-and-count operation. The SDS transfers this count to the shift register. The shift register reads out in nines complement form to its output latches. The latch outputs are read out onto Channel 2 under complement-add control. Because the original factors each had two leading zeros, the resulting product (in Accumulators 1 and 2) has either four or five high-order zeros. If the latch outputs specify a 5 digit (indicating four leading zeros), this digit becomes a three on Channel 2. The 4 added to the 6 on Channel 1 plus the carry insert yields a zero. The carry is ignored. The zero is read into Auxiliary Register 2 and specifies that 50 is to be subtracted from the exponent during the succeeding subcycle.

The operation is the same when five high-order zeros are sensed. A "1" digit however, results from the complement-add operation and is read into Auxiliary Register 2. This indication causes 51 to be subtracted during the following subcycle.

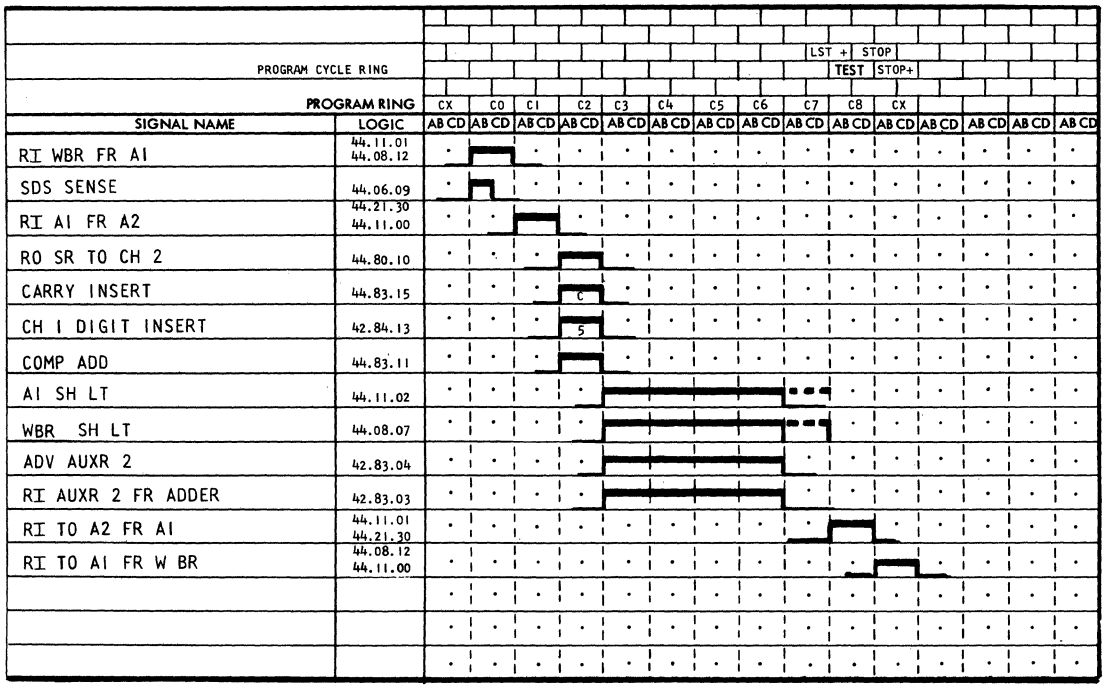


FIGURE 5.0-2 SUBCYCLE 3 NORMALIZE FORCE FIXED POINT DBL SH & COUNT

Objective	Timing	Logic and Location
RI WBR FR A1		
A1 RO AB	CX-0	2D-44.11.01
WBR RI AB	CX-0	3A-44.08.12
SDS Sense	C0-1	2A-44.06.09
RI to A1 FR A2		
A2 RO AB	C1-2	3F-44.21.30
A1 RI AB	C1-2	4E-44.11.00
RO SR to Ch 2	FP 4 Mpy-Add-C2	4E-44.80.10.1
Carry Insert	FP 5 Mpy-Add-C2	4E-44.83.15.1
Ch 1 Digit Ins "6"	FM 5-C2	2C-42.84.13.1
Comp Add	FP Mpy-Add C1-2	4B-44.83.11.1
Adv Auxr 2	C2-3	4A-42.83.04.1
RI Auxr 2 FR Adder	C2-3	4A-42.83.03.1

Objective	Timing	Logic and Location
WBR Sh Lt	C4	44.08.07
A1 Sh Lt	C3-4	5G-44.11.02
RI A2 Fr A1		
A1 AB RO	PC Last +	3E-44.11.01
A2 AB RI	PC Last +	3B-44.21.30
RI A1 FR WBR		
WBR AB RO	PC Stop	44.08.12
A1 AB RI	PC Stop	2F-44.11.00

Correct Exponent-Insert in A1 (Figure 5.0-3)

The function of the fourth subcycle of the multiply operation is to correct the original exponent and insert it in A1, positions 0-1.

The number of high-order zeros in Accumulator 1 is determined during the normalization subcycle. During subcycle 4 (FP 5), 50 is subtracted if 4 high-order zeros are sensed. If five high-order zeros are sensed, 51 is subtracted. Auxiliary Register 2 (low-order position) contains either "0" or "1" depending upon whether four or five high-order zeros are sensed.

At CX time, Auxiliary Register 2 is read out onto Channel 2 gated by complement-add and carry insert. Program Register D, which contains the sum of the original exponents, is read onto Channel 1 simultaneously. The result is read into Accumulator 1, positions 0-1, and Program Register D. Accumulator 1 is shifted right as the digits are read in. The two low-order digits of Accumulator 1 are serially read into the two high-order positions of Accumulator 2. As Accumulator 2 shifts right, the two low-order digit positions are lost.

The exponent is sensed for the overflow-underflow condition at C2 time. This is accomplished by sampling the adder output latches at C10 time. The operation is discussed in Section 2.2.00.

Subcycle 6 Adjust Exponent-Insert in A1

Objective	Timing	Logic and Location
Auxr 2 RO to Ch 2	FP 5-6 Mpy Add FP Stop	4B-44.84.05.1
Auxr 2 Sh RT	FP 5-6 Mpy Add FP Stop	4B-42.83.04.1
Auxr 2 Regen	FP 5-6 Mpy Add FP Stop	4B-42.83.02.1
PR D RO to Ch 1	FM FP 5-6	4E-44.84.01.1

