ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

inf | Informatik
Computer Science

Abstraction techniques for Floating-Point Arithmetic

Angelo Brillout[1], Daniel Kroening[2] and Thomas Wahl[2]

[1]ETH Zurich, [2]Oxford University

Floating-Point Arithmetic (FPA)

- Used for embedded and safety critical systems
- Finite representation of real numbers
  - → Rounding
  - → Deviation causes unintuitive results
  - → Deviation can change control flow

→Behavior of floating-point programs hard to predict

Contributions

→ New effective approximation techniques

- Over- and underapproximation for FPA
- Bit-precise

→ Precise and sound decision procedure for FPA:

- Based on CBMC model checking engine
- SAT solver as the back-end

Floating-Point Arithmetic (FPA)

- Numerical representation of a subset of the reals
- Floating-point format: IEEE-754 standard
  - Triple (s, e, f) stands for the number $(-1)^s \cdot f \cdot 2^e$
  - Represented by a bit-vector

s	e_{r-1} $\cdots$ e_0	f_0 $\cdots$ f_{p-1}
$\longleftrightarrow$ 1	$\longleftrightarrow$ r	$\longleftrightarrow$ p

- Representable numbers $\mathbb{F}_p$
- Floating-point operations $\oplus \ominus \quad \oslash$
  - Differ from real arithmetic. E.g.:

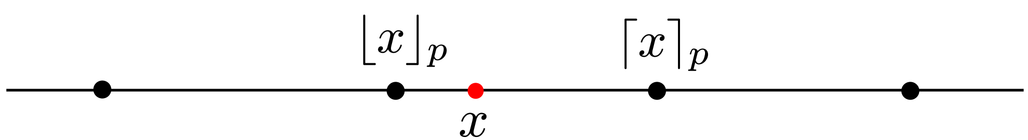
$$(a \oplus b) \oplus c \neq a \oplus (b \oplus c)$$

Floating-Point Arithmetic (FPA)

- Result of FP-operation not always representable

  → Approximations:

$$\begin{aligned} \lfloor x \rfloor_p &:= \max\{f \in \mathbb{F}_p : f \cdot x\}, \quad \text{and} \\ \lceil x \rceil_p &:= \min\{f \in \mathbb{F}_p : f \geq x\}. \end{aligned}$$



  → Rounding function:

$$rd_p(x) \in \{\lfloor x \rfloor_p, \lceil x \rceil_p\}$$

- Rounding based on least significant bits of fraction

Floating-Point Arithmetic (FPA)

- Floating-point operations defined as:

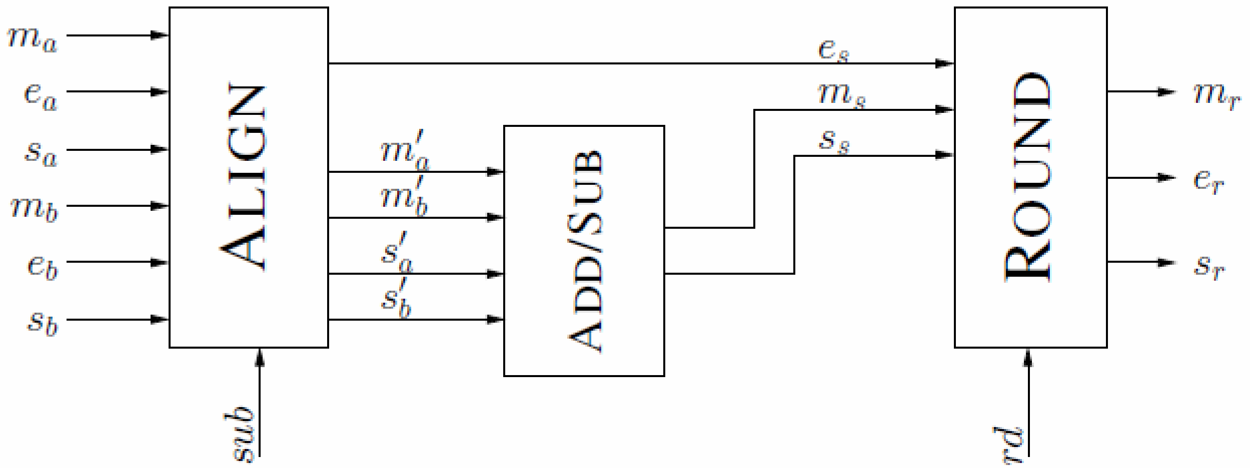
$$x \circledcirc_p y := rd_p(x \circ y)$$

