Chapter 4. General Requirements

Table of Contents

4.1. Data Formats

The instruction set shall support 16-bit fixed point single precision, 32-bit fixed point double precision, 32-bit floating point and 48-bit floating point extended precision data in 2's complement representation.

4.1.1. Single Precision Fixed Point Data

Single precision 16-bit fixed point data shall be represented as a 16-bit 2's complement integer number with the most significant bit (MSB) as the sign bit:

```
 MSB                              LSB
-----------------------------------
| S|                               |
-----------------------------------
  0  1                              15
```

Examples of single precision fixed point numbers are shown in Table I.

Table I. Single Precision Fixed Point Numbers

Integer	16-Bit Hexadecimal Word
32767	`7 F F F`
16384	`4 0 0 0`
4096	`1 0 0 0`
2	`0 0 0 2`

Integer	16-Bit Hexadecimal Word
1	0 0 0 1
-1	F F F F
-2	F F F E
-4096	F 0 0 0
-16384	C 0 0 0
-32767	8 0 0 1
-32768	8 0 0 0

4.1.2. Double Precision Fixed Point Data

Double precision 32-bit fixed point data shall be represented as a 32-bit 2's complement integer number with the most significant bit (MSB) of the first word as the sign bit.

```
 MSB                                                            LSB
------------------------------------------------------------------
| S|            (MSH)              |              (LSH)          |
------------------------------------------------------------------
  0  1                          15 16                             31
```

Examples of machine representation for double precision fixed point numbers are shown in Table II.

Table II. Double Precision Fixed Point Numbers

Integer	32-Bit Hexadecimal Word
2,147,483,647	7 F F F F F F F
1,073,741,824	4 0 0 0 0 0 0 0
2	0 0 0 0 0 0 0 2
1	0 0 0 0 0 0 0 1
0	0 0 0 0 0 0 0 0
-1	F F F F F F F F
-2	F F F F F F F E
-1,073,741,825	C 0 0 0 0 0 0 0
-2,147,483,647	8 0 0 0 0 0 0 1

Integer	32-Bit Hexadecimal Word
-2,147,483,648	8 0 0 0 0 0 0 0

4.1.3. Fixed Point Operands

All operands for fixed point adds, subtracts, multiplies and divides are integer. A fixed point overflow shall be defined as arithmetic overflow if the result is greater than $7FFF_{16}$ or less than 8000_{16} for single precision and greater than 7FFF $FFFF_{16}$ or less than 8000 0000_{16} for double precision.

4.1.4. Results on Fixed Point Overflow

On fixed point operations which cause overflow, the operation shall be performed to completion as if the MSBs are present and the 16 LSBs for single precision or the 32 LSBs for double precision shall be retained in the proper register(s). Division by zero shall produce a fixed point overflow and return results of all zeros.

4.1.5. Floating Point Data

Floating point data shall be represented as a 32-bit quantity consisting of a 24-bit 2's complement mantissa and an 8-bit 2's complement exponent.

```
 MSB                                          LSB  MSB          LSB
-------------------------------------------------------------------
| S|                     Mantissa                 |    Exponent    |
-------------------------------------------------------------------
  0   1                                         23  24            31
```

Floating point numbers are represented as a fractional mantissa times 2 raised to the power of the exponent. All floating point numbers are assumed normalized or floating point zero at the beginning of a floating point operation and the results of all floating point operations are normalized (a normalized floating point number has the sign of the mantissa and the next bit of opposite value) or floating point zero. A floating point zero is defined as 0000 0000_{16}, that is, a zero mantissa and a zero exponent (00_{16}). An extended floating point zero is defined as 0000 0000 0000_{16}, that is, a zero mantissa and a zero exponent. Some examples of the machine representation for 32-bit floating point numbers are shown in Table III.

Table III. 32-Bit Floating Point Numbers

Decimal Number	Hexadecimal Notation
	Mantissa Exp
0.9999998×2^{127}	7FFF FF 7F
0.5×2^{127}	4000 00 7F

Decimal Number	Hexadecimal Notation
	Mantissa Exp
0.625×2^4	5000 00 04
0.5×2^1	4000 00 01
0.5×2^0	4000 00 00
0.5×2^{-1}	4000 00 FF
0.5×2^{-128}	4000 00 80
0.0×2^0	0000 00 00
-1.0×2^0	8000 00 00
$-0.5000001 \times 2^{-128}$	BFFF FF 80
-0.7500001×2^4	9FFF FF 04

4.1.6. Extended Precision Floating Point Data

Extended floating point data shall be represented as a 48-bit quantity consisting of a 40-bit 2's complement mantissa and an 8-bit 2's complement exponent. The exponent bits 24 to 31 lay between the split mantissa bits 0 to 23 and bits 32 to 47. The most significant bit of the mantissa is the sign bit 0, and the least significant bit of the mantissa is bit 47.

```
-----------------------------------------------------
|S|   Mantissa MS       |Exponent|  Mantissa LS    |
-----------------------------------------------------
 0 1                  23 24     31 32             47
```

Some examples of the machine representation of 48-bit extended floating point numbers are shown in Table IV.

Table IV. 48-Bit Extended Floating Point Numbers

Decimal Number	Mantissa (MS)	Exp	Mantissa (LS)
0.5×2^{127}	400000	7F	0000
0.5×2^0	400000	00	0000
0.5×2^{-1}	400000	FF	0000
0.5×2^{-128}	400000	80	0000
-1.0×2^{127}	800000	7F	0000
-1.0×2^0	800000	00	0000
-1.0×2^{-1}	800000	FF	0000

Decimal Number	Mantissa (MS)	Exp	Mantissa (LS)
-1.0×2^{-128}	800000	80	0000
0.0×2^{0}	000000	00	0000
-0.75×2^{-1}	A00000	FF	0000

For both floating point and extended floating point numbers, an overflow is defined as an exponent overflow and an underflow is defined as an exponent underflow.

4.1.7. Floating Point Operands

All operands for floating point instructions must be normalized or a floating point zero. A floating point overflow shall be defined as exponent overflow if the exponent is greater than $7F_{16}$. The results of an operation which causes a floating point overflow shall be the largest positive number if the sign of the resulting mantissa was positive, or shall be the smallest negative number if the sign of the resulting mantissa was negative. Underflow shall be defined as exponent underflow if the exponent is less than 80_{16}. The results of an operation which causes a floating point underflow shall be floating point zero. Separate interrupts are set for overflow and underflow. Only the floating point instructions shall set the underflow interrupt.

4.1.8. Truncation of Floating Point Results

All floating point results shall be truncated toward negative infinity.

4.1.9. Results of Division

The sign of any non-zero remainder is the same as the dividend for all division instructions; the remainder is only accessible for single precision integer divides with 16 bit dividends and for single precision integer divides with 32 bit dividends.

Prev Home Next
Definitions Instruction Formats