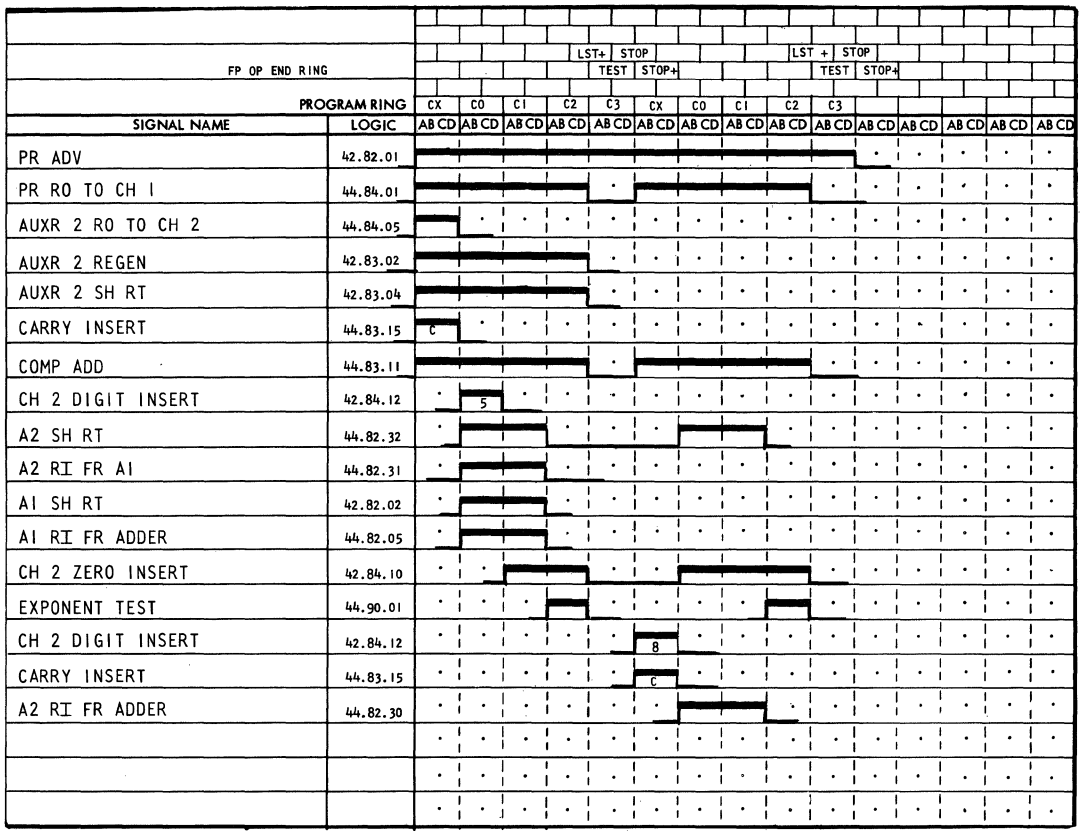


FIGURE 5.0-3 SUBCYCLE 4 CORRECT EXPONENT - INSERT IN A1
SUBCYCLE 5 CORRECT EXPONENT - INSERT IN A2

Objective	Timing	Logic and Location
Adv PR D	Arith Op FP 5-6	4D-42.82.01.1
Carry Ins	Arith Op FP 6 CX	4F-44.83.15.1
Ch 2 Digit Ins 5	FM 6-C0	2C-42.84.12.1
Comp Add	FP5-6	5D-44.83.11.1
PR RI FR Adder	Arith Op-FP6 CX-0	2C-42.82.02.1
A1 Sh RT	Arith Op FP6 CX-0	4B-42.82.02.1
A1 RI FR Adder	Arith Op FP6 CX-0	4B-44.82.05.1
A2 Sh RT	Arith Op FP6 CX-0	4B-44.82.32.1
A2 RI FR A1	CX-0	4B-44.82.31.1

Objective	Timing	Logic and Location
Stop A2 RI A1	C1-2	4G-44.82.31.1
Ch 2 Zero Ins	FM6 C0-1	4F-42.84.10.1
Underflow-Overflow Sense	C2	44.90.01.1-02.1
A2 RI FR Adder	Arith Op-FP7 CX-0	4C-44.82.30.1
Stop Auxr 2 Sh RT RI	C2-3	4G-42.83.04.1
Stop Auxr 2 Regen	C2-3	4F-42.83.02.1

Correct A1 Exponent-Insert in A2 (Figure 5.0-3)

The correcting or updating of the Accumulator 2 exponent, as with the other floating-point codes, it simply a matter of reducing the A1 exponent by 8. The value thus obtained is inserted in Accumulator 2, positions 0-1.

Program Register D which contains the exponent of A1 is read out onto Channel 1 at CX time. Simultaneously an "8" digit is inserted on Channel 2. The entries are gated by complement add with a carry insert. The result is read into A2, positions 0-1, and Program Register D. A2 is right shifted for two digit times to make room for the exponent. The two low-order digits of A2 are lost.

The exponent is tested for the underflow condition of C2 time.

Objective	Timing	Logic and Location
Ch 2 Digit Ins 8	CX	42.84.12.1
PR D RO to Ch 1	FP Mpy Add FP 6-7	4G-44.84.01.1
Carry Ins	Arith Op-FP 7	4H-44.83.15.1
Comp Add	Arith Op-FP 6-7	4E-44.83.11.1
A2 RI FR Adder	CX-0	4C-44.82.30.1
A2 Sh RT	CX-0	4C-44.82.32.1
Underflow Sense	C2	44.90.02.1
Ch 2 Zero Ins	C0-1	42.84.10
PR RI FR Adder	Arith Op-FP6	2C-42.82.02.1

Objective	Timing	Logic and Location
PR Adv	Arith Op-FP 5-6	4D-42.82.01.1
Stop PR RO	FP Last +	4J-44.84.01.1
Stop A2 Sh RT	C1-2	4F-44.82.32.1
Stop A2 RI FR Adder	C1-2	4G-44.82.30.1

End-Data Operation

The end-data operation begins at C2 of FP 7.

Objective	Timing	Logic and Location
Stt End Data Op	FP 7 C2	3F-42.80.25.1

6.0.00 FLOATING DIVIDE - 73

FLOATING DIVDE DOUBLE - 75

The floating-divide and divide-double instruction execution is almost the same.

Floating divide resets Accumulator 2 to all zeros prior to beginning the divide operation. During divide-double, Accumulator 2 contains the low-order eight digits of the dividend.

The actual execution includes comparing the mantissas and subtracting the exponents. Accumulators 1 and 2 are then complemented as in fixed-point divide. The floating-divide-double operation uses eight digits in each of Accumulators 1 and 2. Fixed point divide is then forced on beginning at the start-divide portion (See 5.3.01). Accumulators 1 and 2 are interchanged at the end of the divide operation putting the eight-digit quotient into Accumulator 1 and the eight-digit remainder in Accumulator 2. The operation concludes by correcting the original exponent difference and inserting the updated exponents in A1 and A2.

The floating ring divides the data cycle into seven subcycles and sequences the operation. The subcycles are obtained by switching the Op matrix output, FDD, with the outputs of the floating ring, FP1-FP7.

The floating ring is advanced sequentially through all steps of the instruction. FP 1-FP 7 are switched with the appropriate pulse from the program ring to cause the advance. The Arith Op signal is switched with FP1 to yield Arith Op-FP1. This signal causes Op Compare Abs and Op A1 to be forced on for subcycle 1. The operation is identical to that for floating add-sub't at this point.

Objective	Timing	Logic and Location
Op Compare Abs		
Set Floating Ring PSN 1	DC 3A	4J-42.80.04.1
Arith Op-FP 1	FP 1-FDD	4A-42.86.15.1
Op Compare Abs	Arith Op-FP 1	5A-42.86.12.1
Op A1	Arith Op-FP 1	3F-42.86.12.1

The following major objectives are required to perform the floating-divide and divide-double operations:

