March 30, 1960

MEMORANDUM TO: Mr. G. E. Bontecou

SUBJECT: Meeting on Improved 1401 Divide

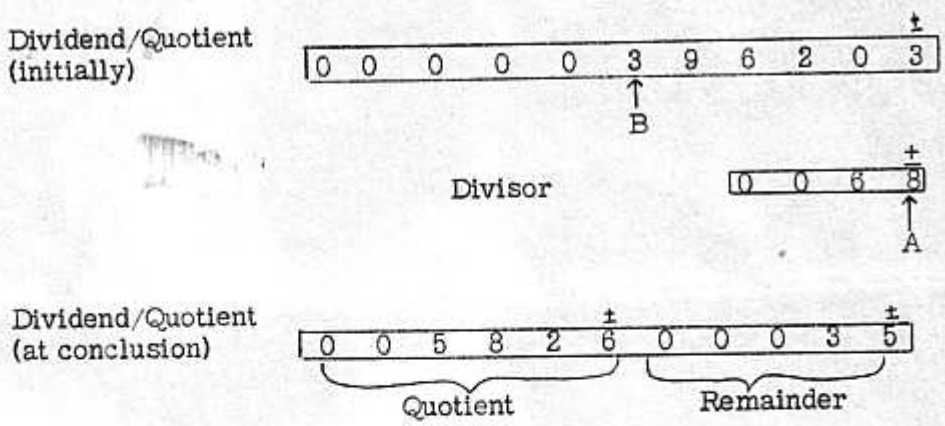
In attendance: Messrs. Bespalko, Grenchus, Smith, Bell, Morrow, McAdon, Carr, Pokoski, and Hanf

A proposal was made to the 1401 engineering group to change the formats used for the Dividend and Quotient fields in the divide instruction. This proposal, if adopted, will require changes in the programming manual but will increase the speed of divide and simplify the divide controls. Billy Carr has stated that the changes are satisfactory from a programming point of view and that the increased speed should compensate for any inconvenience with regard to programs that have already been written.

The proposal is also applicable to the ARS and will result in a substantial cost saving and speed increase as well as eliminating a number of compatibility problems produced by some of the peculiarities of the present 1401 divide instruction.

The dividend and quotient fields can be defined as follows, if the present 1401 convention which allows division by any number different from zero is adhered to:

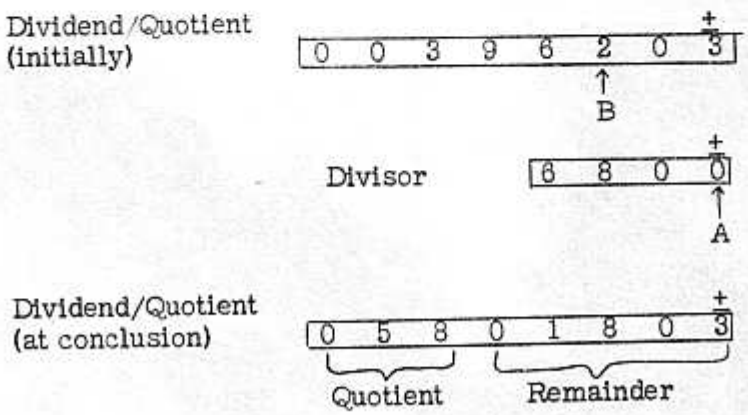
A dividend/quotient field is set up whose length is the length of the divisor plus the length of the dividend plus one. The dividend is located in the low order end of this field with a sign over the units position. The remaining digits of the dividend/quotient field are set to zero (this can be done by reset adding the dividend into the dividend/quotient field). In the divide command, the A address specifies the units position of the divisor (as in the present scheme) and the B address specifies the high order position of the dividend. At the conclusion, the quotient will be located in the high order digits of the dividend/quotient field, will have the same length as the original dividend, and will have its proper sign over units. The remainder will be in the low order digits of the dividend/quotient field and will be smaller in magnitude than the divisor and will have thw same sign as the original dividend. For example, suppose we wish to divide 396, 203 by 0068.



Execution of % (A) (B)

Notice that the result of the division will always be such that if a subsequent multiply was given (specifying the A address and the low order position of the Dividend/Quotient field), the result would be the original dividend less the remainder. Thus, multiply and divide are precise inverses.

If it is known that the divisor does not contain high order zeros, the Dividend/Quotient field can be shortened provided that the B address is always given as a number of positions to the right of the high order position of the Dividend/Quotient field equal to one more than the length of the divisor. The quotient will always be located immediately to the left of the remainder field which is one position longer than thedivisor. For example:



Notice that in this case also, a multiplication (and add in of the remainder) would restore the Dividend/Quotient field to its original value. An overflow can occur only if the Divisor is smaller than the number formed by the digits of the dividend/quotient field which are immediately to the left of the position indicated by the B address. The possibility of such an overflow can always be avoided by setting up the format as shown in the first example and avoiding division by zero.

Further details of the proposed divide operation (such as the final setting of the A and B MARS and the handling of overflows) will be described following a study by engineering of the implementation of this divide scheme.

A minor change to the definition of multiplication was also suggested. This change does not effect the multiply operation in those cases where the 1401 manual is followed: i. e. the product field length is the multiplier length plus the multiplicand length plus one. In fact, the change makes this always true by detecting the units position of the multiplier by means of this rule rather than by looking for a sign bit. It means also that the multiplier does not have to have zone bits over its units position.

Examples of divide times are given below for the present system and the proposed system, assuming the format shown in the first example:

	Present System	Proposed System
20D ÷ 10D	15.00 ms.	12.02 ms.
10D ÷ 5D	4.50 ms.	3.46 ms.
4D ÷ 2D	1.10 ms.	.81 ms.

These times can be reduced even further with the proposed system if the programmer knows something about the value of the divisor and eliminates unnecessary high order zeros in the Dividend/Quotient field.

W. P. Hanf

W. P. Hanf

WPH:mab

CC: Messrs. K. A. Bell
G. M. Benson
S. Bespalko
N. M. Boothe
C. E. Branscomb
B. N. Carr
R. P. Case
P. Fagg
E. J. Grenchus
E. S. Hughes, Jr.
J. J. Ingram
T. M. McAdon
H. W. Morrow
A. M. Nires
J. L. Pokoski
W. A. Reichlen
D. Royse
R. M. Smith
F. O. Underwood
B. E. Updike