

FSCA Floating Point Sine and Cosine Approximate 浮動小数点命令

浮動小数点三角関数

PR	書式	動作概略	命令コード	実行 ステート	Tビット
0	FSCA FPUL,Dm	sin(FPUL) FRn cos(FPUL) FR[n+1]	1111nnn011111101	4	
1					

説明

FPSCR.PR=0 の場合 : FPUL レジスタの内容で示される角度から、sin と cos の近似値 (2 の-21 乗より小さい誤差) を嘆声度浮動小数点値で求め、sin の値を FRn に、cos の値を FR[n+1]に格納します。近似命令のため、常に非正確例外を発生します。(入力値が0 の場合でも、発生します。)

FPSCR.enable.I がセットされている場合、FPU 例外トラップが、例外の発生如何に関わらず発生します。例外発生時は、FPSCR.cause FPSCR.flag には、例外の正しい情報が反映され、FRn/Dm は更新されません。ソフトウェアで適切な処理を行ってください。

動作内容

```

FSCA(int n)
{
    float angle;
    long offset = 0x00010000;
    long fraction = 0x0000ffff;
    case((FPSCR_PR){
        0:clear_cause ();
        set_I();

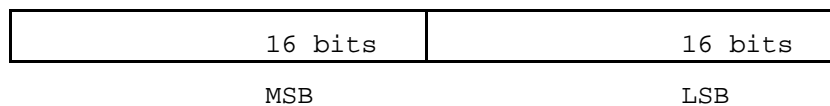
        /* extract sub-rotation (fraction) part */
        fraction &= FPUL;
        /* convert to float */
        angle = fraction;
        /* convert to radian */
        angle = 2*M_PI*angle / offset;
        FR[n]    sin(angle);
        FR[n+1]  cos(angle);
    }
}

```

```
    pc += 2; break;
1: undefined_operation();    /* reserved */
}
}
```

入力値のフォーマット: 角度の記述方法

角度は、符号付きの固定小数点で表されます。上位 16 ビットが整数部、つまり、回転数を示し、下位 16 ビットが角度を示します。



0x7FFF FFFF to 0x0000 0001: $360 \cdot 2^{15}$ - $360/2^{16}$ to $360/2^{16}$ 度

0x0000 0000: 0 度

0xFFFF FFFF to 0x8000 0000: $-360 \cdot 2^{15}$ to $-360/2^{16}$ 度

発生する可能性がある例外

Inexact

FSRRA Floating Point Square Root Reciprocal Approximate 浮動小数点命令

浮動小数点-1/2 平方

根

PR	書式	動作概略	命令コード	実行 ステート	Tビット
0 1	FSRRA FRn	1/ FRn FDn	1111nnnn01111101	3	

説明

FPSCR.PR=0 の場合 : FRn の内容の単精度浮動小数点数の算術平方根の逆数を近似で求め、結果を FRn に格納します。近似命令のため、整数などが入力されても、常に不正確例外が発生します。

FPSCR.enable.I がセットされている場合、FPU 例外トラップが、例外の発生如何に関わらず発生します。例外発生時は、FPSCR.cause FPSCR.flag には、例外の正しい情報が反映され、FRn/Dn は更新されません。ソフトウェアで適切な処理を行ってください。

動作内容

```
FSRRA(int n){
    case(FPSCR_PR){
        0:  fsrra_single(n);          break;
        1:  undefined_operation();    break;
    }
    pc += 2;
}

fsrra_single(int n)
{
    clear_cause();
    case(data_type_of(n)){
        NORM :    if(sign_of(n) == 0)
                    set_I();
                    FR[n]  1/sqrt(FR[n]);
                else    invalid(n); break;
        DENORM:    if(sign_of(n) == 0)
                    fpu_error();    break;
```

```
        else    invalid(n); break;

PZERO :
NZERO :    dz(n, sign_of(n)); break;
PINF  :    FR[n] = 0;    break;
NINF  :    invalid(n);  break;
qNaN  :    qnan(n);      break;
sNaN  :    invalid(n);  break;
    }
}
```

FSQRT Special Cases

FRn	+NORM	-NORM	+0	-0	+INF	-INF	qNaN	sNaN
FSQRT(FRn)	SQRT	Invalid	+0	-0	+INF	Invalid	qnan	Invalid

Note: When DN = 1, a denormalized number is treated as zero.

発生する可能性がある例外

FPU error

Invalid operation

Divide by Zero

Inexact