**4. How many bits would you need to address a 2M × 32 memory if**

**a) the memory is byte addressable?**

221 \* 22 = 223 use 23 bits.

**b) the memory is word addressable?**

221 use 21 bits.

**6. How many bits are required to address a 1M × 8 main memory if**

**a) main memory is byte addressable?**

220 use 20 bits.

**b) main memory is word addressable?**

220 use 20 bits.

**7. Redo Example 4.1 using high-order interleaving instead of low-order interleaving.**

Redo

Suppose we have a 128-word memory that is 8-way high-order interleaved, which means it uses 8 memory banks; 8 = 23, so we use the high-order 3 bits to identify the bank. Because we have 128 words, we need 7 bits for each address (128 = 27). Therefore, an address in this memory has the following structure:

high-order interleaving

|  |  |
| --- | --- |
| 3 bits | 4 bits |
| module | offset |

|  |  |  |  |
| --- | --- | --- | --- |
| Module | Address split per given structure | Module number | Offset in module |
| Module 0 | 000 0000 | 0 | 0 |
|  | 000 0001 | 0 | 1 |
|  | 000 0010 | 0 | 2 |
|  |  |  |  |
|  | 000 1111 | 0 | 15 |
| Module 1 | 001 0000 | 1 | 0 |
|  | 001 0001 | 1 | 1 |
|  | 001 0010 | 1 | 2 |
|  |  |  |  |
| Module 7 | 111 1111 | 7 | 15 |

**8. Suppose we have 4 memory modules instead of 8 in Figures 4.6 and 4.7.**

**Draw the memory modules with the addresses they contain using:**

1. **High-order interleaving**

|  |  |
| --- | --- |
| 2 bits | 2 bits |
| module | offset |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Module | Decimal word address | Binary address | Address split per given structure | Module number | Offset in module |
| Module 0 | 0 | 0000 | 00 00 | 0 | 0 |
|  | 1 | 0001 | 00 01 | 0 | 1 |
|  | 2 | 0010 | 00 10 | 0 | 2 |
|  | 3 | 0011 | 00 11 | 0 | 3 |
| Module 1 | 4 | 0100 | 01 00 | 1 | 0 |
|  | 5 | 0101 | 01 01 | 1 | 1 |
|  | 6 | 0110 | 01 10 | 1 | 2 |
|  | 7 | 0111 | 01 11 | 1 | 3 |
| Module 2 | 8 | 1000 | 10 00 | 2 | 0 |
|  | 9 | 1001 | 10 01 | 2 | 1 |
|  | 10 | 1010 | 10 10 | 2 | 2 |
|  | 11 | 1011 | 10 11 | 2 | 3 |
| Module 3 | 12 | 1100 | 11 00 | 3 | 0 |
|  | 13 | 1101 | 11 01 | 3 | 1 |
|  | 14 | 1110 | 11 10 | 3 | 2 |
|  | 15 | 1111 | 11 11 | 3 | 3 |

1. **Low-order interleaving**

|  |  |
| --- | --- |
| 2 bits | 2 bits |
| offset | module |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Module | Decimal word address | Binary address | Address split per given structure | Offset in module | Module number |
| Module 0 | 0 | 0000 | 00 00 | 0 | 0 |
|  | 4 | 0100 | 01 00 | 1 | 0 |
|  | 8 | 1000 | 00 00 | 2 | 0 |
|  | 12 | 1100 | 00 00 | 3 | 0 |
| Module 1 | 1 | 0000 | 00 01 | 0 | 1 |
|  | 5 | 0101 | 01 01 | 1 | 1 |
|  | 9 | 1001 | 10 01 | 2 | 1 |
|  | 13 | 1101 | 11 01 | 3 | 1 |
| Module 2 | 2 | 0010 | 00 10 | 0 | 2 |
|  | 6 | 0110 | 01 10 | 1 | 2 |
|  | 10 | 1010 | 10 10 | 2 | 2 |
|  | 14 | 1110 | 11 10 | 3 | 2 |
| Module 3 | 3 | 0011 | 00 11 | 0 | 3 |
|  | 7 | 0111 | 01 11 | 1 | 3 |
|  | 11 | 1011 | 10 11 | 2 | 3 |
|  | 15 | 1111 | 11 11 | 3 | 3 |

**10. Suppose that a 2M × 16 main memory is built using 256K × 8 RAM chips and memory is word addressable.**

**a) How many RAM chips are necessary?**

256K X 8 \* 4 = 512K X 16

512K X 16 \* 4 =2M X 16

RAM chips = 4\*4 = 16 chips

1. **If we were accessing one full word, how many chips would be involved?**

2 chips because 2 chips make you can access X 16

**c) How many address bits are needed for each RAM chip?**

256K X 8 = 28\*210 X 23

Bits per each RAM chips = 8+10+3 = 21

**d) How many banks will this memory have?**

256K = 28\*210

banks = 8+10 = 18 // banks per one chip

banks in memory = banks per chip \* N = 18 \* 16 = 288 banks

**e) How many address bits are needed for all memory?**

Bits per each RAM chips\* number of RAM chips = 21\*16 = 336 bits

**f) If high-order interleaving is used, where would address 14 (which is E in hex) be**

**located?**

14 located in module 0 offset 13

**g) Repeat exercise 9f for low-order interleaving.**

14 located in module 13 offset 0

**13. A digital computer has a memory unit with 24 bits per word. The instruction**

**set consists of 150 different operations. All instructions have an operation**

**code part (opcode) and an address part (allowing for only one address). Each**

**instruction is stored in one word of memory.**

1. **How many bits are needed for the opcode?**

150<28 = 8 bits

1. **How many bits are left for the address part of the instruction?**

24-8 = 16 bits

1. **What is the maximum allowable size for memory?**

216 is the maximum size

1. **What is the largest unsigned binary number that can be accommodated in**

**one word of memory?**

Largest unsigned is 224-1 = 33554430

**15. Assume a 220 byte memory.**

**a) What are the lowest and highest addresses if memory is byte addressable?**

Lowest address = 00000000000000000000

Highest address = 11111111111111111111

**b) What are the lowest and highest addresses if memory is word addressable,**

**assuming a 16-bit word?**

Lowest address = 000000000000000000

Highest address = 111111111111111111

**c) What are the lowest and highest addresses if memory is word addressable,**

**assuming a 32-bit word?**

Lowest address = 0000000000000000

Highest address = 1111111111111111

**17. You and a colleague are designing a brand new microprocessor architecture.**

**Your colleague wants the processor to support 509 different instructions. You**

**do not agree, and would like to have many fewer instructions. Outline the**

**argument for a position paper to present to the management team that will**

**make the final decision. Try to anticipate the argument that could be made to**

**support the opposing viewpoint.**

* In case that CPU has same bit CPU that has less instruction can access more memory.
* Less instruction use less bit to contain.
* Less instruction make BigO smaller than more instruction.
* Some instruction can use by combine with N instruction.