A. Compare Mantissas-Subtract Exponents

B. Complement A2

C. Complement A1

D. Divide

E. Interchange A1 & A2

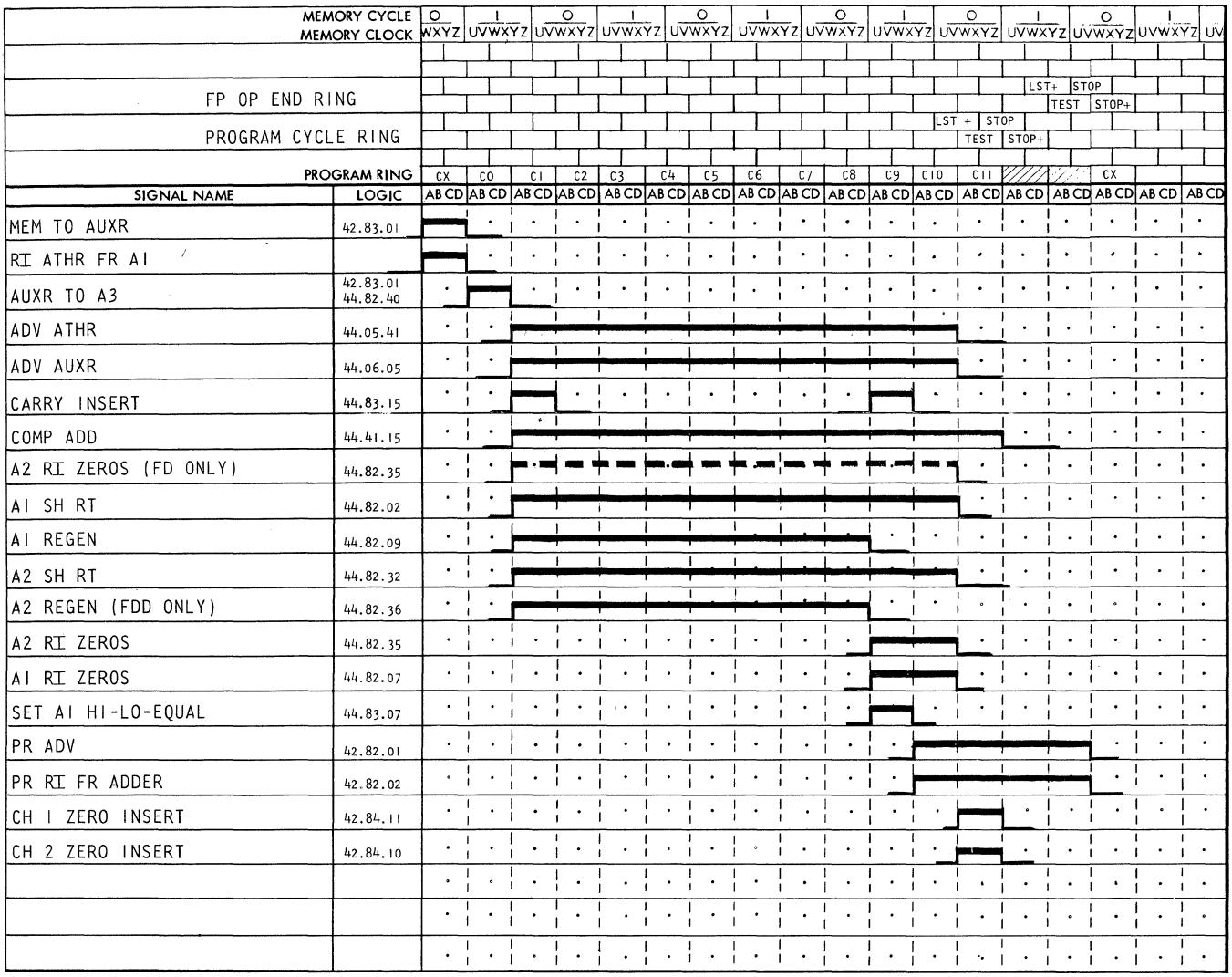


FIGURE 6.0-1 SUBCYCLE 1 - COMPARE MANTISSAS - SUBTRACT EXPONENTS - FORCE FIXED POINT OP COMPARE ABS & OP A1

F. Correct Exponent-Insert in A1

G. Correct Exponent-Insert in A2

H. End Data Operation

Compare Mantissas-Subtract Exponents (Figure 6.0-1)

The Compare Abs operation performs two functions:

1. Determines which of the two mantissas is greater; i.e. A1$\geqslant$ A3 or A3$>$ A1.

2. Subtracts exponents, reads the difference or the complement of the difference into Program Register D.

The mantissas are compared so that the proper adjustment factors for the quotient and remainder exponents may be inserted during a later subcycle. The following shows the resulting conditions and the adjustments indicated for each case, together with sample calculations illustrating the two cases.

If: Dividend $\geq$ Divisor-(carry 1st)

i.e. A1 $\geq$ A3

1. Add 51 to quotient exponent.

2. Subtract 7 from corrected quotient exponent to get correct remainder exponent.

If: Divisor > Divident-(no carry 1st)

i.e. A3 > A1

1. Add 50 to quotient exponent.

2. Subtract 8 from corrected quotient exponent to get correct remainder exponent.

A1 Low

54 125----------Dividend in A1

52 375----------Divisor in A3

54 125

Carry Insert 1 1 Carry Insert

47 625

02 751

PRD-02 ← → No-Carry-A1 Low

A1 Low-Add 50-Sub't 8

A1	A2
52333...	44.....

A1 High

52 375 - Dividend in A1

54 125 - Divisor in A3

52 375

Carry Insert 1 1 Carry Insert

45 875

98 251

PR D 98 ← → Carry Out-A1 High

A1 High - Add 51 - Sub't 7

A1	A2
493000...	42000...

The operation begins by reading the addressed factor (the Divisior) into the auxiliary register at Data Control 4 time. The divisor is then read from the auxiliary register to Accumulator 3 during the following digit time (C0). Accumulator 1 was read into the arithmetic register at CX time. Accumulator 1 contains the high-order eight digits of the dividend plus the dividend exponent. If the operation is floating divide, Accumulator 2 is reset to zeros during the compare function.

Objective	Timing	Logic and Location
STT A2 Zeros	FD-C0-1	4D-44.82.35.1
Stop A2 Regen	FD	5D-44.82.36.1

The floating-divide-double instruction uses Accumulator 2 for the low-order eight digits of the dividend as in fixed-point divide.

With the high-order digits of the dividend in the arithmetic register and the divisor in the auxiliary register, the comparison is initiated. The arithmetic register is read out onto Channel 1 and the auxiliary register to Channel 2. The entries are gated into the adder under complement-add control with a carry insert to form the tens complement of the factor in the auxiliary register.

The adder-carry latches are sensed at C9 time for the existence of the carry. The carry or no-carry condition sets either the A1 High or the A1 Low latch (Logic 44.83.07.1). These latches control subsequent execution of the instruction.

Objective	Timing	Logic and Location
Set A1 High	Carry 1st-C9	4B-44.83.07.1
Set A1 Low	No Carry 1st-C9	4E-44.83.07.1

A carry is inserted at C9 time to form a tens complement number in the units position of the exponent. The two digits making up the exponent difference are read into Program Register D. This number may be the true difference or the complement of the difference as indicated previously.

The field register is set for full field control by Arith Op-FP1 and the Data Control 4 pulse just as for the floating add-sub't instructions.

Objective	Timing	Logic and Location
Mem to Auxr	FDDA-DC 4	6J-42.83.01.1
RI to ATHR FR A1		
Auxr to A3	FDD 1-CX-0	4F-42.83.01.1
	FDD 1-CX-0	4E-44.82.40.1

Objective	Timing	Logic and Location
Adv Athr (Entry A)	C1	4A-44.05.41.1
Adv AUXR (Entry B)	C1	5C-44.06.05.1
A1 Sh RT	FP1-Mpy-Div, C0-1	4C-44.82.02.1
A1 REGEN	FP1-Mpy-Div, C0-1	4B-44.82.09.1
A2 RI Zeros (FD Only)	FD-C0-1 FP1	4D-44.82.35.1
A2 Sh RT	Arith Op-FP1	4A-44.82.32.1
A2 Regen (FDD Only)	C0-1-No FD	4D-44.82.36.1
Carry Insert	C1 + C9	5J-44.83.15.1
Comp Add	C1	4C-44.41.15.1
A2 RI Zeros	C8-9-FDD 1	5D-44.82.35.1
A1 RI Zeros	C9 + C10-FP Mpy Div	4A-4B-44.82.07.1
Set Hi-Lo-Eq	C9	44.83.07.1
PR D Adv	Arith Op-FP1-C9-10	4A-42.82.01.1
PR D RI FR Adder	Arith Op-FP1-C9-10	2A-42.82.02.1
Ch 1 Zero Insert	FP 1 Mpy Div-C10-11	4C-42.84.11.1
Ch 2 Zero Insert	FP 1 Mpy Div-C10-11	4G-42.84.10.1
Stop A1 Regen	C8-9	4G-44.82.09.1
Stop A2 Sh RT	PC Last +	4F-44.82.32.1

Complement A2 (Figure 6.0-2)

The complementing of the dividend, as in fixed-point divide, is accomplished in two steps. The floating-ring outputs, FP2 and FP3, are switched with FDD to give FDD 2 and FDD 3. The FDD 2 and FDD 3 signals define, respectively, the complement A2 and A1 subcycles.

