[How to get from 5 operations to 75 in two easy steps. My actual proposal is to ADD ONLY the 18 operations in blue. Add the following in 7.1 in place of the *flag* type.

[Kahan asked us to distinguish the abstract status flags, which might exist in hardware or in special variables, from their images in ordinary variables]

— *flagImage* is a type of an ordinary program variable that contains an image of a status flag

— *flagGroupImage* is a type of an ordinary program variable that contains an image of a set of status flags

[Kahan also asked for swap operations as you will see below.]

5.7.4 Operations on subsets of flags

Implementations shall provide the following non-computational operations that act upon multiple status flags collectively:

— *void* **raiseFlags**(*exceptionGroup*)

raises the flags corresponding to the exceptions specified in the *exceptionGroup* operand, which can represent any subset of the exceptions. [Default flag value, "true" on all current implementations]

— *void* **lowerFlags**(*exceptionGroup*)

lowers (clears) the flags corresponding to the exceptions specified in the *exceptionGroup* operand, which can represent any subset of the exceptions.

— *boolean* **testFlags**(*exceptionGroup*)

queries whether any of the flags corresponding to the exceptions specified in the *exceptionGroup* operand, which can represent any subset of the exceptions, are raised.

— *boolean* **testSavedFlags**(*flagGroupImage*, *exceptionGroup*)

queries whether any of the flags in the *flagGroupImage* operand corresponding to the exceptions specified in the *exceptionGroup* operand, which can represent any subset of the exceptions, are raised.

— *flagGroupImage* **saveFlags**(*exceptionGroup*)

returns a representation of the state of those flags corresponding to the exceptions specified in the exceptionGroup operand.

— *void* **restoreFlags**(*flagGroupImage*, *exceptionGroup*)

restores the flags corresponding to the exceptions specified in the exceptionGroup operand, which can represent any subset of the exceptions, to their state represented in the flagGroupImage operand.

— *flagGroupImage* **swapFlags**(*flagGroupImage*, *exceptionGroup*)

swaps the flags corresponding to the exceptions specified in the *exceptionGroup* operand, which can represent any subset of the exceptions, saving to the *flagGroupImage* result and restoring from the *flagGroupImage* operand. [for Kahan]

The return value of the saveFlags and swapFlags operations is for use as an operand for the restoreFlags, swapFlags, or testSavedFlags operation in the same program; this standard does not require support for any other use.

The following operations avoid the need for encoding and decoding an exceptionGroup parameter for the common case of all flags. They deal with all the real status flags as a group.

- *void* **lowerAllFlags***(void)*
- *flagGroupImage* **saveAllFlags**(*void*)
- *void* **restoreAllFlags**(*flagGroupImage*)
- *flagGroupImage* **swapAllFlags** (*flagGroupImage*)
  swap a flag group image with the real status flags [for Kahan]

The following operations avoid the need for encoding and decoding an exceptionGroup parameter for the common case of one flag. They deal with the real status flags one at a time. Consequently there are five specific operations, corresponding to the five individual exceptions of clause 7, for each operation with an EXCEPTION macro in its name.

- *void* **raiseEXCEPTION** (*void*)
  raises a status flag with default value ["true" on all existing implementations]
- *void* **lowerEXCEPTION** (*void*)
- *boolean* **testEXCEPTION** (*void*)
- *flagImage* **saveEXCEPTION** (*void*)
- *void* **restoreEXCEPTION**(*flagImage*)
- *flagImage* s**wapEXCEPTION**(*flagImage*)
  swap a flag image with a real status flag [for Kahan]

The following manipulate single flagImages within a *flagGroupImage*; they provide a portable way to perform portable operations on a potentially opaque implementation-dependent type. They do not affect the real status flags [pass by reference is more compact]::

- *void* **raiseEXCEPTIONInGroupImage** (*flagGroupImage **)
- *void* **lowerEXCEPTIONInGroupImage** (*flagGroupImage **)
- *boolean* **testEXCEPTIONInGroupImage** (*flagGroupImage **)
- *flagImage* **saveEXCEPTIONInGroupImage** (*flagGroupImage **)
- *void* **restoreEXCEPTIONInGroupImage** (*flagGroupImage *, flagImage*)
- *flagImage* **swapEXCEPTIONInGroupImage** (*flagGroupImage *, flagImage*)
  swap a flag image with one in a flagGroupImage [for Kahan]

The following manipulate a single *flagImage*; the same operation can be used for each exception; [they could be set up as operand and result (except swap) but pass by reference is more compact]:

- *void* **raiseFlagImage** (*flagImage **)
- *void* **lowerFlagImage** (*flagImage **)
- *boolean* **testFlagImage** (*flagImage **)
- *void* **swapFlagImage** (*flagImage *, flagImage **)
  swap two flag images [for Kahan]