Overflow Flag

Young W. Lim

2024-07-19 Fri

Outline

Based on

1. "Self-service Linux: Mastering the Art of Problem Determination", Mark Wilding
1. "Computer Architecture: A Programmer's Perspective", Bryant & O'Hallaron

Compling 32-bit program on 64-bit gcc

- `gcc -v`
- `gcc -m32 t.c`
- `sudo apt-get install gcc-multilib`
- `sudo apt-get install g++-multilib`
- `gcc-multilib`
- `g++-multilib`
- `gcc -m32`
- `objdump -m i386`

TOC: Overflow flag

TOC Overflow flag in unsigned and signed computations

- Overflow flag

Overflow flag (1)

- overflow flag is based on signed arithmetic
- to decide if the overflow flag is turned on or off, only need to look at the sign bits (leftmost) of the three numbers

augend	+	addend	=	sum
minuend	−	subrahend	=	difference

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

Overflow flag (2)

- in signed arithmetic,
  - watch the overflow flag to detect errors
  - overflow flag on means the result is wrong
  - errors can be detected by examining
    the sign of the result, in the 2's complement arithmetic
    ($P + P \to N$ or $N + N \to P$)
- in unsigned arithmetic,
  - the overflow flag tells you nothing interesting

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

Overflow flag (3)

- when two positive numbers are added
  - if the result is a positive, $(P + P \rightarrow P)$, then no overflow
  - if the result is a negative, $(P + P \rightarrow N)$, then overflow
- when two negative numbers are added
  - if the result is a negative, $(N + N \rightarrow N)$, then no overflow
  - if the result is a positive, $(N + N \rightarrow P)$, then overflow

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

Overflow flag (4)

- adding _negative (N) and positive (P) numbers cannot be wrong, because the sum is between the addends ([N, P]) .
  - if opposite signed numbers are added, then no overflow
  - both of the addends lies
    in the allowable range of numbers,
    their sum is between the opposite signed addends,
    therefore the sum lies also in the allowable range
  - (P + N → P or N) no overflow always
  - (N + P → P or N) no overflow always

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

TOC Rules for the overflow flag

- the 1st rule for setting `OF`
- the 2nd rule for setting `OF`
- cases for clearing `OF` ($1 \sim 6$)

Overflow flag setting and clearing conditions

		Method 1		Method 2
		ADD conditions	SUB conditions	
1	OF=1	$P + P \rightarrow N$	$P - N \rightarrow N$	$c_n \bigoplus c_{n-1} = 1$
2	OF=1	$N + N \rightarrow P$	$N - P \rightarrow P$	$c_n \bigoplus c_{n-1} = 1$
3	OF=0	$P + P \rightarrow P$	$P - N \rightarrow P$	$c_n \bigoplus c_{n-1} = 0$
4	OF=0	$N + N \rightarrow N$	$N - P \rightarrow N$	$c_n \bigoplus c_{n-1} = 0$
5	OF=0	$P + N \rightarrow P$	$P - N \rightarrow P$	$c_n \bigoplus c_{n-1} = 0$
6	OF=0	$P + N \rightarrow N$	$P - P \rightarrow N$	$c_n \bigoplus c_{n-1} = 0$
7	OF=0	$N + P \rightarrow P$	$N - N \rightarrow P$	$c_n \bigoplus c_{n-1} = 0$
8	OF=0	$N + P \rightarrow N$	$N - P \rightarrow N$	$c_n \bigoplus c_{n-1} = 0$

$$+P = -(-P) \quad = -N$$
$$+N = -(-N) \quad = -P$$

1. $P + P \rightarrow N$ $(P - N \rightarrow N)$

If the sum of two signed numbers with the sign bits off (0, 0) yields a result number with the sign bit on (1),

the overflow flag is turned on

```
signed addition          signed subtraction          unsigned addition

0100   carries
 0100   (+4)              0100   (+4)                  0100   ( 4)
+0100  +(+4)             -1100  -(-4)                 +0100  +( 4)
-----  -----             -----  -----                 -----  -----
01000   (-8)             01000   (-8)                 01000   ( 8)
```

