September 27, 1956

PROJECT STRETCH
LINK COMPUTER MEMO NO. 10

TITLE: Floating Point Definition

BY: F. P. Brooks

As a result of conferences between Messrs. W. Wolensky and F. P. Brooks, and the specifications suggested at the meeting with LASL representatives on September 19th, the following definitions of floating point operation in the Link and Stretch are proposed:

1) A floating point number shall be represented by an exponent sign, an exponent magnitude, a fraction sign, and a fraction magnitude.

2) Simple logical addition and subtraction shall be able to modify the exponent and its sign in such a manner as to effect multiplication and division of floating point numbers by powers of two for all floating numbers. Alternatively, a special mode of floating shift that suitably alters exponents shall be provided.

3) For exponent magnitude, at least eight and not more than twelve bits are needed. While the smaller exponent range forces scaling in some few problems, it shortens considerably the time required for a program error or an instability in mathematical method or model to show itself by overflow or underflow. It appears that the seven bit magnitude ($10^{\pm 38}$) presently available in the 704 is satisfactory for most LASL problems. The exponent should express a power of two, since this simplifies circuitry for comparison and arithmetic operations.

4) Underflow conditions (exponent sign negative, exponent magnitude overflow) shall set a selector. It shall be possible to cause break-in (Switch of program control to another instruction counter) at the time of underflow selector transfer without a programmed test in the operating program. This offers to the programmer the options of transferring to an error routine immediately upon underflow or of changing the underflowed quantity to zero.

5) Overflow conditions (exponent sign positive, exponent magnitude overflow) shall set a separate selector which shall have the same control powers as the underflow selector.

6) All words with a zero fraction shall be treated as mathematical zeros, satisfying these equations in all floating point operations and all comparisons with all floating point numbers:

For all $X \neq O$

$$X \pm O = X$$
$$X \cdot O = O$$
$$O/K = O$$

$$-|X| < -O = +O < |X|$$

In order that these conditions be satisfied, it appears desirable to let one bit indicate a zero fraction, and to set this bit when a word with zero fraction leaves the arithmetic unit after a floating point operation. This should permit time saving on additions, multiplications, or divisions of zero, and would simplify comparisons.

7) An attempted division by zero shall set a selector other than those provided for overflow and underflow, and this selector shall have the same break-in property that those do.

8) A set of unnormalized floating point operations shall be provided to aid fixed floating conversions and special operations.

9) The standard floating point operations shall provide full normalization after each operation, so that floating point numbers are ordinarily stored in fully normalized form. LASL representatives pointed out that unnormalized storage with left and right shifting before floating additions, etc., might permit time savings in some common problems. These savings are on the assumption that the time required for shifting and normalization will be linear with the number of places to be shifted. The possible time savings on accumulation are somewhat offset by possible time losses on comparisons, multiplications, and divisions, and by circuitry complications. It does not appear that shifting in the Link and Stretch will necessarily depend linearly upon the number of places to be shifted, and the possible net time savings of the semi-normalized mode of operation does not appear to be worth the complications introduced.

10) If floating point addition is accomplished in such a way that a single-word addend can be summed with a two-word augend in the central registers to provide a two-word sum, provision of this type of operation will simplify programming of long accumulations. These at present demand a consideration of the order of addends.