First Instructions

X86 ARCHITECTURE

Assembly language programming
By xorpd

xorpd.net

Objectives

- You will learn about some basic x86 instructions.
  - Basic data manipulation.
    - MOV
  - Simple Arithmetic.
    - ADD
    - SUB

Basic Instructions structure

- x86 Instructions have numeric representation (Opcode) and textual representation.
- x86 instructions have the following structure:
  - Mnemonic, or shortcut, for the instruction's name.
  - Arguments. (Needed for the operation).
- Written like this:
  - Mnemonic arg1,arg2,arg3,…
  - Usually no more than 2 arguments. (Sometimes even no arguments at all).
- The arguments are somehow encoded into the numeric representation.

Encoding instructions

- There is a computer program that translates the textual representation of an instruction into the numeric representation of the instruction.
  - This program is called **Assembler**.
- While the numeric representation is unique and agreed upon, there are different textual flavors (Syntaxes) to represent the instructions.
- We are going to use the syntax of the **fasm flat assembler**. We will learn more about it later in detail.

MOV

- The MOV instruction allows to “move” data.
  - MOV destination, source
  - Data is copied from source to destination.
- Examples:
  - `mov eax,8CBh`
    - Will store the number 0x8CB inside the 32-bit register eax.
  - `mov ecx,edx`
    - Will copy the number inside edx to ecx. (32 bit copy).
  - `mov si,cx`
    - Will copy the number inside cx to si. (16 bit copy).
  - Invalid example: `mov 13h,ecx`
    - It is not possible to assign ecx into 13h.
  - Invalid Example: `mov ecx,dh`
    - ecx is of size 32 bits, but dh is of size 8 bits. Sizes don’t match.

MOV - Example

- We make a table of the effects of various MOV instructions on eax, ecx and edx.

Instruction	eax	ecx	edx
	????????	????????	????????
mov eax, 3h			
mov edx, ABh			
mov edx, edx			
mov ecx, edx			
mov edx, eax			

MOV - Example

- We make a table of the effects of various MOV instructions on eax, ecx and edx.

Instruction	eax	ecx	edx
	????????	????????	????????
mov eax, 3h	00000003	????????	????????
mov edx, ABh			
mov edx, edx			
mov ecx, edx			
mov edx, eax			

MOV - Example

- We make a table of the effects of various MOV instructions on eax, ecx and edx.

Instruction	eax	ecx	edx
	????????	????????	????????
mov eax, 3h	00000003	????????	????????
mov edx, ABh	00000003	????????	000000AB
mov edx, edx			
mov ecx, edx			
mov edx, eax			

MOV - Example

- We make a table of the effects of various MOV instructions on eax, ecx and edx.

Instruction	eax	ecx	edx
	????????	????????	????????
mov eax, 3h	00000003	????????	????????
mov edx, ABh	00000003	????????	000000AB
mov edx, edx	00000003	????????	000000AB
mov ecx, edx			
mov edx, eax			

MOV - Example

- We make a table of the effects of various MOV instructions on eax, ecx and edx.

Instruction	eax	ecx	edx
	????????	????????	????????
mov eax, 3h	00000003	????????	????????
mov edx, ABh	00000003	????????	000000AB
mov edx, edx	00000003	????????	000000AB
mov ecx, edx	00000003	000000AB	000000AB
mov edx, eax			

MOV - Example

- We make a table of the effects of various MOV instructions on eax, ecx and edx.

Instruction	eax	ecx	edx
	????????	????????	????????
mov eax, 3h	00000003	????????	????????
mov edx, ABh	00000003	????????	000000AB
mov edx, edx	00000003	????????	000000AB
mov ecx, edx	00000003	000000AB	000000AB
mov edx, eax	00000003	000000AB	00000003

MOV – Example (Cont.)

- We make a table of the effects of various MOV instructions on eax, ecx and their partial counterparts.

Instruction	eax	ecx
	????????	????????
mov ax,9Ch		
mov eax,DDDD1234h		
mov cl,E5h		
mov ah,cl		

MOV – Example (Cont.)

- We make a table of the effects of various MOV instructions on eax, ecx and their partial counterparts.

Instruction	eax	ecx
	????????	????????
mov ax,9Ch	????009C	????????
mov eax,DDDD1234h		
mov cl,E5h		
mov ah,cl		

MOV – Example (Cont.)

- We make a table of the effects of various MOV instructions on eax, ecx and their partial counterparts.

Instruction	eax	ecx
	????????	????????
mov ax,9Ch	????009C	????????
mov eax,DDDD1234h	DDDD1234	????????
mov cl,E5h		
mov ah,cl		

MOV – Example (Cont.)

