Memory Interleaving

- As CPU processing speed increased, it started to outpace memory access speed. This resulted in longer wait times as the CPU idled as the slower memory unit retrieved data.

- One strategy developed to overcome this was memory interleaving. It is a design which compensates for the slow speed of RAM by spreading memory addresses across multiple chips.

Memory Interleaving

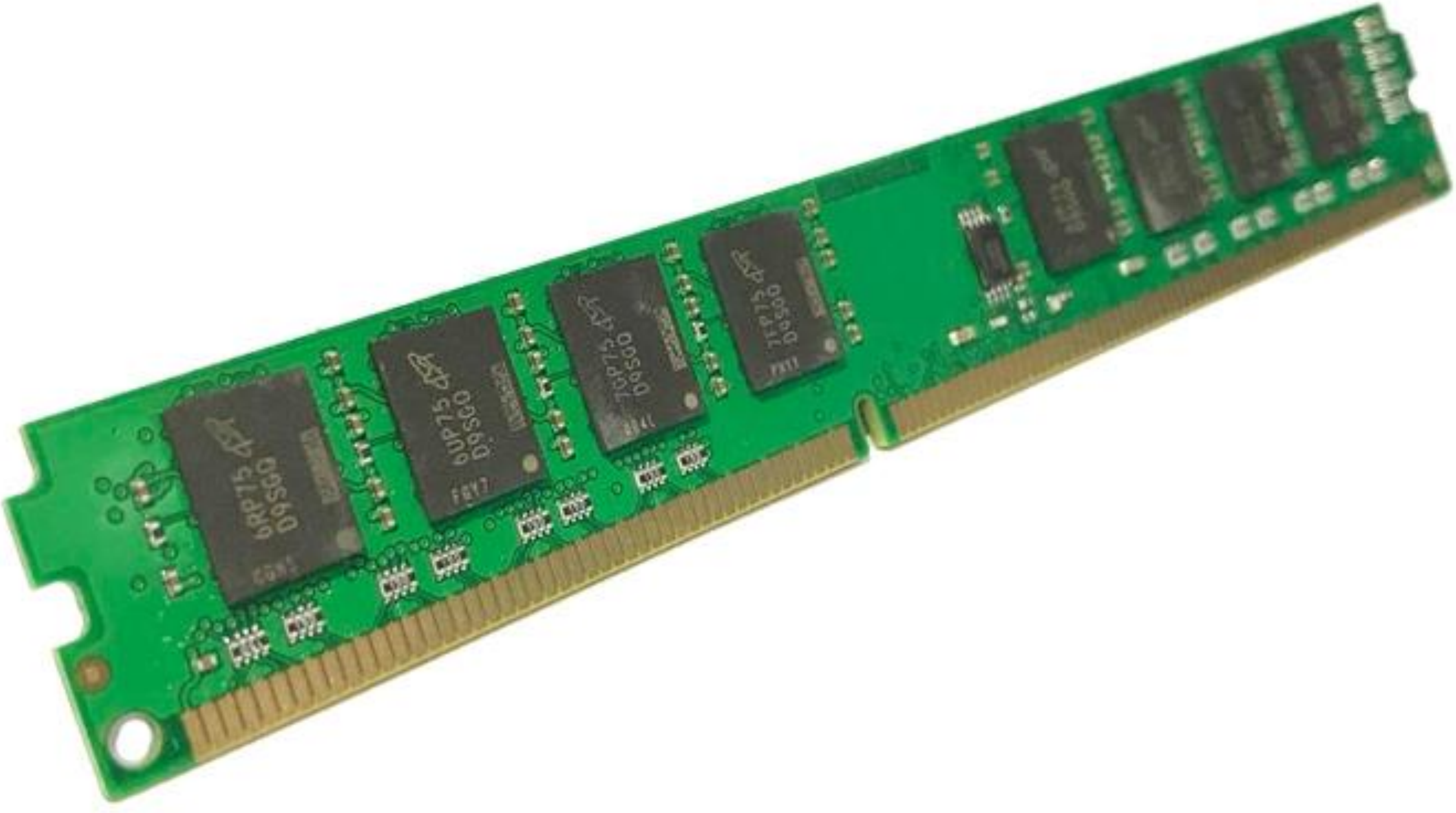
- Computer memory consists of a linear array of addressable storage cells that are similar to registers.
- Memory can be **byte-addressable**, or **word-addressable**, where a word typically consists of two or more bytes.
- Memory is constructed of RAM chips, often referred to in terms of length × width (L x W).
- A **4M** × **8** RAM chip gives us 4,194,304 8-bit memory locations.