The execution of the complement function depends upon the A1 High or A1 Low latch setting. When A1 is high, the read-out of Accumulator 2 onto Channel 2 starts at C1 time. The entries are gated by complement-add and no-carry insert to form a nines complement. The entries are read back into Accumulator 2 with a shift right each digit time. The Regen at C10 time causes A2 to contain the digits indicated below. The digits used represent the digit positions of the original mantissa, i.e. 2-9.

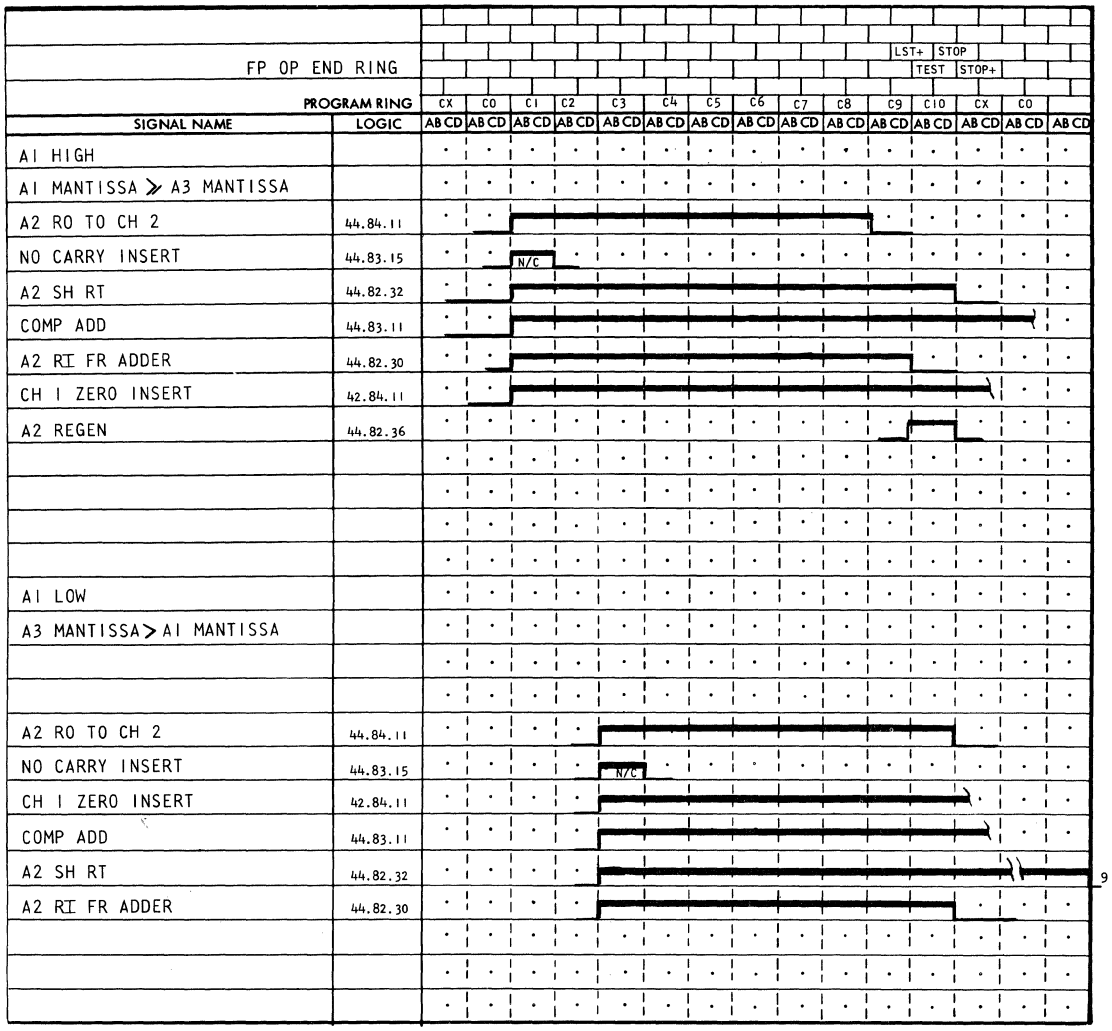


FIGURE 6.0-2 SUBCYCLE 2 - COMPLEMENT A2 DIVIDEND ≫ DIVISOR (A1 ≫ A3)
DIVISOR > DIVIDEND (A3 > A1)

A1 High

	S	0	1	2	3	4	5	6	7	8	9	O. L.
A2-(C10)		0	2	3	4	5	6	7	8	9	0	

The A1 High condition causes the dividend to be shifted right one place during the succeeding subcycle, Complement A1.

During the A1 Low condition, the read-out of Accumulator 2 to Channel 2 occurs at C3 time. The entries are gated into the adder as before under complement-add and no-carry insert to form the nines complement. For this case the read in to Accumulator 2 is shut off as before at C10 time allowing the read-in to Accumulator 2 of only eight digits. The first digit read-in to A2 had been a zero since the first entry had been read out to the adder at C3. Only seven digits, therefore, are read into A2 in addition to the initial zero. Accumulator 2, when A1 is low, appears as follows at C10 time of the complement A2 subcycle:

A1 Low

	S	0	1	2	3	4	5	6	7	8	9
A2-(C10)		3	4	5	6	7	8	9	0	0	0

A1 High - A1 Mantissa ≥ A3 Mantissa

Objective	Timing	Logic and Location
A2 Sh RT	C0-1 FDD 2-A1 High	5C-44.82.32.1
A2 RO to Ch 2	C0-1 FDD 2-A1 High	4B-44.84.11.1
No Carry Insert	C1-FDD 2 A1 High	5A-44.83.15.1
Ch 1 Zero Insert	C2-3 FDD 2 A1 High	4F-42.84.11.1
Comp Add	C0-1 FDD 2 A1 High	4F-44.83.11.1
A2 RI FR Adder	C0-1 FDD 2 A1 High	4B-44.82.30.1
A2 Regen	C9-10 FDD 2 A1 High	4C-44.82.36.1
Stop A2 RO to Ch 2	C8-9	4F-44.84.11.1

A1 Low A3 Mantissa > A1 Mantissa

Objective	Timing	Logic and Location
A2 Sh RT	C0-1 Arith Op FP1	4A-44.82.32.1
A2 RO to Ch 2	C2-3 FDD ± A1 Low	4C-44.84.11.1
No Carry Insert	C3 FDD 2 A1 Low	5C-44.83.15.1
Ch 1 Zero Insert	C2-3 FDD 2 A1 Low	4F-42.84.11.1
Comp Add	C2-3 FDD 2 A1 Low	4G-44.83.11.1
A2 RI FR Adder	C2-3 FDD 2 A1 Low	4E-44.82.30.1
Stop A2 Sh RT	FP Stop	3J-44.82.32.1
Stop A2 RO to Ch 2	FDD A-FP Stop	4G-44.84.11.1

Complement A1 (Figure 6.0-3)

To perform a fixed-point-divide operation, the eight-digit divisor in Accumulator 3 must be larger than the portion of the dividend in Accumulator 1 (high-order eight digits). If this is not the case, the A1 ≥ A3 indication obtained as the result of Compare causes