Method 1	OF = 1	when $\overline{a_{n-1}} \cdot \overline{b_{n-1}} \cdot s_{n-1}$ for addition and $\overline{a_{n-1}} \cdot b_{n-1} \cdot s_{n-1}$ for subtraction
Method 2	OF = $c_n \bigoplus c_{n-1}$	$c_4 \bigoplus c_3 = 0 \bigoplus 1 = 1$

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

The 2nd case - setting the overflow flag

2. $N + N \rightarrow P$ $(N - P \rightarrow P)$

If the sum of two numbers with the sign bits on (1, 1)
yields a result number with the sign bit off (0)

the overflow flag is turned on.

```
signed addition           signed subtraction          unsigned addition

1001    carries
 1001    (-7)              1001    (-7)                1001    ( 9)
+1001   +(-7)             -0111   -(+7)               +1001   +( 9)
-----   -----             -----   -----               -----   -----
10010    ( 2)             10010    ( 2)               10010    (18)
```

Method 1	OF = 1	when $a_{n-1} \cdot b_{n-1} \cdot \overline{s_{n-1}}$ for addition and $a_{n-1} \cdot \overline{b_{n-1}} \cdot \overline{s_{n-1}}$ for subtraction
Method 2	OF = $c_n \bigoplus c_{n-1}$	$c_4 \bigoplus c_3 = 1 \bigoplus 0 = 1$

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

The 3rd case - clearing the overflow flag

3. $P + P \to P$ $(P - N \to P)$

If the sum of two signed numbers with the sign bits off `(0, 0)`
yields a result number with the sign bit off `(0)`,

the overflow flag is turned off

```
signed addition          signed subtraction          unsigned addition

0011   carries
 0011   (+3)              0011   (+3)                  0011   ( 3)
+0011  +(+3)             -1101  -(-3)                 +0011  +( 3)
-----  -----             -----  -----                 -----  -----
00110   (+6)             00110   (+6)                 00110   ( 6)
```

Method 1	`OF = 0`	when $\overline{a_{n-1}} \cdot \overline{b_{n-1}} \cdot \overline{s_{n-1}}$ for addition and $\overline{a_{n-1}} \cdot b_{n-1} \cdot \overline{s_{n-1}}$ for subtraction
Method 2	`OF` $= c_n \bigoplus c_{n-1}$	$c_4 \bigoplus c_3 = 0 \bigoplus 0 = 0$

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

The 4th case - clearing the overflow flag

4. $N + N \rightarrow N$ $(N - P \rightarrow N)$

If the sum of two signed numbers with the sign bits on `(1, 1)` yields a result number with the sign bit on `(1)`,

the overflow flag is turned off

```
signed addition           signed subtraction           unsigned addition

1101   carries
 1101   (-3)               1101    (-3)                  1101    (13)
+1101  +(-3)              -0011   -(+3)                 +1101   +(13)
-----  -----              -----   -----                 -----   -----
11010   (-6)              11010    (-6)                 11010    (26)
```

Method 1	`OF = 0`	when $a_{n-1} \cdot \overline{b_{n-1}} \cdot s_{n-1}$ for addition and $a_{n-1} \cdot b_{n-1} \cdot s_{n-1}$ for subtraction
Method 2	`OF =` $c_n \bigoplus c_{n-1}$	$c_4 \bigoplus c_3 = 1 \bigoplus 1 = 0$

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

The 5th case - clearing the overflow flag

5. $P + N \to P$, $(P - P \to P)$

   If the sum of two signed numbers with the sign bits off and on `(0, 1)` yields a result number with the sign bit off `(0)`,

   the overflow flag is turned off

```
signed addition          signed subtraction          unsigned addition

1100   carries
 0100   (+4)              0100    (+4)                0100    ( 4)
+1101  +(-3)             -0011   -(+3)               +1101   +(13)
-----  -----             -----   -----               -----   -----
10001   (+1)             10001    (+1)               10001    (17)
```

