Subject: Treatment of BOZ is inconsistent, incomplete, and maybe unreasonable
From: Van Snyder

1 Introduction

The treatment of BOZ is inconsistent, incomplete, and maybe unreasonable. This was discussed in section 2.15 of the comment that accompanied the US National Body ballot on the current committee draft.

[35:10] says "Binary, octal and hexadecimal constants are interpreted according to their respective number systems." For the DATA statement at [87:23-25] and the INT intrinsic at [319:13-15], a BOZ literal is interpreted as an integer having the same number of bits as the integer with the largest decimal range. If the kind of that integer is packed decimal, what does [35:10] mean for the constant `z'0B'`? Is this the number 11, which will appear in a memory dump as the byte `z'11'`, or will it result in a bit pattern in the memory of the computer that will appear in a memory dump as the byte `z'0B'`? That is, is the "respective number system"an abstract mathematical concept, or is it determined by the kind ascribed to the binary, octal or hexadecimal literal?

For the REAL intrinsic at [340:9-12], it's clear that a BOZ literal is considered to be a bit pattern that specifies the value of a real variable, for which the kind is taken from the kind of the result. It's also clear for the CMPLX intrinsic at [303:17-31] and the DBLE intrinsic at [308:10-16], which are described in terms of the REAL intrinsic.

If, drawing inspiration from the REAL intrinsic, we assume the latter intrepretation of the meaning of "interpreted according to their respective number systems," the intrepretation of a binary, octal and hexadecimal literals is ambiguous if a processor supports two different integer representations having the same decimal range. At least five integer representations have been used (1's complement, 2's complement, sign magnitude, packed decimal, and excess-3), others have been proposed and are still being seriously considered (three digits in ten bits), and contemporary computers exist that provide two representations (e.g. IBM 390 provides 2's complement binary and packed decimal).

Under the assumption that the intrepretation of the bit pattern ought to depend on the kind imputed to the literal, in the DATA statement, the kind should be taken from the corresponding variable, and in the INT intrinsic, the kind should be taken from the KIND argument if the KIND argument is present, or default kind otherwise.

2 Edits to interpret BOZ literals as in the REAL intrinsic

Edits refer to 02-007r3. Page and line numbers are displayed in the margin. Absent other instructions, a page and line number or line number range implies all of the indicated text is to be replaced by associated text, while a page and line number followed by + (-) indicates that associated text is to be inserted after (before) the indicated line. Remarks are noted in the margin, or appear between [and] in the text.

[Editor: "specifies ... processor." ⇒ "that is the same as the kind type parameter of the corresponding variable and whose value is the value that an integer variable of the same kind has if its value is the bit pattern specified by the *boz-literal-constant*. The interpretation of the value of the bit pattern is processor dependent." This wording is similar to the present wording for the REAL intrinsic – but how is it affected here (and in the REAL intrinsic) by variations in endian-ness?] 87:24-25

[Editor: At the end, add "If A is a *boz-literal-constant*, it is as if A is an integer variable with the same kind as the result and a value equal to the value such a variable has if its value is the bit pattern specified by the *boz-literal-constant*. The interpretation of the value of the bit pattern is processor dependent."] 319:3

[Editor: Delete. Why did we describe the argument in the Result Value paragraph in the first place?] 319:13-15

3 A natural extension

Why do we require a *boz-literal-constant* to be an integer in a DATA statement, while we go to so much trouble to specify its interpretation as a real number in the case that it's an argument for the REAL intrinsic? One can get the same effect by using a parameter, but why make it so hard? It isn't new technology to use a similar interpretation in both places.

If a *data-stmt-constant* is a *boz-literal-constant*, the corresponding variable shall be of type integer or real. 87:23-25
The *boz-literal-constant* is treated as if it is a constant of the same type and kind as the corresponding variable. Its value is the same as a variable of the same type and kind has if its value is the bit pattern specified by the *boz-literal-constant*. The interpretation of the value of the bit pattern is processor dependent.

4 Just for symmetry...

[Editor: Insert ", as the actual argument asociated with the dummy argument I of the intrinic function CHAR" after "INT".] 35:15

If a *data-stmt-constant* is a *boz-literal-constant*, the corresponding variable shall be of type integer, real or character. The *boz-literal-constant* is treated as if it is a constant of the same type and kind as the corresponding variable. Its value is the same as a variable of the same type and kind has if its value is the bit pattern specified by the *boz-literal-constant*. The interpretation of the value of the bit pattern is processor dependent. 87:23-25

[Editor: After "parameter" insert ", or a *boz-literal-constant*".] 303:7

Result Value. 303:12

Case (i): If I is of type integer, the result is the character in position I of the collating sequence associated

---

Case ((i)): If I is a *boz-literal-constant* the value of the result is the value that a variable of the same type and kind as the result has if the value is the bit pattern specified by the *boz-literal-constant*. The interpretation of the value of the bit pattern is processor dependent. 303:15+

Examples. 303:16

Case (i): CHAR(88) has the value 'X' on a processor using the ASCII collating sequence.

Case (ii): CHAR(Z'44') has the value 'P' on a processor using the ASCII collating sequence.