

MAIN MEMORY

ORGAssignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Bernhard Kainz (with thanks to

N. Dulay and E.

Edwards)

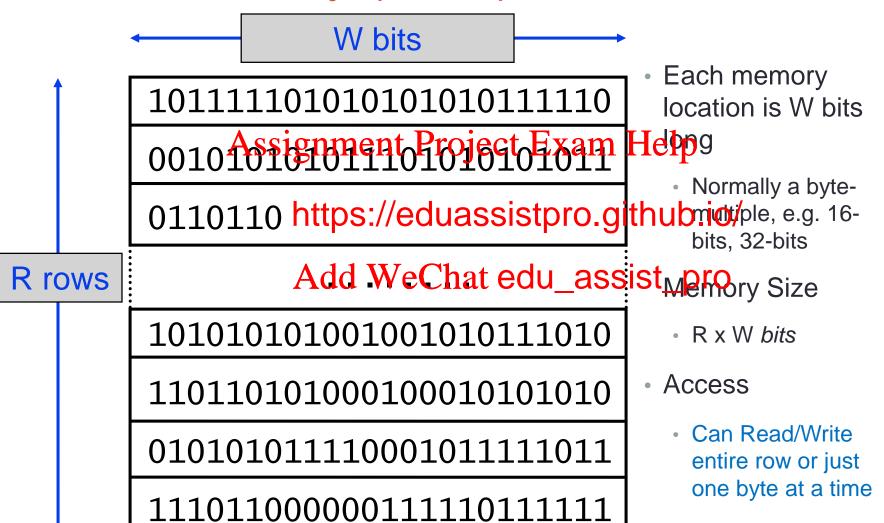
b.kainz@imperial.ac.uk

Main Memory Organisation

- Addressing
- Byte Orderingssignment Project Exam Help
- https://eduassistpro.github.io/
 Memory Module

Add WeChat edu_assist_pro

Main Memory (RAM)



Addressing

Main Memory

 Where in memory is the 16-bit value of 3?

| Project Even Help | 1101 | 1010 | 1101 | 0110 |
|--------------------------------|----------|---------------------|------|------|
| Project Exam Help | 0011 | 0000 | 0000 | 0000 |
| a scher duassistpro.gitիաթ. | ttns://e | 000 h | 0000 | 0000 |
| daassistpro.gievelly. | itpo.//0 | 111 | 1111 | 1111 |
| Chat edu_assist_p | dowe | 000Q <mark>A</mark> | 0000 | 0000 |
| | 0010 | 1010 | 1010 | 1001 |
| ADDRESSING | 0000 | 0000 | 0000 | 0000 |
| Identify memory | 1110 | 1111 | 1111 | 1111 |

a scheme for uniquely uassistpro.githម្រៀ គ្រួ

Chat edu_assist_pro

ADDRESSING

Identify memory locations with a positive number called the (memory) address

Word Addressing

| Main M | Memory | Address |
|-----------|------------------------|--------------------------------------|
| 0110 1101 | 1010 1101 | ← 0 |
| 0000 0000 | Assignment] | Project Exam Help |
| 0000 0000 | ⁰ https://e | duassistpro.ghtfdb.io/es tire row |
| 1111 1111 | 111111111We | Chat edu_assist_pro |
| 0000 0000 | 0000 0000 | <u>←</u> 4 |
| 1001 1010 | 1010 0010 | ← 5 |
| 0000 0000 | 0000 0000 | ← 6 |
| 1111 1111 | 1111 1110 | ← 7 |

