ATTACHMENT B

Summary of X3J3 Actions in Response to Public Comment

The following is a summary of X3J3 actions in response to public comment on the draft proposed revised FORTRAN standard X3J3/76 published for public review in March 1976.

A number of public comments resulted in X3J3 actions to change the draft; others resulted in X3J3 actions affirming the features of the language substsntislly as described in the draft. These two groups of items ere listed separately in this summary.

It should he noted that the draft proposed standard describes two levels of the FORTRAN language. All changes made to the full language in response to public comment were carefully reviewed for their applicability to the subset language, and where necessary appropriate action was taken by X3J3 to change the subset as well. Items in this list marked (*) involved changes to the subset language as well as to the full language.

Part 1. Actions resulting in changes to the March 1976 Draft

* 1. Further restrictions on collating sequence. The letters A - Z and the digits 0 - 9 must not overlap in the collating sequence. This restriction use not included in the Draft.

* 2. Blank lines. A completely blank line among the lines of a program unit is considered to he a comment line. Formerly it was the initial line of s statement.

* 3. Statement ordering. A further restriction was adopted, to the effect that symbolic names of constants, or variables appearing in dimension hound expressions, must not he explicitly typed farther down in a program unit.

* 4. Form of a constant. The rules were relaxed to permit an integer constant (as well as a real constant) as either pert of a complex constant.

* 5. Data type of a symbolic (PARAMETER) name. Type rules for PARAMETER statements are changed, so that the type of the symbolic name of a constant depends upon the form of the name rather then upon the form of the constant.

* 6. Character length for a symbolic (PARAMETER) name. A symbolic name for e constant of character type may he declared in a CHARACTER statement with s length specification consisting of an asterisk. The actual length for such s symbolic name is determined from the length of the character constant in the PARAMETER statement. This way of specifying length was not provided in the Draft.

* 7. Restriction to integer type, for subscripts etc. In a number of places where the Draft permitted integer, real, or double precision expressions, only integer expressions are now permitted. These are: subscript expressions, substring expressions, label selectors for computer GO TO, external unit identifiers, record number specifiers, record length specifiers, end alternate return specifiers.

* 8. Intrinsic function names. The Draft provided that certain type statements could change an intrinsic function name to external. This has been changed so that only an EXTERNAL statement can remove a function name from the intrinsic category in this way.

* 9. PARAMETER syntax. The following changes were made to the Draft:

An expression, which may include symbolic names of previously defined constants, may he included in the definition of a symbolic constant in a PARAMETER statement.

The entire list of definitions must he enclosed in parentheses.

* 10. Exponentiation. An integer, real, or complex expression may be raised to a real or complex power. Complex exponentiation was prohibited in the Draft. Rules have been included to determine the principal value of the result.

Exponentiation with integer powers is now permitted in an arithmetic constant expression.

* 11. Logical operators. Two logical operators were added to those described in the Draft. These are .EQV. and .NEQV. , and both of these operators have lower precedence than any of the previous logical operators. If L1 and L2 are logical expressions, then Ll .EQV. L2 is true when Ll and L2 ere both true or both false, and Ll .NEQV. L2 is true when one is true and the other is false.

* 12. SAVE statement. The following changes were made to the Draft:

An integer variable specified in a SAVE statement, that is defined with a statement label value, no longer becomes undefined upon return from a procadure.

When a labelled common block is specified in a SAVE statement in one subprogram of an executable program, the block must be specified in a SAVE statement in every subprogram in which it appears.

* 13. DATA statements. The rules for correspondence between the list of names and the list of constants were relaxed, so that an entity of complex type (as well as integer, real, or double precision) in either list may correspond to an entity of any of these types in the other list.

* 14. Block IF. The following statements were added to the language:

```
IF (e) THEN
ELSE IF (s) THEN
ELSE
END IF
```

For each IF-THEN statement there must be a corresponding END IF statement. Between the IF-THEN and the corresponding END IF there may appear any number of ELSE IF-THEN statements, followed by at most one ELSE. Groups of statements delimited by IF-THEN and END IF must be properly nested, with respect to other such group of statements and with respect to DO loops.

* 15. Restrictions on transfer of control. The following rules were adopted:

Transfer of control into the range of a DO loop is prohibited.

