DATA GENERAL
CORPORATION
Southboro,
Massachusetts 01772
(617) 485-9100

PROGRAM

Logical Inclusive OR

TAPES

ASCII Source: 090-000026

ABSTRACT

This routine computes the logical inclusive OR of two 16-bit numbers.

093-000022-01

1. REQUIREMENTS

1.1 Memory

1K or larger memory

1.2 Equipment

NOVA central processor

1.3 External Subroutines

None

1.4 Other

None

2. OPERATING PROCEDURE

2.1 Calling Sequence

```
JSR  .OR
return
```

2.2 Input Format

One 16-bit quantity is passed in ACØ, the second in AC1.

2.3 Output Format

The inclusive OR of the two quantities is returned in ACØ.

2.4 Error Returns

None

2.5 State of the Active Registers upon Exit

AC2, AC3, and Carry are unchanged. ACØ, and AC1 are destroyed.

2.6 Cautions to User

None

3. DISCUSSION

3.1 Algorithms

With an inclusive OR, a bit of the result is 1 if either of the corresponding bits is 1. Otherwise, the result bit is Ø. The algorithm for full words is

$$A \vee B = A \wedge \sim B + B$$

Taking the arguments as single bits, if B is 1, $A \wedge \sim B$ is Ø regardless of the state of A, and the expression on the left is 1. If B is Ø, the expression is 1 or Ø as A is 1 or Ø. In no case are $A \wedge \sim$ B and B both 1, so the full word addition generates no carries.

3.2 Limitations and Accuracy

The routine is exact.

3.3 Size and Timing

.OR is 4 words in length.

Execution time is 20.0 μseconds.

3.4 References

Section 2.2 of "How to Use the NOVA" contains a further discussion of logical arithmetic.

3.5 Flow Diagrams

None

4. EXAMPLES AND APPLICATIONS

The ASCII source of .OR is provided with the NOVA software. If a user routine requires inclusive OR, this tape should be edited into his software.

5. PROGRAM LISTING

A listing of .OR follows. No origin is given in the source, enabling the tape to be edited anywhere within a user routine.

```
                ; LOGICAL OP
                ; COMPUTES THE LOGICAL INCLUSIVE OR OF
                ; TWO UNSIGNED NUMBERS

                ; INPUT:         A IN AC0, B IN AC1

                ; OUTPUT:        A .OR. B IN AC0

                ; CALLING SEQUENCE:
                ;       JSR     .0R
                ;       RETURN

                ; DESTROYED:     AC0, AC1
                ; UNCHANGED:     AC2, AC3, CARRY

                ; METHOD:        A .OR. B = .NOT.B .AND. A + B


00000 124000 .OR:       COM 1,1         ; .NOT. B
00001 123400            AND 1,0         ; A .AND. .NOT. B
00002 122000            ADC 1,0         ; .NOT. B .AND. A + B
00003 001400            JMP 0,3         ; RETURN
```