- Verification of FPA programs:
  - Naïve method: Bit-vector model of an FPU and bit-blasting
  - BMC (Unrolling, Bit-blasting, SAT-solving)

→ Does not scale for FPA

FPA Verification

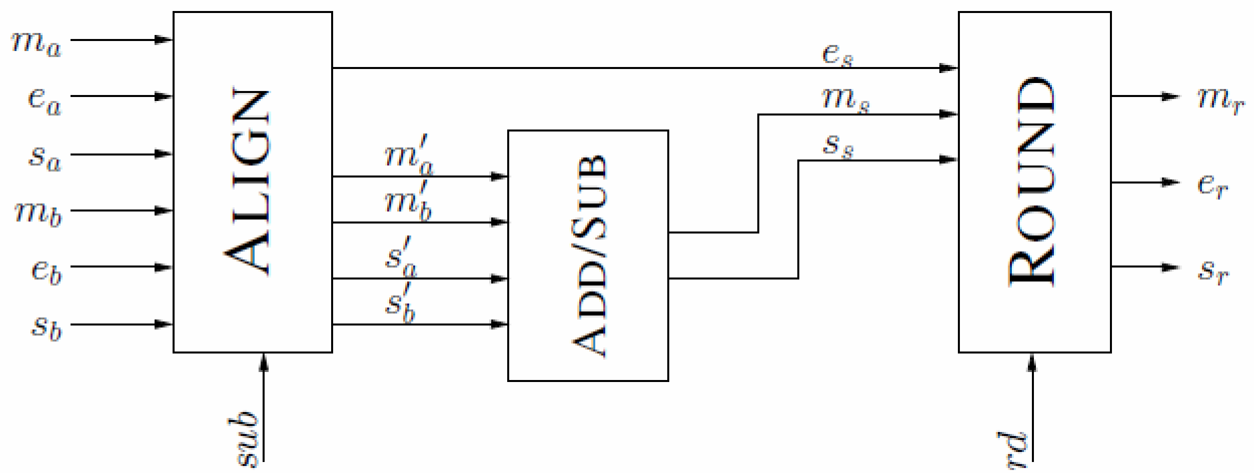
- FPU-Implementation of Add/Sub



- Align: mantissa shifted, rendering exponents equal
- Add/Sub: resulting mantissas are added/subtracted
- Round: shortening mantissa to obtain a number in $\mathbb{F}_p$

