Endianness

Little Endian and Big Endian Data Storage

- Big-endian: is an order in which the "big end" (most significant value in the sequence) is stored first (at the lowest storage address).

- Little-endian is an order in which the "little end" (least significant value in the sequence) is stored first (at the lowest storage address).

- Example: For Hex Number 0x4F52
  - **Big Endian:** if 4F is stored at storage address 1000, 52 will be at address 1001)

  - **Little Endian:** it would be stored as 524F (52 at address 1000, 4F at 1001)

https://searchnetworking.techtarget.com/definition/big-endian-and-little-endian

Exercise 1

- For the number `0xFF00AA11`, fill in the following:

Big Endian	
Address	Value
8003	
8002	
8001	
8000	

Little Endian	
Address	Value
8003	
8002	
8001	
8000	

Exercise

- For the number `0xFF00AA11`, fill in the following:

Big Endian	
Address	Value
8003	11
8002	AA
8001	00
8000	FF

Little Endian	
Address	Value
8003	FF
8002	00
8001	AA
8000	11

https://chortle.ccsu.edu/assemblytutorial/Chapter-15/ass15_4.html

Exercise 2

- Consider an array of bytes `[0x11, 0x12, 0x13, 0x14]`. Fill in the following:

Big Endian	
Address	Value
8003	
8002	
8001	
8000	

Little Endian	
Address	Value
8003	
8002	
8001	
8000	

Exercise 2

- Consider an array of bytes `[0x11, 0x12, 0x13, 0x14]`. Fill in the following:

Big Endian	
Address	Value
8003	14
8002	13
8001	12
8000	11

Little Endian	
Address	Value
8003	14
8002	13
8001	12
8000	11

Exercise 3

- Consider an array of half words: [0x1234, 0x5678], fill in the following:

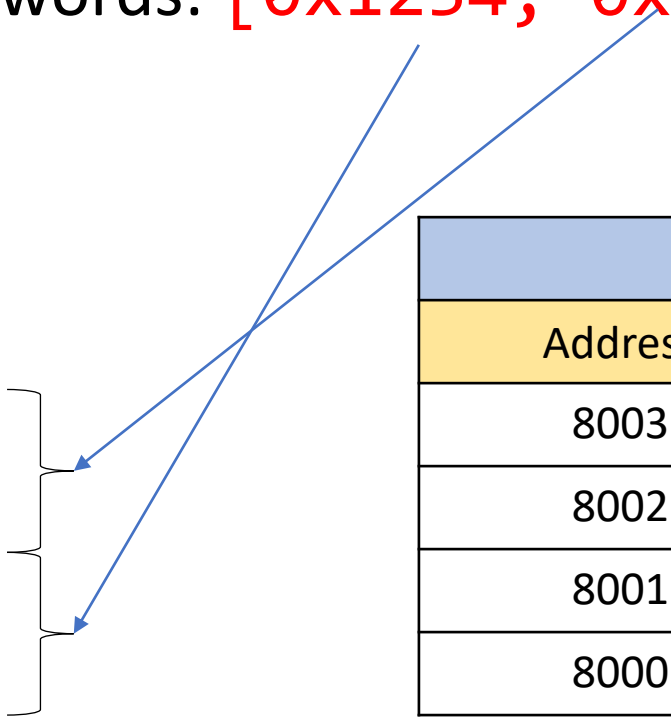
Big Endian	
Address	Value
8003	
8002	
8001	
8000	

Little Endian	
Address	Value
8003	
8002	
8001	
8000	

Exercise 3

- Consider an array of half words: `[0x1234, 0x5678]`, fill in the following:

