CGTM No. 21

A Study of Swap Instructions for the IBM System/360

by

John H. Welsch

July 1967

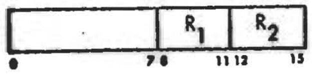
SLAC - Stanford University

There are two logically different types of arithmetic registers, the general registers and the floating-point registers with no direct way to exchange information between them. This report discusses the need for such instructions and describes a few proposals.

The IBM System/360 has two sets of addressable arithmetic registers: 16 general registers (32 bits long) and 4 floating-point registers (64 bits long). There is no direct way to transmit information between the two kinds of registers. To transfer 64 bits between one floating-point register and an even-odd pair of general registers takes two 4-byte instructions and 8 bytes of temporary core storage. This transfer could be done by one 2-byte instruction and no temporary storage. Several such instructions are described below. The SWPR instruction has been installed on several model 65's. The other two instructions (LDGR and STGR) are suggestions for one-way communication which would be a little faster and possibly more convenient in some cases.

Load from General Register

LDGR R_1, R_2 *[RR, Long Operands]*



The contents of the even-odd general register pair specified by the second operand are placed in the first operand location.

Condition Code: The code remains unchanged.

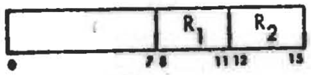
Program Interruptions:

Operation (if floating-point feature is not installed)

Specification

Store in General Register

STGR R_1, R_2 *[RR, Long Operands]*



The first operand is stored in the even-odd general register pair specified by the second operand.

Condition Code: The code remains unchanged.

Program Interruptions:

Operation (if floating-point feature is not installed)

Specification

Swap Registers

SWPR R_1,R_2 [RR, Long Operands]

	R_1	R_2
0 7	8 11	12 15

The first operand is a double word contained in an adjacent pair of general registers. R_1 designates the leftmost register of the pair. R_1 must be even, otherwise a specification exception is recognized. The second operand is a double word contained in a floating point register. R_2 must be 0, 2, 4 or 6, otherwise a specification exception is recognized.

The first operand replaces the second operand, and the second operand replaces the first operand.

Condition Code: The code remains unchanged.

Program Interruptions: Specification

Execution Times (approximate, in μ seconds)

	SWPR	LDGR and STGR	[STD LM]
Model 50	4.50	3.50	13.00
Model 65I	1.75	1.23	2.33
Model 75I	.60	.40	2.62
			+ 8 bytes

The last column simulates the STGR with existing instructions.

Uses

Most of the subroutines in the function libraries of FORTRAN, PL/I and ALGOL could make good use of instructions of this type, where the argument is broken into the characteristic part and fraction part and each part operated on separately in the different sets of registers. Subroutines for FIX, FLOAT and rounding can be improved with the STGR and LDGR instructions.

Of increasing importance is the ability to write re-entrant subroutines. These could be more easily written since the need for local storage is decreased by juggling all work in the general and floating-point registers.