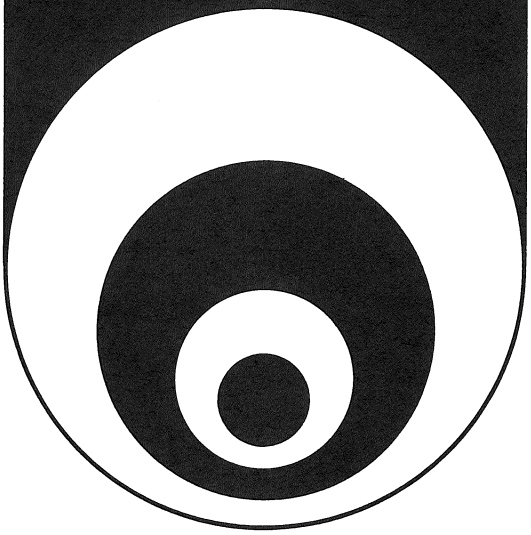
signetics

# MOS MICROPROCESSOR



# BIT AND BYTE TESTING PROCEDURES AS51

## SUMMARY

This applications memo describes several methods of testing the contents of the internal registers in the Signetics 2650 Microprocessor.

The following test examples are given:

- Specific bit(s) in a register.
- Positive, negative, or zero-contents of a register.
- Contents of a register compared with a value (equals, greater than, or less than).
- Interdigit-carry (IDC), overflow (OVF), and carry (C) flags in the program status word.

## INTRODUCTION

As a result of an operation on register(s) of the 2650 register bank, five bits (bits 7, 6, 5, 2, and 0) in the Program Status Lower (PSL) portion of the Program Status Word (PSW) register can be affected.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CC1 | CC0 | IDC | RS | WC | OVF | COM | C |

PROGRAM STATUS LOWER (PSL)

These bits are affected as follows:

### CC1, CC0: Condition Code Bits

| CONDITION CODE | | RESULT OF | | |
|---|---|---|---|---|
| CC1 | CC0 | LOAD/STORE, ARITHMETIC, LOGICAL INSTRUCTIONS | COMPARE INSTRUCTION | SELECTIVE TESTS ON BITS (TMI, TPSU, & TPSL) |
| 0 | 0 | Zero | Equal | All bits 1 |
| 0 | 1 | Positive | Greater Than | ——— |
| 1 | 0 | Negative | Less Than | Not all bits 1 |

### IDC: Interdigit Carry/$\overline{\text{Borrow}}$ Bit

The IDC bit is affected by arithmetic operations as well as rotation.

0 = Interdigit borrow/no interdigit carry
1 = Interdigit carry/no interdigit borrow

### OVF: Overflow Bit. Arithmetic Operation

The overflow bit in arithmetic operations is set as follows:

Operand 1 ± Operand 2→ Result

| SIGN | | | ADD OVF | SUB OVF |
|---|---|---|---|---|
| OPERAND 1 | OPERAND 2 | RESULT | | |
| + | + | + | 0 | 0 |
| + | + | – | 1 | 0 |
| + | – | + | 0 | 0 |
| + | – | – | 0 | 1 |
| – | + | + | 0 | 1 |
| – | + | – | 0 | 0 |
| – | – | + | 1 | 0 |
| – | – | – | 0 | 0 |

### OVF: Overflow Bit. Rotate Operation

Condition: WC = 1; if WC = 0, the OVF bit is not affected.

The overflow bit is set as follows:

| OPERAND SIGN | | OVF |
|---|---|---|
| BEFORE ROTATE | AFTER ROTATE | |
| + | + | 0 |
| + | – | 1 |
| – | + | 0 |
| – | – | 0 |

### C: Carry/$\overline{\text{Borrow}}$ Bit

The Carry bit is affected by arithmetic operations as well as rotation.

0 = borrow/no carry
1 = carry/no borrow

## BIT TESTING PROCEDURES

The bits of a register Rx (register zero Ro or any register R1, R2 or R3 in the selected register bank) can be tested as follows:

| | | | | BYTES | CYCLES |
|---|---|---|---|---|---|
| **TEST FOR '0' IN BIT 3 OF Rx** | | | | | |
| TMI, Rx | H'08' | | 1) | 2 | 3 |
| BCTR, 2 | LBL | *Branch if bit 3 is zero. | | 2 | 3 |
| | | | | 4 | 6 |
| or: | | | | | |
| ANDI, Rx | H'08' | | 2) | 2 | 2 |
| BCTR, 0 | LBL | *Branch if bit 3 is zero. | | 2 | 3 |
| | | | | 4 | 5 |

While the second test is faster, it affects the contents of Rx.

## BIT TESTING PROCEDURES (Continued)

### TEST FOR '1' IN BIT 3 OF Rx

```
TMI, Rx   H'08'                          1)  2 | 3
BCTR, 0   LBL    *Branch if bit 3 is one.    2 | 3
                                             4 | 6
```

or:

```
ANDI, Rx  H'08'                          2)  2 | 2
BCFR, 0   LBL    *Branch if bit 3 is one.    2 | 3
                                             4 | 5
```

While the second test is faster, it affects the contents of Rx.