Method 1	`OF = 0`	when $\overline{a_{n-1}} \cdot \underline{b_{n-1}} \cdot \overline{s_{n-1}}$ for addition and $\overline{a_{n-1}} \cdot b_{n-1} \cdot \overline{s_{n-1}}$ for subtraction
Method 2	`OF` $= c_n \bigoplus c_{n-1}$	$c_4 \bigoplus c_3 = 1 \bigoplus 1 = 0$

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

The 6th case - clearing the overflow flag

6. $P + N \rightarrow N$ $(P - P \rightarrow N)$

If the sum of two signed numbers with the sign bits off and on (0, 1) yields a result number with the sign bit on (1),

the overflow flag is turned off

```
signed addition          signed subtraction          unsigned addition

0000   carries
 0011   (+3)              0011    (+3)                0011    ( 3)
+1100  +(-4)             -0100   -(+4)               +1100   +(12)
-----  -----             -----   -----               -----   -----
01111   (-1)             01111    (-1)               01111    (15)
```

Method 1	OF = 0	when $\overline{a_{n-1}} \cdot b_{n-1} \cdot s_{n-1}$ for addition and $\overline{a_{n-1}} \cdot \overline{b_{n-1}} \cdot s_{n-1}$ for subtraction
Method 2	OF = $c_n \bigoplus c_{n-1}$	$c_4 \bigoplus c_3 = 0 \bigoplus 0 = 0$

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

The 7th case - clearing the overflow flag

7. $N + P \to P$, $(N - N \to P)$

   If the sum of two signed numbers with the sign bits on and off `(1, 0)` yields a result number with the sign bit off `(0)`,

   the overflow flag is turned off

```
signed addition          signed subtraction          unsigned addition

1100    carries
 1101    (-3)              0011    (-3)               1101    (13)
+0100    (+4)             -1100   -(-4)              +0100   +( 4)
-----   -----             -----   -----              -----   -----
10001    (+1)             10001    (+1)              10001    (17)
```

Method 1	`OF = 0`	when $a_{n-1} \cdot \overline{b_{n-1}} \cdot \overline{s_{n-1}}$ for addition and $a_{n-1} \cdot b_{n-1} \cdot \overline{s_{n-1}}$ for subtraction
Method 2	`OF` $= c_n \bigoplus c_{n-1}$	$c_4 \bigoplus c_3 = 1 \bigoplus 1 = 0$

`http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt`

The 8th case - clearing the overflow flag

8. $N + P \rightarrow N$, ($N + P \rightarrow N$)

If the sum of two signed numbers with the sign bits on and off (1, 0) yields a result number with the sign bit on (1),

the overflow flag is turned off

```
signed addition          signed subtraction          unsigned addition

0000   carries
 1100   (-4)              0100   (-4)                 1100   (12)
+0011  +(+3)             -1101  -(-3)                +0011  +( 3)
-----  -----             -----  -----                -----  -----
01111   (-1)             01111   (-1)                01111   (15)
```

Method 1	OF = 0	when $a_{n-1} \cdot \overline{b_{n-1}} \cdot s_{n-1}$ and $a_{n-1} \cdot b_{n-1} \cdot s_{n-1}$
Method 2	OF = $c_n \bigoplus c_{n-1}$	$c_4 \bigoplus c_3 = 0 \bigoplus 0 = 0$

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

TOC Method 1 for computing the overflow flag

Adding two numbers with the same sign

- overflow can only happen when adding two numbers of the same sign results in a different sign ($P + P \to N, \quad N + N \to P$)
- n-bit signed binary arithmetic $A + B = S$

$$A = (a_{n-1}, \cdots, a_1, a_0)$$
$$B = (b_{n-1}, \cdots, b_1, b_0)$$
$$S = (s_{n-1}, \cdots, s_1, s_0)$$

- to detect overflow
    - only the sign bits are considered
    - msb (most significant bit) $a_{n-1}, b_{n-1}, s_{n-1}$
    - the other bits are ignored

