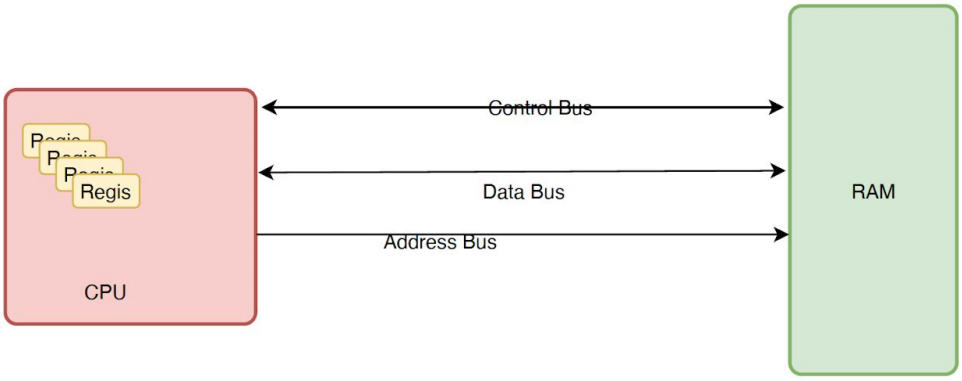
Computer Organization and Assembly Language - Lab Fall 2021

Concept of Address, Data and Control bus:



Register explanation:

AX	AH	AL	Accumulator	Data
BX	BH	BL	Base	
CX	CH	CL	Count	
DX	DH	DL	Data	
	SP		Stack Pointer	Pointer and index group
	BP		Base Pointer	
	SI		Source Index	
	DI		Destination Index	
	IP		Instruction Pointer	
	Flags		Status and control flags	
	ES		Extra	Segment group
	CS		Code	
	DS		Data	
	SS		Stack	

Instruction Format:

Assembly instructions are made up of an operation code (op-code) and a set of operands. The op-code identifies the action to be taken. The operands identify the source and destination of the data. The operands identify CPU registers, memory locations or I/O ports. The complete form of an instruction is:

```
[Op-code] [Destination Operand] [Source operand]
```

Let's start coding!

Example 01

English Version:

. Mov 5 to ax
. Mov 10 to bx
. Add ax to bx

Assembly version:

```
[org 0x01000]  ; leave it for upcoming labs
    mov ax, 5   ; moving 5 into ax
    mov bx, 10  ; moving 10 into bx
    add bx, ax  ; adding ax into bx
    mov ax, 0x4c00  ; leave it for upcoming labs
    int 0x21   ; leave it for upcoming labs
```

Compile the code:

```
nasm ch02-ex01.asm -o ex01.com -l ex01.lst
```

Debugger:

```
afd ex01.com
```

Example 02 (Swapping two numbers)

English version:

```
[org 0x01000]  ; leave it for upcoming labs
    mov ax,5
    mov bx,10
    mov cx, ax
    mov ax, bx
    mov bx, cx
    mov ax, 0x4c00  ; leave it for upcoming labs
    int 0x21   ; leave it for upcoming labs
```

Compile the code:

```
nasm ch02-ex02.asm -o ex02.com -l ex02.lst
```

Debugger:

```
afd ex02.com
```

Lab Task: Write a program in assembly language that calculates the square of six by adding six to the accumulator six times.