Byte Addressing

Main Memory

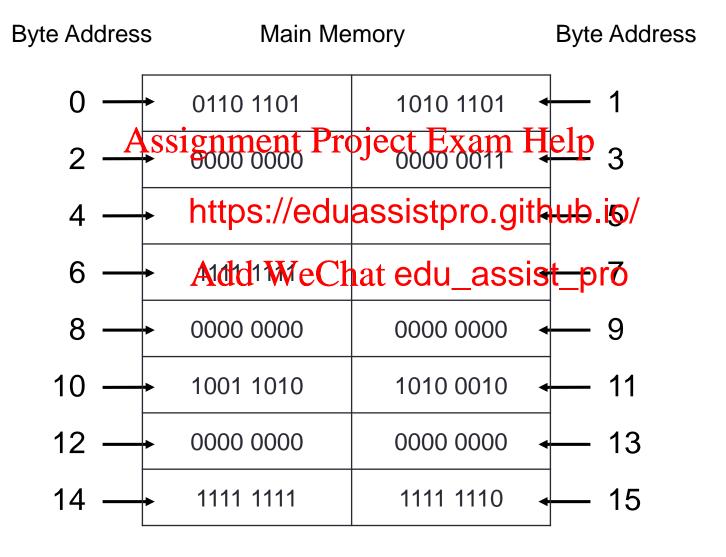
Word Address

| 0110 1101 | 1010 1101 | ← 0 • With byte roject Exam Helpressing, every |
|-----------|------------|--|
| 0000 0000 | | byte in main |
| 0000 0000 | https://ed | uassistpro.githubmon/ has an address |
| 1111 1111 | 1Add1We | hat edu_assist_pro |
| 0000 0000 | 0000 0000 | ← 8 • In this example |
| 1001 1010 | 1010 0010 | which is byte 0 and which is byte 1? |
| 0000 0000 | 0000 0000 | ← 12 |
| 1111 1111 | 1111 1110 | ← 14 |

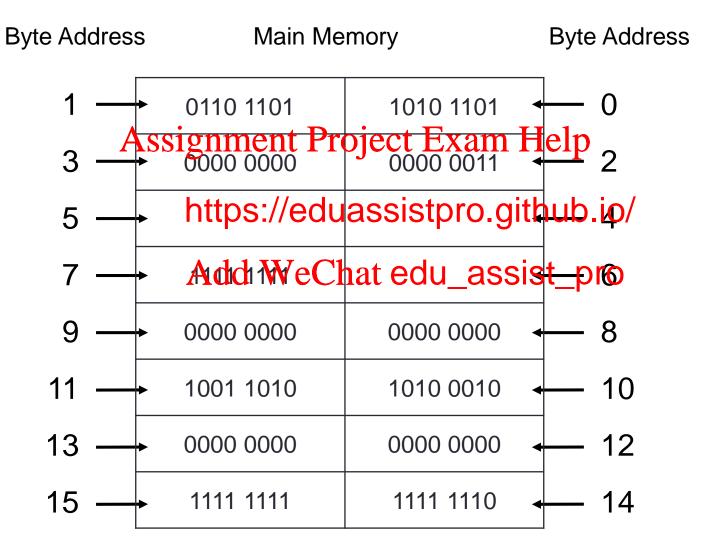
Byte Addressing

- Two formats
 - Big Endian Assignment Project Exam Help
 - Stores Most Sighttps://eduassistpro.github.io/
 - Motorola 6800, Ishala WerChart edu_assist_ARINA
 - Little Endian
 - Stores Least Significant Byte first
 - x-86, ARM, DEC Alpha, VAX, PDP-11

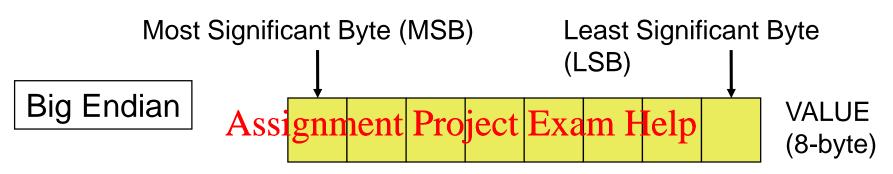
Byte Addressing (Big Endian)



Byte Addressing (Little Endian)

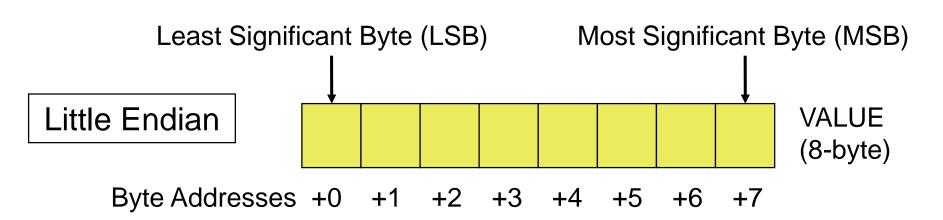


Byte Ordering – Multibyte Data Items



Byte Addresses https://eduassistpro.github.io//

Add WeChat edu_assist_pro



Example 1: 16-bit Integer

(View 1)

