Integer Overflow

Lecture 8
Section 2.5

Robb T. Koether

Hampden-Sydney College

Mon, Jan 27, 2014

Outline

Addition of Signed Integers

- To add signed integers,
  - Express any negative values in two's complement form.
  - Add them, using the ordinary rules of addition.
  - To catch overflow, check two bits:
    - The carry-in bit of the last column.
    - The carry-out bit of the last column.

Subtraction of Signed Integers

- To subtract signed integers,
  - Express any negative values in two's complement form.
  - Replace the subtrahend with its two's complement.
  - *Add* them, using the ordinary rules of addition.
  - To catch overflow, check two bits:
    - The carry-in bit of the last column.
    - The carry-out bit of the last column.

Outline

Outline

Signed Overflow

- If the correct result is too large to fit in the allotted space, the condition is called overflow.
- Integer overflow under addition occurs when
  - The sum of two positive integers is too large.
  - The sum of two negative integers is too large.

Detecting Signed Overflow

a	b	$a+b$	c-in	c-out	Valid?
P	P	P	0	0	Yes
P	P	N	1	0	No
P	N	P	1	1	Yes
P	N	N	0	0	Yes
N	P	P	1	1	Yes
N	P	N	0	0	Yes
N	N	P	0	1	No
N	N	N	1	1	Yes

- What characterizes overflow?

Detecting Signed Overflow

a	b	$a+b$	c-in	c-out	Valid?
P	P	P	0	0	Yes
P	P	N	1	0	No
P	N	P	1	1	Yes
P	N	N	0	0	Yes
N	P	P	1	1	Yes
N	P	N	0	0	Yes
N	N	P	0	1	No
N	N	N	1	1	Yes

- What characterizes overflow?

Detecting Signed Overflow

- Integer overflow under addition of signed integers is detected when *the carry-in bit does not match the carry-out bit in the high-order position*.

Detecting Signed Overflow

- Using 8-bit signed integers, when we add $80 + 90$ we get -86.

$$\begin{aligned} 127 + 1 &= \texttt{01111111} + \texttt{00000001} \\ &= \texttt{10000000} \\ &= -128. \\ (127 + 1) - 256 &= -128. \end{aligned}$$

Detecting Signed Overflow

- Using 8-bit signed integers, when we add $(-80) + (-90)$ we get 86.

$$\begin{aligned}(-128) + (-1) &= \texttt{10000000} + \texttt{11111111} \\ &= \texttt{01111111} \\ &= 127. \\ ((-128) + (-1)) + 256 &= 127.\end{aligned}$$

The x86 Processor

- The x86 processor has a 32-bit EFLAGS register.
- Among the 32 bits are four flags:

  Carry Flag (CF) – Set if an arithmetic operation generates a carry or a borrow out of the most-significant bit of the result; cleared otherwise. This flag indicates an overflow condition for unsigned-integer arithmetic.

  Sign Flag (SF) – Set equal to the most-significant bit of the result, which is the sign bit of a signed integer (0 indicates a positive value and 1 indicates a negative value).

  Overflag Flag (OF) – Set if the integer result is too large a positive number or too small a negative number (excluding the sign bit) to fit in the destination operand; cleared otherwise. This flag indicates an overflow condition for signed-integer (two's complement) arithmetic.

  Zero Flag (ZF) – Set if the result is zero; cleared otherwise.

Signed Addition

a		b		$a+b$		CF	SF	OF	ZF
01000000	64	00100000	32	01100000	96	0	0	0	0
01000000	64	01100000	96	10100000	-96	0	1	1	0
01000000	64	11100000	-32	00100000	32	1	0	0	0
00100000	32	11000000	-64	11100000	-32	0	1	0	0
11100000	-32	01000000	64	00100000	32	1	0	0	0
11000000	-64	00100000	32	11100000	-32	0	1	0	0
11000000	-64	10100000	-96	01100000	96	1	0	1	0
11000000	-64	11100000	-32	10100000	-96	1	1	0	0

Signed Addition

a		b		$a+b$		CF	SF	OF	ZF
01000000	64	00100000	32	01100000	96	0	0	0	0
01000000	64	01100000	96	10100000	-96	0	1	1	0
01000000	64	11100000	-32	00100000	32	1	0	0	0
00100000	32	11000000	-64	11100000	-32	0	1	0	0
11100000	-32	01000000	64	00100000	32	1	0	0	0
11000000	-64	00100000	32	11100000	-32	0	1	0	0
11000000	-64	10100000	-96	01100000	96	1	0	1	0
11000000	-64	11100000	-32	10100000	-96	1	1	0	0

The Relational and Equality Operators

Mnemonic	Condition Tested For	Status Flags Setting
O	Overflow	OF = 1
NO	No Overflow	OF = 0
B	Below	CF = 1
NB	Not Below	CF = 0
E	Equal	ZF = 1
NE	Not Equal	ZF = 0
S	Sign	SF = 1
NS	No Sign	SF = 0
BE	Below or Equal	(CF OR ZF) = 1
NBE	Neither Below nor Equal	(CF OR ZF) = 0
L	Less	(SF XOR OF) = 1
NL	Not Less	(SF XOR OF) = 0
LE	Less or Equal	(SF XOR OF) OR ZF = 1
NLE	Neither Less nor Equal	(SF XOR OF) OR ZF = 0