FPA Verification

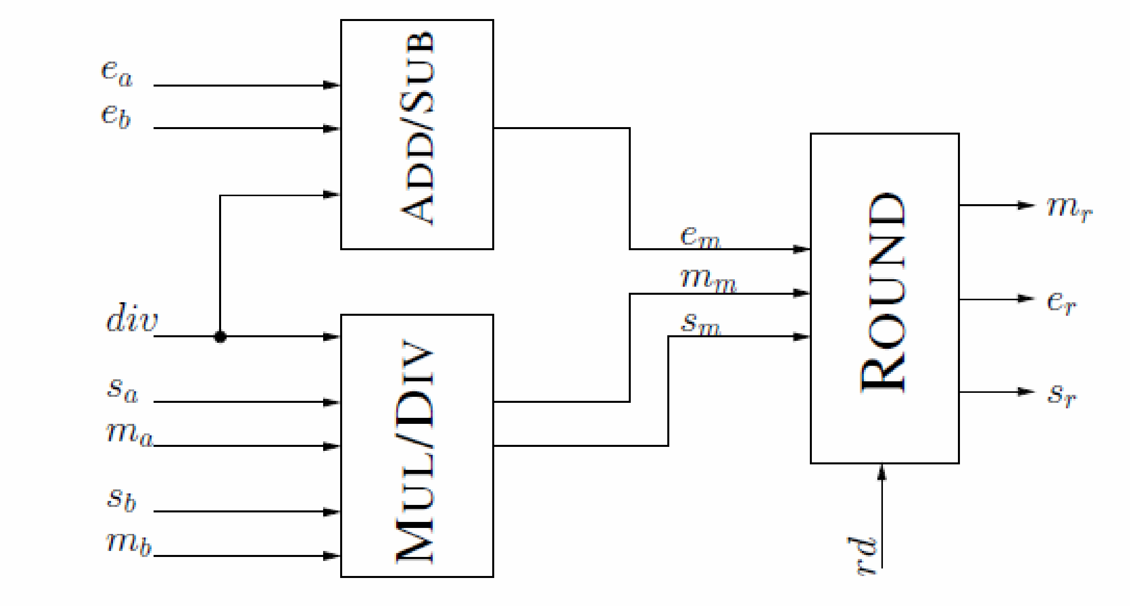
- FPU-Implementation of Add/Sub



Precision	ALIGN	ADD/SUB	ROUND	Total
$p = 5$	295	168	572	1035
$p = 23$	687	420	1447	2554
$p = 52$	1404	826	2923	5153

FPA Verification

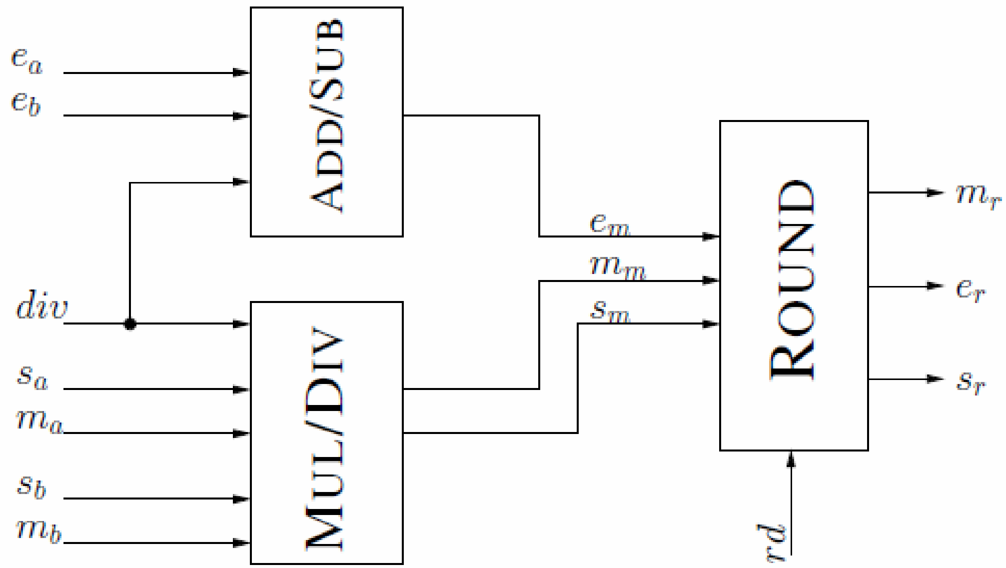
- FPU-Implementation of Mul/Div



- Add/Sub: exponents added/subtracted (Mul/Div)
- Mul/Div: mantissas multiplied/divided (Mul/Div)

FPA Verification

- FPU-Implementation of Mul/Div



Precision	MUL/DIV	ADD/SUB	ROUND	Total
$p = 5$	280	94	674	1048
$p = 23$	3898	94	2258	6550
$p = 52$	19268	94	5742	25104

FPA Verification

→ Need for approximate FP-operations

Can we approximate FP-operations by reducing the precision p ?