Memory Organization

- How does the computer access a memory location corresponding to a particular address?
- We observe that 4MB can be expressed as $2^2 \times 2^{20} = 2^{22}$ words.
- The memory locations for this memory are numbered 0 through $2^{22} - 1$.
- Thus, the memory bus of this system requires 22 address lines.
  - The address lines "count" from 0 to $2^{22} - 1$ in binary. Each line is either "on" or "off" indicating the location of the desired memory element.

Memory Organization

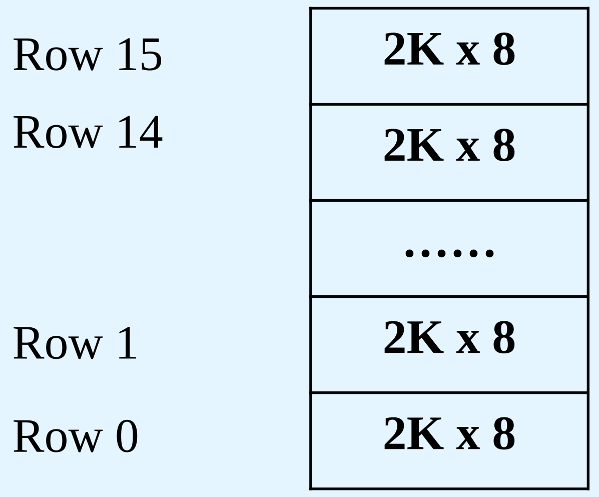
- Physical memory usually consists of more than one RAM chip.
- Access is more efficient when memory is organized into banks of chips with the addresses **interleaved** across the chips.


6RP75 D9SGO
6UP75 D9SGO
7DP75 D9SGO
7FP75 D9SGO

Memory Organization

- Suppose we need to build 32K x 8, byte-addressable memory, but we only have 2K x 8 RAM chips.

- Connect 16 rows of chips together:



Memory Organization

- Each chip addresses 2K bytes.

- Because we have a total of 32K addresses for this memory, must have 15 bits (2^5 x 2^{10} = 2^{15} bytes to access).

- Each chip only holds 2^{11} bytes, so 4 bits are used to determine the address of each chip (16 of them = 2^4).

Memory Organization

- Either the leftmost bits or the rightmost bits are used, depending on the architecture.
- A decoder is used to decode these 4 bits to determine which chip holds the desired data.
- Once the proper chip has been located, the remaining 11 bits are used to determine the offset on the chip (where the data is).

Memory Organization

- A single memory module can only be accessed sequentially so memory is normally split across multiple modules (or banks) which can be accessed simultaneously.

- This process is called **Memory Interleaving**.

- With **low-order interleaving**, the low-order bits are used to address the banks.

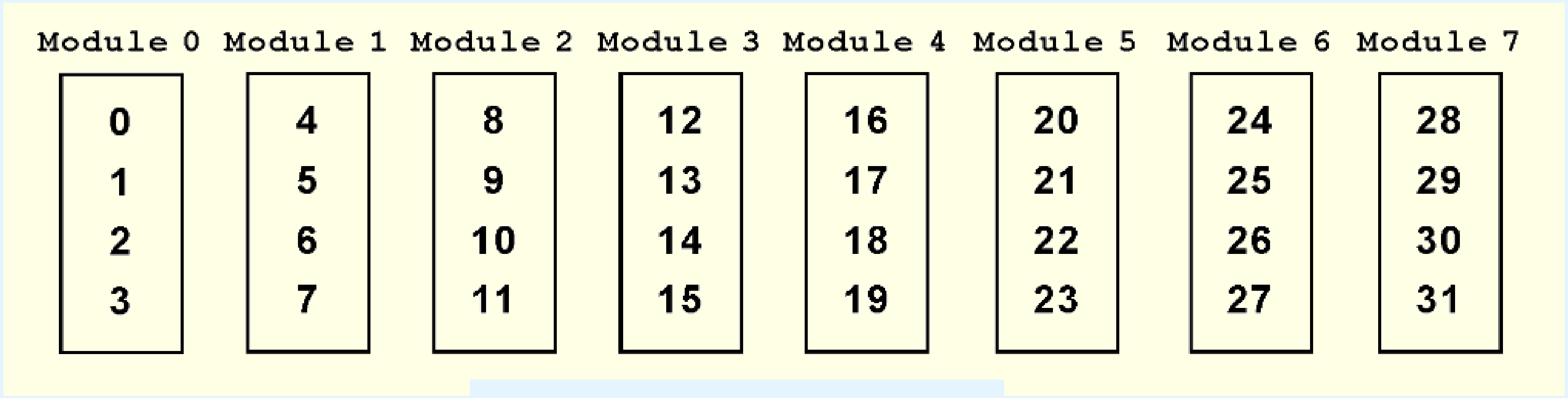
- With **high-order interleaving**, the high-order bits are used to address the banks.