Transfer into a group of statements delimited by IF-THEN, ELSE IF-THEN, ELSE, or END IF, from outside the immediate group, is prohibited. However, transfer to an END IF statement is not prohibited.

16. Input and output statements. The following changes were made:

An IOSTAT specifier may be included in any input or output statement (including auxiliary and file positioning input or output statements) to permit determination of the cause of an error or end-of-file condition without requiring a transfer of control. (The ERR = and END = specifiers are retained.)

* An asterisk may be used as a unit specifier in a WRITE statement, or in a READ statement that contains a control information list. The asterisk identifies the same processor-determined external unit (which must be preconnected for sequential access) that is used for a PRINT statement or for a READ statement without a control information list, respectively.

* An empty list is no longer prohibited.

The "array block" list item form was deleted. (An unqualified array name is still permitted.)

17. Input and output errors. The following changes were made:

There are no longer any "mandatory" errors that must be treated as errors by all standard conforming processors. Instead, the list of error conditions is processor dependent.

When an error or end condition occurs during an input or output operation, any implied-DO variables in the input or output list become undefined.

18. File properties. The following changes were made:

The Stream access method is no longer provided.

* A file may have more than one allowed access method (Sequential or Direct) and more than one allowed form (Formatted or Unformatted). However, for any connection at most one access method and one form are established.

* 19. Connection properties. Properties formerly associated with a file are now properties of a connection between a file and a unit. This includes the following:

An access method (Sequential or Direct) is established for the connection.

A form (Formatted or Unformatted) is established for a connection to a file that exists. (Note that a Sequential connection to a file that exists now has a form property.)

A record length is established for a connection whose access method is Direct.

A blank significance property (ZERO or NULL) is established for a connection whose form is Formatted. If the connection results from execution of an OPEN statement that does not explicitly specify a blank significance property, the default is BLANK = NULL.

* 20. Auxiliary input and output statements. The following changes were made:

New specifiers SEQUENTIAL, DIRECT, FORMATTED, and UNFORMATTED were added to the INQUIRE statement to provide information concerning the act of allowed access methods and the set of allowed forms for a file.

* The MAXREC specifier is no longer included in the OPEN and INQUIRE statements.

A BLANK specifier is now included in the INQUIRE statement, to permit determination of whether the currently established blank significance property is ZERO or NULL.

An inquiry specifier must not be referenced by any other inquiry specifier in the same INQUIRE statement (to avoid side effects).

A file that is opened as a scratch file must not be closed with KEEP status.

21. Records. The following changes were made:

Records written by a suitable explicit format may be read by list-directed formatting.

* The length of an unformatted record is measured in processor-dependent units. If the list of an unformatted WRITE does not fill the record on a file connected for direct access, the remainder of the record in undefined.

* In a formatted record, a value corresponding to an input list item of logical type may contain an optional period before the T or F (so that the logical constant forms .TRUE. and .FALSE. may be used as input values). For list-directed input, the input form corresponding to a list item of complex type may have blanks before or after the real or imaginary pert.

22. Format specification. The following changes were made:

* Negative zero is prohibited in the exponent field for E and D output.

* The form Gw.dEe was added which has the effect of Fw.d or Ew.dEe depending upon the magnitude of the datum. The form Ew.dDe was removed, but Ew.dEe was retained. [In the subset language the forms Ew.dEe and Gw.dEe, which did not appear in the Draft, were added.]

The forms +nX and -nX were changed to TRc and TLc respectively. TLc has exactly the same effect as nX, if c and n have the same values. Neither effects the output record length.

23. Intrinsic functions. The following changes were made:

Intrinsic functions ICHAR and CHAR were added, to provide conversion of a character to an integer and vice versa. The pattern matching function INDEX was also added.

* REAL is now permitted as a specific intrinsic function with integer argument. (This should encourage use of REAL and INT instead of FLOAT and IFIX in the subset language). REAL with a complex argument must now be considered a generic intrinsic function. DBLE is retained as a generic function only; DFLOAT is eliminated entirely.

* The type conversion functions (INT, IFIX, IDINT, FLOAT, SNGL, REAL, CHAR, ICBAR) are no longer permitted as actual arguments.

* The intrinsic function SIGN is no longer undefined when its second argument is zero; its value is now defined as the absolute value of the first argument.

* The generic arctangent function with two arguments was changed to ATAN2.