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

Overflow conditions for additions and subtractions

- with two operands (A and B) and one result (S), three sign bits ($a_{n-1}, b_{n-1}, s_{n-1}$) are considered $\rightarrow 2^3 = 8$ possible combinations
- only two cases result in overflow for an addition
  - 0 0 1 $(p + p \rightarrow n)$
  - 1 1 0 $(n + n \rightarrow p)$
- only two cases are considered as overflow for an subtraction
  - 0 1 1 $(p - n \rightarrow n)$
  - 1 0 0 $(n - p \rightarrow p)$

s
`http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt`

Overflow condition for an addition

- Overflow in an addition $(A + B)$

	a_{n-1}	b_{n-1}	c_{n-1}	
	0	0	0	$p + p \rightarrow p$
OVER	0	0	1	$p + p \rightarrow n$
	0	1	0	$p + n \rightarrow p$
	0	1	1	$p + n \rightarrow n$
	1	0	0	$n + p \rightarrow p$
	1	0	1	$n + p \rightarrow n$
OVER	1	1	0	$n + n \rightarrow p$
	1	1	1	$n + n \rightarrow n$

  - adding two positives should be positive
  - adding two negatives should be negative

`http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt`

Overflow conditions for a subtraction

- Overflow in a subtraction $(A - B)$

	a_{n-1}	b_{n-1}	c_{n-1}	
	0	0	0	$p - p \rightarrow p$
	0	0	1	$p - p \rightarrow n$
	0	1	0	$p - n \rightarrow p$
OVER	0	1	1	$p - n \rightarrow n$
OVER	1	0	0	$n - p \rightarrow p$
	1	0	1	$n - p \rightarrow n$
	1	1	0	$n - n \rightarrow p$
	1	1	1	$n - n \rightarrow n$

- subtracting a *negative* is the same as adding a *positive*
- subtracting a *positive* is the same as adding a *negative*

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

Overflow in signed computations

- ALU might contain a small logic
  that sets the overflow flag to "1"
  if and only if any one of the above four OV conditions is met.

- in signed computations,
  adding two numbers of the same sign
  must produce a result of the same sign,
  otherwise overflow happened.

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

TOC Method 2 for computing the overflow flag

- Carry into and carry out of the sign bit
- Overflow in 2's complement arithmetic
- Overflow flag $= c_n \bigoplus c_{n-1}$
- Examples of 4-bit signed additions
- c_n and c_{n-1} in a n-bit addition
- Overflow flag computation
- Examples of computing overflow flag
- Hexadecimal carry, octal carry, decimal carry
- No carry into the sign bit

Carry into and carry out of the sign bit

- When adding two n-bit binary values, consider
  - the carry *coming into* the most significant bit (msb)
    c_{n-1} : carry into the sign bit
  - the carry *going out of* the most_significant bit (msb)
    c_n : carry out of the sign bit
    this is the carry flag (CF) in the processor

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

Overflow in 2's complement arithmetic

- overflow in 2's complement happens (`OF=1`) when
  - there is a carry *into* the sign bit ($c_{n-1} = 1$) but no carry *out of* the sign bit ($c_n = 0$)
  - there is no carry *into* the sign bit ($c_{n-1} = 0$) but a carry *out of* the sign bit ($c_n = 1$)

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

Overflow flag = $c_n \bigoplus c_{n-1}$

- the overflow flag is the XOR ($c_n \bigoplus c_{n-1}$) of
  - of the carry *coming into* the sign bit (c_{n-1})
  - with the carry *going out of* the sign bit (c_n)
- overflow happens when
  the carry in (c_{n-1}) does not equal
  to the carry out (c_n)

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

Examples of 4-bit signed additions (1)

- 4-bit 2's complement addition examples

```
0000
 0100 (+4) (pos sign 0)
+1000 (-8) (neg sign 1)
=======================
01100 (-4) (neg sign 1)

C4 carry out 0 (1+0+0)
C3 carry in  0 (0+1+0)
0 XOR 0 = NO OVERFLOW
```

```
0100
 0100 (+4) (pos sign 0)
+0100 (+4) (pos sign 0)
========================
01000 (-8) (neg sign 1)

C4 carry out 0 (0+0+1)
C3 carry in  1 (1+1+0)
0 XOR 1 = OVERFLOW!
```

```
1100
 1100  (-4) (neg sign 1)
+0100  (+4) (pos sign 0)
========================
10000  ( 0) (pos sign 0)

C4 carry out 1 (1+0+1)
C3 carry in  1 (1+1+0)
1 XOR 1 = NO OVERFLOW
```

```
1000
 1100 (-4) (neg sign 1)
+1000 (-8) (neg sign 1)
=======================
10100 (+4) (pos sign 0)

C4 carry out 1 (1+1+0)
C3 carry in  0 (1+0+0)
1 XOR 0 = OVERFLOW!
```