Memory Organization

- Suppose we have a byte-addressable memory consisting of 8 modules of 4 bytes each, giving a total of 32 bytes of memory.
- We need 5 bits to uniquely identify each byte ($32 = 2^5$) in the total memory space.
- We have 8 modules so we need 3 bits to address each of them ($2^3 = 8$).
- The remaining 2 bits are used to determine the offset within the module.

Memory Organization

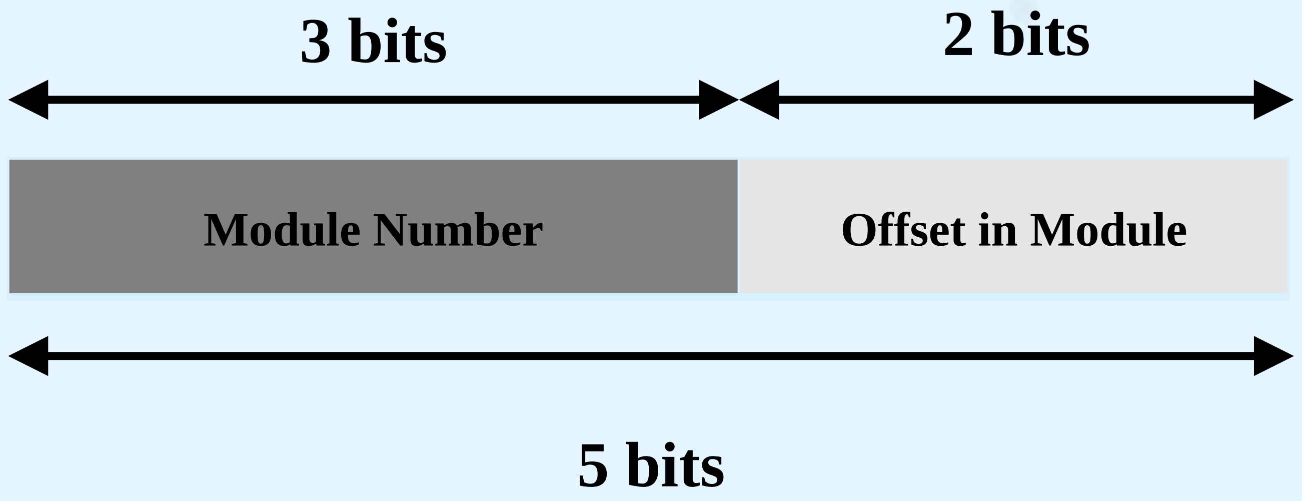
- **High-order interleaving** distributes the addresses so that each module contains consecutive addresses.
- Module 0 contains the data stored at addresses 0, 1, 2 and 3
- Module 1 contains the data stored at addresses 4, 5, 6 and 7.
- ….
- Module 7 contains the data stored at addresses 28, 29, 30 and 31



High-Order Interleaving

Memory Organization

- High-order interleaving uses the first 3 bits to determine the address of the memory module and the remaining 2 bits are used to determine the offset within the module.

