DISTRIBUTION LIST

Messrs.	W. N. Barrett	Dept. 705
	K. A. Bell	Dept. 280
	G. M. Benson	Dept. 299
	A. A. Bissell	Dept. 617
	G. E. Bontecou	Dept. 299
	N. M. Boothe	Dept. 299
	C. E. Branscomb	Dept. 191
	B. M. Carr	Dept. 299
	R. P. Case	Dept. 663
	J. S. Conzola	Dept. 191
	J. F. Dite	Dept. 617
	P. Fagg	Dept. 663
	P. Farbanish	Dept. 280
	A. Glover	Dept. 191
	E. J. Grenchus	Dept. 663
	W. F. Hanf	Dept. 299
	E. S. Hughes, Jr.	Dept. 454
	J. Ingram	Dept. 280
	T. Macklulin	Dept. 625
	T. M. McAdon	Dept. 294
	H. W. Morrow	Dept. 294
	W. R. Murray	Dept. 268
	J. L. Pokoski	Dept. 280
	G. H. Rees	Dept. 280
	D. Royse	Dept. 194
	W. S. Schaffer	Dept. 631
	R. M. Smith	Dept. 663
	K. B. Stoffel	Dept. 294
Miss	J. Swanson	Dept. 171
Messrs.	W. Wheeler	Dept. 391
	E. T. Zielinski	Dept. 614
	F. O. Underwood	Dept. 631

May 4, 1960

MEMORANDUM FOR FILE 1401

SUBJECT: Engineering Description of the Revised Multiply-Divide Option

Introduction

Multiply-Divide is a single optional feature which can be installed on the 1401 machine. Certain basic rules must be observed in assigning storage fields and in using word marks and plus-minus signs to insure valid answers at all times. It will be noticed that there is a marked similarity in the execution of the respective operations and in the methods for determining the assignment of storage fields.

Multiply

Prior to initiating a multiply operation, it is necessary to provide a B-field which has a length equal to the multiplier + multiplicand +1 with a WM in the highest order position. The multiplier must be moved into the high order positions of this field; the product will be developed in the low order positions of this field. A WM must also be placed in the high order position of the A-field (Multiplicand). Sign bits in the units position of the multiplier or multiplicand are not necessary for the execution of this operation, but minus signs must be indicated by a B bit. See example below - a bar over a digit denotes a WM.

				9̄	8	7	Multiplicand
3̄	0	1					Multiplier
0̄	2	9	7	0	8	7̄	Product

When the above conditions have been satisfied, the instruction

© (AAA) (BBB) Multiply,

where (AAA) and (BBB) are the addresses of the units positions of the A-field and B-field respectively, will initiate the operation. Primarily, the multiply operation is one in which the multiplicand is added to the product a number of times dependent on the value of the multiplier, and the multiplier being destroyed during the process.

Two additional address registers are required for this operation; A-Aux STORAGE ADDRESS REGISTER (A-AUX STAR) and B-Aux STORAGE ADDRESS REGISTER (B-AUX STAR). The A-Aux STAR always retains the address of the units position of the multiplicand. After each addition of the multiplicand to the product, a transfer of the A-Aux STAR contents to the A-STAR will return the A-Star to its initial starting address.

The B-Aux STAR retains the address of the position of the product into which the units position of the multiplicand is to be added and is decremented by 1, to effect a shift in the product, as each multiplier digit is reduced to zero. The B-Aux STAR address is used to select the storage position only during the first B cycle of each addition and when shifting the product field due to a multiplier digit of zero.

The detailed operation is as follows:

After the Multiply instruction, @ (AAA) (BBB), has been stored in the proper registers, the execution of the instruction begins by having the contents of the A-STAR and B-STAR regenerated and also transferred to the A-Aux STAR and B-Aux STAR respectively during the first A and B cycles. Thus the addresses of the positions of the multiplier and product are stored for future reference. During the first A cycle the sign of the multiplicand is also analyzed and stored in the Sign Trigger.

Consecutive A and B cycles are then taken until an A Register WM is detected and further A cycles are eliminated. The location of the multiplier digit is always the third B cycle following the A cycle in which the WM is detected, at which time the multiplier sign is analyzed if it is the units position. Zeros are put in memory during B cycles to clear the product field up to but not including the units position of the multiplier.

Multiplier digits are analyzed for the following conditions:

a. Blank or zero
b. "1" bit only
c. "2" bit only
d. Multiplier digit greater than 2

Based upon the above results, the following action takes place:

a. Blank or zero - A zero is written back in storage and a B cycle is taken to decrement the B-Aux STAR by "1" which effects a shift of the partial product. Another B cycle is then taken to analyze the next multiplier position.

b. "1" bit only - A zero is written back in storage and consecutive A and B cycles are taken to add once the multiplicand to the partial product. On the first A and B cycles the A-Aux STAR and B-Aux STAR are used to address storage and the decremented addresses enter only the A-STAR and B-STAR respectively.

c. "2" bit only - The action here is the same as for the "1" bit, but twice the multiplicand is added to the partial product.

d. Multiplier digit greater than 2 - The multiplier digit is reduced by two prior to being written in memory. Otherwise the action is the same as for the "2" bit.

