STRETCH ARITHMETIC

Registers. At the beginning of arithmetic operations the operands involved are assumed to be in the registers S and A. The result is now to be formed in the adders and other circuitry and then placed in A. The basic adder is assumed to be of single precision length. The A'-register is introduced mainly to preserve the A-operand for checking purposes. The registers may be presented in the following form with floating point arithmetic in mind. The high order part of A and A', i.e.. the exponent part and the first mantissa part, will be denoted by A_h and A'_h, and the exponent combined with the last mantissa by A_l on A'_l.

Bits	11	45	45	Add	Mpy	Div
S	i,s,σ,e_S	m_S		Augend	Mplr	Divr
A	i,s,σ,e_A	m_A	$\bar{m}_A$	Addend Sum	Mplcd Product	Divd Quotient
A'	$i,s,\sigma,e_{A'}$	$m_{A'}$	$\bar{m}_{A'}$	.......Previous A.......		
				—	—	Rdr

It is assumed that we have Load to and Store from S, A_h or A'_h, with or without clearing the corresponding low order mantissa, also that we have full sign control on S or A for these operations as well as those given below. We may want to extend this to read "on S and A and on the result in A" or take some intermediate position.

Normalized operations. The three basic single-precision normalized arithmetic operations are defined below. In the notation we assume that e and m include their respective signs.

Addition (A and S) (1) Examine operands for exceptions. (2) Form $p = e_S - e_A$. If p is negative interchange A_h and S and set $\bar{m}_A = 0$. (3) Send A to A'. Shift $(m_A, \bar{m}_A)$ by min $(/p/, 90)$ steps to the right. Set $e_A = e_S$. (4) Add A and S mantissas to form the correct double precision result in A. (5a) If sum overflows "adjust by one" as for the 704. Examine for ∞-exception. If $m_A = 0$ recognize a 0-exception. (5b) Normalize $(m_A, \bar{m}_A)$ to eliminate leading zeroes. Examine for underflow exception.

Multiplication (M) (1) Examine operands for exceptions. (2) Send A to A'. Set $e_A = e_S + e_A$. Examine for exceptions. (3) Multiply m_S and m_A to form a double precision product in A. (4) If m_A has a leading zero adjust by one. Examine for exception.

Division (D) (1) Examine operands for exceptions. (2) If $/m_A/ \geq /m_S/$ shift A mantissas one step to the right and increase e_A by unity. Examine for exception. (3) If $/m_A/ \geq /m_S/$ recognize an ∞-exception. Otherwise form $e_A = e_A - e_S$. Examine for exceptions. (4) Form the correct single precision quotient in A_h and the remainder in $\bar{m}_A$. Shift $\bar{m}_A$ to $\bar{m}_{A'}$.

Unnormalized Operations (AU, SU, MU). Include the three unnormalized operations present in the 704, with AU and SU similar to A and S except that step (5b) is omitted, and with MU similar to M except that step (4) is omitted. These orders have many special purpose applications and are cheap.

Positioning Operations. Include a Position instruction where the "long address" (see Addressing) specifies the location of desired exponent e'. (1) Form $p = e' - e_A$. If p is positive, shift the mantissas right p steps. Set $e_A = e'$. Examine for $m_A = 0$. (2) If p is negative, shift left $/p/$ steps but not beyond normalization. Set $e_A = e'$ or if the full shift is not possible to the e corresponding to the normalized result. In the latter case set a special shift trigger to signal "full positioning not possible".

We also want the special positioning instruction "Normalize".

With these instructions one can simulate fixed point work including the shifting normally associated with it. In addition, positioning has many other special applications, e.g., separating integral and fractional parts of a number.

Double precision. If we include a Load to and a Store from A_1 and A'_1 and associate the Store with an exponent reduction of 45 we will have double precision facilities in Stretch at least equivalent to those in the 704. Since double precision work will be rare it is not worth going beyond this except to include inexpensive facilities which obviously will benefit multiple precision programming in general. The reduction of 45 connected with the Store should be associated with examination for O-exceptions.

Significance. To get some sort of hold on the significance of results it is proposed that Stretch be capable of operating in a "significance mode" which would alter the result of at least those instructions which may introduce lead zeroes. Specifically one would modify step (5b) under floating point addition to read as follows: Complement bit # q in m_A and then perform the normalization, etc. On entering the mode q would be specified by the programmer. We intend to test out the usefulness of this idea on some of our 704 programs.