CS430 Computer Architecture

Spring 2015

Arithmetic Logic Unit

- The ALU performs arithmetic and logical operations on data
- All other elements … control unit, registers, memory, I/O mainly bring data to ALU for processing and then take the results back

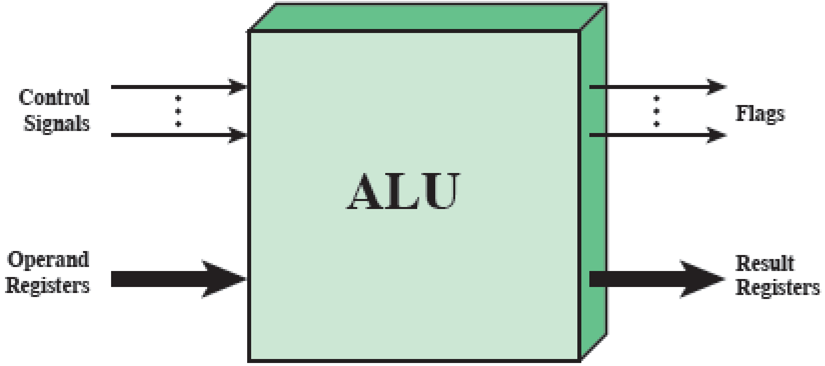


Figure 10.1 ALU Inputs and Outputs

Chapter 10
Integer Arithmetic

- In general, we know the following is true:

```
0 + 0 = 0 0
0 + 1 = 0 1
1 + 0 = 0 1
1 + 1 = 1 0
```

Integer Arithmetic

- Perform the following addition and interpret the result in:
  a. modulo 2^n
  b. two's complement notation.

```
  10101010
+ 10101010
  --------
```

Carry-in & Carry-out

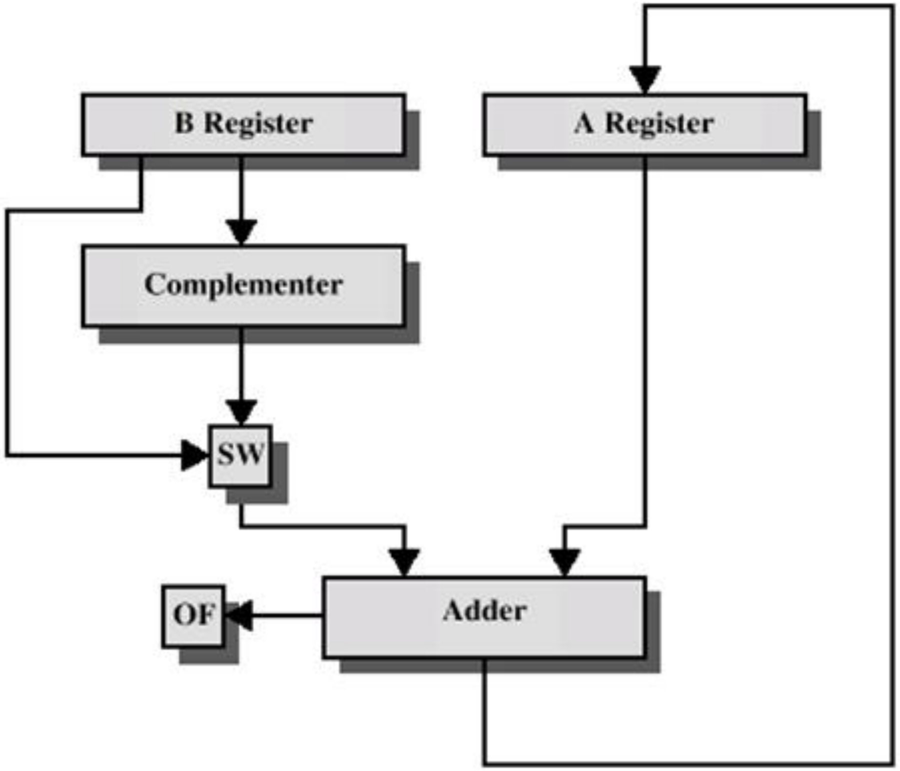
- Consider the following:

```
 00001111
+01010101
 --------
```

1. What is the carry-in and carry-out of bit 3 (bit numbering starts at bit 0)
2. The carry-out of the MSb during an addition is the value of the external carry in the flags register for an addition
3. What is the external carry for the above example?

Subtraction

- Subtraction is performed by taking the two's complement of the subtrahend and adding this value to the minuend.



OF = overflow bit
SW = Switch (select addition or subtraction)

Problem

- Perform the following subtraction:

```
 00110011     (Minuend)
-00001111     (Subtahend)
 --------
```

1. Before performing the subtraction, identify the two numbers being subtracted. Assume the numbers are represented in modulo 2^n.
2. Perform the subtraction.
3. Interpret the result. Is it what you would expect it to be?

Arithmetic Overflow

- Remember that the range of values that can be represented using 8-bits for:
  - modulo 2^n numbers is 0 to 255
  - two's complement is -128 to 127.
- The microprocessor will perform the addition or subtraction of two numbers, but the question is how do we know if the result is correct?
- The answer lies with two flags: (a) the external carry flag and (b) the overflow flag.
- First we will define overflow as a condition such that an arithmetic operation produces a result outside the range of the number system being used.

