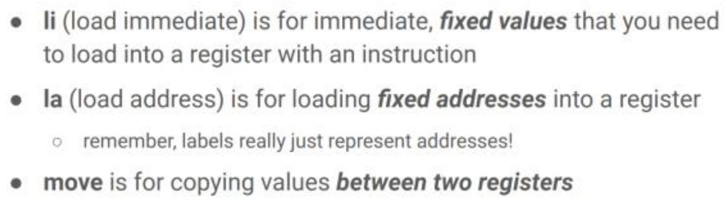
General notes

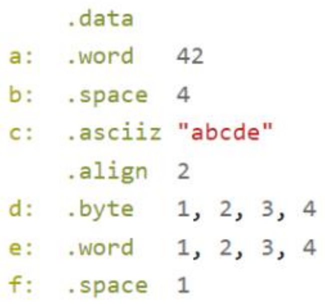
Monday, 23 September 2024 3:41 PM



- **li** (load immediate) is for immediate, ***fixed values*** that you need to load into a register with an instruction
- **la** (load address) is for loading ***fixed addresses*** into a register
    - remember, labels really just represent addresses!
- **move** is for copying values ***between two registers***

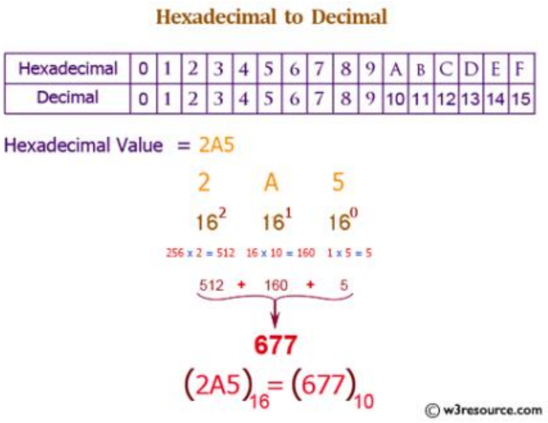
Q2. memory

Monday, 23 September 2024 3:45 PM



```
     .data
a:   .word    42
b:   .space   4
c:   .asciiz "abcde"
     .align   2
d:   .byte    1, 2, 3, 4
e:   .word    1, 2, 3, 4
f:   .space   1
```

Label	address	Contents	Contents in hex
a	0x10010020	42	0x 2A 00 00 00
b	0x10010024	?? ??? ???	0x ?? ?? ?? ??
c	0x10010028	'a', 'b', 'c', 'd'	0x 61 62 63 64
	0x1001002C	'e', '\0' X X	0x 65 00 ?? ??
d	0x10010030	1, 2, 3, 4	0x 01 02 03 04
e	0x10010034	1	0x 01 00 00 00
	0x10010038	2	0x 02 00 00 00
	0x1001003C	3	
	0x10010040	4	
f	0x10010044	?? ?? ?? ??	0x ?? ?? ?? ??

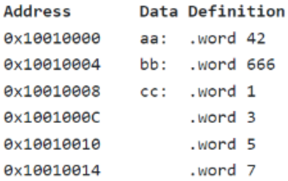

Hexadecimal to Decimal
Hexadecimal 0 1 2 3 4 5 6 7 8 9 A B C D E F
Decimal 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Hexadecimal Value = 2A5
2 A 5
16² 16¹ 16⁰
256 x 2 = 512 16 x 10 = 160 1 x 5 = 5
512 + 160 + 5
677
(2A5)16 = (677)10
© w3resource.com

.align k
Next memory address divisble by 2^k

Q4. memory

Monday, 23 September 2024 3:52 PM

```
a. la   $t0, aa
   $t0 = 0x10010000

b. lw   $t0, bb
   $t0 = 666

c. lb   $t0, bb
   $t0 = 9A

d. lw   $t0, aa+4
   $t0 = 666

e. la   $t1, cc
   $t1 = 0x10010008

f. lw   $t0, ($t1)
   $t0 = 1

g. la   $t1, cc
   lw   $t0, 8($t1) # $t1 + 8
   $t0 = 5

h. li   $t1, 8
   lw   $t0, cc($t1) # cc + $t1 = 0x10010010
   $t0 = 5


i. la   $t1, cc
   lw   $t0, 2($t1) # cc + 2
```



Address	Data	Definition
0x10010000	aa:	.word 42
0x10010004	bb:	.word 666
0x10010008	cc:	.word 1
0x1001000C		.word 3
0x10010010		.word 5
0x10010014		.word 7

Bb	9A	02	00	00

Cc:	01 00 00 00
	03 00 00 00

Int cc[4] = {1, 3, 5, 7}

666 = 0x 00 00 02 9A

La = load address
Li = load immediate
Move = copy between registers

Address
- Label
- ($reg)

Lw $t0, address
Lb

Sw $t0, address
Sb

Arrays

Monday, 23 September 2024 5:38 PM

```
int numbers[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
```

Numbers[0] => lw $t0, numbers

Numbers[1]

- Get the address

&numbers[1] = &numbers + 4
& numbers[2] = &numbers + 8
&numbers[i] = &numbers + i*4

label	Memory address	Contents (1 word)	How we would access in c
numbers:	1000	0	Numbers[0]
	1004	1	Numbers[1]
	1008	2	Numbers[2] etc.
	1012	3	
	1016	4	
	1020	5	
	1024	6	
	1028	7	
	1032	8	
	1036	9	

How to get array[i]
Get the address
&array[i] = &array + i*sizeof(element)

Lw, $reg, &array[i]