**TP D1(L1,B1)**

# Test Decimal

**RSL-a**

| | | | | | |
|---|---|---|---|---|---|
| EB | $L_1$//// | $B_1D_1$ | $D_1D_1$ | //////// | C0 |

**TP** is an RSL-a instruction that is used to test a field to see if it is in a packed decimal format. A field of length 16 or less is in a packed decimal format if each byte contains two hex digits in the range 0-9, except the rightmost byte which contains a hex digit (0-9) followed by valid sign (hex values A, B,C, D, E, or F). The hex signs A, C, E, and F are positive, while hex signs B and D are negative.

This instruction is a little unusual in that it has a single operand, which denotes the field that is being tested. As a result, some of the space in the object code is unused as shown in the object code diagram above. TP also has a two-byte op-code – EBC0. TP examines the target field and sets the condition code to indicate the results:

Condition Code:

| | | |
|---|---|---|
| 0 | - | All digit codes and the sign valid |
| 1 | - | Sign invalid |
| 2 | - | At least one digit code invalid |
| 3 | - | Sign invalid and at least one digit code invalid |

As you can see, in those cases where the target field is not packed, the instruction also provides information about what is wrong with the tested field.

Consider the following TP example.

```
           TP   APK
           BNZ  NOTPACKED
           …
NOTPACKED EQU  *
```

Assume that APK is in the following state when tested.

```
APK      DC  XL4'0000123C'
```

After executing the TP instruction above, the condition code is set to 0, since the field is packed. As a result, the branch is not taken. On the other hand if APK had contained X'00000223', then TP would have set the condition code to 1 (bad sign), and the branch would have been taken.

Occasionally, date information is stored in a non-native format called "packed no-sign". For instance, the month, day, and year can be stored as four bytes |mm|dd|yy|yy| where the numeric portion of the rightmost byte is used to store a digit instead of a sign. If the same information were stored in a packed decimal field, it might look like |mm|dd|yy|yy|0C|which requires an extra, wasted byte. The problem with packed no-sign fields is that being a non-native format, there aren't any machine instructions specifically designed to support it. Programmers have resorted to using TRT tables to recognize when a field contains valid packed no-sign data. But TP provides some support for this data format if we are clever. By moving a packed no-sign field adjacent to a one byte packed zero (X'0C') and then using TP to test the entire area, only two conditions codes are possible: 1) 0 – all the packed no-sign digits are valid, and 2) 2 – one of the packed no-sign digits was invalid.

Here's an example. Assume X is supposed to be a packed no-sign field with 4 bytes. We define the following fields,

```
WORK      DS  0XL5
XWORK     DS   XL4
          DC   X'0C'
```

We can then test the field like this,

```
       MVC  XWORK,X    BUILD THE WORK AREA
       TP   WORK       IS XWORK PACKED NO-SIGN?
       BNE  BADFIELD   BRANCH IF BAD
```

**Examples**

**Some unrelated TP's:**

```
A         DC  X'12345C'     GOOD DIGITS, GOOD SIGN
B         DC  X'123456'     GOOD DIGITS, BAD SIGN
C         DC  X'1A345C'     BAD DIGIT, GOOD SIGN
D         DC  X'1A3456'     BAD DIGIT, BAD SIGN

     Results:
          TP  A     C.C. = 0 - PACKED FIELD
          TP  B     C.C. = 1 - BAD SIGN
          TP  C     C.C. = 2 - BAD DIGIT
          TP  D     C.C. = 3 - BAD DIGIT, BAD SIGN
```

## Tips

1. Avoid S0C7s by using TP to test fields which are required to be packed.
2. TP can be used to test packed no-sign fields instead of using the more complicated TRT technique.