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

## PROGRAM

Logical Exclusive OR

## TAPES

ASCII Source: 090-000025

## ABSTRACT

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

# 1. REQUIREMENTS

## 1.1 Memory

1K or larger alterable memory

## 1.2 Equipment

NOVA central processor

## 1.3 External Subroutines

None

## 1.4 Other

None

# 2. OPERATING PROCEDURE

## 2.1 Calling Sequence

```
JSR  .XOR
return
```

## 2.2 Input Format

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

## 2.3 Output Format

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

## 2.4 Error Returns

None

## 2.5 State of Active Registers upon Exit

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

### 2.6 Cautions to User

None

## 3. DISCUSSION

### 3.1 Algorithms

With an exclusive OR, a bit of the result is 1 if the corresponding operand bits differ. Otherwise, the result bit is Ø. This is equivalent to the arithmetic sum if carries from one bit position to the next are ignored. A Carry from the ith position occurs only if the ith bits are both 1. If a Carry occurs, it has twice the value of a 1 in the ith position. We can therefore cancel the effects of any carries using the following:

$$A \oplus B = A + B - 2(A \wedge B).$$

### 3.2 Limitations and Accuracy

The routine is exact.

### 3.3 Size and Timing

.XOR is 7 words in length.

Execution time is 34.Ø $\mu$ seconds.

### 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 .XOR is provided with the NOVA software. If a user routine requires exclusive OR, this source should be edited into the user routine.

## 5. PROGRAM LISTING

A listing of .XOR follows. No origin is given in the source, enabling the tape to be edited anywhere within a user's routines.

```
                  ; LOGICAL EXCLUSIVE OR
                  ; COMPUTES THE EXCLUSIVE OR OF TWO UNSIGNED NUMBERS

                  ; INPUT:          A IN ACØ, B IN AC1

                  ; OUTPUT:         A .XOR. B IN ACØ

                  ; CALLING SEQUENCE:
                  ;       JSR     .XOR
                  ;       RETURN

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

                  ; METHOD:         A .XOR. B = A + B -2(A .AND. B)

ØØØØØ Ø54ØØ6 .XOR:    STA 3,.CDØ3        ; SAVE RETURN
ØØØØ1 135ØØØ          MOV 1,3            ; B TO AC2
ØØØØ2 11752Ø          ANDZL Ø,3          ; 2(A .AND. B)
ØØØØ3 123ØØØ          ADD 1,Ø            ; A + B
ØØØØ4 1624ØØ          SUB 3,Ø            ; A + B -2(A .AND. B)
ØØØØ5 ØØ2ØØ6          JMP @.CDØ3         ; RETURN


ØØØØ6 ØØØØØØ .CDØ3:   Ø                  ; SAVE RETURN
```