Arithmetic Overflow

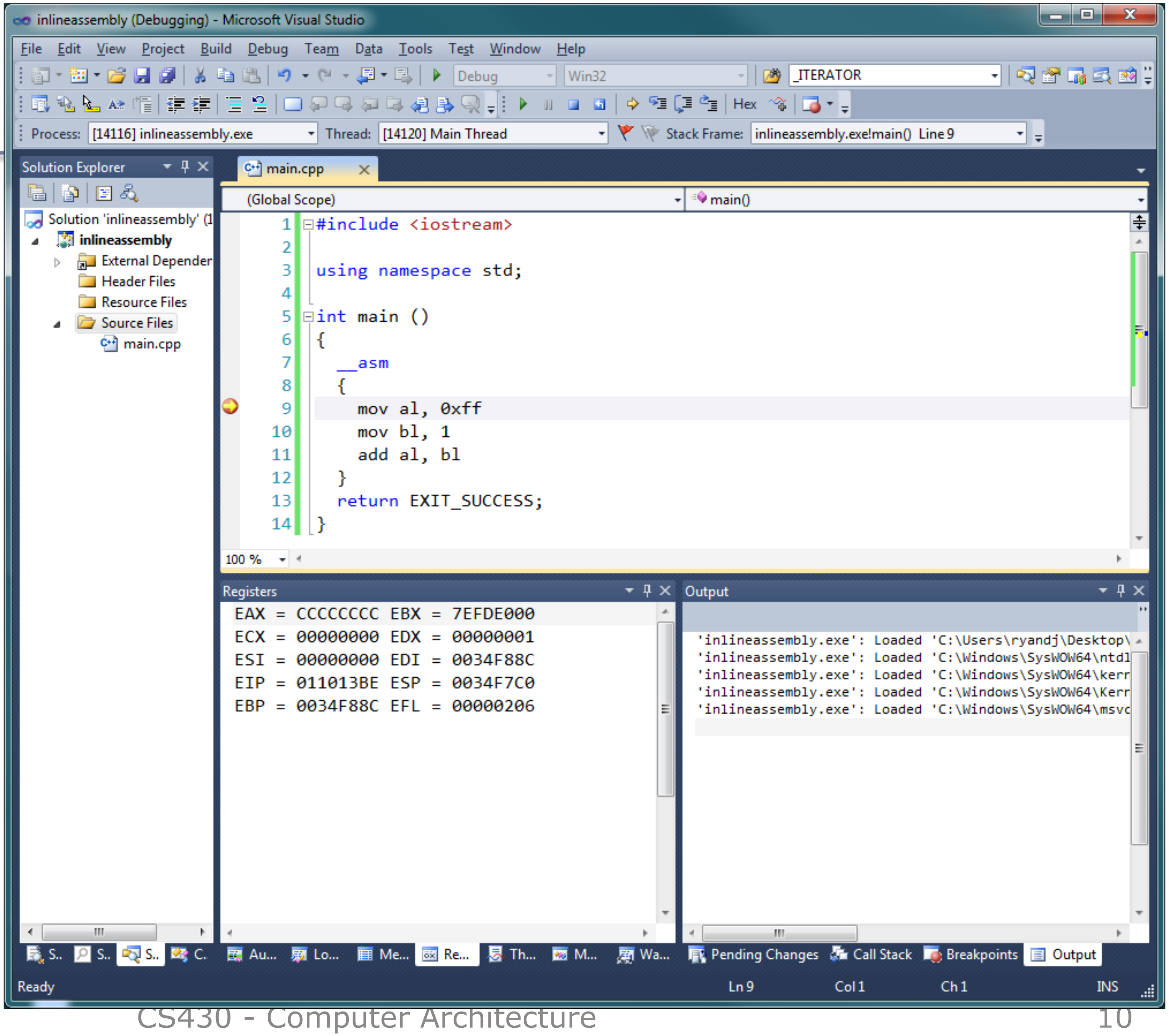
- Perform the operations below and interpret the result in:
  1. modulo 2^n
  2. two's complement notation.

```
 11111111      01111111
+00000001     +00000001
 --------      --------
```

- Were there any examples of overflow? Identify each case and briefly explain why.

