  - Real type (default real type, double-precision real type, quadruple precision real type)
  - Complex type (default complex type, double precision complex type, quadruple precision complex type)

- A variable specified in a VOLATILE statement cannot be given initial values by a block data program unit, a DATA statement, or a type declaration statement.
- *variable-name* must not be an automatic array or an assumed-shape array.
- *variable-name* must not have the ALLOCATABLE, OPTIONAL, POINTER or TARGET attributes.
- *variable-name* must not be a variable that has the host association.
- *variable-name* must not be a variable that has the use association.

4. Issues

+ An important issue which needs to be addressed is the interaction of the VOLATILE statement and attribute with asynchronous I/O and COPYIN/COPYOUT.
+ Furthermore, the question needs to addressed whether the variables specified in a VOLATILE statement also can have use association and/or host association ?
+ We also should discuss whether a block specification needs to be implemented.

To: X3J3
From: Matthijs van Waveren
Subject: VOLATILE requirement
Date: February 14, 1997

1. Goal

The goal of this requirement is to be able to specify that variables or arrays need to be referenced from memory and not from local registers. The rationale is that for applications where the memory version of the variables might be different from the local version (e.g. asynchronous I/O, MPI I/O, device drivers, ...) this would be a very useful feature. We expect that the workload of this requirement will be relatively light.

2. Illustrative Syntax

- Statement form:
  VOLATILE [::] A
- Attribute form:
  REAL, VOLATILE :: A

Example of usage:

```
REAL, VOLATILE :: A
REAL B, C

CALL MPI_RECV(..., A, ...)

B = A + C
```

This will lead to the situation that, after receipt of the variable A, this variable will be referenced from memory and not from registers.

3. Implementation of the statement form

volatile-stmt
is VOLATILE [::] *volatile-decl-spec-list*

volatile-decl-spec
is *variable-name*

The VOLATILE statement declares that the variable expressed by *variable-name* is a volatile variable. The VOLATILE statement also specifies that this variable is to be located in global space.

Constraints

- The type and kind type parameters of a variable specified in a VOLATILE statement must be one of the following:
  - Integer type (default integer type, eight-byte integer type)
  - Logical type (default logical type)