Taslim Dosunmu

TA: Boyu Hou

Computer Organization and Programming

31 Aug 2017

Lab 2

**Exercise 1:**

The purpose of learning the x86 architecture before diving in and programming with it is to understand how and why the machine works first. By understanding the underlying machinery, this gives us insight into why we will have to do things certain ways. Given the general experience level of the students in this class, where our knowledge of programming likely is experience with a high level language such as Java, many of the students do not know about the nuances of the hardware level development and might run into “weird” errors without understanding what’s going on.

DH Register DL Register

[\_ \_ 8 bits\_ \_ \_] [\_ \_ 8 bits \_ \_]

DX Register: [\_ \_ \_ \_ \_ \_ \_ 16 bits \_ \_ \_ \_ \_]

EDX Register: [\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ 32 bits \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_]

**Exercise 2:**

Byte: 4 3 2 1

[ 12 | 78 | 45 |69 ]

**Exercise 3:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Contents of EAX (before) | Instruction | Contents of EAX (after) | CF | OF | SF | ZF |
| 00000040 | add EAX, 40 | 00000068 | 0 | 0 | 0 | 0 |
| FFFFFF40 | add EAX, 40 | FFFFFF68 | 0 | 0 | 0 | 0 |
| 00000040 | add EAX, -40 | FFFFFFF8 | 0 | 0 | 1 | 0 |
| FFFFFF40 | add EAX, -40 | FFFFFEF8 | 1 | 1 | 1 | 0 |
| FFFFFFFF | add EAX, 1 | 00000000 | 1 | 0 | 0 | 1 |

**Exercise 4:**

1. CF is set when the result of an unsigned arithmetic operation is too large for the destination
2. CF and OF is set for signed arithmetic operations results too large for their destinations
3. SF is set when a negative result is generated. CF and OF can also be set.

**Exercise 5:**

|  |  |
| --- | --- |
| Addresses | Contents |
| 1996 | 23 |
| 1997 | DD |
| 1998 | DA |
| 1999 | AB |

**Exercise 6:**

Bytes used for the following data types:

1. Word: 2 bytes
2. Double Word: 4 bytes
3. Quad Word: 8 bytes

**Exercise 7:**

1. Signed— **From:** 0 **To:** 63
2. Unsigned— **From:** 0 **To:** 31 (+)

**From:** -1 **To:** -32 (-)