Inline Assembly Using Studio 2010

- Grab inlineassembly from CS430Public



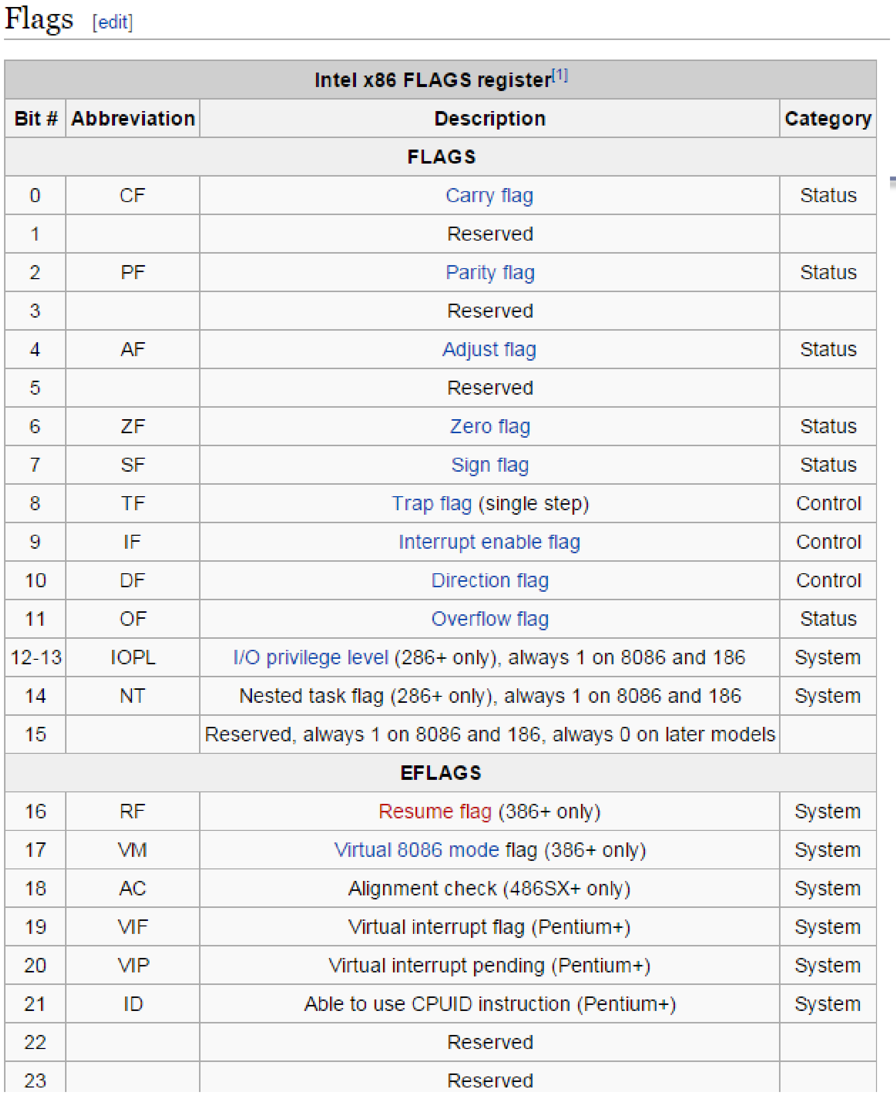
Debug
Win32
_ITERATOR
Process: [14116] inlineassembly.exe
Thread: [14120] Main Thread
Stack Frame: inlineassembly.exe!main() Line 9

```
#include <iostream>

using namespace std;

int main ()
{
  __asm
  {
    mov al, 0xff
    mov bl, 1
    add al, bl
  }
  return EXIT_SUCCESS;
}
```

```
EAX = CCCCCCCC EBX = 7EFDE000
ECX = 00000000 EDX = 00000001
ESI = 00000000 EDI = 0034F88C
EIP = 011013BE ESP = 0034F7C0
EBP = 0034F88C EFL = 00000206
```

```
'inlineassembly.exe': Loaded 'C:\Users\ryandj\Desktop\
'inlineassembly.exe': Loaded 'C:\Windows\SysWOW64\ntdl
'inlineassembly.exe': Loaded 'C:\Windows\SysWOW64\kern
'inlineassembly.exe': Loaded 'C:\Windows\SysWOW64\Kern
'inlineassembly.exe': Loaded 'C:\Windows\SysWOW64\msvc
```

Partial EFL Register



Flags [edit]

Intel x86 FLAGS register[1]			
Bit #	**Abbreviation**	**Description**	**Category**
		FLAGS	
0	CF	Carry flag	Status
1		Reserved	
2	PF	Parity flag	Status
3		Reserved	
4	AF	Adjust flag	Status
5		Reserved	
6	ZF	Zero flag	Status
7	SF	Sign flag	Status
8	TF	Trap flag (single step)	Control
9	IF	Interrupt enable flag	Control
10	DF	Direction flag	Control
11	OF	Overflow flag	Status
12-13	IOPL	I/O privilege level (286+ only), always 1 on 8086 and 186	System
14	NT	Nested task flag (286+ only), always 1 on 8086 and 186	System
15		Reserved, always 1 on 8086 and 186, always 0 on later models	
		EFLAGS	
16	RF	Resume flag (386+ only)	System
17	VM	Virtual 8086 mode flag (386+ only)	System
18	AC	Alignment check (486SX+ only)	System
19	VIF	Virtual interrupt flag (Pentium+)	System
20	VIP	Virtual interrupt pending (Pentium+)	System
21	ID	Able to use CPUID instruction (Pentium+)	System
22		Reserved	
23		Reserved	

Practice

- Let's perform the following operations and determine where any overflows occurred for both unsigned and 2’s complement representations.

```
 1001                1100
+0101               +0100
-----               -----

 0011                1100
-0100               -1111
-----               -----

 1000                1000
+0001               -0001
-----               -----
```