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

Examples of 4-bit signed additions (2)

- same sign addition → possible overflow

```
------    ------    ------    ------
+ +, -    - -, +    + +, +    - -, -
------    ------    ------    ------
    +5        -5        +5        -5
    +5        -5        +1        -1
 -----    ------    ------     -----
    -6(OF)    +6(OF)    +6        -6

 0101      1011      0001      1111
  0101      1011      0101      1011
  0101      1011      0001      1111
------    ------    ------    ------
 01010     10110     00110     11010
------    ------    ------    ------
C4 = 0    C4 = 1    C4 = 0    C4 = 1
C3 = 1    C3 = 0    C3 = 0    C3 = 1
------    ------    ------    ------
OF = 1    OF = 1    OF = 0    OF = 0
```

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

Examples of 4-bit signed additions (3)

- mixed sign addition → no overflow

```
------    ------    ------    ------
+ -, +    + -, -    - +, +    - +, -
------    ------    ------    ------
    +5        +5        -5        -5
    -1        -6        +6        +1
 -----    ------    ------     -----
    +4        -1        +1        -4

 1111      0000      1110      0011
  0101      0101      1011      1011
  1111      1010      0110      0001
------    ------    ------    ------
 10100     01111     10001     01100
------    ------    ------    ------
C4 = 1    C4 = 0    C4 = 1    C4 = 0
C3 = 1    C3 = 0    C3 = 1    C3 = 0
------    ------    ------    ------
OF = 0    OF = 0    OF = 0    OF = 0
```

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

c_n and c_{n-1} in a n-bit addition

$(n-1)^{th}$ bit – MSB

- adding operations
  at the $(n-1)$ bit position
- $\{c_n, s_{n-1}\} = a_{n-1} + b_{n-1} + c_{n-1}$

$$\begin{array}{cc} & \text{msb} \\ & a_{n-1} \\ & b_{n-1} \\ & c_{n-1} \\ \hline c_n & s_{n-1} \end{array}$$

- c_n :
  carry coming *out of* the msb

$(n-2)^{th}$ bit

- adding operations
  at the $(n-2)$ bit position
- $\{c_{n-1}, s_{n-2}\} =$
  $a_{n-2} + b_{n-2} + c_{n-2}$

$$\begin{array}{cc} \text{msb} & \\ & a_{n-2} \\ & b_{n-2} \\ & c_{n-2} \\ \hline c_{n-1} & s_{n-2} \end{array}$$

- c_{n-1} :
  carry coming *into* the msb

Overflow flag computation

ADD (addition)	SUB (subtraction)
OF $= c_n \bigoplus c_{n-1}$	OF $= c_n \bigoplus c_{n-1}$
a 2's complement addition $A + B = A + B + 0 \ (c_0 = 0)$	the transformed addition $A - B = A + \overline{B} + 1 \ (c_0 = 1)$
$\{c_n, s_{n-1}\} = a_{n-1} + b_{n-1} + c_{n-1}$	$\{c_n, s_{n-1}\} = a_{n-1} + \overline{b_{n-1}} + c_{n-1}$
$\{c_{n-1}, s_{n-2}\} = a_{n-2} + b_{n-2} + c_{n-2}$	$\{c_{n-1}, s_{n-2}\} = a_{n-2} + \overline{b_{n-2}} + c_{n-2}$

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

Hexadecimal carry, octal carry, decimal carry

- Note that this XOR method
  only works with the binary carry that goes into the sign bit.
- not works with hexadecimal carry
  decimal carry, octal carry
  - the carry doesn't go into the sign bit
  - can't XOR that non-binary carry with the outgoing carry.