- We make a table of the effects of various MOV instructions on eax, ecx and their partial counterparts.

Instruction	eax	ecx
	????????	????????
mov ax,9Ch	????009C	????????
mov eax,DDDD1234h	DDDD1234	????????
mov cl,E5h	DDDD1234	??????E5
mov ah,cl		

MOV – Example (Cont.)

- We make a table of the effects of various MOV instructions on eax, ecx and their partial counterparts.

Instruction	eax	ecx
	????????	????????
mov ax,9Ch	????009C	????????
mov eax,DDDD1234h	DDDD1234	????????
mov cl,E5h	DDDD1234	??????E5
mov ah,cl	DDDDE534	??????E5

MOV – Example (Cont.)

- We make a table of the effects of various MOV instructions on eax, ecx and their partial counterparts.

Instruction	eax	ecx
	????????	????????
mov ax,9Ch	????009C	????????
mov eax,DDDD1234h	DDDD1234	????????
mov cl,E5h	DDDD1234	??????E5
mov ah,cl	DDDDE534	??????E5

ADD

- The ADD instruction allows to add numbers.
  - ADD destination, source
  - $destination \leftarrow destination + source$
  - The result wraps around if larger than the size of the arguments.
- Examples:
  - `add eax,edx`
    - Adds the contents of eax and edx. Stores the result in eax. ($eax \leftarrow eax + edx$).
  - `add esi,11b`
    - Adds the number $11b = 3_{10}$ to esi. ($esi \leftarrow esi + 3$).
  - `add dx,si`
    - Adds the contents of si to dx, and stores the result in dx. ($dx \leftarrow dx + si$). Note that this is a 16 bit addition.
  - Invalid example: `add 532h,ecx`
    - 532h can not be the destination of the addition operation. (Where will the result be stored?)
  - Invalid example: `add bx,eax`
    - bx is of size 16 bit, but eax is of size 32 bit. Sizes don't match.

ADD - Example

Instruction	esi	eax	ebx
	00000001	00000002	00000003
add eax,ebx			
add eax,eax			
mov esi,0FFFFFFFFh			
add ebx,esi			
add esi,eax			

ADD - Example

Instruction	esi	eax	ebx
	00000001	00000002	00000003
add eax,ebx	00000001	00000005	00000003
add eax,eax			
mov esi,0FFFFFFFFh			
add ebx,esi			
add esi,eax			

ADD - Example

Instruction	esi	eax	ebx
	00000001	00000002	00000003
add eax,ebx	00000001	00000005	00000003
add eax,eax	00000001	0000000A	00000003
mov esi,0FFFFFFFFh			
add ebx,esi			
add esi,eax			

ADD - Example

Instruction	esi	eax	ebx
	00000001	00000002	00000003
add eax,ebx	00000001	00000005	00000003
add eax,eax	00000001	0000000A	00000003
mov esi,0FFFFFFFFh	FFFFFFFF	0000000A	00000003
add ebx,esi			
add esi,eax			

ADD - Example

Instruction	esi	eax	ebx
	00000001	00000002	00000003
add eax,ebx	00000001	00000005	00000003
add eax,eax	00000001	0000000A	00000003
mov esi,0FFFFFFFFh	FFFFFFFF	0000000A	00000003
add ebx,esi	FFFFFFFF	0000000A	00000002
add esi,eax			

ADD - Example

Instruction	esi	eax	ebx
	00000001	00000002	00000003
add eax,ebx	00000001	00000005	00000003
add eax,eax	00000001	0000000A	00000003
mov esi,0FFFFFFFFh	FFFFFFFF	0000000A	00000003
add ebx,esi	FFFFFFFF	0000000A	00000002
add esi,eax	00000009	0000000A	00000002

ADD – Example (Cont.)

- Addition of partial registers:

Instruction	edi	ecx	eax
	AB29FFFF	00000703	000000FF
add al,ch			
add di,cx			
mov edi,0AB29FFFFh			
add edi,ecx			

ADD – Example (Cont.)

- Addition of partial registers:

Instruction	edi	ecx	eax
	AB29FFFF	00000703	000000FF
add al,ch	AB29FFFF	00000703	00000006
add di,cx			
mov edi,0AB29FFFFh			
add edi,ecx			

ADD – Example (Cont.)

- Addition of partial registers:

Instruction	edi	ecx	eax
	AB29FFFF	00000703	000000FF
add al,ch	AB29FFFF	00000703	00000006
add di,cx	AB290702	00000703	00000006
mov edi,0AB29FFFFh			
add edi,ecx			

ADD – Example (Cont.)

