A U T O N E T I C S
A DIVISION OF NORTH AMERICAN AVIATION, INC.
INDUSTRIAL PRODUCTS
3584 Wilshire Blvd., Los Angeles 5, Calif.

May 4, 1960

RECOMP TECHNICAL BULLETIN NO. 10

TITLE: Translation of FORTRAN to SALT

PURPOSE: To indicate how certain FORTRAN programs may be converted to SALT language.

EFFECTIVE DATE: May 4, 1960

CONTENTS:

1. <u>Constants, variables, and subscripts</u>

1.1 Fixed point constants in FORTRAN are represented by floating point words in SALT in fixed point format. Since SALT permits more digits than does FORTRAN, no conversion is required.

1.2 Floating point constants without exponents in FORTRAN may have any number of digits. Numbers of this type must be truncated to at most 15 symbols, including decimal point and must not contain more than 11 digits in either the integer or fraction.

1.3 Floating point constants with exponents in FORTRAN have general form

$$n\ E \pm X$$

where n is a floating point constant without an exponent, E is the letter "E", and x is an integer. To convert to SALT write

$$t\ \&\ 10'\ (\pm X)$$

where t equals n truncated as per 1.2.

1.4 FORTRAN differentiates between fixed and floating point variables. SALT does not make this distinction, and hence no conversion is required.

1.5 If a FORTRAN identifier contains the letter "C", another letter must be chosen to replace it such that the identifier remains unique.

1.6 FORTRAN programs involving three dimensional arrays must be entirely recoded to include at most two dimensional arrays.

1.7 SALT subscripts must not contain any arithmetic operations. If a FORTRAN subscript involves arithmetic operations, a new unique subscript must be defined equal to the arithmetic function. This definition must precede the use of the new subscript.

2. Arithmetic statements

2.1 The following is a conversion table for the arithmetic operations.

Operation	FORTRAN	SALT
Addition	+	+
Subtraction	-	-
Multiplication	*	&
Division	/	/
Exponentiation	**	'

3. Control Statements

3.1 Unconditional "GØ TØ n"

The FORTRAN "GØ TØ n" where n is a statement number requires no conversion.

3.2 Assigned GØ TØ cannot be converted to SALT.

3.3 ASSIGN cannot be converted to SALT.

3.4 Computed GØ TØ cannot be converted to SALT.

3.5 IF requires no conversion.

3.6 SENSE LIGHT and IF (SENSE LIGHT) cannot be converted to SALT. A numerical switch, however, can be substituted for the sense light, in which case 3.5 may be used.

3.7 The FORTRAN statement IF (SENSE SWITCH i) n_1, n_2 must be replaced by IF (SENSE j) n_1, n_2, where i=1, ..., 6 and j=B, C, or D.

3.8 IF ACCUMULATØR ØVERFLØW and IF QUØTIENT ØVERFLØW cannot be converted to SALT.

3.9 IF DIVIDE CHECK cannot be converted to SALT.

3.10 PAUSE requires no conversion. PAUSE n should be translated as PAUSE.

3.11 STØP requires no conversion. STØP n should be translated as STØP.

3.12 The FORTRAN statement DØ n i=n_1, n_2 should be translated as DØ n FØR i m_1(1)m_2. The FORTRAN statement DØ n i=m_1, m_2, m_3 should be translated as DØ n FØR i m_1 (m_2) m_3.

3.13 CØNTINUE requires no conversion

3.14 EXITF, GØ, and GØGØ are examples of exit instructions which vary from company to company; they should be translated as END.

4. Input - output instructions: There are no analogies among the FORTRAN and SALT input-output instructions. Input-output regions should be entirely recoded.

5. Specification Statements

5.1 DIMENSIØN V_1, V_2, V_3, ..., V_n, where the V_i are subscripted variables, should be translated as n statements of the form ARRAY V_i.

5.2 EQUIVALENCE, FREQUENCE, and CØMMØN cannot be translated, but may in general be deleted.

6. Subroutines

6.1 Function definition formula. In FORTRAN, expressions of the type A(X)=B where A is an identifier with last letter "F", X is a set of arguments, and B is an arithmetic expression based on the arguments, define subroutines in the main program. SALT permits functions of this form only if the function identifier is defined in both the SALT and the SCRAP II macro list. X may involve exactly one argument.