16-bit integer '5' stored at memory address 24

Big Endian Projectoborous Help

Byte https://eduassistpro.gathub.io/

Add WeChat edu_assist_pro

Little Endian

0000 0101 0000 0000

Byte Addresses

24

25

Example 1: 16-bit Integer

(View 2)

16-bit integer '5' stored at memory address 24

Big Endian ignment Brown Helpword address 24

Byte Addrhttps://eduassistpro.github.io/

Add WeChat edu_assist_pro

Little Endian

0000 0000 0000 0101

Word address 24

Byte Addresses

25

24

Example 2: 32-bit Value

(View 1)

32-bit hex value 54 BC FE 30 stored at memory address 24

```
Big Endian Assignment Project Exam Help
                                            FF
                                                    30
                 https://eduassistpro.github.jo/
            0101
                                                 0011 0000
                       WeChat edu_assist_
25
Byte Addresses
                                                    27
 Little Endian
                 30
                         FF
                                            BC
                                                    54
            0011 0000 1111 1110
                                        1011 1100
                                                 0101 0100
 Byte Addresses 24
                                            26
                        25
                                                    27
```

Example 2: 32-bit Value

(View 2)

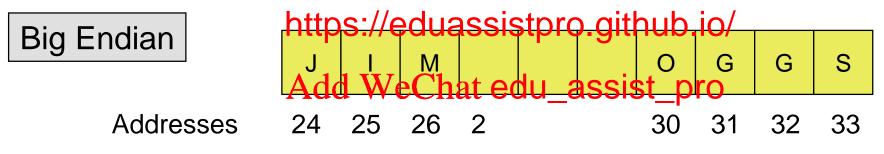
32-bit hex value 54 BC FE 30 stored at memory address 24

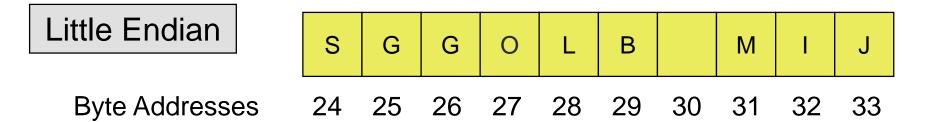
```
Big Endian Assignment Project Exam Help
                 https://eduassistpro.github.jo/
0 0011 0000
                                            FF
                                                     30
            0101
                        WeChat edu_assist_26
Byte Addresses
 Little Endian
                 54
                         BC
                                            FE
                                                     30
            0101 0100
                      1011 1100
                                        1111 1110 0011 0000
Byte Addresses 27
                                            25
                                                    24
                        26
```

Example 3: ASCII String

(View 1)

- String "JIM BLOGGS" stored at memory address 24
- Treat a string as an array of (ASCII) bytes
 - Each byte is Aconsidered into Probled byts Examiff February en is 2-bytes)

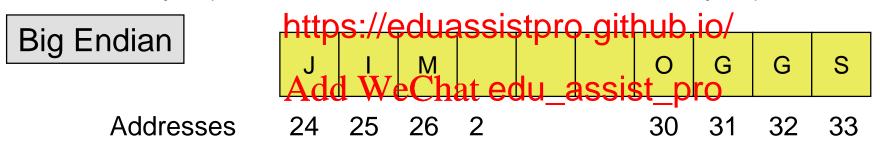


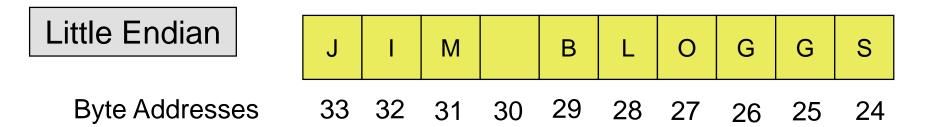


Example 3: ASCII String

(View 2)

- String "JIM BLOGGS" stored at memory address 24
- Treat a string as an array of (ASCII) bytes
 - Each byte is Aconsidered into Probled byts Examiff February en is 2-bytes)





Potential Problems

- How do we tran https://eduassistpro.glffiub.go/"JIM BLOGGS") from to a Little-Endian memory and vice Werswe Chat edu_assist_pro
- How do we transfer an object which holds both types of values above and vice-versa?
- Why is it necessary?

