

fsqrt rationale (or at least, some sense for the magic constant)

I. J. Mermet

December 11, 2016

1 Explicacion de la obtencion de la constante

Sea f un nmero flotante de precisin simple conforme norma IEEE 754.

$$f = (-1)^s \cdot (1 + M) \cdot 2^{exp-127} \quad (1)$$

Queremos encontrar una forma de calcular \sqrt{f} , por lo tanto asumiremos que s vale 0.

Sea $y = \sqrt{f} = f^{1/2}$, entonces:

$$\log_2(y) = \frac{1}{2} \log_2(f) \quad (2)$$

La ecuacion (2) nos interesa en particular, pues es facil de calcular:

$$f = (1 + M) * 2^{exp} \quad (3)$$

$$\log_2(f) = \log_2((1 + M) * 2^{exp}) = exp + \log_2(1 + M) \quad (4)$$

Conforme IEEE 754, $M \in [0, 1) \Rightarrow (1 + M) \in [1, 2) \Rightarrow \log_2(1 + M) \in [0, 1) \Rightarrow \log_2(1 + M) \simeq M \Rightarrow \log_2(1 + M) = M + \sigma$

Tomando en cuenta el anterior resultado, y la ecuacion (4), reemplazando:

$$\log_2(f) = exp + M + \sigma \quad (5)$$

Sea I_f la representacion del mapa de bits de f como un numero entero. Entonces, tenemos que:

$$I_f = E_f * L + M_f \quad (6)$$

Donde

- $E_f = exp + B$
- B = bias del exponente (127)
- $M_f = M * L$

- $L = 2^{23}$ (23 es la cantidad de bits de la mantisa)

$$I_f = (exp+B)*L+M*L = L*(exp+B+M) = L*(exp+B+M+\sigma-\sigma) = L*(\log_2(f) + B - \sigma) \Rightarrow \frac{I_f}{L} - (B - \sigma) = \log_2(f)$$

Para dejar en claro:

$$I_f = \log_2(f) \quad (7)$$

$$\text{Volviendo a (2): } \frac{I_y}{L} - (B - \sigma) = \frac{1}{2} \left(\frac{I_f}{L} - (B - \sigma) \right) \Rightarrow \frac{I_y}{L} = \frac{1}{2} \left(\frac{I_f}{L} - (B - \sigma) \right) + (B - \sigma) \Rightarrow \frac{I_y}{L} = \frac{I_f}{2L} + \frac{(B-\sigma)}{2} \Rightarrow I_y = \frac{I_f}{2} + \frac{(B-\sigma)*L}{2}$$

Tomando $\sigma = 0$, para simplificar el razonamiento, el termino $\frac{B*L}{2}$ es 0x3f800000. Convirtiendo I_y a una interpretacion de su mapa de bits como flotante de precision simple, tenemos una aproximacion del valor buscado. Su precision puede mejorarse buscando un σ que minimice el error medio para todos los puntos o aplicando n iteraciones de Newton-Raphson.

2 Codigo inicial

```
float fsqrt(float n) {
    unsigned i = *(unsigned*)&n;
    i = (i + 0x3f800000) >> 1;
    float y = *(float*)&i;
    return y;
}
```