* The definitions of ATAN2 and CLOG were corrected.

24. Subprograms involving character data type. The following changes were made:

Character function names may now he used as actual arguments.

A statement function of type character, having constant length, is now permitted.

Character strings of "adjustable" length (specified by a non-constant length expression) ere now prohibited as dummy arguments. However, the "passed" length (determined by the length of the actual argument) is retained.

The length specification for a statement function dummy argument of character type must be an integer constant expression.

25. Other subprogram features. The following changes were made:

* An asterisk is permitted as the upper dimension bound for the last dimension of a dummy array. The "assumed size" of such a dummy array is determined from the size of the actual argument array.

If a function subprogram contains ENTRY statements, a reference to one of the function procedures defined by the subprogram must not cause definition of an associated function procedure name (as a variable) of a different type.

* A subroutine with no arguments may he defined with empty parentheses following the subprogram name. The same applies to an ENTRY statement in a subroutine. The parentheses in a FUNCTION statement are mandatory even if there is no argument list.

26. Scope of a name. The scope of the name of an implied-DO variable in a DATA statement was not specified in the Draft. The scope of such a name is now specified to be the implied-DO list.

Part 2. Actions resulting in no change to the March 1976 Draft

During the processing of public comments, proposals were rejected by the committee which would have made the following changes to language features described in the Draft published for public review in Hatch 1976:

1. Arrangement of statement lines.

Permit more than one statement to be written in the sequence of an initial line and its continuation lines, with a semicolon between statements and an optional semicolon at the and.

Permit a # character on any initial line or continuation line except an initial line that contains an END statement. Characters on the line to the right of the # would be treated as a comment.

2. Longer symbolic names.

Permit variable names up to 8 characters in length.

Permit variable names, array names, end symbolic constant names up to 16 characters in length.

3. Alternative forms for logical constants.

Permit .T. and .F. as alternative forms for logical constants.

4. Additional data types.

Add Double Precision Complex data type, or Bit data type, to FORTRAN, along with the necessary intrinsic functions, etc.

5. Different substring syntax.

Change the notation for a substring of an array element, by placing the substring specifier inside the parentheses that delimit the subscript, with the substring specifier preceding the subscript expressions.

Change the substring specifiers to first character and substring length, instead of first character and last character.

Replace the parentheses that delimit the substring specification, with angular brackets < and > .

6. Integer DO variables.

Require that a DO variable be of integer type, and that the DO loop parameters be integer expressions.

7. Expressions _and Assignment.

Provide .NG. and .NL. relational operators, as alternatives to .LE. and .GT. respectively.

Provide .XOR. logical operator, with the same precedence as .OR., instead of .NEQV.

Permit "full array references" (of the same form as an array element reference, but with asterisks in place of subscript expressions) on the left side, and in the expression on the right side, in an assignment statement. All dimensions would be required to match throughout an assignment statement containing such references. The right side would be prohibited from containing scalars or other entities partially associated with the array on the left. The interpretation would be equivalent to a sequence of arithmetic assignment statements, one for each of the elements of the arrays so referenced.

8. Control statements.

Delete the assigned GO TO statement (hut retain statement label assignment for use with FORMAT statements).

Delete the keyword THEN from the block IF and ELSE IF statements.

Add CASE, WHILE, or UNTIL statements.

Change the minimum DO loop iteration count from zero to one.

Delete the alternate return feature in subroutine calla.

Provide an internal procedure mechanism (within a program unit).

9. Input and output.

Prohibit execution of an OPEN statement specifying a unit that is already connected.

Permit a RECL specifier, to establish e maximum record length, in an OPEN statement for a sequential connection.

Provide the NAMELIST input and output features.

Provide the R edit descriptor.

10. Additional intrinsic functions.

Add the intrinsic functions EPSLN, INTXP, and SETXP, to provide information concerning precision parameters of the host computer.

Add the VERIFY function with two character string arguments, which is true if each character of the first string is also present in the second.

Add "lexical comparison" functions of logical type, one of which is true if the first argument precedes the second in the ASCII collating sequence, and the other one of which is true if the first argument follows the second in the ASCII collating sequence.

Add complex trigonometric intrinsic functions CTAN, CSINH, CCOSH, CTANH.

Add a one-argument sign function SIGNUM, whose value is -1, 0, or +1 according as the argument is negative, zero, or positive.