6.2 Function subprograms in FORTRAN are similar to routines in SALT except for nomenclature and one restriction:

FORTRAN	SALT
CALL NAME	GØ TØ NAME
SUBRØUTINE NAME	RØUTINE NAME
RETURN	RETURN NAME

6.2.1 The SALT routinepresumes that the arguments have been stored in fixed locations prior to execution.

REFERENCES:

1. "Signal Corps RECOMP Algebraic Translater" Program No. 1034, T. J. Tobias, U.S. Army Signal Engineering Agency, Arlington, Virginia.

2. "FORTRAN", International Business Machines.

INFORMATION TO: SALT Users.

WRITTEN BY: M. F. Berman
Applied Mathematics
Autonetics Industrial Products

APPENDIX

EXAMPLE 1.

Find the approximate numerical solution of the ordinary differential equation

$$\frac{dy}{dx} = xy+1$$

in the interval $0 \leq x \leq 1$ given that $y(0) = y_o = 0$

$$\Delta y = \Delta x \Delta (x_o \; y_o + 1)$$

$$y_2 = y \text{ (at } x_1 + \Delta x) \; y_1 + \Delta x \, (x_1 \; y_1 + 1)$$

$$y_{i+1} = y_i + \Delta x \, (x_i \; y_i + 1)$$

Print at intervals of 0.01

FORTRAN

```
  READ 1, DELTAX
  PRINT 1, DELTAX
  XPRINT = 0.01
  X = 0.0
  Y = 0.0
3 Y = Y + DELTAX * (X*Y+1.0)
  X = X + DELTAX
  IF (X-XPRINT) 3, 4, 4
4 PRINT 2, X, Y,

  XPRINT = XPRINT + 0.01
  IF (X-1.0) 3, 5, 5
5 STØP
  GØ GØ
```

SALT

```
   READ DELTAX $
   PRINT DELTAX $
   XPRINT:  0.01 $
   X : 0 $
   Y : 0 $
3, Y: Y+DELTAX & (X&Y+1.0) $
   X: X+ DELTAX $
   IF (X-XPRINT) 3, 4, 4, $
4, PRINT X $
   PRINT Y $
   XPRINT : XPRINT + 0.01 $
   IF (X-1.0) 3, 5, 5 $
5, STØP $
   END $
```

EXAMPLE 2.

Multiply two matrices A and B together and leave the result in memory at C. A and B have order N by N, where N may not exceed 15.

FORTRAN

```
DIMENSIØN A(15,15), B(15,15), C(15,15)
DØ 2 I = 1, N
DØ 2 J = 1, N
C(I,J) = 0.0
DØ 2 K = 1, N
C(IJ) = C(I,J) + A(I,K)*B(K,J)
STØP
GØGØ
```

SALT

```
   ARRAY A(15, 15) $
   ARRAY B(15, 15) $
   ARRAY D(15, 15) $
   DØ 2 FØR I 1(1) N $
   DØ 2 FØR J 1(1) N $
   D(I,J) : 0 $
   DØ 2 FØR K 1(1) N $
2, D (I,J):D(I,J)+A(I,K)&B(K,J) $
   STØP $
   END $
```

EXAMPLE 3.

Write a subprogram to compute

$$Y = X^A + Z \quad \text{if } X > Z$$

$$Y = Z^A + X \quad \text{if } X \leq Z$$

FORTRAN

```
   SUBRØUTINE FINDY
   IF  (X-Z) 4, 4, 5
 4 Y = Z ** A + X
   GØ TØ 10
 5 Y = X ** A + Z
10 RETURN
```

SALT

```
     ROUTINE FINDY $
     IF  (X-Z) 4, 4, 5 $
 4,  Y: Z' A + X $
     GØ TØ 10 $
 5,  Y: X' A + Z $
10,  RETURN FINDY $
```

Enter subprogram by:

```
CALL FINDY
```

Enter subprogram by:

```
GØ TO FINDY $
```