- Addition of partial registers:

Instruction	edi	ecx	eax
	AB29FFFF	00000703	000000FF
add al,ch	AB29FFFF	00000703	00000006
add di,cx	AB290702	00000703	00000006
mov edi,0AB29FFFFh	AB29FFFF	00000703	00000006
add edi,ecx			

ADD – Example (Cont.)

- Addition of partial registers:

Instruction	edi	ecx	eax
	AB29FFFF	00000703	000000FF
add al,ch	AB29FFFF	00000703	00000006
add di,cx	AB290702	00000703	00000006
mov edi,0AB29FFFFh	AB29FFFF	00000703	00000006
add edi,ecx	AB2A0702	00000703	00000006

ADD – Example (Cont.)

- Addition of partial registers:

Instruction	edi	ecx	eax
	AB29FFFF	00000703	000000FF
add al,ch	AB29FFFF	00000703	00000006
add di,cx	AB290702	00000703	00000006
mov edi,0AB29FFFFh	AB29FFFF	00000703	00000006
add edi,ecx	AB2A0702	00000703	00000006

- Wraparound is done according to the size of arguments.

SUB

- The SUB instruction subtracts numbers.
  - SUB destination,source.
  - $destination \leftarrow destination - source$
  - The result wraps around if needed.
    - Equivalent to $destination \leftarrow destination + (-source)$, where $-source$ is found using the two's complement method.
- Examples:
  - `sub eax,edx`
    - Subtracts edx from eax, and stores the result in eax. ($eax \leftarrow eax - edx$)
  - `sub cl,dl`
    - Subtracts dl from cl. Stores the result inside cl. ($cl \leftarrow cl - dl$).
  - `sub esi,4h`
    - Subtracts 4 from esi, and stores the result back in esi. ($esi \leftarrow esi - 4$).
  - Invalid example: `sub eax,dl`
    - eax is of size 32 bits. dl is of size 8 bits. Sizes mismatch.
  - Invalid example: `sub 1Ah,dl`
    - It is impossible to store the result inside 1Ah. No such opcode exists.

SUB - Example

Instruction	eax	ebx	ecx
	0000001A	00000003	00000002
sub eax,ebx			
add eax,ebx			
sub ecx,ebx			
add ecx,eax			
sub cl,al			

SUB - Example

Instruction	eax	ebx	ecx
	0000001A	00000003	00000002
sub eax,ebx	00000017	00000003	00000002
add eax,ebx			
sub ecx,ebx			
add ecx,eax			
sub cl,al			

SUB - Example

Instruction	eax	ebx	ecx
	0000001A	00000003	00000002
sub eax,ebx	00000017	00000003	00000002
add eax,ebx	0000001A	00000003	00000002
sub ecx,ebx			
add ecx,eax			
sub cl,al			

SUB - Example

Instruction	eax	ebx	ecx
	0000001A	00000003	00000002
sub eax,ebx	00000017	00000003	00000002
add eax,ebx	0000001A	00000003	00000002
sub ecx,ebx	0000001A	00000003	FFFFFFFF
add ecx,eax			
sub cl,al			

SUB - Example

Instruction	eax	ebx	ecx
	0000001A	00000003	00000002
sub eax,ebx	00000017	00000003	00000002
add eax,ebx	0000001A	00000003	00000002
sub ecx,ebx	0000001A	00000003	FFFFFFFF
add ecx,eax	0000001A	00000003	00000019
sub cl,al			

SUB - Example

Instruction	eax	ebx	ecx
	0000001A	00000003	00000002
sub eax,ebx	00000017	00000003	00000002
add eax,ebx	0000001A	00000003	00000002
sub ecx,ebx	0000001A	00000003	FFFFFFFF
add ecx,eax	0000001A	00000003	00000019
sub cl,al	0000001A	00000003	000000FF

SUB - Example

Instruction	eax	ebx	ecx
	0000001A	00000003	00000002
sub eax,ebx	00000017	00000003	00000002
add eax,ebx	0000001A	00000003	00000002
sub ecx,ebx	0000001A	00000003	FFFFFFFF
add ecx,eax	0000001A	00000003	00000019
sub cl,al	0000001A	00000003	000000FF

- Wraparound is done according to arguments size.

Summary

- MOV copies data from place to place.
- ADD adds numbers.
- SUB subtracts numbers.

Exercises

- Some code reading and predicting the resulting values of registers.
- Some code writing.
- Make sure to solve everything before moving on.
  - Very important for your understanding of the instructions and registers.