Big Endian	
Address	Value
8003	
8002	
8001	
8000	



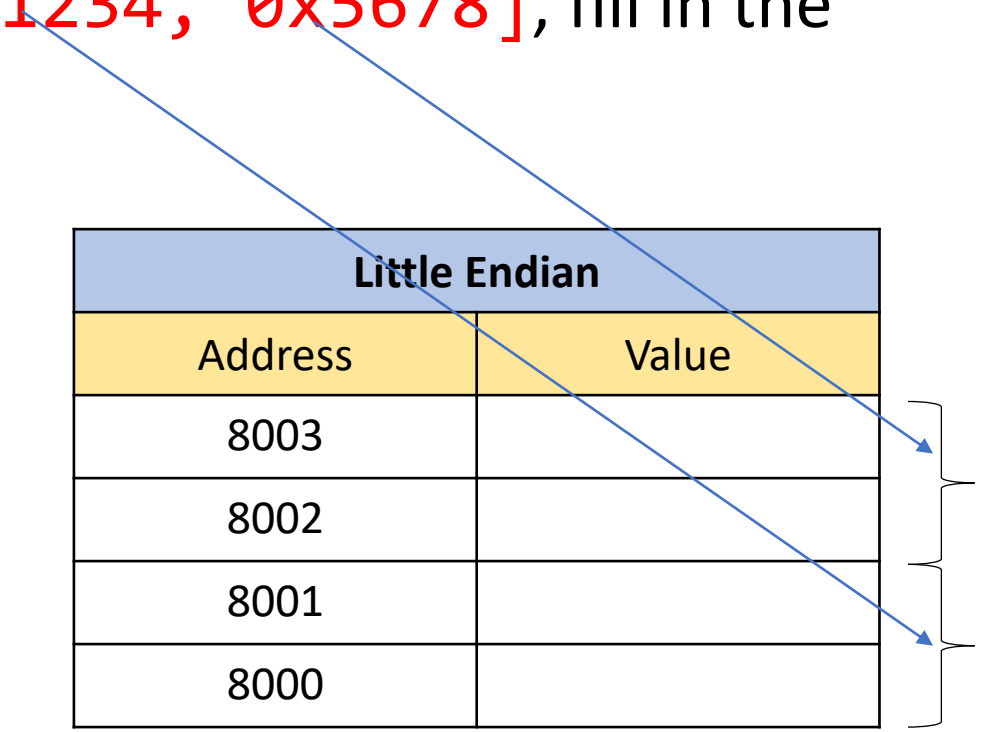
Little Endian	
Address	Value
8003	
8002	
8001	
8000	

Exercise 3

- Consider an array of half words: `[0x1234, 0x5678]`, fill in the following:

Big Endian	
Address	Value
8003	
8002	
8001	
8000	

Little Endian	
Address	Value
8003	
8002	
8001	
8000	



Exercise 3

- Consider an array of half words: `[0x1234, 0x5678]`, fill in the following:

Big Endian	
Address	Value
8003	78
8002	56
8001	34
8000	12

Little Endian	
Address	Value
8003	56
8002	78
8001	12
8000	34

Exercise 4

- Consider the following MIPS code snippet:

```
    .data
        array_of_bytes:        .byte   0x12, 0x34, 0x56, 0x78
        word:                  .word 0x12345678
```

➢In a Big Endian Machine, what does the memory look like?

➢In Little Endian Machine, what does the memory look like?

Lower to Higher Address

Exercise 4

- Consider the following MIPS code snippet:

```
    .data
        array_of_bytes:        .byte   0x12, 0x34, 0x56, 0x78
        word:                  .word   0x12345678
```

➢In a Big Endian Machine, what does the memory look like?

12	34	56	78	12	34	56	78

➢In Little Endian Machine, what does the memory look like?

12	34	56	78	78	56	34	12

Lower to Higher Address

Exercise 5

```
.data
    array_of_bytes:        .byte   0x12, 0x34, 0x56, 0x78
    word:                  .word   0x12345678
.text
.globl main
main:
    lw      t0, array_of_bytes
    lw      t1, word
```

In a LE machine, what are the values loaded in registers *t0* and *t1*?

Exercise 5 - Solution

Let’s draw what the memory looks like…

Little Endian	
Address	Value(in Hex)
8007	12
8006	34
8005	56
8004	78
8003	78
8002	56
8001	34
8000	12

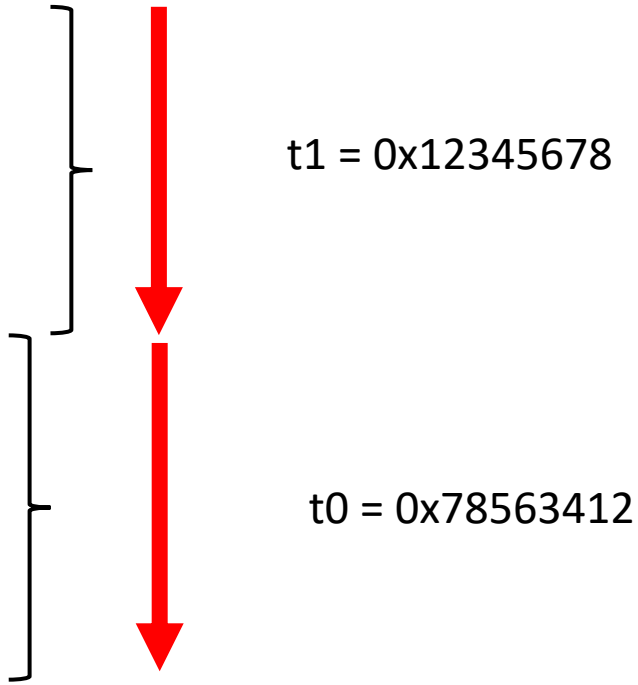
word

array_of_bytes

Exercise 5 - Solution

Let’s draw what the memory looks like…

Little Endian	
Address	Value(in Hex)
8007	12
8006	34
8005	56
8004	78
8003	78
8002	56
8001	34
8000	12



Lab 3