Question

 What is the maximum amount of memory we can have in a 32-bit machine with byte addressing?

Assignment Project Exam Help

Each address pe

https://eduassistpro.github.io/

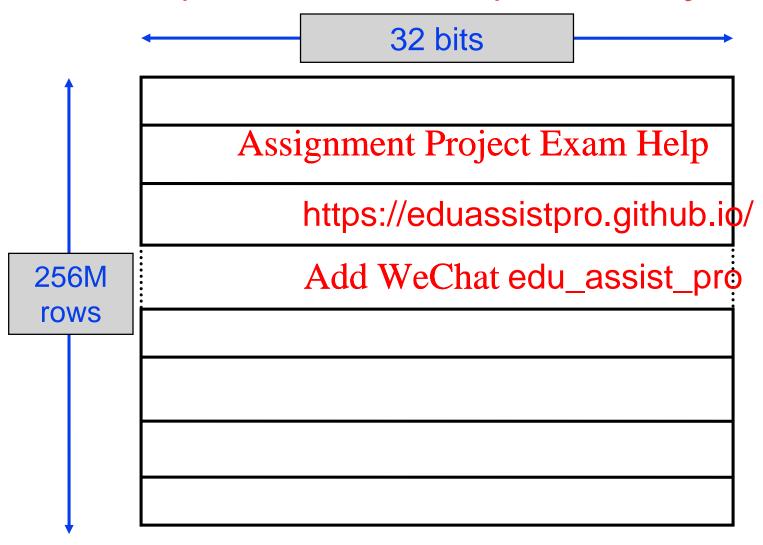
- Number of available addresses = Add WeChat edu_assist_pro
- Recall: Kilo = 2^{10} (10³), Mega = 2^{20} (10⁶) and Giga = 2^{30} (10⁹)
- Hence, we have $2^{32} = 2^2 \times 2^{30} = 4 \times 2^{30}$ bytes = 4 Gigabytes = 4GB
- How much memory for 64-bit addressing?