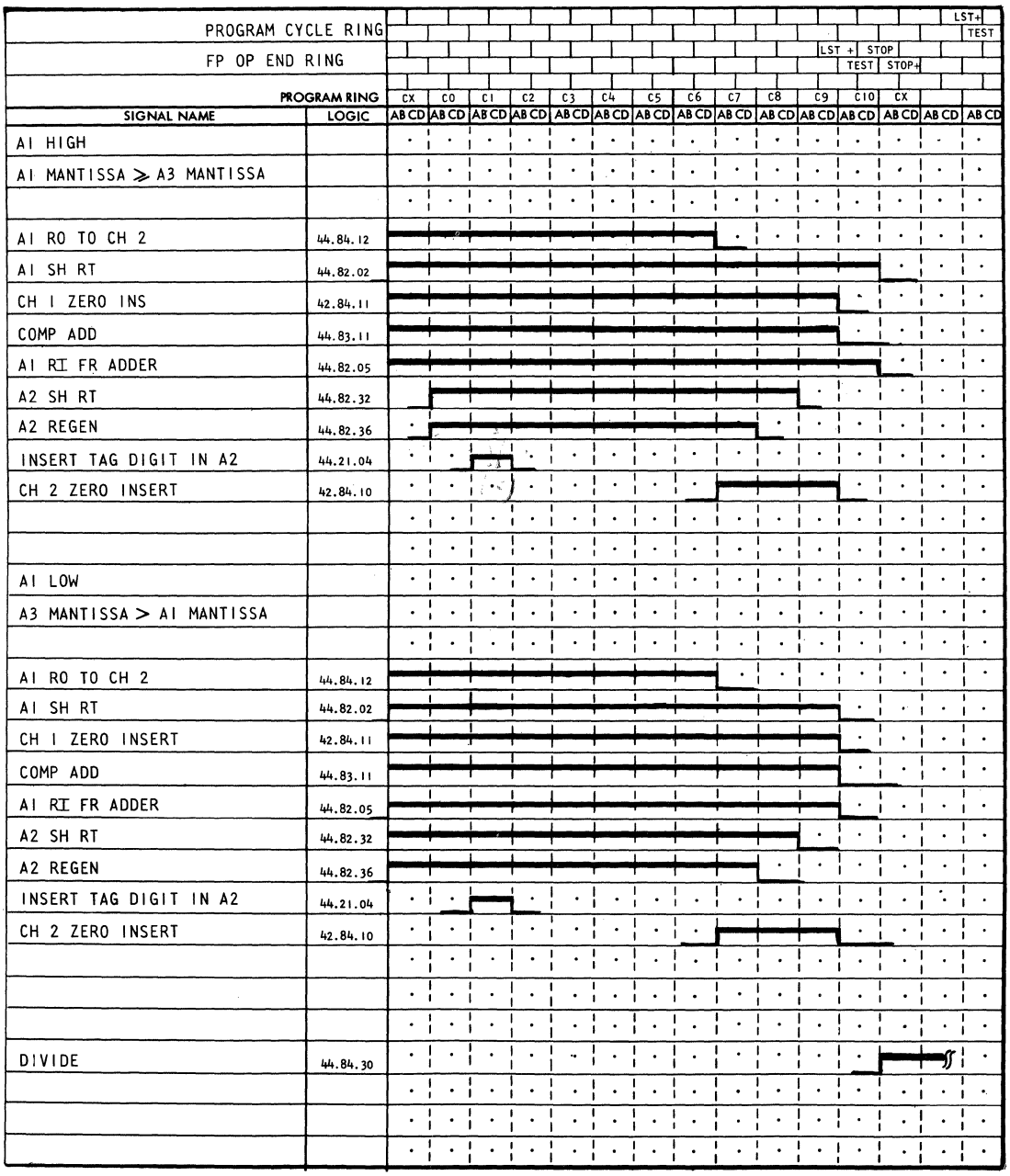


FIGURE 6.0-3 SUBCYCLE 3 - COMPLEMENT A1 DIVIDEND ≥ DIVISOR (A1 ≥ A3)
DIVISOR > DIVIDEND (A3 > A1)
SUBCYCLE 4 - DIVIDE

A1 to be shifted right one extra place during the complementing of A1. This action effectively divides the dividend by 10 to satisfy the conditions for a fixed point divide. During a later subcycle, 51 rather than 50 is added to the exponent difference to adjust the quotient value. The exponent of the remainder in Accumulator 2 is 7 less than the exponent of the quotient in Accumulator 1 to correct for the dividend right shift.

The A1 High condition reads out Accumulator 1 to Channel 2 under complement-add and no-carry insert to form the nines complement. The result is read back into A1 with a right shift each digit time. The Channel 2 zero insert under complement-add causes nines to be read into the high-order positions. At C7 and C10, the register looks as follows:

	S	0	1	2	3	4	5	6	7	8	9	O. L.
A1												
C7		2	3	4	5	6	7	8	9	0		
C10		9	9	9	2	3	4	5	6	7	8	9

If A1 is low, the A1 read-in from adder turns on at CX and reads the high-order digit of A2 into A1. The read-in to A1 continues for eight digit times. The register is shifted right for a total of ten digit times. Thus, the high-order digit of A2 has shifted into the A1 output latches. A1, therefore, looks as follows at C10 time:

	S	0	1	2	3	4	5	6	7	8	9	O. L.
A1		0	0	2	3	4	5	6	7	8	9	2 (High order of A2)

Accumulator 2 is operated simultaneously. The register is right shifted for ten positions with a regen in the first nine positions. At C1 time, the five-bit tag is inserted into A2.

The register contains the digits specified as follows for the A1 High condition upon completion of the operation:

	S	0	1	2	3	4	5	6	7	8	9	O. L.
A2		□	3	4	5	6	7	8	9	X	0	2

Blank ← (□) Tag ← (X)

The A1 Low condition sets the following digits into Accumulator 2 upon completion of complement A1:

	S	0	1	2	3	4	5	6	7	8	9	O. L.
A2		□	4	5	6	7	8	9	X	0	0	3

Blank ← (□) Tag ← (X)

A1 High - A1 Mantissa ≥ A3 Mantissa

Objective	Timing	Logic and Location
A1 Sh RT	FDD 2-3 FP Stop	4D-44. 82. 02. 1
A1 RO to Ch 2	FP 2-3-FP Stop	4B-44. 84. 12. 1
Comp Add	FDD 2 A1 Hi or Low	4F-4C-44. 83. 11. 1

Objective	Timing	Logic and Location
A1 RI FR Adder	FDD 2-3-FP Stop	4C-44.82.05.1
Stop A1 RI FR Adder	FDD 3-C9-10	4F-44.82.05.1
A2 Sh RT	FDD 3 A1 High CX-0	2C-44.82.32.1
A2 Regen	FDD 3-A1 High CX-0	4A-44.82.36.1
Stop A2 Regen	FDD 3-C7-8	4G-44.82.36.1
Insert 5 Bits in A2	C1	44.21.04.1
Ch 1 Zero Insert	C2-3 FDD 2	4F-42.84.11.1
Ch 2 Zero Insert	FDD 3-C6-7	4A-42.84.10.1
Stop A1 RO to Ch 2	FDD 3-C6-7	4F-44.84.12.1

A1 Low A3 Mantissa $\geqslant$ A1 Mantissa

Objective	Timing	Logic and Location
A1 Sh RT	FDD 2-3 FP Stop	4D-44.82.02.1
A1 RO to Ch 2	FP 2-3 FP Stop	4B-44.84.12.1
Comp Add	FDD 2 A1 High or Low	4F-4G-44.83.11.1
A1 RI FR Adder	FDD 2-3-FP Stop	4C-44.82.05.1
A2 Sh RT	C2-3	5D-44.82.32.1
Stop A2 Sh RT	C8-9	5J-44.82.32.1
A2 Regen	FDD 2-3 A1 Low	4D-44.82.36.1
Insert 5 Bits in A2	C1	44.21.04.1
Ch 1 Zero Insert	C2-3 FDD 2	4F-42.84.11.1
Ch 2 Zero Insert	C6-7	4A-42.84.10.1
Stop A1 Sh RT	C9-10	4E-44.82.02.1

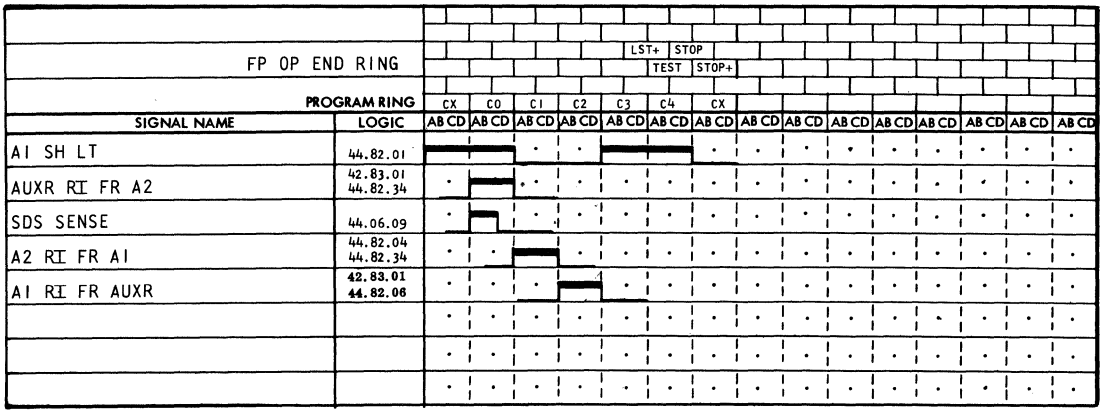


FIGURE 6.0-4 SUBCYCLE 5 - INTERCHANGE A1 & A2

Divide (Figure 6.0-3)

As the floating ring is advanced to FP 4, fixed-point divide is initiated.