### TEST FOR '0' IN BIT 1 OR BIT 3 OR BIT 6 OF Rx

```
TMI, Rx   H'4A'                          1)  2 | 3
BCTR, 2   LBL    *Branch if one of the       2 | 3
                 tested bits is zero.        4 | 6
```

### TEST FOR '1' IN BIT 1 OR BIT 3 OR BIT 6 OF Rx

```
ANDI, Rx  H'4A'                          2)  2 | 2
BCFR, 0   LBL    *Branch if one of the       2 | 3
                 tested bits is one.         4 | 5
```

### TEST FOR '0' IN BIT 1 AND BIT 3 AND BIT 6 OF Rx

```
ANDI, Rx  H'4A'                          2)  2 | 2
BCTR, 0   LBL    *Branch if all tested       2 | 3
                 bits are zero.              4 | 5
```

### TEST FOR '1' IN BIT 1 AND BIT 3 AND BIT 6 OF Rx

```
TMI, Rx   H'4A'                          1)  2 | 3
BCTR, 0   LBL    *Branch if all tested       2 | 3
                 bits are one.               4 | 6
```

### TEST FOR PATTERN IN Rx; e.g., x10xx01x

x = don't care

```
IORI, Rx  H'99'                          2)  2 | 2
COMI, Rx  H'DB'                              2 | 2
BCTR, 0   LBL    *Branch if pattern          2 | 3
                 is true.                    6 | 7
```

1) Contents of register Rx kept
2) Contents of register Rx lost

## BYTE TESTING PROCEDURES

### TEST FOR POSITIVE, NEGATIVE AND ZERO

All of the tests described below must be preceded by an operation on Rx which updates the contents of the condition register, e.g., by instructions such as LOAD, ADD, AND, COMPARE, ROTATE, I/O, etc.

| | CC | OPERATION |
|---|---|---|
| Test for (Rx) $\geq$ 0 | 00 or 01 | BCFR, 2 |
| Test for (Rx) $>$ 0 | 01 | BCTR, 1 |
| Test for (Rx) = 0 | 00 | BCTR, 0 |
| Test for (Rx) $<$ 0 | 10 | BCTR, 2 |
| Test for (Rx) $\leq$ 0 | 00 or 10 | BCFR, 1 |

### TESTS ON THE CONTENTS OF A REGISTER BY USING COMPARE INSTRUCTIONS

**Logical compare:** (COM = 1 in PSL)
Comparison is made between two 8-bit unsigned binary numbers.

**Arithmetic compare:** (COM = 0 in PSL)
Comparison is made between two 8-bit signed numbers.

After execution of the logic or arithmetic compare instruction, the condition register (CC) is set to a specific value and tested as follows:

| REGISTER-TO-REGISTER COMPARE | | |
|---|---|---|
| **Instruction used:** COMZ Rx | | |
| **RESULT** | **CC** | **TEST** |
| (Ro) $\geq$ (Rx) | 00 or 01 | BCFR, 2 |
| (Ro) $>$ (Rx) | 01 | BCTR, 1 |
| (Ro) = (Rx) | 00 | BCTR, 0 |
| (Ro) $<$ (Rx) | 10 | BCTR, 2 |
| (Ro) $\leq$ (Rx) | 00 or 10 | BCFR, 1 |

| REGISTER TO CONSTANT OR MEMORY LOCATION | | |
|---|---|---|
| **Instructions used:** COMI, Rx DATA; COMR, Rx RELATIVE LOCATION OF DATA; COMA, Rx LOCATION OF DATA | | |
| **RESULT V=VALUE** | **CC** | **TEST** |
| (Rx) $\geq$ V | 00 or 01 | BCFR, 2 |
| (Rx) $>$ V | 01 | BCTR, 1 |
| (Rx) = V | 00 | BCTR, 0 |
| (Rx) $<$ V | 10 | BCTR, 2 |
| (Rx) $\leq$ V | 00 or 10 | BCFR, 1 |

Whenever a compare instruction is used, the IDC, OVF, or C bits in the PSL are *not* affected.

## TEST ON OVERFLOW (OVF in PSL)

The overflow bit is affected whenever arithmetic or rotate instructions are executed.

The *OVF bit* is set during an addition whenever the two operands have the same sign and the result has a different sign. During a subtraction, the *OVF bit* is set when the operands differ in sign and the result has a different sign than the first operand.

*Examples:*

| | |
|---|---|
| (+A) + (+B) = (−C) | OVF |
| (−A) + (−B) = (+C) | OVF |
| (+A) − (−B) = (−C) | OVF |
| (−A) − (+B) = (+C) | OVF |

```
Test:   TPSL      H'04'   *OVF test
        BCTR, 0   LBL     *Branch if OVF = set
```

The *OVF bit* is set during rotate instructions with WC = 1 whenever the sign changes from positive to negative. If WC = 0, then rotate instructions do not affect the OVF bit.

*Example:*

```
        RRR, Rx           *Rotate right
        TPSL      H'04'   *Test OVF bit
        BCTR, 0   LBL     *Branch if OVF = set
```

## TEST ON CARRY (C in PSL)

The carry bit is set to 1 by an add instruction that generates a carry and a sub-instruction that does *not* generate a borrow.

*Example:*

**ADDITION**

```
        LODI, Rx  H'88'
        ADDI, Rx  H'99'
        TPSL      H'01'   *Test carry
        BCTR, 0   LBL     *Branch if carry
```

**SUBTRACTION**

```
        LODI, Rx  H'40'
        SUBI, Rx  H'30'
        TPSL      H'01'   *Test borrow
        BCTR, 0   LBL     *Branch if no borrow
```

When a rotate instruction is executed with WC = 1, the carry bit is also affected. Refer to the Signetics 2650 Microprocessor manual for a description of this operation.

signetics
a subsidiary of U.S. Philips Corporation

Signetics Corporation
811 East Arques Avenue
Sunnyvale, California 94086
Telephone 408/739-7700