



## Plans for IEEE Standard 754-2028

E. Jason Riedy

Georgia Institute of Technology

25<sup>th</sup> IEEE Symposium on Computer Arithmetic, 26 June 2018

---

# Outline

## Smaller Work Items

- Documenting Decisions

- Graduating / Retiring Functionality

- Debatable Decisions

## Bigger Picture Items

- Ubiquitous Parallelism

- New Driving Applications: Machine Learning

- Oddball Architectures

- IEEE Standardization Process

- Summary

# Smaller Work Items

---

# Documenting Decisions

- 1985 was good
  - **Many** publications detailing decisions
- 2008 was not
  - Oral history
  - Minutes (thanks to David Bindel)
  - Some email archives
- 2018 will be better
  - Working notes to be available
- Ramp up to 2028?

# Graduating Recommendations?

Which recommended operations graduate?

- Fixed min/max?
- Correctly rounded special functions?
- Augmented arithmetic operations?
- Reductions? (to be mentioned later)
- NaN payload operations?

Oh, and “security.”

# Retiring Unused Pieces?

- Extended and extensible precisions?
- Nail down underflow?
- (Sure others will have more opinions...)

**I'm dodging new inclusions for now.**

# Debatable Decisions

- Special function special cases
  - Power,  $x^y$ . All the joy for integral values of  $y$ .
  - Preference for conformal mappings
  - “Much ado about nothing’s sign bit...”
- **abs**, **negate** as numeric rather than “bit”?
  - So raise invalid on signaling NaNs.
  - What about **copy**? Traditionally left to implementations.

# NaN v. “Missing”

The endless argument. Not now.

Clearly there's a need, but there is no clear path.

# Bigger Picture Items

---

# Ubiquitous Parallelism

- We need compose-able operations.
  - Recommended reduction operations cannot be used to build higher-level parallel operations.
- Reproducibility in the face of dynamic scheduling.
- Traps are out already, replaced by alternate exception handling.
- Can a more data-flow approach help?
- And then there's energy use...

# Driving Apps in Machine Learning

- binary16: The right split?
  - binary8? binary5?
- Will five-eight years be enough for convergence?
  - (Vanishing gradients are an issue.)
- And there is little numerical analysis...

# Oddball / Novel Architectures

- Accelerators (via CAPI, NVLINK, ...)
- “Easy” FPGA programming: OpenCL, Chisel, SPIRAL
- Data-centric, memory-centric proposals and systems
- Quantum, neuromorphic, analog



## What are the arithmetic and debugging considerations?

Testbeds:

- **CRNCH Rogues Gallery at Georgia Tech**
- CENATE at PNNL
- ExCL at ORNL
- JLSE at Argonne
- ASC at Sandia

# Novel Architectures Still Need FPUs

- Not well taught
- Many analyses do not support “special cases” like overflow (not naming names)
- Others never even try running **paranoia**, *etc.*
- Thankfully hardfloat / rocket core exists

# IEEE Standardization Process

---

# Overall IEEE Process

1. PAR: Project Authorization Request, defines scope
  - Current revision's PAR includes backwards compatibility.
2. Form a committee.
  - Officers: Chair, vice chair, secretary, editor. Lacking...
3. Then, eventually, agree to send a revision to the sponsor (MSC).

# Summary

---

# Summary

There's work to do.

There's thinking involved. Correct scope?

And effort. (Committee officers...)

Timeframe: 2023 for a PAR if desired.

And if we're optimistic about five years being enough time to find convergence, then again time to decide.

**IEEE 754 is not the only approach.  
Can have others!**