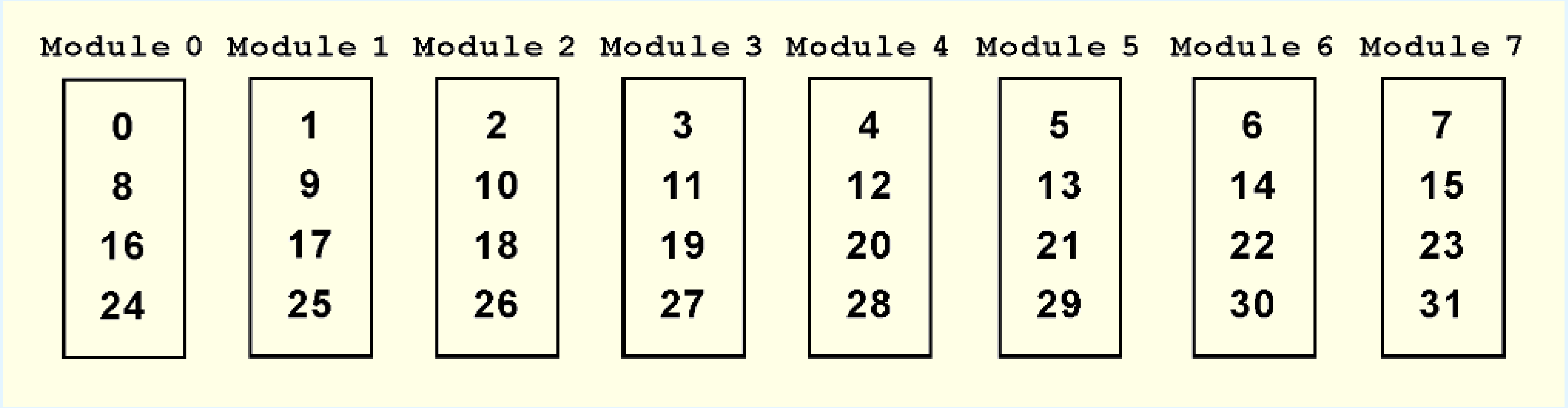


Memory Organization

Module	Decimal Address	Binary Address	Split	Module Number	Offset
Module 0	0	00000	000 00	0	0
	1	00001	000 01	0	1
	2	00010	000 10	0	2
	3	00011	000 11	0	3
Module 1	4	00100	001 00	1	0
	5	00101	001 01	1	1
	6	00110	001 10	1	2
	7	00111	001 11	1	3

Memory Organization

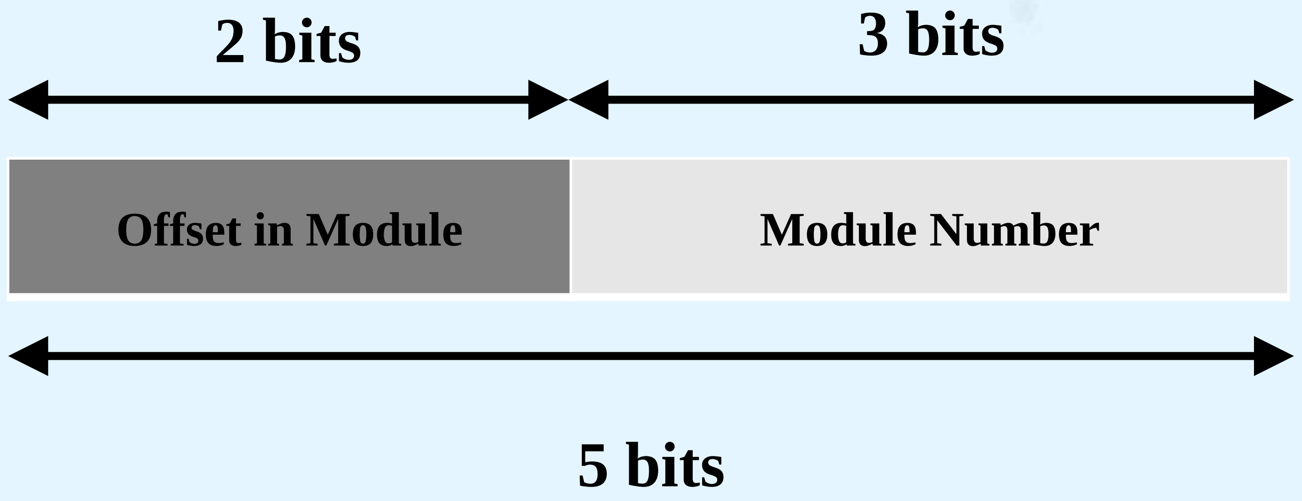
- **Low-order interleaving** places consecutive memory addresses in different modules
- Module 0 contains the data stored at addresses 0, 8, 16 and 24
- Module 1 contains the data stored at addresses 1, 9, 17 and 25.
- ….
- Module 7 contains the data stored at addresses 7, 15, 23 and 31



