May 29, 1957

SIGMA FILE MEMO No. 2

SUBJECT:

Reduction of the Number of Add operations in a multiply operation.

PURPOSE:

The purpose of this memo is to present a method of reorganizing the multiplier in a multiply operation in order to reduce the number of additions required.

BASIC CONSIDERATIONS:

The machine time required to execute a multiply instruction is almost entirely determined by the number of additions required to perform the multiplication. The proposed mantissa for the Stretch machines is 48; therefore an average of 24 additions would normally be expected in a multiply operation. One method of reducing the quantity of additions is to generate sub multiples of the multiplicand and apply these discretely as dictated by the multiplier. For instance, if all of the 16 multiples 0-15 (hexadecimal) of the multiplicand were made available, it would be possible to examine the multiplier four bits at a time and add the particular multiple which is called for by each set of four bits. This would then require only 12 additions for a multiply operation. There are certain methods of reorganizing the multiplier such that the number of additions required is less. For instance multiplying by 11111 is the same as 10000 (-1). That is to say multiplying by 31 is the same as multiplying by 32 and subtracting 1. Therefore 2 additions are required instead of 5 additions. The number 3 (011) and -3 (-011) can also be applied with good results. This of course requires that the multiplication of the multiplicand by 3 be generated and appropriately stored for application as dictated by the multiplier. For instance 1101 becomes 1000 (-3) and 0110 becomes 0030.

METHOD:

One method of reorganizing the multiplier is to scan 4 bits at a time and reorganize these four bits according to the above methods. The reorganized multiplier then dictates the multiplication process. As illustrated by the situation where there is a string of ones, the examination and conversion of these ones will result in what amounts to a carry into the higher order digit positions of the multiplier. The Multiplier-Quotient register therefore has a carry propagation system attached to it such that as the lower order bits are reorganized, the higher order bits are altered by the carries resulting from this reorganization.

A special situation is encountered where, after reorganizing the four bits, a one bit is in the high order position followed by a zero such as in 1000. Instead of performing the indicated addition the bit is saved and included in the examination of the next four bits (making five).

Figure 1 is the truth table of the multiplier digits before and after reorganization. Section 1 contains the 16 possible combinations representing the digits before reorganization. Section 2 contains the results of reorganizing the 4 bits if the bit to the right of these was zero. Section 3 is the reorganization if the bit to the right was a one. In each section is indicated under A, the number, under B the number of additions required, under C the fact that a carry was propagated to higher order bits, and under D the fact that the high order one was saved for reorganization of the next four bits. The average number of additions required for four bits of multiplier without reorganization is two. Column 2B shows an average of 14/16 adds per four bits; and column 3B shows an average of 24/16 adds per 4 bits. An examination of the D columns shows that if we are reorganizing aparticular set of four bits according to section 2, 5/16 of the time we will have to save a one and therefore will be reorganizing the next four bits according to section 3; if we are reorganizing according to section 3, 4/16 of the time we will be obligated to reorganize the following 4 bits according to section 3 again. The probability of reorganizing according to section 3 is 5/17 and the probability of reorganizing according to section 2 is 12/17. The average number of additions required for four bits of multiplier is:

$$5/17 \cdot 24/16 + (1-5/17) \cdot 14/16 \doteq 1.0588$$

For 48 bits the average number of additions will be 12.7 and the maximum number will be 24.

CONCLUSION:

The development of the multiplication of the multiplicand by 3, and the application of 1, -1, 3, and -3 in reorganizing the multiplier, results in the reduction of the average number of additions required for a multiplication operation to 13 and the maximum number to 24. Since the number of shifts is increased, a fast shifting unit is presumed. The scheme requires a slightly complicated multiplier mechanism and what amounts to a programmed multiplication. The reorganizing of the multiplier into 3 bit or 5 bit groups was investigated. These schemes required 12.8 additions.

E. I. Jordan

E. I. Jordan
Engineering Planning

EIJ:bea

Distribution list:

Dr. G. A. Blaauw
Dr. F. P. Brooks
Dr. W. Buchholz
Mr. R. A. Gregory
Mr. J. E. Griffith
Mr. K. W. Kaeli

Mr. S. Pitkowsky
Mr. R. Preiss
Mr. D. Sweeney
Mr. W. Winger
Mr. W. Wolensky

FIGURE I

Table for Reorganizing Multiplier by Applying 1, -1, 3, and -3.

1		2.				3			
A	B	A	B	C	D	A	B	C	D
0000	0	0000	0			00001	1		
0001	1	0001	1			00003	1		
0010	1	0010	1			00101	2		
0100	1	0100	1			01001	2		
1000	1	*000	0		X	*0001	1		X
0011	2	0003	1			00103	2		
0110	2	0030	1			*000-3	1		X
1100	2	0300	1			03001	2		
1101	3	000-3	1	X		03003	2		
1011	3	*003	1		X	0300-1	2		
1110	3	00-10	1	X		0000-3	1	X	
0111	3	*00-1	1		X	*000-1	1		X
1111	4	000-1	1	X		0000-1	1	X	
0101	2	*00-3	1		X	01003	2		
1010	2	00-30	1	X		0300-3	2		
1001	2	*001	1		X	*0003	1		X
Total	32		14		5		24		4

* This multiplier position is not determined until the next 4 bits have been examined.

1. Original multiplier.
2. Multiplier reorganized if a zero appeared in previous high order position.
3. Multiplier reorganized if a one was saved from previous 4 bits.

A. 4 bits of multiplier.
B. No. of additions called for.
C. Carry propagated to higher order positions.
D. High order one saved for next reorganization.