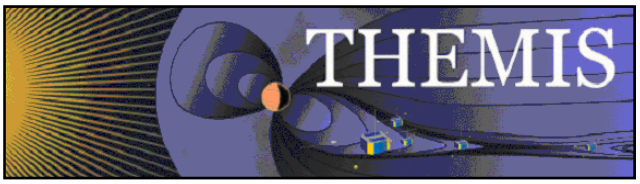


1. Fast Floating Point (FFP.A)

The FFP module is over two decades old and has over four decades of on-orbit history. The source code and description was written by UCB for the AMPTE and CRRES projects. Upon delivery it was then classified by the US government in 1986 until successfully appealed by the University of California. It has been flown on AMPTE, CRRES, Polar EFI, and the four Cluster II spacecraft.

The FFP module meets the needs of the instrument in performing on-orbit data analysis of the DC electric and magnetic fields. Specifically, the module supports the Sine Wave least squares fit subroutine with sufficient range and precision of floating point. The module is able to perform calculations at rates included below.

1.1 Dynamic Range and Precision

The format of the FFP data is Sign(s), a 7-bit Exponent (E) and 16-bits of Mantissa (HL) as described below:

SEEEEEE	HHHHHHHH	LLLLLLLL

Table 21-1. Fast Floating Point Format

The Exponent shall be in Excess-64 format. The sign bit S will indicate negative numbers using a 1. The HL bits shall be fully expressed, with no hidden bit.

1.2 Performance

The module shall perform calculations according to the following table:

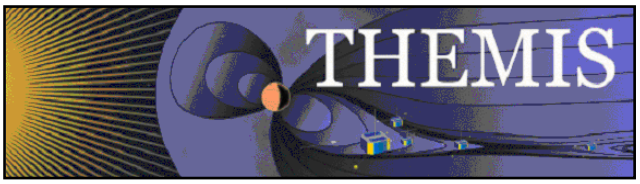
FUNCTION	Minimum Cycles	Average Cycles	Worst Case (Cycles)
FADD	76	300	465
FSUB	97	400	716
FMUL	48	600	1003
FDIV	48	1600	2030
MU21	197	250	298

Figure 21-2. Fast Floating Point Execution Times

The above data were created using software simulations and are useful for estimating the amount of time it will take for the CPU to calculate formulae. To convert these cycle times into microseconds on the 4MHz 8085 (2 MHz cycle-time), multiply by 0.50. Thus, a multiply is 1003*.5=500 usec.

1.3 Protocols

The module uses the following interface protocols. The current value is always held in registers [CDE] and the second parameter is addressed by the [HL]



register pair. When floating values are stored in memory, they are stored with the exponent byte first and the low mantissa last.

1.4 Functions Provided

The module shall provide the following functions:

LODFP	Loads [CDE] from memory at [HL]
STOFP	Stores [CDE] to memory at [HL]
FMUL	Multiplies [CDE] by the value at [HL], leaving the result in [CDE]
FDIV	Divides [CDE] by the value at [HL], leaving the result in [CDE]
FADD	Adds [CDE] to the value at [HL], leaving the result in [CDE]
FSUB	Subtracts the value at [HL] from [CDE], leaving the result in [CDE]
FCMP	Compares the value at [HL] to [CDE], returning carry & zero flags.
FNEG	Negates the value in [CDE]
FLT32	Floats a signed 32-bit value in [DEHL] leaving the result in [CDE]
FIX32	Fixes a floating value in [CDE] leave the signed result in [DEHL]
FSQUA	Squares the value in [CDE]
FSQRT	Takes the Square Root of [CDE]
MU21	Fast fixed point 8-bit by 16-bit unsigned multiply.

Underflow and overflow conditions are treated by returning zero and maximum values, respectively.