Objective	Timing	Logic and Location
STT Divide	FDD A-CX FP 3-4	4B-44.84.30.1

From here on, the operation proceeds as in the circuit description of fixed-point divide (Section 5.3.01 of 7601 MI, Vol. II).

Interchange A1 and A2 (Figure 6.0-4)

At the conclusion of the fixed-point divide portion, the quotient is in Accumulator 2 positions 2-9, and the remainder is in Accumulator 1 positions 2-9. The factors in the two registers must be interchanged.

The contents of Accumulator 2 are read into the auxiliary register at C0 time, which provides temporary storage for this factor. Accumulator 1 then reads its contents into Accumulator 2 at C1. The auxiliary register transfers its contents to Accumulator 1 at C2 to complete the operation.

The significant digit scanner senses for a zero condition as Accumulator 2 transfers to the auxiliary register at C0. This function checks for a zero quotient.

Accumulator 1 is left shifted at CX and again at C3 to place both quotient and remainder in positions 0-7 of A1 and A2.

Objective	Timing	Logic and Location
A1 Sh Lt	FDD 2-C 2-3 FDD B FP 4-5	4C-44.82.01.1 4D-44.82.01.1
Auxr RI FR A2		
Auxr RI AB	CX-0	4E-42.83.01.1
A2 RO AB	CX-0	5G-44.82.34.1

Objective	Timing	Logic and Location
SDS Sense	C0	XX-44.06.09
A2 RI FR A1		
A1 AB RO	C0-1	4B-44.82.04.1
A2 RI AB	C0-1	5F-44.82.34.1
A1 RI FR Auxr		
Auxr RO AB	C1-2	4H-42.83.01.1
A1 RI AB	C1-2	4D-42.82.06.1

Correct Exponent-Insert in A1 (Figure 6.0-5)

The quotient is transferred to Accumulator 1 during the previous subcycle. Program Register D contains the difference of the two original exponents if A3 is greater than A1. If A1 is greater than or equal to A3, Program Register D contains the complement of the original difference. Program Register D is read out onto Channel 1 at CX time. The Channel 2 Digit insert provides a "1" digit at CX time if A1 $\geq$ A3 and a "5" digit at C0 gated by true-add and no-carry insert. If A3 $>$ A1, the Channel 2 zero insert inserts a zero at CX time and the "5" digit at C0. The result is read into both the Program Register D and Accumulator 1 positions 0-1. Accumulator 1 is simultaneously shifted right each digit time to make room for the exponent.

The overflow-underflow test is made in the usual way by sampling the adder output latches at C1-2 time.

Objective	Timing	Logic and Location
Insert Digits in Fld Reg (0-1)	C0	XX-44.80.05.1- 44.80.06.1
PR D ADV	Arith Op-FP-5-6FP Stop	4D-42.82.01.1
PR D RO to Ch 1	FDD B-FP 5-6	4F-44.84.01.1
Ch 2 Digit Ins (A1 $\geq$ A3)	CX A1 High (1) C0 (5)	2G-42.84.12.1 2B-42.84.12.1
Ch 2 Zero Ins (A3 $>$ A1)	C0-1	5E-42.84.10.1
No Carry Ins	FDD 6-CX	5D-44.83.15.1
True-Add	FDD B-FP 56	4B-44.83.10.1
PR RI FR Adder	Arith Op-FP 6 CX-0	2C-42.82.02.1
Ch 2 Digit Ins (A3 $>$ A1)	FDD 6 C0 (5)	2B-42.84.12.1

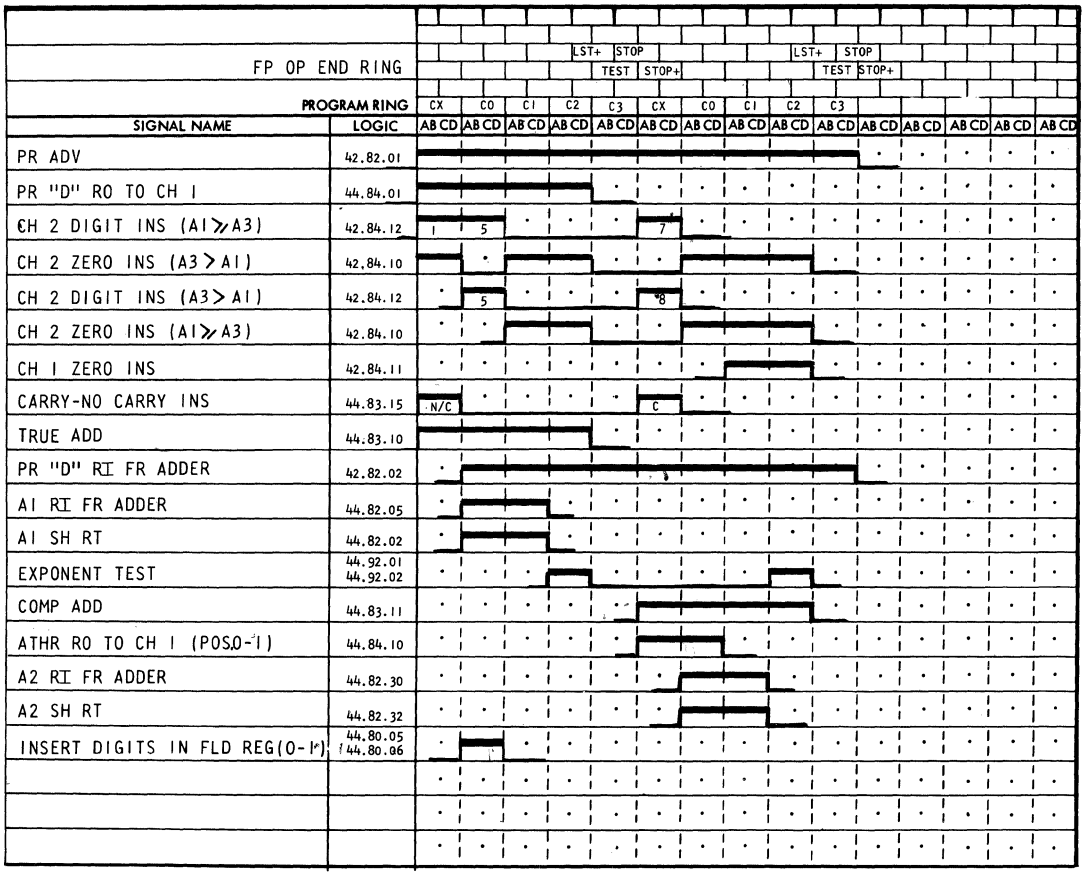


FIGURE 6.0-5 SUBCYCLE 6 - CORRECT EXPONENT - INSERT IN A1
SUBCYCLE 7 - CORRECT EXPONENT - INSERT IN A2

Objective	Timing	Logic and Location
A1 Sh RT	Arith Op FP 6 CX-0	4B-44.82.02.1
A1 RI FR Adder	Arith Op FP 6 CX-0	4B-44.82.05.1
Insert Digits in Fld Reg	FDD 6-C0 (Tens) FDD 6-C0 (Units)	2A-44.80.05.1 2F-44.80.06.1
Ch 2 Zero Ins (A3 > A1)	FP 5-6 A1 Low	5F-5G-42.84.10.1
Stop Ch 2 Zeros (A3 > A1)	CX-0 FDD 6 A1 Low	4J-42.84.10.1
Ch 2 Zero Ins (A1 ≥ A3)	FDD 6 C0-1	5E-42.84.10.1
Exponent Test	C1-2	XX-44.92.01.1 44.92.02.1
Stop PR RO	FP Last +	4J-44.84.01.1

Correct Exponent-Insert in A2 (Figure 6.0-5)

Subcycle 7, just as for the other floating-point operations, causes the correct exponent to be inserted in Accumulator 2.