Low-Order Interleaving

Memory Organization

- Low-order interleaving uses the last 3 bits to determine the address of the memory module and the remaining 2 bits are used to determine the offset within the module.



Memory Organization

Module	Decimal Address	Binary Address	Split	Offset	Module Number
Module 0	0	00000	00 000	0	0
	8	01000	01 000	1	0
	16	10000	10 000	2	0
	24	11000	11 000	3	0
Module 1	1	00001	00 001	0	1
	9	01001	01 001	1	1
	17	10001	10 001	2	1
	25	11001	11 001	3	1

Memory Organization

- The order of interleaving refers to the number of memory banks used in the memory:
  - 4-way uses 4 banks
  - 8-way uses 8 banks
  - 16-way uses 16 banks, etc.

- The number of bits needed to identify the banks is k where 2^k = the number of memory banks.

- So, for 8-way interleaving, k=3 as $2^3 = 8$, so we need 3 bits.

- For 16-way interleaving, we need 4 bits.

Memory Organization

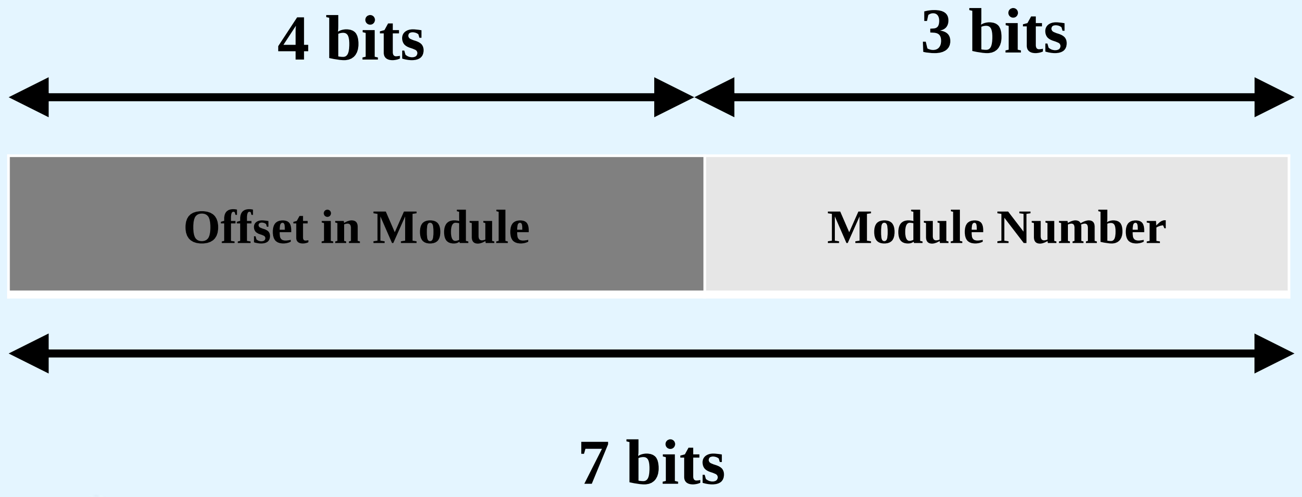
- So for n-way interleaving, we need $n = 2^k$ bits for addressing the memory banks.

EXAMPLE:

Suppose we have a 128 byte memory and we are using **8-way, low-order interleaving**. What is the structure of the addresses for this memory?

Memory Organization

- We have 128 bytes, so we need 7 bits for each individual address (128 – 2^7).
- The interleaving is low order, so the lowest-order bits are used for the memory bank address.
- We are using 8-way interleaving, so we have 8 memory banks (modules), so we need 3 bits to address all of them (8 = 2^3).