Approximation techniques

- Reducing the precision $p' < p$
  - → Least significant bits are lost
- Overapproximation by open rounding:

$$\overline{rd}_{p,p'}(X) \ := \ [\, \lfloor X \rfloor_{p'}, \lceil X \rceil_{p'} \,] \cap \mathbb{F}_p$$

- New FP-operations

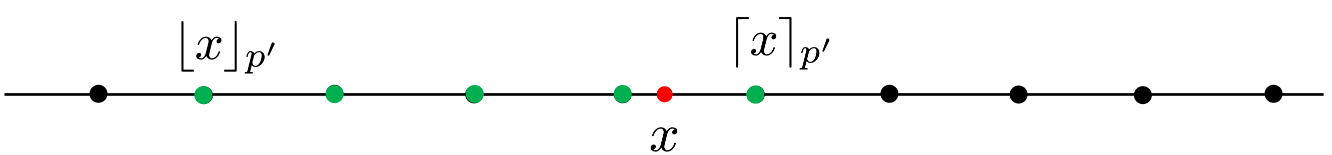
$$X \,\overline{\circledcirc}_{p,p'}\, Y := \overline{rd}_{p,p'}(X \circ Y)$$

- Replace $\circledcirc_p$ by $\overline{\circledcirc}_{p,p'}$ for some precision $p' < p$

Approximation techniques

- Overapproximation: visualization

$$\overline{rd}_{p,p'}(\{x\}) = \left[\, \lfloor x \rfloor_{p'}, \lceil x \rceil_{p'} \,\right] \cap \mathbb{F}_p$$



precision $p' < p$

$$\overline{rd}_{p,p'}(\{x\}) = \{\bullet, \bullet, \bullet, \bullet, \bullet\}$$

Approximation techniques

- Reducing the precision $p' < p$
  - → Least significant bits are lost
- Underapproximation by inhibiting rounding:

$$\underline{rd}_{p,p'}(X) \ := \ X \cap \mathbb{F}_{p'}$$

- New FP-operations

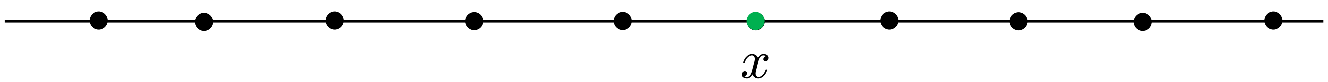
$$X \underline{\circledcirc}_{p,p'} Y := \underline{rd}_{p,p'}(X \circ Y)$$

- Replace $\circledcirc_p$ by $\underline{\circledcirc}_{p,p'}$ for some precision $p' < p$