The field register is set for 0-1 control during the previous subcycle.

The arithmetic register contains the original dividend exponent. This exponent must be reduced by 7 or 8, depending upon the conditions set by the problem, to yield the correct remainder exponent. Positions 0-1 of the arithmetic register are scanned out onto Channel 1. An 8 digit is inserted on Channel 2 if A3 $>$ A1. If A1 $\geqslant$ A3, a 7 is inserted. Both digits are gated by complement-add and carry insert. Zeros are inserted for the remaining Channel 2 entries for both cases. The result is read into Accumulator 2 positions 0-1. At the end of the operation, Program Register D contains the exponent of Accumulator 2.

The underflow test only is made on the exponent of Accumulator 2. This is accomplished in the usual way by sampling the adder output latches at C1-2 time.

Correct Exponent & Insert in A2

Objective	Timing	Logic and Location
ATHR RO to Ch 1 (Pos 0-1)	FDD B FP 6-7 FP Stop	4C-44.84.10.1
Ch 2 Digit Ins 8 (A3 $>$ A1)	CX A1 Low	3B-42.84.12.1
Ch 2 Digit Ins 7 (A1 $\geqslant$ A3)	CX A1 High	3E-42.84.12.1
Carry Ins	Arith Op-FP 7 CX	4H-44.83.15.1
Comp Add	Arith Op FP 6-7 FP Stop	4E-44.83.11.1
A2 Sh RT	Arith Op FP 7 CX-0	4C-44.82.32.1
A2 RI FR Adder	Arith Op FP 7 CX-0	4C-44.82.30.1
Ch 2 Zero Ins (A3 $>$ A1)	FDD 7 CX-0	5B-42.84.10.1
Ch 2 Zero Ins (A1 $\geqslant$ A3)		
Ch 1 Zero Ins	FDD 7-C0-1	5B-42.84.11.1
Exponent Test	C2	XX-44.90.02.1
PR RI FR Adder	FP 6	2C-42.82.02.1
ADV PR	FP 6	4D-42.82.01.1
Stop ATHR RO Ch 1	FDD 7 C0-1	4H-44.84.10.1

End-Data Operation

The end of the floating-divide-double operation occurs at C2 time of FP 7.

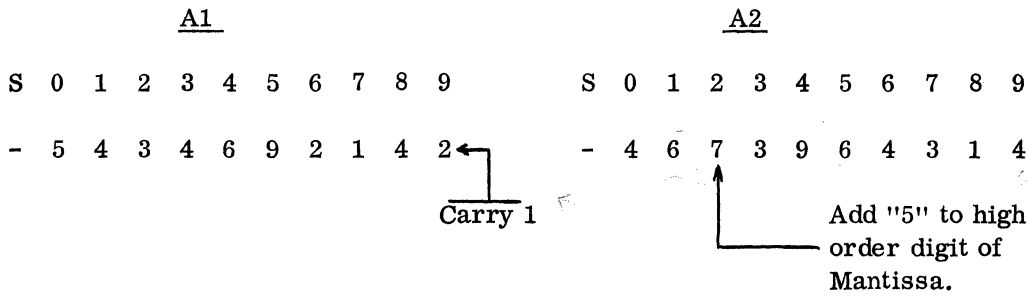
Objective	Timing	Logic and Location
STT End Data Op	FP 7-C2	3F-42.80.25.1

7.0.00 FLOATING ROUND + 71

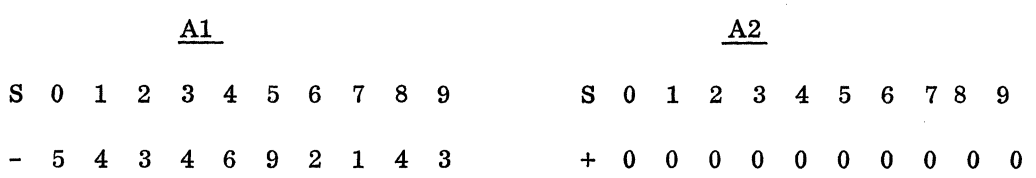
The floating-round instruction is a no-access code, i.e., the command requires no operand and, therefore, no access to core storage.

The main objective of the operation is to round off the mantissa in Accumulator 1. At the conclusion of the instruction the mantissa of Accumulator 2 is reset to all zeros and carries a zero exponent. The sign of Accumulator 2 is also set to plus during the execution of the instruction.

The operation is illustrated below. Accumulators 1 and 2 are shown at the beginning and the end of the instruction. The operation is initiated by adding the "5" digit to the high order of the Accumulator 2 mantissa. The carry as shown is added to the Accumulator 1 low order digit to cause the round-off.



Result:



The instruction utilizes only one cycle defined by the FRD output of the Op Matrix. The operation first loads the field register with 2-2 (Logics 44.80.05-06) so that the high-order digit of the arithmetic register may be scanned out.

The operation is described for the two possible cases: Floating Round Carry and Floating Round No-Carry.

No-Carry Condition (Figure 7.0-1)

Accumulator 2 is transferred to the arithmetic register at CX time. Accumulator 1 is read into the auxiliary register at C0. Also at C0, Accumulator 2 shifts right, simultaneously reading in zeros.

At C1, the field ring scans out the high-order mantissa digit of the arithmetic register onto Channel 1. A 5 digit is injected by the Channel 2 digit insert. The entry is gated by true-add and no-carry insert. This function tests the value of the high-order digit of Accumulator 2. If the digit is five or over, a carry is generated and added to the the next digit position entering the adder at C2.