Overflow/No Overflow

a		b		$a+b$		CF	SF	OF	ZF
01000000	64	00100000	32	01100000	96	0	0	0	0
01000000	64	01100000	96	10100000	-96	0	1	1	0
01000000	64	11100000	-32	00100000	32	1	0	0	0
00100000	32	11000000	-64	11100000	-32	0	1	0	0
11100000	-32	01000000	64	00100000	32	1	0	0	0
11000000	-64	00100000	32	11100000	-32	0	1	0	0
11000000	-64	10100000	-96	01100000	96	1	0	1	0
11000000	-64	11100000	-32	10100000	-96	1	1	0	0

- Overflow/No Overflow (OF = 1/0)

Sign/No Sign

a		b		$a+b$		CF	SF	OF	ZF
01000000	64	00100000	32	01100000	96	0	0	0	0
01000000	64	01100000	96	10100000	-96	0	1	1	0
01000000	64	11100000	-32	00100000	32	1	0	0	0
00100000	32	11000000	-64	11100000	-32	0	1	0	0
11100000	-32	01000000	64	00100000	32	1	0	0	0
11000000	-64	00100000	32	11100000	-32	0	1	0	0
11000000	-64	10100000	-96	01100000	96	1	0	1	0
11000000	-64	11100000	-32	10100000	-96	1	1	0	0

- Sign/No Sign (SF = 1/0)

Less/Not Less

a		b		$a+b$		CF	SF	OF	ZF
01000000	64	00100000	32	01100000	96	0	0	0	0
01000000	64	01100000	96	10100000	-96	0	1	1	0
01000000	64	11100000	-32	00100000	32	1	0	0	0
00100000	32	11000000	-64	11100000	-32	0	1	0	0
11100000	-32	01000000	64	00100000	32	1	0	0	0
11000000	-64	00100000	32	11100000	-32	0	1	0	0
11000000	-64	10100000	-96	01100000	96	1	0	1	0
11000000	-64	11100000	-32	10100000	-96	1	1	0	0

- Less/Not Less ((SF XOR OF) = 1/0)

title

- Explain why the combinations

$$\text{CF} = 1, \text{SF} = 1, \text{OF} = 1$$
$$\text{CF} = 0, \text{SF} = 0, \text{OF} = 1$$

  never occur.

Outline

Signed Overflow of Subtraction

- Overflow occurs under signed subtraction when *the carry-in bit does not match the carry-out bit in the high-order position*.

Signed Subtraction

a		b		$a - b$		CF	SF	OF	ZF
01000000	64	00100000	32	00100000	32	1	0	0	0
00100000	32	01000000	64	11100000	-32	0	1	0	0
01000000	64	11100000	-32	01100000	96	0	0	0	0
01000000	64	10100000	-96	10100000	-96	0	1	1	0
11000000	-64	01100000	96	01100000	96	1	0	1	0
11000000	-64	00100000	32	10100000	-96	1	1	0	0
11100000	-32	11000000	-64	00100000	32	1	0	0	0
11000000	-64	11100000	-32	11100000	-32	0	1	0	0

Overflow/No Overflow

a		b		$a+b$		CF	SF	OF	ZF
01000000	64	00100000	32	00100000	32	1	0	0	0
00100000	32	01000000	64	11100000	-32	0	1	0	0
01000000	64	11100000	-32	01100000	96	0	0	0	0
01000000	64	10100000	-96	10100000	-96	0	1	1	0
11000000	-64	01100000	96	01100000	96	1	0	1	0
11000000	-64	00100000	32	10100000	-96	1	1	0	0
11100000	-32	11000000	-64	00100000	32	1	0	0	0
11000000	-64	11100000	-32	11100000	-32	0	1	0	0

- Overflow/No Overflow (OF = 1/0)

Less/Not Less

a		b		$a+b$		CF	SF	OF	ZF
01000000	64	00100000	32	00100000	32	1	0	0	0
00100000	32	01000000	64	11100000	-32	0	1	0	0
01000000	64	11100000	-32	01100000	96	0	0	0	0
01000000	64	10100000	-96	10100000	-96	0	1	1	0
11000000	-64	01100000	96	01100000	96	1	0	1	0
11000000	-64	00100000	32	10100000	-96	1	1	0	0
11100000	-32	11000000	-64	00100000	32	1	0	0	0
11000000	-64	11100000	-32	11100000	-32	0	1	0	0

- Less/Not Less ((SF XOR OF) = 1/0)

Outline

Unsigned Addition

- Overflow occurs under unsigned subtraction when *the carry-out bit in the high-order position is 1*.
- Overflow occurs under unsigned subtraction when *the carry-out bit in the high-order position is 0*.

Unigned Addition

a		b		$a+b$		CF	SF	OF	ZF
01000000	64	00100000	32	01100000	96	1	0	0	0
01000000	64	01100000	96	10100000	160	0	1	0	0
01000000	64	10100000	160	11100000	224	0	0	0	0
01100000	96	11100000	224	01000000	64	0	1	1	0
10100000	160	11000000	192	01100000	96	1	0	1	0
11100000	224	11100000	224	11000000	192	1	1	0	0

Outline

Collected

Collected

- Sec. 2.3: 11, 23, 40.
- Sec. 2.4: 15, 19.
- Sec. 2.5: 2, 18.

Assignment

Assignment

- Read Section 2.5.
- Perform signed addition and check for overflow.

```
  10010011    01110010
+11011101   +01110101
```

- Perform signed subtraction and check for overflow.

```
  10010011    01110010
−11011101   −01110101
```

- Repeat the above sets using unsigned addition and subtraction.