FLOATING-POINT ARITHMETICAL METHODS

The methods used in the 803B to perform arithmetical operations in the floating-point mode are described. Familiarity with fixed-point methods is assumed.

1. FLOATING-POINT REPRESENTATION

In the conventional fixed-point representation used in the 803, numbers are held in the range $1 > x \geq -1$, using twos complement representation for negative values of x. One disadvantage of a fixed-point representation is that the precision of any number is dependent upon its scale, being proportionately less for small numbers than for large within the range. Also the programmer must keep a careful watch on the scale of the numbers he is using, lest overflow occur.

Both disadvantages are to a great extent overcome if floating-point representation is used, in which a binary number x is expressed as $x = a.2^b$, where a is often called the mantissa and b the exponent. The mantissa is a fixed-point number which contains the significant digits of x, and the exponent is another fixed-point number which in effect indicates the true position of the binary point in x: that is, b indicates the scale of a.

1.1 Floating-point Numbers in the 803B

When an 803B word of 39 digits represents a floating-point number, the mantissa occupies the most significant 30 places and the exponent occupies the remaining nine places. Since the radix is always 2 it need not be explicit.

The mantissa is an ordinary fixed-point number, with a sign digit which is zero if the mantissa is positive and one if it is negative. In order that the maximum number of 30 significant digits may always be present in the mantissa it is arranged always to be kept as large as possible. The standard range for a non-zero mantissa is therefore

$1 > a \geq 1/2$ if it is positive

and $-1/2 > a \geq -1$ if it is negative

After every arithmetical operation the mantissa is shifted left or right as required ("standardised") to bring it to the standard range.

Standardising the mantissa necessitates a compensating adjustment to the value of the exponent which must be reduced if the mantissa is shifted left, or increased if the mantissa is shifted right, one being subtracted from or added to the exponent for each place the mantissa is shifted.

There are nine exponent digits available, so that if the normal fixed-point representation were used the range would be

$$255 \geq b \geq -256$$

A negative exponent expressed in this way is inconvenient, as will shortly be explained, and hence all exponents are treated as if they were greater by 256 than their real value. Thus the range actually used is

$$511 \geq b' \geq 0$$

Since all exponents are treated in the same way, this "displaced zero" gives little trouble.

The reason for avoiding the use of negative exponents is that the floating-point representation of zero is most conveniently made the same as the fixed-point representation. A zero mantissa cannot be standardised, since no amount of left shift will alter it, and hence zero is represented as 0.2^{-256}, that is, a completely zero word.

2. ARITHMETICAL OPERATIONS

In this section we shall write the floating-point number $a.2^b$ as a;b, where a and b are assumed to be in the conventional 803B form described above.

2.1 Addition and Subtraction

Two floating-point numbers cannot be added or subtracted unless their exponents are equal. In the general case, therefore, the exponents must be equalised before addition takes place. So that the mantissae remain within range (as fixed-point numbers) it is convenient to shift right the mantissa having the smaller exponent so that the exponent, which must be increased to compensate for the right shift, becomes equal to that of the other number. Thus if the numbers are $a_1;b_1$ and $a_2;b_2$ where $b_1 > b_2$, a_2 is divided by 2^n such that $b_2 + n = b_1$. The exponent of the result is b_1, provided that $a_1 \pm a_2 2^{-n}$ is still in standard form.

Note that the maximum scaling shift allowed is of 31 places since if the exponent difference is greater than this the smaller number cannot affect the result. In fact, 29 places would suffice, but 31 is simpler to detect.

2.1.1 Standardisation of the Result

There are three possible cases. If the exponent difference $b_1 - b_2$ is one or zero the result mantissa may be very small. Taking five-digit mantissae as an example, consider 0.1000;b - 0.1111; b-1. Equalising the exponents gives 0.1000;b - 0.01111;b = 0.00001;b.

A four-place standardising shift is needed, giving 0.1; b-4. When a 30-digit mantissa is used, the standardising shift needed may be up to 29 places.

If the exponent difference is more than one the maximum standardising shift is one place:

0.1000; b - 0.1111; b-2

= 0.10000; b - 0.001111; b

= 0.010001; b

= 0.10001; b-1

It is also possible for the result mantissa to be out of range. Thus, for example, 0;1 b + 0.1; b = 1.0; b. This can be standardised by a right shift of at most one place, giving 0.1; b+1. To assist in detecting this condition, the sign digits of the mantissa are duplicated before addition or subtraction. Mantissa overflow can only affect the 2^0 digit, so if the 2^0 and 2^1 digits of the result differ the mantissa is shifted one place to the right.

2.2 Multiplication

It will be remembered that in the fixed-point multiplication process of the 803B the partial product is built up in its correct position without progressive shifting so that the operation may be terminated as soon as all significant multiplier digits have been used. When standardised mantissae are used, there can be no non-significant digits in the multiplier and hence there is no advantage in using the fixed-point method of multiplication.

In the multiplication of floating-point numbers, the exponents are added. Since the conventional exponent is greater by 256 than its correct value, it is necessary to subtract 256 from the sum of the exponents to give the correct representation.

