754-754R Compatibility

Jim Thomas

Hewlett-Packard Company

Cupertino, CA

408 447 5781

jwthomas@cup.hp.com

Appeared in first several meeting minutes:

“Knowledgeable persons with a desire to contribute positively toward an upward-compatible revision are encouraged to subscribe and participate.”

From Jan 2001:

... “Items which are not quite upward-compatible, strictly speaking, would presumably be "should," and perhaps "shall" for new implementations, in order to grandfather in existing implementations.”

Implies:

Upward compatibility is an important goal

Existing implementations will remain conforming

But from March 2001:

“754R-200x will be a new standard, and it is conceivable that it might not be strictly upward compatible.”

What does “upward compatible” mean?

754 implementations can be upgraded to 754R without invalidating correct programs, in a manner that is (one of)

a) technically feasible, or

b) technically and commercially feasible

754 implementations should be upgradeable to 754R without invalidating conformance to standard language bindings to 754

Language standard bindings for 754 should be upgradeable to 754R so that conforming implementations can be upgraded without invalidating correct programs

Where upward compatibility matters (1)

Hardware implementors

formally protected by grand-fathering

grand-fathering for architectures (processor families) vs specific processors

some incompatibilities can be corrected by software

commercially feasible for whom?

competitive disadvantage if software support runs too slow

or if contracts specify new FP standard (perhaps unnecessarily)

Programming environments

compilers and libraries

greater implementation flexibility than with HW

commercially feasible for whom?

support for existing user code is essential -- impedes change and begs for upward compatibility

support for language standards might impede or drive change

Where upward compatibility matters (2)

Language standards

- C99, Unix 2000 support 754
  - 5-10 year lead time for substantial change
  - even if not breaking upward compatibility, additional similar but different features may burden implementors, confound programmers, and weaken the standard
- Fortran, C++, Java FP models are evolving
  - language committees are not FP committees
  - need expert FP advice (may or may not know it)
  - FP standard carries weight
  - FP standard in flux may confuse issues and weaken support for FP features

Where upward compatibility matters (3)

Applications software developers

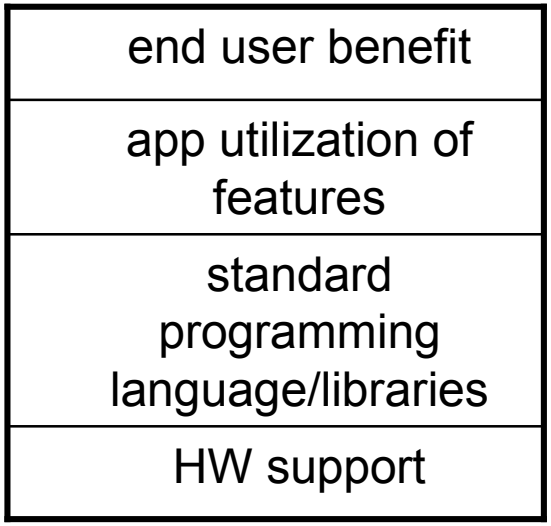
hate platform diversity

use of 754 limited by slow emergence of standard language support

even more wary if FP standard changes undermine language standards

End users

top of food chain



Where upward compatibility matters (4)

Other FP programmers

research, education

more tolerant of change and platform diversity

language standard provides lingua franca

language standard facilitates application of research

What does upward compatibility *goal* imply for 754R?

Don’t break upward compatibility unless benefit is compelling*

Each proposal should explain how implementation upgrades would be technically and commercially feasible

Each proposal should include consideration of compatibility impact

Proposals that would entail changes to language standards should show a feasible upward compatible extension to the standard

Don’t add similar but different features (even if legally upward compatible) unless benefit is compelling

* “too low a bar”