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

No carry into the sign bit

- Hexadecimal addition example
  (showing that XOR doesn't work for hex carry):
```
 8Ah
+8Ah
====
114h
```
- The hexadecimal carry of 1 resulting from A+A
  does not affect the sign bit.
- If you do the math in binary, you'll see
  that there is no carry into the sign bit;
  but, there is carry out of the sign bit.
  Therefore, the above example sets OVERFLOW on.
  (The example adds two negative numbers and gets a positive number.)

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

Summary I

```
------------------------------------------------------------------------
 unsigned add/sub     | signed addition      signed subtraction | CF OF
------------------------------------------------------------------------
  1101   (13)         |  1101   (-3)           1101   (-3)      |
 +1110  +(14)  ADD    | +1110  +(-2)  ADD     -0010  -(+2)      |
 -----  -----         | -----  -----          -----  -----      |
 11011   (11) (+16)   | 11011   (-5)          11011   (-5)      | 1  0
                      |                                         |
  0011   ( 3)         |  0011   (+3)           0011   (+3)      |
 -1110  -(14)  SUB    | +0010  +(+2)          -1110  -(-2)  SUB |
 -----  -----         | -----  -----          -----  -----      |
 10101   ( 5) (-16)   | 00101   (+5)          00101   (+5)      | 1  0
                      |                                         |
  0011   ( 3)         |  0011   (+3)           0011   (+3)      |
 +0010  +( 2)  ADD    | +0010  +(+2)  ADD     -1110  -(-2)      |
 -----  -----         | -----  -----          -----  -----      |
 00101   ( 5) (+ 0)   | 00101   (+5)          00101   (+5)      | 0  0
                      |                                         |
  1101   (13)         |  1101   (-3)           1101   (-3)      |
 -0010  -( 2)  SUB    | +1110  +(-2)          -0010  -(+2)  SUB |
 -----  -----         | -----  -----          -----  -----      |
 11011   (11) (-16)   | 11011   (-5)          11011   (-5)      | 0  0
------------------------------------------------------------------------
```

Summary II

```
---------------------------------------------------------------------
 unsigned add/sub     | signed addition      signed subtraction | CF OF
---------------------------------------------------------------------
  1011   (11)         |  1011   (-5)          1011   (-5)       |
 +1100  +(12)  ADD    | +1100  +(-4)  ADD    -0100  -(+4)       |
 -----  -----         | -----  -----         -----  -----       |
 10111   ( 7) (+16)   | 10111   (+7)         10111   (+7)       | 1  1
                      |                                         |
  0101   ( 5)         |  0101   (+5)          0101   (+5)       |
 -1100  -(12)  SUB    | +0100  +(+4)         -1100  -(-4)  SUB  |
 -----  -----         | -----  -----         -----  -----       |
 11001   ( 9) (-16)   | 01001   (-7)         01001   (-7)       | 1  1
                      |                                         |
  0101   ( 5)         |  0101   (+5)          0101   (+5)       |
 +0100  +( 4)  ADD    | +0100  +(+4)  ADD    -1100  -(-4)       |
 -----  -----         | -----  -----         -----  -----       |
 01001   ( 9) (+ 0)   | 01001   (-7)         01001   (-7)       | 0  1
                      |                                         |
  1011   (11)         |  1011   (-5)          1011   (-5)       |
 -0100  -( 4)  SUB    | +1100  +(-4)         -0100  -(+4)  SUB  |
 -----  -----         | -----  -----         -----  -----       |
 00111   ( 7) (  0)   | 10111   (+7)         10111   (+7)       | 0  1
--------------------------------------------------------------------
```

