

When a coprocessor access is made, the System Bus FC bits are set to 111, but also, bits 19-16 of the address lines are set to 0010. This indicates a reference to a coprocessor. Bits 15-13 are the coprocessor id (001 for the FPC).

31	30	...	20	19	18	17	16	15	14	13	12	...	5	4	3	2	1	0
0	0	...	0	0	0	1	0	x	x	x	0	...	0	x	x	x	x	x
Coprocessor access								Coprocessor type				Coprocessor register						

The format of a coprocessor instruction is:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	Cp-ID			Type			Type-Dependent					
F-line				Cp-type			Inst.								

The instruction type field includes some specific classes of instructions. They are defined as:

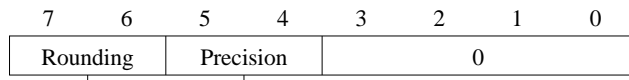
Type bits			Mnemonic	Meaning
8	7	6		
0	0	0	cpGen	General instructions
0	0	1	cpDBCC, cpScc, cpTrapcc	DBcc, set and Trap on condition
0	1	0	cpBcc.W	Branch on condition cc
0	1	1	cpBcc.L	Branch on condition cc
1	0	0	cpSave	Save context
1	0	1	cpRestore	Restore context
1	1	0	Not defined	
1	1	1	Not defined	

The floating point coprocessor has 8 80-bit registers. All operands are converted to this internal format.

There is a FP control register (FPCR) and FP status register (FPSR). They have the following format:

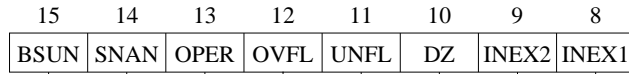
0		Exception enable	Mode Control	FPCR
Condition Code	Quotient	Exception Status	Accrued exception	FPSR

### FP Status and Control Registers



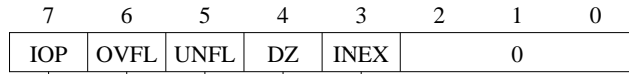
Rounding Precision

00 – Nearest	00 – Extended
01 – to zero	01 – Single
10 – to + infinity	10 – Double
11 – to – infinity	11 – Reserved

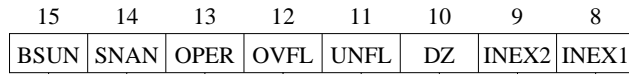


INEX1 — Inexact decimal input  
 INEX2 — Inexact operation  
 DZ — Divide by zero  
 UNFL — Underflow  
 OVFL — Overflow  
 OPER — Operand error  
 SNAN — Signaling not a number  
 BSUN — Branch/set on unordered

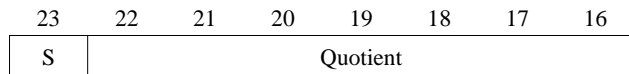
**FP Control Register**



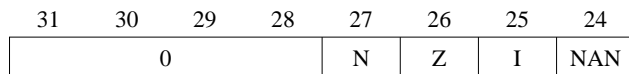
INEX — Inexact  
 DZ — Divide by zero  
 UNFL — Underflow  
 OVFL — Overflow  
 IOP — Invalid operation



INEX1 — Inexact decimal input  
 INEX2 — Inexact operation  
 DZ — Divide by zero  
 UNFL — Underflow  
 OVFL — Overflow  
 OPER — Operand error  
 SNAN — Signaling not a number  
 BSUN — Branch/set on unordered



