

# Faster floating-point square root for integer processors

**Guillaume Revy**<sup>a</sup>, jointly with Claude-Pierre Jeannerod<sup>a</sup>,  
Hervé Knochel<sup>b</sup> and Christophe Monat<sup>b</sup>.

Arénaire INRIA project-team<sup>a</sup> / STMicroelectronics Compilation Expertise Centre<sup>b</sup>



# Context and motivation

## Context

- ▶ **Fast** and **accurate** software for floating-point mathematical functions

$$+, -, \times, /, \sqrt{\phantom{x}}, 1/\sqrt{\phantom{x}}, \dots$$

- ▶ ST231 = **integer processor** for embedded media systems → **no FPU**
- ▶ Floating-Point Library for Integer Processors  
→ single precision / various rounding modes (IEEE-754 standard)

<https://lipforge.ens-lyon.fr/projects/flip/>

## Why start with square root ?

- ▶ ST231 : used in audio/video domains (HD-IPTV, cell phones, ...)  
→ highly demanding on **floating-point square root** computations
- ▶ The square root : one of the simplest functions

# Goal

Exploit at best the ST231 architecture features : **ILP / 32-bit registers**

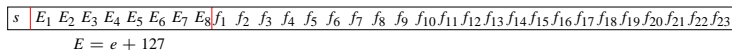
- to implement **fast algorithms** using **polynomial approximation evaluation**,
- to achieve **correct rounding-to-nearest** of square root.

# Some properties of square root computation

**Input :** normal single precision floating-point number (IEEE-754 standard)

$$x = (-1)^s \cdot 1.f \cdot 2^e,$$

with  $s \in \{0, 1\}$ ,  $e \in \mathbb{N} \cap [-126, 127]$ , and  $f = 0.f_1f_2\dots f_{23} \in [0, 1)$ .



**Output :** correct rounding-to-nearest of  $\sqrt{x} : \circ(\sqrt{x})$  or exception.

---

**Main step :** computation of  $\sqrt{1.f}$

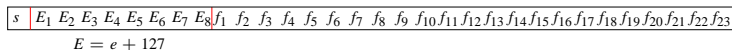
- Restoring/Non-restoring methods
- Newton-Raphson/Goldschmidt iterations

# Some properties of square root computation

**Input :** normal single precision floating-point number (IEEE-754 standard)

$$x = (-1)^s \cdot 1.f \cdot 2^e,$$

with  $s \in \{0, 1\}$ ,  $e \in \mathbb{N} \cap [-126, 127]$ , and  $f = 0.f_1f_2\dots f_{23} \in [0, 1)$ .



**Output :** correct rounding-to-nearest of  $\sqrt{x} : \circ(\sqrt{x})$  or exception.

---

**Main step :** computation of  $\sqrt{1.f}$

- Restoring/Non-restoring methods
- Newton-Raphson/Goldschmidt iterations

# Results and perspectives

## Evaluation of polynomial approximation

- ▶ **Approximation** of  $\sqrt{1+X}$ , for  $X \in [0, 1)$ , using one or several polynomials
- ▶ **Evaluation** using fast, **parallel** schemes (Estrin-like)

## Results for rounding-to-nearest and normal numbers

- ▶ 2 polynomials of degree 6
- ▶ **25** cycles  $\rightarrow$  speedup  $\approx$  **50 %** (previous version : 48 cycles)

## Actual works

- ▶ Other rounding modes / Subnormal floating-point numbers

## Perspectives

- ▶ Extension to other functions :  $1/\sqrt{x}$ ,  $x^{-1/4}$ ,  $1/x$ ,  $x/y$ , ...

# Results and perspectives

## Evaluation of polynomial approximation

- ▶ **Approximation** of  $\sqrt{1+X}$ , for  $X \in [0, 1)$ , using one or several polynomials
- ▶ **Evaluation** using fast, **parallel** schemes (Estrin-like)

## Results for rounding-to-nearest and normal numbers

- ▶ 2 polynomials of degree 6
- ▶ **25** cycles  $\rightarrow$  speedup  $\approx$  **50 %** (previous version : 48 cycles)

## Actual works

- ▶ Other rounding modes / Subnormal floating-point numbers

## Perspectives

- ▶ Extension to other functions :  $1/\sqrt{x}$ ,  $x^{-1/4}$ ,  $1/x$ ,  $x/y$ , ...