The principle underlying the method used to multiply the mantissae is similar to that of fixed-point multiplication except that to save time three digits of the multiplier are examined at each step, instead of two. This is equivalent to performing two stages of the usual method at once, and the partial product is built up according to the following scheme, where d_{n+1}, d_n and d_{n-1} are the digits of the multiplier, P is the partial product and M is the multiplicand.

d_{n+1}	d_n	d_{n-1}	Form
0	0	0	P + 0
0	0	1	P + M
0	1	0	P + M
0	1	1	P + 2M
1	0	0	P - 2M
1	0	1	P - M
1	1	0	P - M
1	1	1	P - 0

The following rules are deduced from the above scheme:

(a) if $d_{n+1} = 0$, add
if $d_{n+1} = 1$, subtract

(b) if $d_n \neq d_{n-1}$, add or subtract M

(c) if $d_{n+1} \neq d_n$ and $d_n = d_{n-1}$, add or subtract 2M.

Since two steps are carried out at once, the partial product and the multiplier are shifted two places after each operation. The sign digit of the multiplier is examined once only, and at the last stage the shift of the product should be of one place instead of two. In 803B practice it is simpler to shift two places, and compensate in the following standardising shift.

2.2.1 Standardisation of the Result

In general, standardisation may require a shift of one place in either direction, or of no places. A right shift is only needed if the mantissae are both equal to -1.

Since the product mantissa is in fact shifted one extra place to the right, the standardising shift is of 0, 1 or 2 places to the left. To allow for a possible two-place shift, two extra low-order digits are retained in the mantissa during the formation of the partial product. Less significant digits are lost, but they may affect round-off (section 2.5). The exponent is corrected for the excess mantissa shift by subtracting 255 instead of 256 initially.

Mantissa overflow is possible due either to -1×-1, or to the fact that the doubled multiplicand may be added or subtracted. Two extra sign digits are therefore used to permit the correction of this overflow.

2.3 Division

The division process used is the same as in fixed-point operation. There are three slight practical differences however, the first being that it is not possible to require that the divisor mantissa shall be greater than the dividend mantissa. To avoid this restriction, the dividend mantissa is shifted one place to the right before division begins.

The second difference is that the error introduced in the fixed-point method due to the conventional positive sign of zero is overcome in floating-point division by treating a zero remainder as if it had the same sign as the divisor. The third difference is a consequence of the second: if the quotient is exact the remainder will be zero, and when this is found the process can be terminated.

In floating-point division the exponent of the divisor is subtracted from that of the dividend. To preserve the conventional representation, 256 should then be added to the quotient mantissa. But since the dividend mantissa is halved it is necessary to add 257.

2.3.1 Standardisation of the Result

To standardise the quotient of two standardised mantissae may require a shift of one place either way, or of no places. The left shift is required only when the dividend is $+1/2$ and the divisor is -1. Since the dividend is halved before division begins, the standardising shift is of 0, 1 or 2 places to the left. One extra quotient digit is formed (making 31 in all, including sign), to allow for standardisation. It is not necessary to form two extra digits since in the one case requiring a two-place shift all but the three most significant quotient digits are zero.

Mantissa overflow cannot occur.

2.4 Fixed point to Floating-point Conversion

The "standardise" instruction in the 803B is used to convert the representation of a number from fixed-point to floating-point.

The number in the accumulator is treated as a signed integer. This is equivalent to a 39-digit mantissa with an exponent equal to 38. An initial exponent equal to 38 + 256 is therefore formed, and the number is shifted until the sign and the next significant digits differ. The number of places shifted are counted in the usual way and subtracted from the initial exponent. Only 30 digits are retained in the mantissa.

2.5 Round-Off

The result mantissa of all floating-point arithmetical processes in the 803B, including conversion from fixed-point, is rounded to 30 digits. Any ones in the mantissa of order 2^{-30} or less cause the 2^{-29} digit to be made a one. This is, so to speak, a logical round-off, not an arithmetical one, and is very simple to perform. To prevent rounding errors being increased, it is necessary to perform round-off after standardisation, and it should be noted that rounding cannot make the standardisation incorrect or cause exponent overflow.

In addition and subtraction, round-off will occur if there are any ones shifted below 2^{-29} during the initial scaling of the mantissae.

In multiplication a round-off digit is inserted if any digit one is formed in the less significant part of the product.

In division the least significant digit of the quotient is always made a one unless the quotient is exact.

In fixed-point conversion any digit one of order 2^{-30} or less remaining after standardisation of the mantissa will cause the 2^{-29} digit to be made a one.

3. SPECIAL CASES

Three possible special cases exist. If the mantissa of the result is zero the exponent must be made zero too, to give the correct representation.

If the exponent result becomes negative (i.e. if the true exponent is less than -256), the result is too small to be represented and both the mantissa and exponent are made zero. This condition is known as exponent underflow.

If the exponent becomes greater than 511 (i.e. if the true exponent is greater than 255), the result is too great to be represented. In this case the 803B stops. This condition is known as exponent overflow, and it may be noted that since an exponent greater than 511 appears to be negative the observed result in cases of exponent overflow will have a small exponent.

Temporary underflow and overflow can occur during arithmetical operations, an exponent within range being obtained when later corrections are made. In such cases, the actions described above do not take place, being only initiated when the exponent of the final result is out of range.