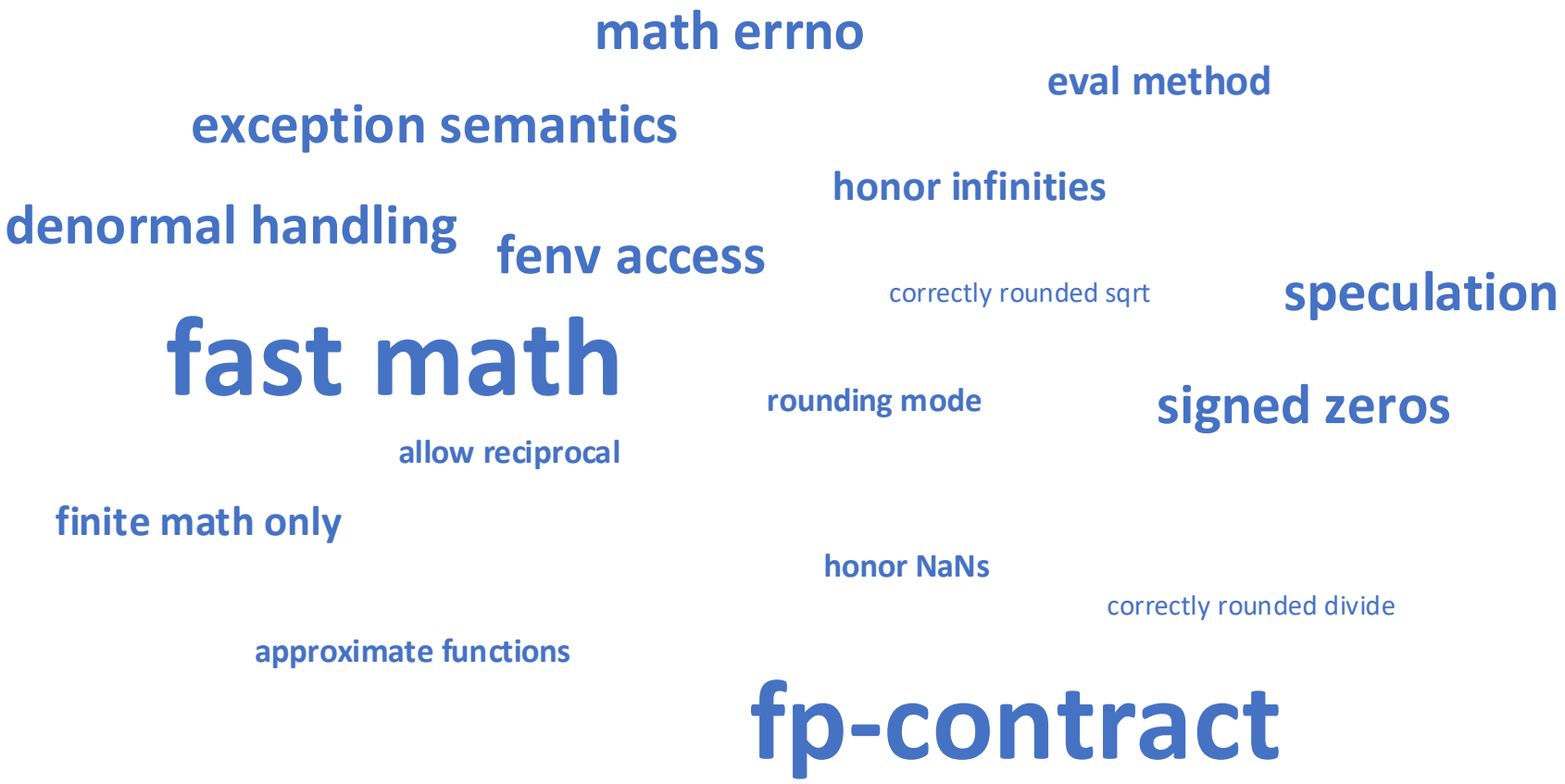
fp-model

Single option floating point control

Andy Kaylor
Intel Corporation

So many choices...



fp-model – One option to rule them all

- Select from a small number of models
  - precise, fast, strict
  - We can add more as needed
- Models are intended to represent common use cases
- Each model provides consistent defaults for all floating point settings
- Individual settings can be overridden if needed
- Front ends apply the model to generate IR

precise – reliable and accurate results

- Most value changing optimizations are disabled
- FP contraction is enabled if allowed by source language standard and supported by target architecture
- Denormals are preserved
- No floating point environment access
  - Default rounding mode is assumed
  - Exception semantics are not preserved
- Conforms to IEEE-754

fast – best optimization

- All fast-math optimizations are enabled
- Not a value safe mode!
- FP contraction anywhere the backend thinks it is profitable
- Denormals are flushed to zero
- No floating point environment access
  - Default rounding mode is assumed
  - Exception semantics are not preserved

strict – preserves all source semantics

- All value changing optimizations are disabled
- FP contraction is disabled
- Denormals are preserved
- Floating point environment access is allowed
  - Rounding mode is not assumed
  - Exception semantics are preserved
- Many optimizations are inhibited

Call to action

- Use this in your front ends
  - clang supports -ffp-model=[precise|fast|strict]
  - Consistency among front ends would be great!
- Provide feedback!
  - Do we need other models?
  - Do you like the current settings?