Cases for setting the overflow flag (1) CF=1, OF=1

- singed integer overflow (OF=1 means incorrect S)

```
------------------------------------------------------------------------
* unsigned addition      |  * signed addition          signed subtraction
------------------------------------------------------------------------
                         |    1000
   1011    (11)          |     1011    (-5)               1011    (-5)
  +1100   +(12)   ADD    |    +1100   +(-4)   ADD        -0100   -(+4)
  -----   -----          |    -----   -----              -----   -----
  10111    ( 7) (+16)    |    10111    (+7)              10111    (+7)
                         |
  OF=1                   |    n + n -> p (OF=1)          n - p -> p (OF=1)
                         |
  OF meaningless         |    -> incorrect S             -> incorrect S
   S = 0111              |     S = 0111                   S = 0111
------------------------------------------------------------------------
 * think hand            |  * OF <- C4 XOR C3 = 1 XOR 0 = 1
   addition              |          of signed addition
------------------------------------------------------------------------

* CF=1, S=0111, OF=1 for all three interpretations
```

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

Cases for setting the overflow flag (2) CF=1, OF=1

- singed integer overflow (OF=1 means incorrect S)

```
-----------------------------------------------------------------------
* unsigned subtraction |    signed addition        * signed subtraction
-----------------------------------------------------------------------
                       |    0100
   0101    ( 5)        |     0101    (+5)            0101    (+5)
  -1100   -(12)   SUB  |    +0100   +(+4)           -1100   -(-4)   SUB
  -----   -----        |    -----   -----           -----   -----
  11001    ( 9)  (-16) |    01001    (-7)           01001    (-7)
                       |
  OF=1                 |    p + p -> n (OF=1)       p - n -> n (OF=1)
                       |
  OF meaningless       |    -> incorrect S          -> incorrect S
   S = 1001            |     S = 1001                S = 1001
-----------------------------------------------------------------------
* think hand           |  * OF <- C4 XOR C3 = 0 XOR 1 = 1
  subtraction          |          of signed addition
-----------------------------------------------------------------------

* CF=1, S=1001, OF=1 for all three interpretations
```

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

Cases for setting the overflow flag (3) CF=0, OF=1

- singed integer overflow (OF=1 means incorrect S)

```
-----------------------------------------------------------------------
* unsigned addition     |  * signed addition          signed subtraction
-----------------------------------------------------------------------
                        |  0100
  0101    ( 5)          |   0101    (+5)                0101    (+5)
 +0100   +( 4)   ADD    |  +0100   +(+4)   ADD         -1100   -(-4)
 -----   -----          |  -----   -----               -----   -----
 01001    ( 9)  (+ 0)   |  01001    (-7)               01001    (-7)
                        |
 OF=1                   |  p + p -> n (OF=1)           p - n -> n (OF=1)
                        |
 OF meaningless         |  -> incorrect S              -> incorrect S
  S = 1001              |   S = 1001                    S = 1001
-----------------------------------------------------------------------
 * think hand           |  * OF <- C4 XOR C3 = 0 XOR 1 = 1
   addition             |        of signed addition
-----------------------------------------------------------------------

* CF=0, S=1001, OF=1 for all three interpretations
```

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

Cases for setting the overflow flag (4) CF=0, OF=1

- singed integer overflow (OF=1 means incorrect S)

```
------------------------------------------------------------------------
* unsigned subtraction |    signed addition        * signed subtraction
------------------------------------------------------------------------
                       |    1000
   1011    (11)        |     1011    (-5)               1011    (-5)
  -0100   -( 4)   SUB  |    +1100   +(-4)              -0100   -(+4)   SUB
  -----   -----        |    -----   -----              -----   -----
  00111    ( 7) (   0) |    10111    (+7)              10111    (+7)
                       |
  OF=1                 |    n + n -> p (OF=1)          n - p -> p (OF=1)
                       |
  OF meaningless       |    -> incorrect S             -> incorrect S
   S = 0111            |     S = 0111                   S = 0111
------------------------------------------------------------------------
* think hand           |  * OF <- C4 XOR C3 = 1 XOR 0 = 1
  subtraction          |          of signed addition
------------------------------------------------------------------------

* CF=0, S=0111, OF=1 for all three interpretations
```

