Applesoft: Internals--Floating Point Math Package (2 of 3)

```
Revised:        11/7/84
Security:       Everyone
```

Applesoft: Internals--Floating Point Math Package (2 of 3)

===

FLOATING POINT REGISTER MOVE ROUTINES

MOVFM   EAF9
Moves the number in memory Y,A points to into FAC. On exit A and Z reflect
FACEXP.

MOV2F   EB1E
Pack FAC and temporarily moves it into register 2. On exit A and Z reflect
FACEXP.

MOV1F   EB21
Pack FAC and moves it temporarily into register 1. On exit A and Z reflect
FACEXP.

MOVML   EB23
Pack FAC and move it into zero page area starting at X. On exit A and Z
reflect FACEXP.

MOVMF   EB2B
Pack FAC and move it into memory pointed to by Y,X. On exit A and Z reflect
FACEXP.

MOVFA   EB53
Move ARG into FAC. On exit A and Z reflect FACEXP.

MOVAF   EB63
Move FAC into ARG. On exit A and Z reflect FACEXP.

CONUPK   E9E3
Load ARG from memory pointed to by Y,A. On exit A and Z reflect
FACEXP.

SUMMARY OF MOVES:

 FAC   => (Y,A) EB2B   MOVMF
 FAC   => (0,X) EB23   MOVML

```
 FAC   => TEMP1  EB21    MOV1F
 FAC   => TEMP2  EB1E    MOV2F
 FAC   =>  ARG   EB63    MOVAF


 ARG   =>  FAC   EB53    MOVFA


 (Y,A) =>  FAC   EAF9    MOVFM
 (Y,A) =>  ARG   E9E3    CONUPK
```

FLOATING POINT OPERATORS

The following routines require that A and Z reflect FACEXP. Even though FAC
move routines set up A and Z, a LDA $9D will insure their proper values.

```
FMULT           E97F
```
Multiply the FAC by the number in memory pointed to by Y,A.

```
FMULTT          E982
```
Multipy FAC and ARG. On entry A and Z must reflect FACEXP.

```
FDIV            EA66
```
Divide the number in memory pointed to by Y,A by FAC.

```
FDIVT           EA69
```
Divide ARG by FAC. On entry A and Z must reflect FACEXP.

```
FADD            E7BE
```
Add the number Y,A points to in memory to FAC.

```
FADDT           E7C1
```
Add FAC and ARG. On entry A and Z must reflect FACEXP.

```
FSUB            E7A7
```
Subtract FAC from the number Y,A points to in memory.

```
FSUBT           E7AA
```
Subtract FAC from ARG. On entry A and Z must reflect FACEXP.

TRANSCENDENTAL OPERATORS

```
FPWRT           EE97
```
Raise ARG to the FAC power. On entry A and Z must reflect the value of
FACEXP.

```
SQR             EE8D
```
Takes the square root of FAC

```
LOG             E941
```
Log base e of FAC

```
EXP             EF09
```
Raise e to the FAC power

```
COS             EFEA
Returns the cosine of the FAC in radians.


SIN             EFF1
Returns the sine of the FAC in radians.


TAN             F03A
Returns the tangent of the FAC in radians.


ATN             F09E
Returns the inverse-tangent of the FAC in radians.


                OTHER OPERATORS


SGN             EB90
Modifies the value of the FAC depending on its value on entry.  On exit:


     FAC =  1  if FAC was greater than 0
     FAC =  0  if FAC was equal to 0
     FAC = -1  if FAC was less than 0


ABS             EBAF
Absolute value of FAC.  If FAC is less than 0 then FAC = -FAC.


INT             EC23
Greatest integer value of FAC.  Uses QINT and floats the result.


RND             EFAE
Form a pseudo-random number in FAC and update RND at 00C9.


SIGN            EB82
Set A according to the value of FAC.  On exit:


     A = 1  if FAC is positive
     A = 0  if FAC = 0
     A = FF if FAC is negative


FCOMP           EBB2
Compare FAC and a packed number in memory pointed to by Y,A.  On exit:


     A = 1  if (Y,A) < FAC
     A = 0  if (Y,A) = FAC
     A = FF if (Y,A) > FAC


NEGOP           EED0
FAC = -FAC


FADDH           E7A0
Add 1/2 to FAC


DIV10           EA55
Divide FAC by 10. Returns positive numbers only.
```

MUL10   EA39
Multiply FAC by 10. Works for both positive and negative numbers.

Apple Tech Notes

Tech Info Library Article Number:75