

Computer Organization

Second Class

Lab 2

Mov Instruction/ Memory Access

Dr. Samar Amil Yousif

Data Transfer Instructions (XCHG inst.)

Variables

array

XCHG Instruction

Exchange values of two operands.

XCHG Destination, Source

XCHG REG, memory

XCHG memory, REG

XCHG REG, REG

XCHG Instruction

- It **cannot directly** exchange the content of two memory locations.
- The **source** and **destination** must both be of the **same type** (bytes or words).
- The **segment registers cannot be used** in this instruction.
- Can not exchange with immediate no.
- This instruction **does not** affect any flag.

Example 1:

```
MOV AL, 5
```

```
MOV AH, 2
```

```
XCHG AL, AH
```

```
XCHG AL, AH
```

```
RET
```

What is variable ?

Variable is a memory location

Variables

Variable is a memory location. For a programmer it is much easier to have some value be kept in a variable named "**var1**" then at the address 5A73:235B, especially when you have 10 or more variables.

Our compiler supports two types of variables: **BYTE** and **WORD**.

Syntax for a variable declaration:

name **DB** *value*

name **DW** *value*

DB - stays for Define Byte.

DW - stays for Define Word.

name - can be any letter or digit combination, though it should start with a letter. It's possible to declare unnamed variables by not specifying the name (this variable will have an address but no name).

value - can be any numeric value in any supported numbering system (hexadecimal, binary, or decimal), or "?" symbol for variables that are not

Example 1:

Note: Compiler is not case sensitive, so "VAR1" and "var1" refer to the same variable.

```
MOV AL, var1
```

```
MOV BX, var2
```

```
RET ; stops the program.
```

```
VAR1 DB 7
```

```
var2 DW 1234h
```

Example1:

```
mov cl, 8
```

```
xchg cl,a
```

```
MOV AL, a
```

```
MOV BL, b
```

```
RET
```

```
a DB 3
```

```
b DB 5
```

Example 2:

```
mov al, a
```

```
xchg al,b
```

```
mov cl,b
```

```
ret
```

```
a db 6
```

```
b db 5
```

Arrays

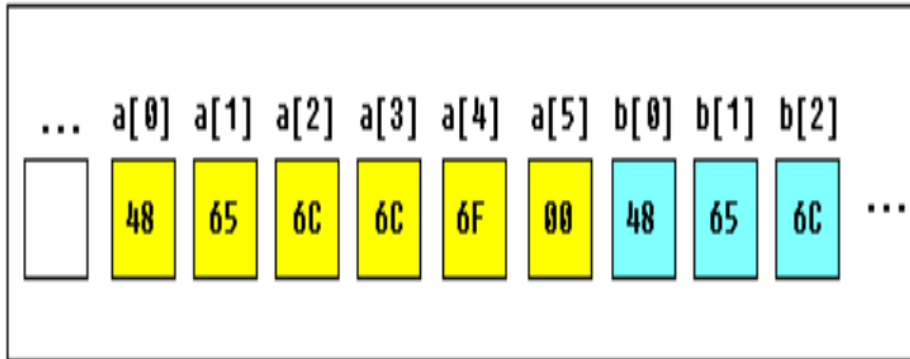
Arrays can be seen as chains of variables. A text string is an example of a byte array, each character is presented as an ASCII code value (0..255).

Here are some array definition examples:

```
a DB 48h, 65h, 6Ch, 6Ch, 6Fh, 00h
```

```
b DB 'Hello', 0
```

b is an exact copy of the *a* array, when compiler sees a string inside quotes it automatically converts it to set of bytes. This chart shows a part of the memory where these arrays are declared:



You can access the value of any element in array using square brackets, for example:

```
MOV AL, a[3]
```

Example

```
mov cl, a[2]  
ret  
a db 1,2,3,4,5,6
```

Note: You can also use any of the memory index registers **BX, SI, DI, BP**, for example:

```
MOV SI, 3
```

```
MOV AL, a[SI]
```

```
ret
```

```
a db 1, 2, 3 , 4, 5
```