Memory Organization

- Note that each module must be of size 2^4.

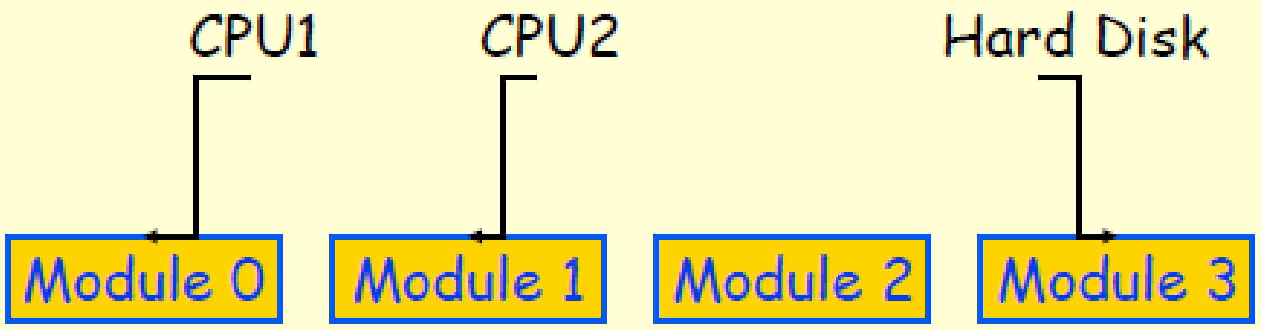
- This can be concluded in two ways:

1. We have 128 bytes and we have 8 modules, so each module will hold 16 bytes: ($128/8 = 2^7/2^3 = 2^4 = 16$).

2. We can also see from the address structure that the offset in the module requires 4 bits, allowing for $2^4 = 16$ bytes per module.

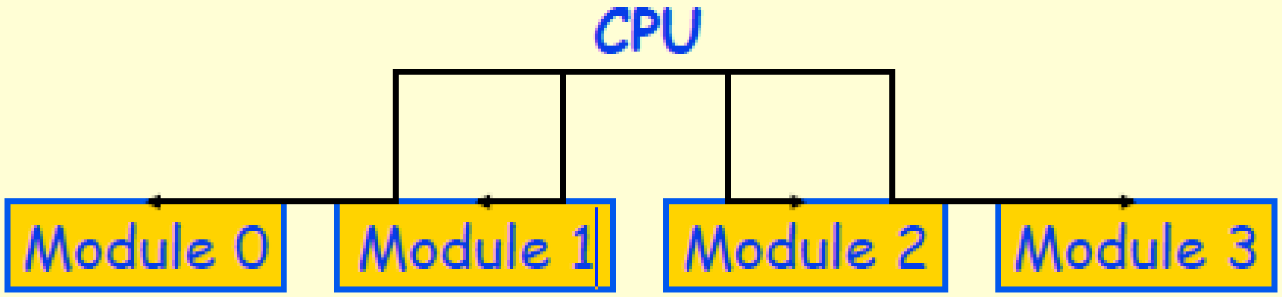
High-Order Interleave

- Good if Modules can be accessed independently by different units, e.g. by the CPU and a Hard Disk (or a second CPU) AND the units use different Modules
- => Parallel operation => Higher Performance



Low-Order Interleave

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



- Since adjacent memory locations lie in different Modules 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.

Memory Organization

Exercise 1:

For a 32K x 8 memory that uses **16-way, high-order interleaving**, find the location (chip and offset) of the following address:

001000000100111

Memory Organization

Answer:

32K memory by 16-way = 16 x 2K chips.

32K = 2^5 x 2^{10} = 2^{15} = 15 bits for all addresses.

16 = 2^4 = 4 bits for chip addresses.

15 – 4 = 11 bits for offset in each chip.

0010 00000100111

Chip: 2, Offset: 39.

Exercise 2: Repeat for 8-way, low-order interleaving.

Memory Organization

Answer:

32K memory by 8-way = 8 x 4K chips.

32K = 2^5 x 2^{10} = 2^{15} = 15 bits for all addresses.

18 = 2^3 = 3 bits for chip addresses.

15 – 3 = 12 bits for offset in each chip.

001000000100 111

Chip: 7, Offset: 516.