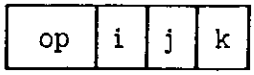
LOGICAL OPERATIONS

A comprehensive set of logical operations is included on the arithmetic registers, the index registers, and condition bits. For most of the logical instructions the two operands are treated as either 1-, 24-, or 48-bit quantities and a logical connective is applied bit by bit. However, for the "count" instructions a function is computed, not on corresponding pairs of bits of different operands, but on all 24 or 48 bits of one operand.

All logical operations have the short format:



where the j- and k-fields designate the operand registers or bits and the i-field designates the result register or bit. The contents of the operand registers or bits are not changed by the execution of a logical operation.

The basic set of logical operations provides for eight logical connectives, applied bit by bit on the operands. The truth tables for these eight functions are:

function	function value a 0 0 1 1 b 0 1 0 1	common names of function	base mnemonic
$a \wedge b$	0 0 0 1	and, logical product	AND
$a \wedge \bar{b}$	0 0 1 0	logical difference	TAF
$\bar{a} \wedge \bar{b}$	1 0 0 0	nor, Peirce stroke	FAF
$a \vee b$	0 1 1 1	or, logical sum	OR
$a \vee \bar{b}$	1 0 1 1	cover	TOF
$\bar{a} \vee \bar{b}$	1 1 1 0	nand, Scheffer stroke	FOF
$a = b$	1 0 0 1	equivalence	EQ
$a \neq b$	0 1 1 0	not equal, exclusive or, modulo 2 sum	XOR

It should be noted that all sixteen possible Boolean functions of two variables can be computed by these eight operations by interchanging the names in the j- and k-fields or by setting k equal to j. In particular are the following common functions (where R may be interpreted as either A, X, or c):

move	$R^i \leftarrow R^j$	$R^i \leftarrow R^j \wedge R^j$
complement and move	$R^i \leftarrow \bar{R}^j$	$R^i \leftarrow \bar{R}^j \wedge \bar{R}^j$
set to 0's	$R^i \leftarrow 0$'s	$R^i \leftarrow R^i \wedge \bar{R}^i$
set to 1's	$R^i \leftarrow 1$'s	$R^i \leftarrow R^i \vee \bar{R}^i$

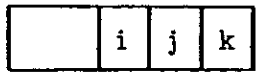
In addition to the operations included in this section, the shift instructions and certain move instructions provide logical (i. e. bit by bit) functions.

Logical Operations, Arithmetic Unit

ANDA	$A^i \leftarrow A^j \wedge A^k$
TAFA	$A^i \leftarrow A^j \wedge \bar{A}^k$
FAFA	$A^i \leftarrow \bar{A}^j \wedge \bar{A}^k$
ORA	$A^i \leftarrow A^j \vee A^k$
TOFA	$A^i \leftarrow A^j \vee \bar{A}^k$
FOFA	$A^i \leftarrow \bar{A}^j \vee \bar{A}^k$
EQA	$A^i \leftarrow A^j = A^k$
XORA	$A^i \leftarrow A^j \neq A^k$

Exceptions: none

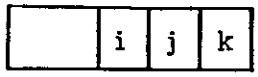
Logical Operations, Index Unit



ANDX	$X^i \leftarrow X^j \wedge X^k$
TAFX	$X^i \leftarrow X^j \wedge \bar{X}^k$
FAFX	$X^i \leftarrow \bar{X}^j \wedge \bar{X}^k$
ORX	$X^i \leftarrow X^j \vee X^k$
TOFX	$X^i \leftarrow X^j \vee \bar{X}^k$
FOFX	$X^i \leftarrow \bar{X}^j \vee \bar{X}^k$
EQX	$X^i \leftarrow X^j = X^k$
XORX	$X^i \leftarrow X^j \neq X^k$

Exceptions: none

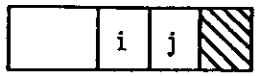
Logical Operations, Condition Bits



ANDC	$c_i \leftarrow c_j \wedge c_k$
TAFC	$c_i \leftarrow c_j \wedge \overline{c}_k$
FAFC	$c_i \leftarrow \overline{c}_j \wedge \overline{c}_k$
ORC	$c_i \leftarrow c_j \vee c_k$
TOFC	$c_i \leftarrow c_j \vee \overline{c}_k$
FOFC	$c_i \leftarrow \overline{c}_j \vee \overline{c}_k$
EQC	$c_i \leftarrow c_j = c_k$
XORC	$c_i \leftarrow c_j \neq c_k$

Exception	Exception bit
c_{24} set to 0 or c_{25} set to 1	CC

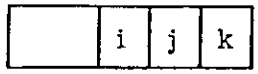
Count Total Ones, Arithmetic CNTT



The contents of register A^i are replaced by the number of bits of register A^j which have the value 1.

Exceptions: none

Count Leading Alike, Arithmetic CNTAA



The contents of register A^i are replaced by the number of leading bits of register A^j which have the value of the bit A^k_0. The bits of A^j are examined in the order A^j_0, A^j_1, A^j_2, and so on.

Note that if the k-field specifies A^0, the effect is to count leading 0's.

Exceptions: none

Count Leading Different, Arithmetic CNTDA

The contents of register A^i are replaced by the number of leading bits of register A^j which are different in value from the value of bit A^k_0 (that is, have the value $\bar{A}^k_0$). The bits of A^j are examined in the order A^j_0, A^j_1, A^j_2, and so on.

Note that if the k-field specifies A^0, the effect is to count leading 1's.

Exceptions: none

Count Leading Alike, Index CNTAX

The contents of register X^i are replaced by the number of leading bits of register X^j which have the value of the bit X^k_0. The bits X^j are examined in the order X^j_0, X^j_1, X^j_2, and so on.

Note that if the k-field specifies X^0, the effect is to count leading 0's.

Exceptions: none

Count Leading Different, Index CNTDX

The contents of register X^i are replaced by the number of leading bits of register X^j which are different in value from the value of bit X^k_0 (that is, have the value $\bar{X}^k_0$). The bits of X^j are examined in the order X^j_0, X^j_1, X^j_2, and so on.

Note that if the k-field specifies X^0, the effect is to count leading 1's.

Exceptions: none