

## Proposed Modification Of Floating-Point Multiplication

The University of Michigan Computing Center

Leonard J. Harding, Jr.

The MD and MDR instructions currently provide for a 14 hexadecimal digit intermediate product; consequently, when postnormalization is necessary the fourteenth digit of the result will always be zero. It would appear that this situation is not necessary, and that minor alterations in the ROS program of the Model 67 will provide for a fifteenth digit in the intermediate product.

Currently, the fifteenth digit of the product is lost two bits at a time. The lower two bits are "shifted" off the end of the PAL, and hence, B register extension. This occurs when the R4 shift is performed after the last multiple selected by the SEL-MPL\*E3 micro-order has been added into the partial product (at 14C). The high order bits of the fifteenth digit are not lost, but remain unused in B(64-65). This is because postnormalization is based on the contents of DT rather than those of AB (at 78B).

A method for retaining the fifteenth digit of the product is thus clear. It suffices to delay the issuance of the MPLY-END micro-order one CPU cycle in order that the partial product resulting from the last SEL-MPL\*E3 micro-order may

be shifted L2 before it is shifted R4. So that the MPLY-END will function properly, this necessitates that the status of the TX-trigger be retained through this additional cycle. If postnormalization is then based on the AB register, the desired result is obtained.

The above modification is realized in the enclosed patch of the microprogram as we know it. The appropriate page from the CAS that we are working from is included for identification purposes. This patch lengthens the existing program by one ROS word. Since this section of the ROS microprogram is common to both short and long operand multiplies, the execution times of all four floating point multiplies will be increased by one CPU cycle, i.e., 200 ns. This represents about a 5% increase for the ME and MER, and less than 3% for the MD and MDR instructions.

It will be noted that the proposed "patch" decrements the AB-byte counter to 7 and later restores it to zero. This manipulation of the AB-byte counter will prevent the reset of the TX-trigger that would normally occur during clock time of the cycle in which the ROS word in 14C is in control. This AB-byte counter control of the TX-trigger was set up for the extended precision arithmetic package that has been RPQ-ed into our Model 67's.

14C

A	ABL2+0
A	→AB
C	ABC-1
R	

\*  
LAST MULTIPLE ADDED TO PARTIAL PRODUCT. MUST SHIFT R2. SET AB-BYTE COUNTER TO 7 TO SAVE TX-TRIGGER

NEW

A	ABL2+0
A	R→AB
C	MPLY-END
R	

\*  
ACCOMPLISHES R2 SHIFT WITHOUT LOSING THE LAST TWO PRODUCT BITS. SELECT +DT IF THE TX-TRIGGER IS ON.

24E

A	AB+0
A	→AB,DT
C	ABC+1
R	

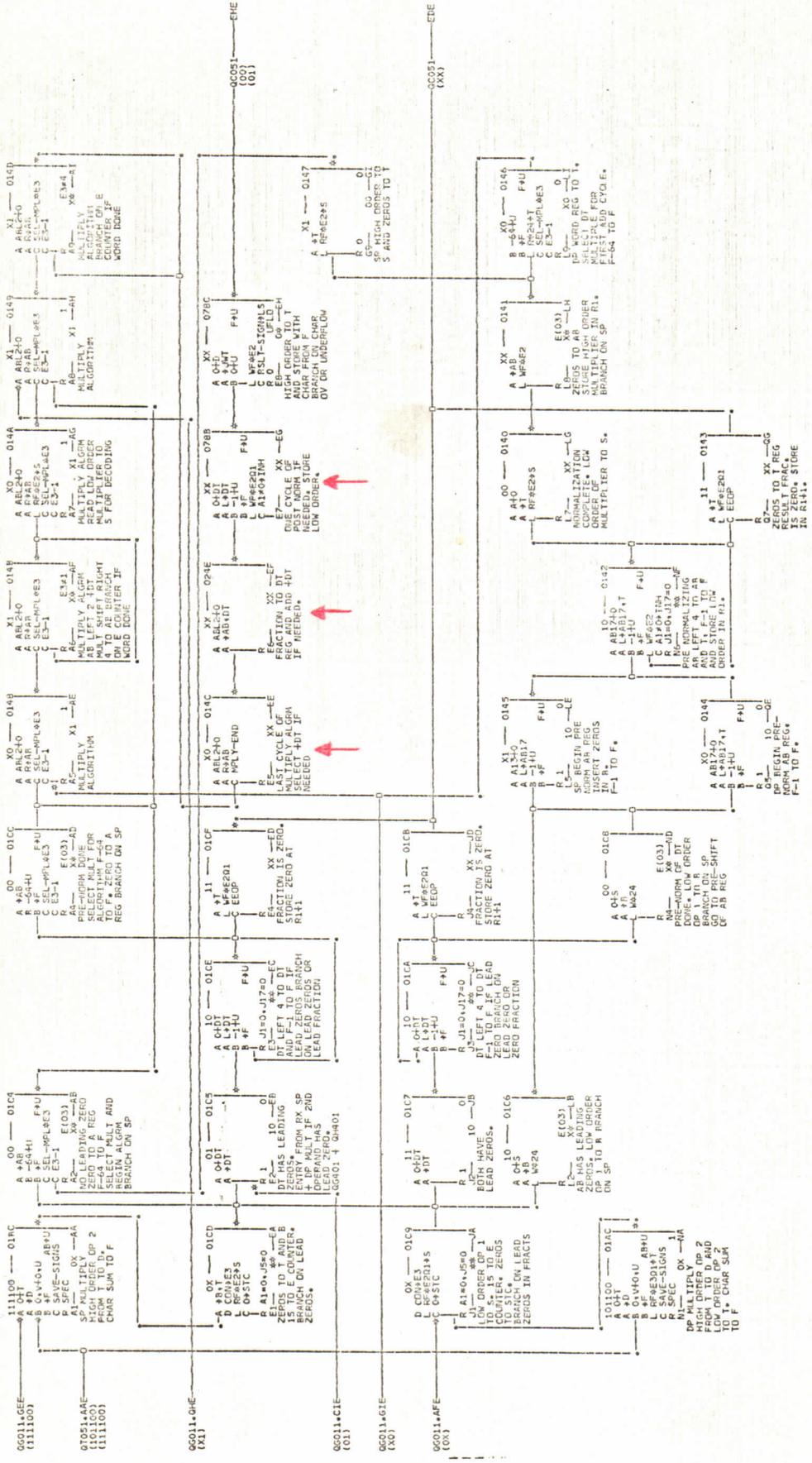
\*  
15 HEXADECIMAL DIGIT INTERMEDIATE PRODUCT IS IN AB(04-67), 14 DIGIT VERSION IS IN DT. SET AB-BYTE COUNTER BACK TO ZERO.

78B

A	AB+0
A	L→DT
B	-1+U
B	F
L	WF*E2Ω1
C	A1#0→INH
R	F→U

\*  
POSTNORMALIZE FROM AB IF NECESSARY. STORE THE LOW ORDER WORD OF THE RESULT.

SUGGESTED REVISION OF THE ROS PROGRAM FOR FLOATING-POINT MULTIPLICATION



705059 07/21/65 MACH 100031  
 DATE 07/27/65 SHEET 1  
 LDC 6.31 VERSION  
 C2056 MACH 100031  
 MULTIPROG 100031  
 SLD 100031  
 FLT. PT. MULTIPLY

If this "patch" does not function properly or in some way interacts incorrectly with other parts of the micro-program, other modifications that would provide a guard digit will be considered.

dh