The product sign is always set on the first B cycle after the multiplier sign is analyzed. If the units digit of the multiplier is zero, the units position of the product is addressed when a B cycle is taken to decrement the B-Aux STAR by 1 and shift the partial product. If the units digit of the multiplier is other than zero, the sign is set during the first B cycle of the addition when the units position of the product is agin addressed.

The multiply operation is terminated when the WM and a zero is detected in the high order position of the multiplier. At the completion of the operation, sign bits appear in the units position of the product only. All other zone information, excluding WM's, are removed in the B-field and zeros are written in all positions of the B-field which do not have significant digits. All information in the A-field (multiplicand) is regenerated. (BBB) is address of the units position of the product.

Divide

Prior to initiating a divide operation, it is necessary to provide a B-field which has a length equal to the divisor + dividend +1, a WM may be placed in the high order position of this field but it is not necessary for the execution of this operation. The dividend must be moved by a Reset Add instruction, into the low order positions of this field with a B bit in the units position; the quotient will be developed in the high order positions. Zeros must be placed in the high order positions of this field.

A WM must be placed in the high order position of the A-field (divisor). Sign bits in the units position of this field are not necessary for the execution of this operation, but minus signs must be indicated by B bits. See example below- a bar over a digit denotes a WM.

Quotient			Remainder			
	±				±	
$\bar{0}$	4	3	0	0	$\bar{7}$	Quotient
					±	
			4	8	0	Dividend
				$\bar{1}$	1	Divisor

When the above conditions have been satisfied, the instruction

% (AAA) (BBB) DIVIDE,

where (AAA) is the address of the units position of the A-Field and (BBB) is the address of the high order position of the dividend, will initiate the operation. The divide operation is primarily one in which the divisor is subtracted from the dividend a number of times till the dividend is reduced to a sum less than the divisor. A total of the number of subtractions forms the quotient and a final partial dividend less than the divisor forms the remainder.

Two additional address registers are required for this operation; A-Aux STAR and B-Aux STAR. The A-Aux STAR always retains the address of the units position of the divisor. After each subtraction of the divisor from the dividend and/or comparison of the divisor to the dividend, a transfer of the A-Aux STAR contents to the A-STAR will return the A-STAR to its initial starting address.

The B-Aux STAR retains the address of the positions of the partial dividend from which the units position of the divisor will be subtracted and is incremented by 1 when the partial dividend is less than the divisor. Storage is addressed by the B-Aux STAR during the first B cycle of each subtraction and/or comparison and when shifting due to a dividend being less than the divisor.

The detailed operation is as follows:

After the Divide instruction, % (AAA) (BBB), has been stored in the proper registers, the execution of the instruction begins by having the contents of the A-STAR and B-STAR regenerated and also transferred to the A-Aux STAR and B-Aux STAR respectively during the first A and B cycles. Thus the address of the units positions of the divisor and the first partial dividend are stored for future reference. During the first A cycle the sign of the divisor is also analyzed and stored in the sign Trigger.

Consecutive A and B cycles are then taken to compare the divisor to the first partial dividend. The comparison is completed during the second B cycle following the detection of a A Register WM which eliminated further A cycles. A number, resulting from the comparison, is added to the quotient digit which is always located during the third B cycle following the A cycle in which the WM detected. The comparison can determine the following conditions:

a. $B < A$ - Dividend less than the divisor.
b. $A \leq B < 2A$ - Dividend is equal to or greater than the divisor but less than twice the divisor.
c. $B \geq 2A$ - Dividend is equal to or greater than twice the divisor.

Based on the above results, the following action takes place.

a. $B < A$ - The quotient digit read out of storage is regenerated into storage and a B cycle is taken to increment the B-Aux STAR by "1" which effects a shift in the dividend. Consecutive A and B cycles are then taken to compare the divisor to a new partial dividend.

b. $A \leq B < 2A$ - The quotient digit read out of storage is increased by "1" and consecutive A and B cycles are taken to subtract once the divisor from the dividend and to make a comparison of the divisor to the reduced dividend.

c. $B \geq 2A$ - The quotient digit read out of storage is increased by "2" and consecutive A and B cycles are taken to subtract twice the divisor from the dividend and to make a comparison of the divisor to the reduced dividend.

The quotient sign is always set with the quotient digit produced by a comparison of the divisor and a partial dividend which includes the units position of the dividend. A minus sign in the units position of the dividend will change the status of the Sign Trigger. The B bit detected in the dividend field will also turn on the End of Divide Latch and a comparison result of $B < A$ when this latch is on will terminate the divide operation.

At the completion of the operation; sign bits appear in the units positions of the quotient and remainder (the sign of the dividend is retained in the remainder) and zeros are written in all positions of the B-field which do not have significant digits. All information in the A-field (divisor) is regenerated. The location of the units position of the quotient is the address of the units position of the dividend minus the positions in the divisor minus 1.

The Overflow Latch is turned on if a carry is generated in a quotient position due to division by zero or if an improper (BBB) address is given such that the first partial dividend is more than 9 times the divisor. A Branch instruction must be initiated to test this latch before the next arithmetic instruction.

S. Bespalko