BITBYT	CERN Program Library	M421

Author(s) : C. Letertre, J. Zoll **Library:** KERNLIB
Submitter : **Submitted:** 28.01.1971
Language : Fortran or Assembler **Revised:** 12.06.1987

Package for Handling Bits and Bytes

This package manipulates individual bits and bytes in a word.

A *bit* in a word is specified by giving its position `J` $(= 1, 2, \ldots, 32[, \ldots, 64])$ in the word, bit `1` being the least significant bit.

A *byte* in a word is a group of `NBITS` consecutive bits. The byte is specified by giving `NBITS` and the bit position `J` of the least significant bit of the byte.

Structure:

`SUBROUTINE` and `FUNCTION` subprograms

User Entry Names: `JBIT`, `SBIT0`, `SBIT1`, `SBIT`, `MSBIT0`, `MSBIT1`, `MSBIT`, `JBYT`, `SBYT`, `MSBYT`, `CBYT`, `MCBYT`, `JBYTET`, `JBYTOR`, `SBYTOR`, `MBYTOR`, `MBYTET`, `JRSBYT`

Usage:

`IX = JBIT(IW,J)`	returns `IX` = `0` or `1`, the value of bit `J` in `IW`.
`CALL SBIT0(IX,J)`	sets `0` into bit `J` of `IX`.
`CALL SBIT1(IX,J)`	sets `1` into bit `J` of `IX`.
`CALL SBIT(IA,IX,J)`	copies bit `1` of `IA` into bit `J` of `IX`.
`IX = MSBIT0(IW,J)`	returns `IW` in `IX` with bit `J` set to `0`.
`IX = MSBIT1(IW,J)`	returns `IW` in `IX` with bit `J` set to `1`.
`IX = MSBIT(IA,IW,J)`	returns `IW` in `IX` with bit `J` set to the value of bit `1` in `IA`.
`IX = JBYT(IW,J,NBITS)`	returns in `IX` right-justified the byte at `J` in `IW`.
`CALL SBYT(IA,IX,J,NBITS)`	copies the byte at `1` of `IA` into the byte at `J` of `IX`.
`IX = MSBYT(IA,IW,J,NBITS)`	returns `IW` in `IX` with the byte at `J` replaced by the byte at `1` of `IA`.
`CALL CBYT(IA,JA,IX,J,NBITS)`	copies the byte at `JA` of `IA` into the byte at `J` of `IX`.
`IX = MCBYT(IA,JA,IW,J,NBITS)`	returns `IW` in `IX` with the byte at `J` replaced by the byte at `JA` of `IA`.
`IX = JBYTET(IA,IW,J,NBITS)`	returns in `IX` the logical `AND` of `IA` and the byte at `J` of `IW` right-justified.
`IX = JBYTOR(IA,IW,J,NBITS)`	returns in `IX` the logical `OR` of `IA` and the byte at `J` of `IW` right-justified.
`CALL SBYTOR(IA,IX,J,NBITS)`	replaces the byte at `J` in `IX` by the logical `OR` of this byte and the byte at `1` of `IA`.
`IX = MBYTOR(IA,IW,J,NBITS`	returns `IW` in `IX` with the byte at `J` replaced by the logical `OR` of this byte and the byte at `1` of `IA`.
`IX = MBYTET(IA,IW,J,NBITS)`	returns `IW` in `IX` with the byte at `J` replaced by the logical `AND` of this byte and the byte at `1` of `IA`.
`IY = JRSBYT(IA,IX,J,NBITS)`	read and reset byte; equivalent to `IY = JBYT(IX,J,NBITS)` `CALL SBYT(IA,IX,J,NBITS)`.

Notes:

The subroutines

```
SBIT0   SBIT1   SBIT    SBYT    CBYT   SBYTOR
```

are duplicated by the functions

```
MSBIT0  MSBIT1  MSBIT   MSBYT   MCBYT   MBYTOR
```

to allow implementation by statement functions. Such implementations can be picked up from the ZEBRA CDE Pam-file for different machines as sequence definitions

```
Q$JBIT    :   JBIT,   JBYT
Q$SBIT    :   MSBIT0, MSBIT1, MSBIT
Q$SBYT    :   MSBYT
Q$CBYT    :   MCBYT
Q$JBYTET  :   JBYTET, JBYTOR, MBYTET, MBYTOR
```

•