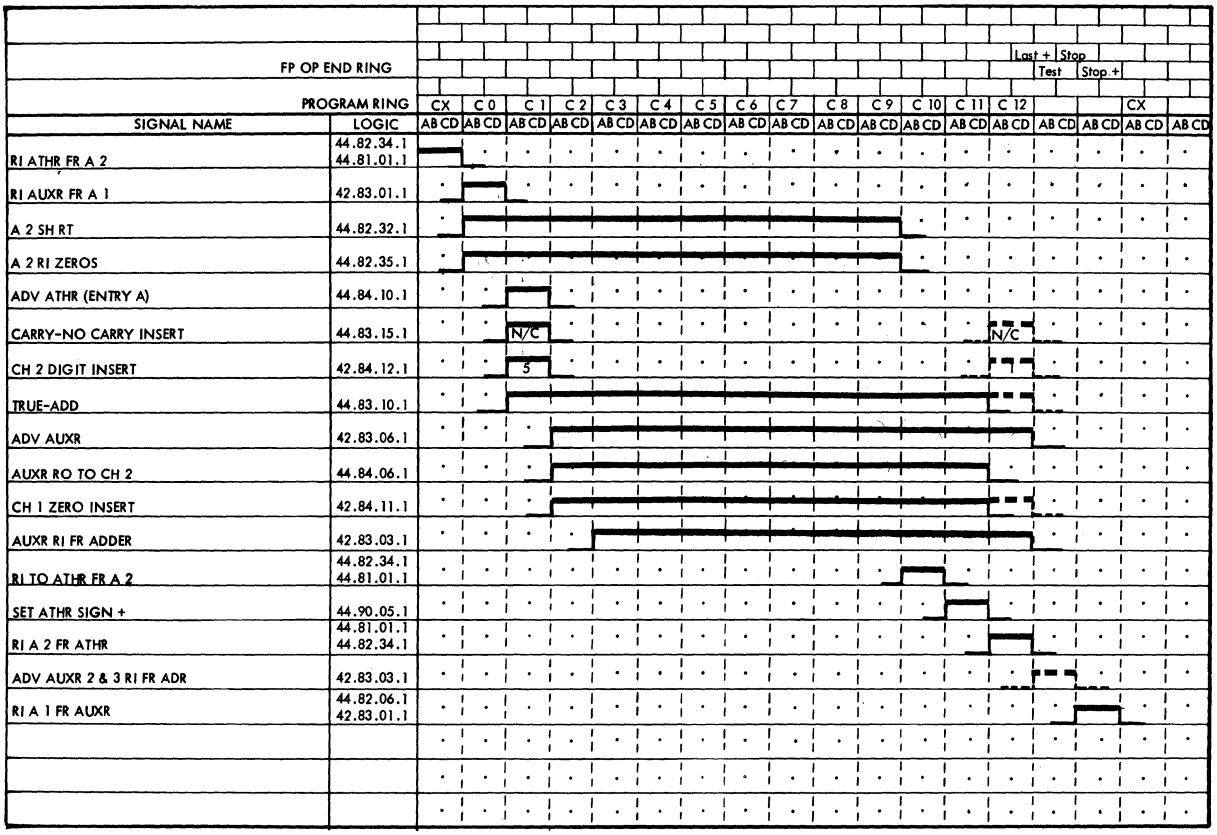


FIGURE 7.0-1 FLOATING ROUND

At C2, the auxiliary register is read out to Channel 2. The auxiliary register contains the contents of Accumulator 1 at this time. The low-order digit is read into the adder at C2 coincident with the carry, if a carry exists. The operation then proceeds as in a normal add. Zeros are inserted for the Channel 1 entries.

Carry Condition (Figure 7.0-1)

The carry may be extended through all eight digit positions of the mantissa if the original mantissa of Accumulator 1 is all nines. If this occurs, the units position of the exponent is increased by one and the mantissa is reset to all zeros except for the high order position which becomes a 1. The following is an illustration:

A1										
S	0	1	2	3	4	5	6	7	8	9
+	5	6	9	9	9	9	9	9	9	9

Carry "1"

A2										
S	0	1	2	3	4	5	6	7	8	9
+	4	8	6	4	3	9	2	1	4	4

Add "5" to high order digit of Mantissa.

Result:

	A1											A2										
S	0	1	2	3	4	5	6	7	8	9		S	0	1	2	3	4	5	6	7	8	9
+	5	7	1	0	0	0	0	0	0	0		+	0	0	0	0	0	0	0	0	0	0

The FR D Carry signal controls the operation for this case. The adder output latches are sensed at C10 for the carry out of the high-order position of the mantissa.

Objective	Timing	Logic and Location
FR D Carry	Carry 1st FR D A, C10	4B-44.83.17.1
FR D No-Carry	No Carry 1st FR D A, C10	4F-44.83.17.1

If a no-carry occurs, the true-add latch is shut off at C11, and the adder operation is terminated.

The carry condition, however, specifies that the original mantissa is all nines. The mantissa must have a 1 inserted in its high-order position to satisfy the requirement for a minimum mantissa value.

The true-add latch remains on for one additional digit time for the carry condition. The auxiliary register is being advanced through C12 time. At C12, a 1 digit on Channel 2 and a no-carry insert result in a 1 output from the adder at FP Last + time. The Channel 1 zero insert furnishes an extra 0 for the FR D carry condition for this digit time. The adder output is read into auxiliary register 2 high order corresponding to position 2 of the mantissa. Auxiliary Registers 2 and 3 are advanced as one eight-position register during the operation to read the digit into Auxiliary Register 2, high-order position.

At the end of the operation, the auxiliary register contains the result of the round-off. The register is parallel-transferred to Accumulator 1 by FP Stop to conclude the operation.

FRD CARRY AND NO-CARRY

Objective	Timing	Logic and Location
Insert Digits in Fld Reg (2-2)	DC 4	2C-2E-44.80.05- 44.80.06
RI Athr FR A2		
A2 AB RO	DC 4	3J-44.82.34.1
Athr AB RI	DC 4	4F-44.81.01.1

Objective	Timing	Logic and Location
A2 Sh RT	CX-0	3C-44.82.32.1
A2 RI Zeros	C1-2	4B-44.82.35.1
RI Auxr FR A1	FR D-CX-0	4D-42.83.01.1
Adv Athr (Entry A)	C0-1	4B-44.84.10.1
Ch 2 Digit Ins "5"	C1-FR D	2D-42.84.12.1
No-Carry Insert	C1	4A-44.83.15.1
True-Add	C0-1	4C-44.83.10.1
Ch 1 Zero Insert	C1-2	4E-42.84.11.1
Adv Auxr	C1-2	4B-42.83.06.1
Auxr RO to Ch 2	C1-2	4B-44.84.06.1
Stop Athr RO to Ch 1	C1-2	4G-44.84.10.1
Auxr RI FR Adder	C2-3	4B-42.83.07.1
Carry Sense	C10	4G-44.41.01
RI Athr FR A2		
A2 AB RO	C9-10	4J-44.82.34.1
Athr AB RI	C9-10	4E-44.81.01.1
Set Athr Sign +	C10-11	4C-44.90.05.1
Stop A2 Sh RT	C9-10	3F-44.82.32.1
True Add Off	No Carry-C11-12	4E-44.83.10.1
Stop Auxr 3 RO to Ch 1	FR D-C 11-12	4E-44.84.06.1
Stop Auxr Adv	FR D-FP Last +	4G-42.83.06.1
RI A2 FR Athr		
Athr AB RO	C11-12	5D-44.81.01.1
A2 AB RI	C11-12	4G-44.82.34.1

Objective	Timing	Logic and Location
RI A1 FR Auxr		
Auxr AB RO	FP Stop-FR D	4G-42.83.01.1
A1 AB RI	FP Stop-FR D	4A-44.82.06.1

FR D CARRY

Objective	Timing	Logic and Location
No-Carry Insert	C12-FR D Carry	XX-44.83.15.1
Ch 2 Digit Ins "1"	C12-FR D Carry	2J-42.84.12.1
Stt Auxr 3 RI 2	FR D Carry-FP Last +	4B-42.83.10.1
Stt Auxr 3 Sh RT RI	FR D Carry-FP Last +	4F-42.83.10.1
Stop Auxr 3 RI 2	FR D Carry-FP Stop	4D-42.83.10.1
Stop Auxr 3 Sh RT RI	FR D Carry-FP Stop	4H-42.83.10.1
Stt Auxr 2 Adder RI	FR D Carry-FP Last +	4C-42.83.03.1
Stop Auxr 2 Adder RI	FR D Carry-FP Stop	4F-42.83.03.1

8.0.00 FLOATING BRANCH OVERFLOW + 70

FLOATING BRANCH UNDERFLOW - 70

These instructions cause a branching of the problem program to the location in core storage specified by the address portion of either the FBV or FBU instruction.

The overflow or underflow condition sets a floating-overflow or floating-underflow latch. The conditions for an underflow or overflow are outlined in Section 2.2.00.

As a result of the command to branch, the address portion of either the FBU or FBV instruction is transferred to the instruction counter. The operation from here on is as detailed in the description of the Branch codes, Section 5.8.00 of Arithmetic and Program Unit.

Objective	Timing	Logic and Location
FP BR to D	CX Op FBV FP Overflow	XX-44.92.01.1
FP No Br to D	CX Op FBV No FP Overflow	XX-44.92.01.1
FP Br to D	CX Op FBU FP Underflow	XX-44.92.02.1
FP No Br to D	CX Op FBU No FP Underflow	XX-44.92.02.1