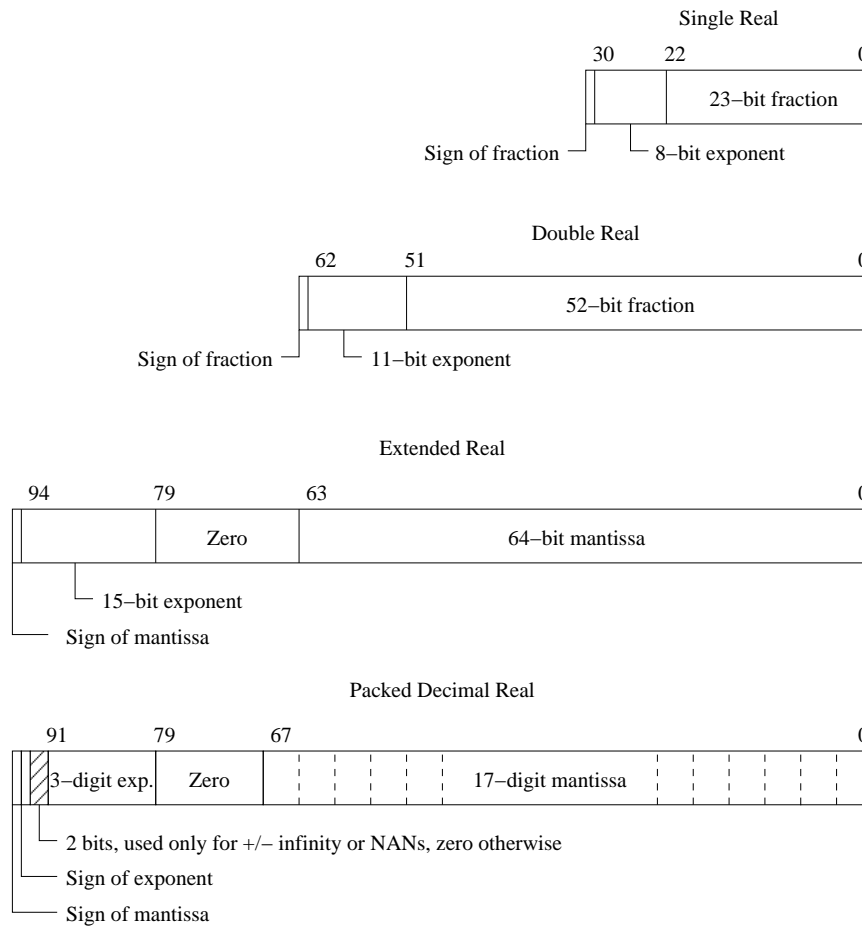
Quotient — Seven least significant bits of quotient  
 S — Sign of quotient



NAN — Not a number  
 I — Infinity  
 Z — Zero  
 N — Negative

**FP Status Register**

Supported data types: byte integer, word integer, long integer, and the following:



Calculate a vector times a constant plus a vector

For  $i = 1$  to  $N$   
 $X(i) = Y(i) * C + X(i)$

$C$  = address of constant  
 $XVec$  = address of vector  $X$   
 $YVec$  = address of vector  $Y$

```
MOVE.W    #N-1,D0    ; D0 contains loop counter
FMOVE.D   C,FP0      ; FP0 contains the constant
LEA       XVec,A0
LEA       YVec,A1
```

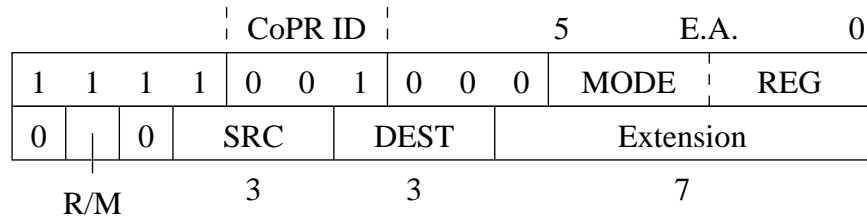
```
Again:   FMOVE.X   FP0,FP1
          FMUL.D    (A1)+,FP1    ; Calculate Y(i) * C
          FADD.D    (A0),FP1     ; Calculate Y(i) * C + X(i)
          FMOVE.D   FP1,(A0)+
          DBRA      D0,Again
```

**FADD Source + FPn -> FPn**

---

**FADD.<fmt> <ea>,FPn**

**FADD.X FPm,FPn**



R/M = 0 Effective Address Field all 0's

R/M = 1 Normal addressing modes (An not legal)

DEST specifies FPn

R/M = 0 SRC specifies FPm

R/M = 1 SRC specifies operand type

- 000 – L Long Integer
- 001 – S Single Prec. Real
- 010 – X Extended Prec. Real
- 011 – P Packed Decimal Real
- 100 – W Word Integer
- 101 – D Double Prec. Real
- 110 – B Byte Integer

Extension – 0100010 for floating–point add

Other combinations are for other operations

– FSUB, FDIV, FMUL, etc.