Approximation techniques

- Underapproximation: visualization

$$\underline{rd}_{p,p'}(\{x\}) = \{x\} \cap \mathbb{F}_{p'}$$



precision $p' < p$

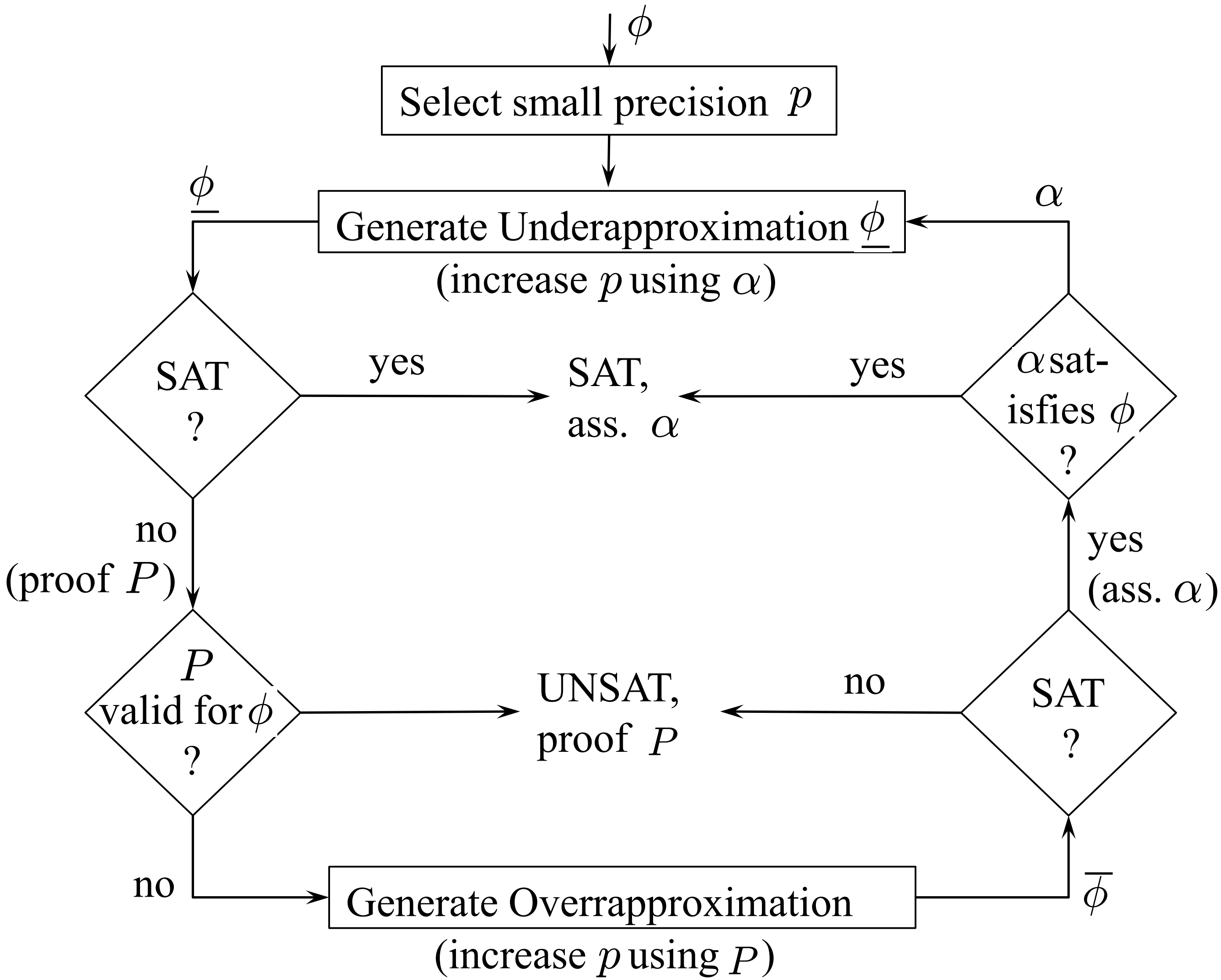
$$\underline{rd}_{p,p'}(\{x\}) = \{x\} \text{ if } x \in \mathbb{F}_{p'}, \quad \emptyset \text{ otherwise}$$

Alternating abstractions for FPA

- Over-approximation
  - Permits *more* execution traces than original program
  - SAT: no conclusion, UNSAT: assertions OK
- Under-approximation
  - Permits *less* execution traces than original program
  - SAT: assertion violated, UNSAT: no conclusion
- Refinement: increase p

→ Alternation yields complete procedure

Alternating abstractions for FPA



Alternating abstractions for FPA

Refinement for FPA:

- Spuriously SAT:

  r result of $\overline{\circledcirc}_{p,p'}$. If $r \neq \circledcirc_p$ then increase precision

- Spuriously UNSAT:
  - Recall:

$$\underline{rd}_{p,p'}(X) \;:=\; \boxed{X \cap \mathbb{F}_{p'}}$$

  - If the constraint $X \cap \mathbb{F}_{p'}$ occurs in P , then increase precision

Summary

- Model Checking with FPA
  - Effective over- and underapproximation hard to find
  - Slow (model checking)
  - ✓ Fully automatic
  - ✓ Provides counterexample

→ Implemented in CBMC

State of the Art

- Proof assistants
  - ✓ Very powerful
  - Require interaction
  - No counterexample

- Interval arithmetic $[1,2]+[4,6]=[5,8]$
  - ✓ Fully automated
  - Too coarse
  - No counterexample

Issues

- E.g. the formula $(a \oplus b) \oplus c \neq a \oplus (b \oplus c)$ is SAT
  - Every overapproximation based on $\overline{\circledcirc}$ is SAT
  - Every underapproximation based on $\underline{\circledcirc}$ is UNSAT

→ Some formulae do not have effective over- or underapproximations

Conclusion

- New algorithm for iteratively approximating complex FPA –formulae
  - New under- and over-approximations for FP-operations
- Ability to generate counterexamples
  - Debugging
  - Automated test-vector generation
- Promising experiments, future work

Thank you!