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).

<table>
<thead>
<tr>
<th>31 30 ... 20</th>
<th>19 18 17 16</th>
<th>15 14 13 12</th>
<th>11 10 9 8 7 6 5 4 3 2 1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 ... 0</td>
<td>0 0 1</td>
<td>x x x</td>
<td>0 ... 0</td>
</tr>
</tbody>
</table>

Coprocessor access | Coprocessor type | Coprocessor register

The format of a coprocessor instruction is:

<table>
<thead>
<tr>
<th>15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 1 1</td>
</tr>
</tbody>
</table>

F-line | Cp-type | Inst.

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

<table>
<thead>
<tr>
<th>Type bits</th>
<th>Mnemonic</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>cpGen</td>
<td>General instructions</td>
</tr>
<tr>
<td>0 0 1</td>
<td>cpDBCC, cpScc, cpTrappcc</td>
<td>DBcc, set and Trap on condition</td>
</tr>
<tr>
<td>0 1 0</td>
<td>cpBcc.W</td>
<td>Branch on condition cc</td>
</tr>
<tr>
<td>0 1 1</td>
<td>cpBcc.L</td>
<td>Branch on condition cc</td>
</tr>
<tr>
<td>1 0 0</td>
<td>cpSave</td>
<td>Save context</td>
</tr>
<tr>
<td>1 0 1</td>
<td>cpRestore</td>
<td>Restore context</td>
</tr>
<tr>
<td>1 1 0</td>
<td>Not defined</td>
<td></td>
</tr>
<tr>
<td>1 1 1</td>
<td>Not defined</td>
<td></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Condition Code</th>
<th>Quotient</th>
<th>Exception Status</th>
<th>Accrued exception</th>
</tr>
</thead>
</table>

FP Status and Control Registers
Inexact decimal input
Inexact operation
Divide by zero
Underflow
Overflow
Operand error
Signaling not a number
Branch/set on unordered

FP Control Register

Inexact
Divide by zero
Underflow
Overflow
Invalid operation

FP Status Register

Seven least significant bits of quotient
Sign of quotient

Not a number
Infinity
Zero
Negative

Rounding Precision
00 – Nearest
01 – to zero
10 – to + infinity
11 – to – infinity
00 – Extended
01 – Single
10 – Double
11 – Reserved
Supported data types: byte integer, word integer, long integer, and the following:

**Single Real**

- 30 22 0
- 23-bit fraction
- 8-bit exponent
- Sign of fraction

**Double Real**

- 62 51 0
- 52-bit fraction
- 11-bit exponent
- Sign of fraction
- Sign of exponent

**Extended Real**

- 94 79 63 0
- 64-bit mantissa
- 15-bit exponent
- Sign of mantissa
- Sign of exponent

**Packed Decimal Real**

- 91 79 67 0
- 3-digit exp. Zero
- 17-digit mantissa
- 2 bits, used only for +/- infinity or NANs, zero otherwise
- Sign of exponent
- Sign of mantissa

Calculate a vector times a constant plus a vector

For i = 1 to N

\[ X(i) = Y(i) \times C + X(i) \]

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) \times C
FADD.D (A0),FP1 ; Calculate Y(i) \times C + X(i)
FMOVE.D FP1,(A0)+
DBRA D0,Again
```
**FADD**  
Source + FPn → FPn

FADD.<fmt>  <ea>,FPn  
FADD.X     FPm,FPn

<table>
<thead>
<tr>
<th>CoPR ID</th>
<th>5</th>
<th>E.A.</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 1 1</td>
<td>0 0 1</td>
<td>0 0 0</td>
<td>MODE</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>SRC</td>
<td>DEST</td>
</tr>
<tr>
<td>R/M</td>
<td>3</td>
<td>3</td>
<td>7</td>
</tr>
</tbody>
</table>

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.