Memory Modules and Chips

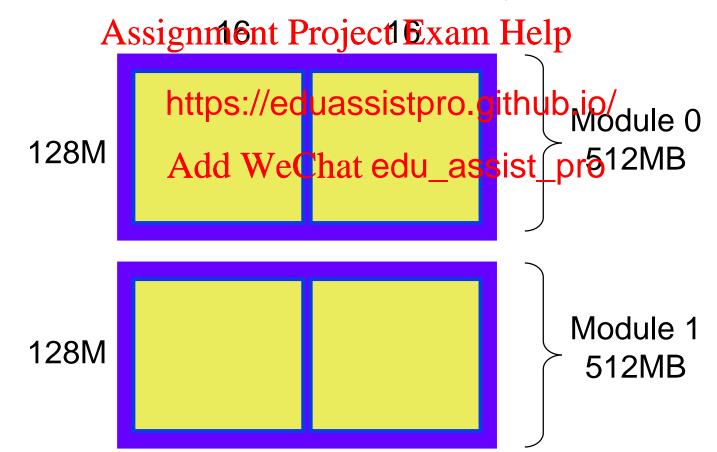
Assignment Project Exam Help

https://eduassistpro.github.io/

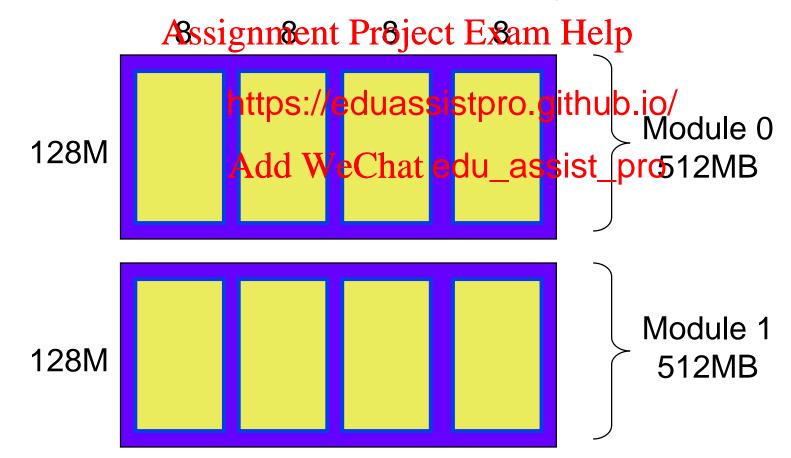
Add WeChat edu_assist_pro



- Two 512MB memory modules
 - Each module has two 128M x 16-bit RAM Chips

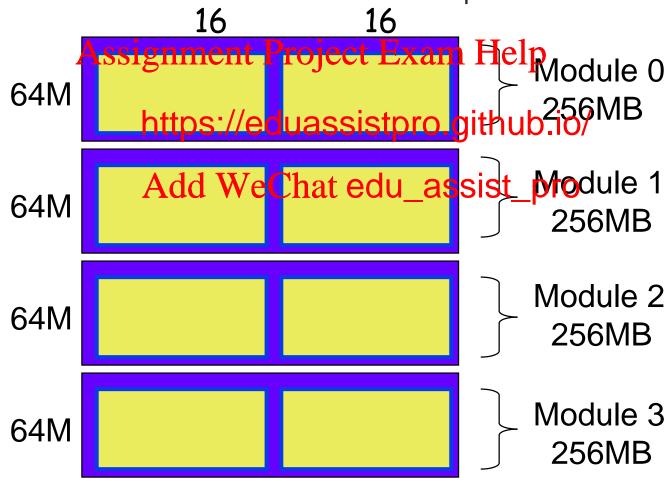


- Two 512MB memory modules
 - Each module has four 128M x 8-bit RAM Chips

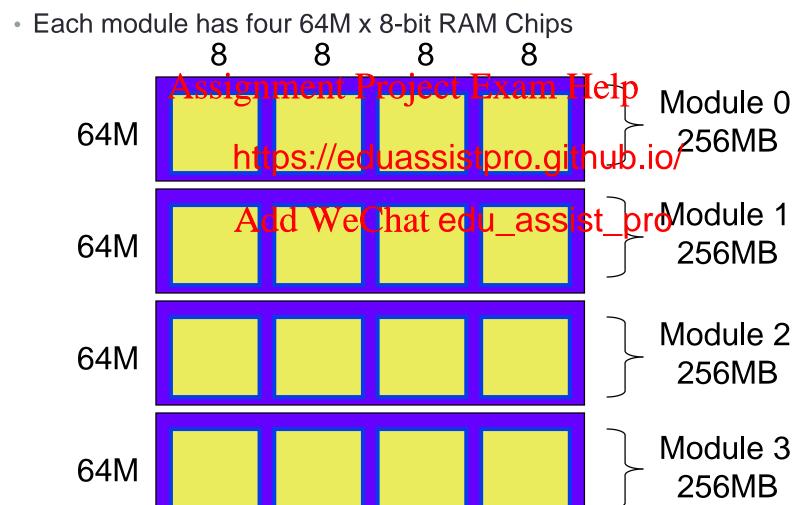


Four 256MB memory modules





Four 256MB memory modules



Memory Interleaving

- Example:
 - Memory = 4M words, each word = 32-bits
 - · Built with 4 Alyiga heim Project Leam Help
 - For 4M words w

S

22 bits = 2 bits (thttps://eduassistpro.githuelegitor/ow within Module)

