Published: 06/01/67

Identification

The EPL run-time routine, bool_
bool_$bool_
Ruth A. Weiss

Purpose

Bool_ implements the PL/I assignment

Z = bool(x,y,w)

where Z, x, y are bit strings, varying or nonvarying and w is a 4 bit string varying or nonvarying.

If x and y are of different lengths, the shorter is extended with zeros to the length of the longer.

Usage

The call is:

call bool_$bool_(x,y,w,z)

Errors

If any argument is not a string, will stop on oct 0.
If the length of w is ≠ 4, will stop on oct 0.

Implementation

Bool_ calls movstr_ twice (See BN.7.10 for movstr_). First the longer of x or y is moved (movstr_$movb_) or complemented (movstr_$not_) into z. Then the shorter string is moved into z thru the entry to movstr_ shown in the table. For this second move z is given a dummy length. This length is either the shorter (min) or longer (max) length as shown in the table. For this second move in the cases of 0000 and 1111 the string being moved is given a length of 0.

TABLE OF MOVES FOR BOOL_

	move 1		move 2			
bool	str1 longer	str2 longer	str2 shorter		str1 shorter	
0000	movb_		movb_	max		
0001	movb_		and_	max		
0010	movb_	not_	notand_	min	and_	max
0011	movb_	not_	return		movb_	max
0100	not_	movb_	and_	max	notand_	min
0101	not_	movb_	movb_	max	return	
0110	movb_		exclor_	min		
0111	movb_		or_	min		
1000	not_		notand_	min		
1001	not_		exclor_	min		
1010	movb_	not_	nnot_	max	return	
1011	movb_	not_	notor_	max	or_	min
1100	not_	movb_	return		nnot_	max
1101	not_	movb_	or_	min	notor_	max
1110	not_		notor_	max		
1111	not_		nnot_	max		