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

Cases for clearing the overflow flag (1) CF=1, OF=0

- no signed integer overflow (CF=0 means correct S)

```
-----------------------------------------------------------------------
* unsigned addition     |  * signed addition          signed subtraction
-----------------------------------------------------------------------
                        |   1100
   1101   (13)          |    1101   (-3)               1101   (-3)
  +1110  +(14)   ADD    |   +1110  +(-2)   ADD        -0010  -(+2)
  -----  -----          |   -----  -----              -----  -----
  11011   (11)  (+16)   |   11011   (-5)              11011   (-5)
                        |
  OF=0                  |   n + n -> n (OF=0)         n - p -> n (OF=0)
                        |
  OF meaningless        |   -> correct S              -> correct S
   S = 0000             |     S = 0000                  S = 0000
-----------------------------------------------------------------------
* think hand            |  * OF <- C4 XOR C3 = 1 XOR 1 = 0
  addition              |          of signed addition
-----------------------------------------------------------------------

* CF=1, S=1011, OF=0 for all three interpretations
```

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

Cases for clearing the overflow flag (2) CF=1, OF=0

- no signed integer overflow (CF=0 means correct S)

```
-----------------------------------------------------------------------
* unsigned subtraction |    signed addition        * signed subtraction
-----------------------------------------------------------------------
                       |    0010
   0011    ( 3)        |     0011    (+3)              0011    (+3)
  -1110   -(14)   SUB  |    +0010   +(+2)             -1110   -(-2)   SUB
  -----   -----        |    -----   -----             -----   -----
  10101    ( 5)  (-16) |    00101    (+5)             00101    (+5)
                       |
  CF=1                 |    p + p -> p (OF=0)         p - n -> p (OF=0)
                       |
  OF meaningless       |    -> correct S              -> correct S
   S = 0101            |      S = 0101                  S = 0101
-----------------------------------------------------------------------
* think hand           |  * OF <- C4 XOR C3 = 0 XOR 0 = 0
  subtraction          |          of signed addition
-----------------------------------------------------------------------

* CF=1, S=0101, OF=0 for all three interpretations
```

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

Cases for clearing the overflow flag (3) CF=0, OF=0

- no signed integer overflow (CF=0 means correct S)

```
------------------------------------------------------------------------
* unsigned addition      |  * signed addition          signed subtraction
------------------------------------------------------------------------
                         |    0010
   0011    ( 3)          |     0011    (+3)              0011    (+3)
  +0010   +( 2)   ADD    |    +0010   +(+2)   ADD       -1110   -(-2)
  -----   -----          |    -----   -----             -----   -----
  00101    ( 5)  (+ 0)   |    00101    (+5)             00101    (+5)
                         |
  OF=0                   |    p + p -> p (OF=0)         p - n -> p (OF=0)
                         |
  OF meaningless         |    -> correct S              -> correct S
   S = 0101              |     S = 0101                  S = 0101
------------------------------------------------------------------------
* think hand             |  * OF <- C4 XOR C3 = 0 XOR 0 = 0
  addition               |          of signed addition
------------------------------------------------------------------------

* CF=0, S=0101, OF=0 for all three interpretations
```

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt

Cases for clearing the overflow flag (4) CF=0, OF=0

- no signed integer overflow (CF=0 means correct S)

```
------------------------------------------------------------------------
* unsigned addition      |  * signed addition          signed subtraction
------------------------------------------------------------------------
                         |    1100
   1101    (13)          |     1101    (-3)             1101    (-3)
  -0010   -( 2)   SUB    |    +1110   +(-2)            -0010   -(+2)   SUB
  -----   -----          |    -----   -----            -----   -----
  11011    (11)  (-16)   |    11011    (-5)            11011    (-5)
                         |
  OF=0                   |    n + n -> n (OF=0)        n - p -> n (OF=0)
                         |
  OF meaningless         |    -> correct S             -> correct S
   S = 1011              |     S = 1011                 S = 1011
------------------------------------------------------------------------
* think hand             |  * OF <- C4 XOR C3 = 1 XOR 1 = 0
  subtraction            |          of signed addition
------------------------------------------------------------------------

* CF=0, S=1011, OF=0 for all three interpretations
```

http://teaching.idallen.com/dat2343/10f/notes/040_overflow.txt