Add WeChat edu_assist_pro

Module Row within Module

High-Order Interleave

20 2

Row within Module

Module

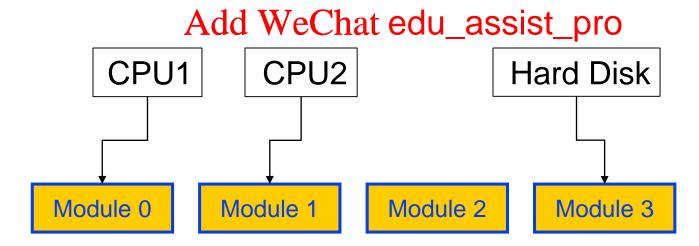
Low-Order Interleave

High-Order Interleave

| Address Decimal | | | | dress nary | | | | |
|------------------------|----|-------------|--------------|---------------|--------|--------------|---|------------------------|
| 0 | 00 | ASS 0000 | ignme | ent Pr | oject | Exan 0000 | n <mark>Help</mark> Module= 0 | Row=0 |
| 1 | 00 | 0000 | httns | ·//edi | ıaccio | storo | Module=0/ | Row=1 |
| 2 | 00 | 0000 | пцро |).// Cat | JUSSIC | stpro. | odule=0 | Row=2 |
| 3 | 00 | 0000 | 60000 | M000 | have o | du_as | sistdybato | Row=3 |
| 4 | 00 | 0000 | 0000 | 0000 | 000 | | odule=0 | Row=4 |
| 5 | 00 | 0000 | 0000 | 0000 | 0000 | 0101 | Module=0 | Row=5 |
| ••• | | | | | | | | |
| 2 ²⁰ -1 | 00 | 1111 | 1111 | 1111 | 1111 | 1111 | Module=0 | Row=2 ²⁰ -1 |
| 2 ²⁰ | 01 | 0000 | 0000 | 0000 | 0000 | 0000 | Module=1 | Row=0 |
| 2 ²⁰ +1 | 01 | 0000 | 0000 | 0000 | 0000 | 0001 | Module=1 | Row=1 |

High-Order Interleave

- Good if Modules can be accessed independently by different units, e.g. by the CPU and a Hard Disk (or a second CPU) styphthen units use different Modules
- Parallel operatiohttps://eduassistpro.github.io/

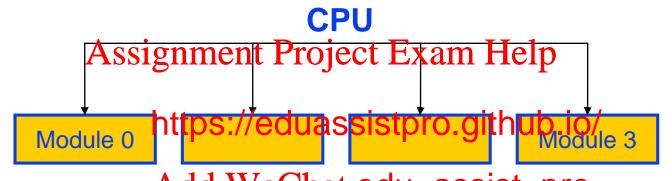


Low-Order Interleave

| Address Decimal | Address Binary | |
|--------------------|--|--------------------|
| 0 | 00 odssignment Project Exam Help=0 | Row=0 |
| 1 | 00 0000 odule=1 | Row=0 |
| 2 | _{00 0000} https://eduassistpro.gitիկի <u>եւ</u> թ/ | Row=0 |
| 3 | 00 0000 0000 0000 000 dule= $\frac{3}{2}$ | Row=0 |
| 4 | 00 0000 0000 0000 000 edu_assist_pro | Row=1 |
| 5 | 00 0000 0000 0000 0000 01 <mark>01</mark> Module=1 | Row=1 |
| | | |
| $2^{20}-1$ | 00 1111 1111 1111 1111 11 <mark>11</mark> Module= <mark>3</mark> | $Row = 2^{18} - 1$ |
| 2 ²⁰ | 01 0000 0000 0000 0000 00 <mark>00</mark> Module= 0 | $Row=2^{18}$ |
| 2 ²⁰ +1 | 01 0000 0000 0000 0000 00 <mark>01</mark> Module=1 | $Row=2^{18}$ |

Low-Order Interleave

Good if the CPU (or other unit) can request multiple adjacent memory locations



- Since adjacent memory locations lie odules an "advanced" memory system can perform the accesses in parallel
 - Such adjacent accesses often occur in practice, e.g.
 - i. Elements in an array, e.g.. Array[N], Array[N+1], Array[N+2],
 - ii. Instructions in a Programs, InstructionN, InstructionN+1,...
- In the above situations, an "advanced" CPU can pre-fetch the adjacent memory locations → higher performance