A 2KB Memory Has A Starting Address Of 2000h. Calculate The Final Address. Repeat For 4K And 16K Memories

A 2KB Memory Has A Starting Address Of 2000h. Calculate The Final Address. Repeat For 4K And 16K Memories

Understanding memory addressing is fundamental in computer architecture and embedded systems. Accurate calculations of memory addresses help in efficient memory management, addressing hardware requirements, and optimizing system performance. In this article, we'll explore how to determine the final address of different memory sizes—specifically 2KB, 4KB, and 16KB—when the starting address is given as 2000h. We will delve into the concepts of memory size, address calculation, and hexadecimal arithmetic to provide a clear and comprehensive understanding.

Understanding Memory Size and Addressing

Before diving into calculations, it’s essential to understand what memory size and addresses represent.

Memory Size

- Memory size indicates the total amount of storage available, usually expressed in bytes.
- Common memory sizes include 1KB (1024 bytes), 2KB (2048 bytes), 4KB (4096 bytes), 8KB, 16KB, and so on.
- The size determines the range of addresses that the memory can occupy.

Memory Addressing

- Each memory location is identified by a unique address.
- Addressing typically starts at a specific starting address, often 0000h in systems, but can be any address depending on the design.
- The addresses are represented in hexadecimal (base 16), which makes handling large numbers more manageable.

Calculating Final Address for Different Memory Sizes

Given the starting address and memory size, the final address indicates the last memory location occupied by the memory block.

The general formula:
$$ \text{Final Address} = \text{Starting Address} + \text{Memory Size} - 1 $$

Since memory addresses are zero-based, the last address is one less than the total size added to the starting address.

Let's apply this formula to each case.

Case 1: 2KB Memory Starting at 2000h

Step 1: Convert Memory Size to Bytes
- 2KB = 2 × 1024 = 2048 bytes

Step 2: Convert Starting Address to Decimal
- Starting address: 2000h
- 2000h in decimal:
- 2 × 16^3 + 0 × 16^2 + 0 × 16^1 + 0 × 16^0
- 2 × 4096 = 8192
- So, 2000h = 8192 decimal

Step 3: Calculate Final Address in Decimal
- Final address = 8192 + 2048 - 1 = 8192 + 2047 = 10239

Step 4: Convert Final Address Back to Hex
- 10239 decimal to hexadecimal:
- 10239 ÷ 16 = 639 remainder 15 (F)
- 639 ÷ 16 = 39 remainder 15 (F)
- 39 ÷ 16 = 2 remainder 7
- 2 ÷ 16 = 0 remainder 2

Reading remainders from last to first: 2 7 F F

Thus, 10239 decimal = 0x27FF

Final Address: 27FFh

Summary for 2KB Memory

- Starting Address: 2000h
- Memory Size: 2048 bytes
- Final Address: 27FFh

Case 2: 4KB Memory Starting at 2000h

Step 1: Convert Memory Size to Bytes
- 4KB = 4 × 1024 = 4096 bytes

Step 2: Starting Address in Decimal
- 2000h = 8192 decimal (as above)

Step 3: Calculate Final Address in Decimal
- Final address = 8192 + 4096 - 1 = 8192 + 4095 = 12287

Step 4: Convert 12287 to Hex
- 12287 ÷ 16 = 768 remainder 15 (F)
- 768 ÷ 16 = 48 remainder 0
- 48 ÷ 16 = 3 remainder 0
- 3 ÷ 16 = 0 remainder 3

Reading remainders from last to first: 3 0 0 F

Therefore, 12287 decimal = 0x300F

Final Address: 300Fh

Summary for 4KB Memory

- Starting Address: 2000h
- Memory Size: 4096 bytes
- Final Address: 300Fh

Case 3: 16KB Memory Starting at 2000h

Step 1: Convert Memory Size to Bytes
- 16KB = 16 × 1024 = 16384 bytes

Step 2: Starting Address in Decimal
- 2000h = 8192 decimal

Step 3: Calculate Final Address in Decimal
- Final address = 8192 + 16384 - 1 = 8192 + 16383 = 24575

Step 4: Convert 24575 to Hex
- 24575 ÷ 16 = 1535 remainder 15 (F)
- 1535 ÷ 16 = 95 remainder 15 (F)
- 95 ÷ 16 = 5 remainder 15 (F)
- 5 ÷ 16 = 0 remainder 5

Reading remainders from last to first: 5 F F F

Thus, 24575 decimal = 0x5FFF

Final Address: 5FFFh

Summary for 16KB Memory

- Starting Address: 2000h
- Memory Size: 16384 bytes
- Final Address: 5FFFh

Visual Summary of Address Calculations

Memory Size	Starting Address (Hex)	Starting Address (Decimal)	Final Address (Hex)	Final Address (Decimal)
2KB	2000h	8192	27FFh	10239
4KB	2000h	8192	300Fh	12287
16KB	2000h	8192	5FFFh	24575

Important Points to Remember

- The calculation for the final address always subtracts 1 from the total size added to the starting address because addresses are zero-based.
- Hexadecimal conversions require understanding of base-16 arithmetic and remainders.
- Memory sizes are typically expressed in powers of two, making binary and hexadecimal calculations straightforward.
- The starting address can be any valid hexadecimal address, and the methodology remains the same regardless of the starting point.

Practical Applications of Address Calculations

- These calculations are crucial in low-level programming, such as assembly language, embedded system development, and operating system memory management.
- Accurate address calculation helps prevent memory overlaps and ensures data integrity.
- In hardware design, understanding address ranges assists in configuring memory modules, addressing decoding, and designing memory maps.

Conclusion

Calculating the final address of a memory block given its size and starting address involves straightforward hexadecimal arithmetic and understanding of memory addressing principles. For a memory starting at 2000h:
- The 2KB memory ends at 27FFh.
- The 4KB memory ends at 300Fh.
- The 16KB memory ends at 5FFFh.

Mastering these calculations enhances one's capability to work effectively with memory management in various computing contexts, ensuring efficient utilization of hardware resources and robust system design.

Frequently Asked Questions

What is the starting address of a 2KB memory device?

The starting address of the 2KB memory is 2000h.

How do you calculate the final address of a 2KB memory starting at 2000h?

Add the size of the memory in bytes minus one to the starting address: 2000h + (2KB - 1) = 2000h + 07FFh = 27FFh.

What is the final address of a 2KB memory starting at 2000h?

The final address is 27FFh.

How do you determine the final address for a 4KB memory starting at 2000h?

Add 4KB - 1 to the starting address: 2000h + (4KB - 1) = 2000h + 0FFFh = 2FFFh.

What is the final address for a 4KB memory starting at 2000h?

The final address is 2FFFh.

How do you find the final address of a 16KB memory starting from 2000h?

Add 16KB - 1 to the starting address: 2000h + 3FFFh = 5FFFh.

What is the final address of a 16KB memory starting at 2000h?

The final address is 5FFFh.

Why do we subtract one when calculating the final address of memory ranges?

Because the starting address is inclusive; subtracting one ensures the address range covers exactly the memory size without overlapping.

Can the same calculation method be used for different memory sizes?

Yes, the method involves adding the size minus one to the starting address, regardless of memory size.

What is the general formula to find the last address of a memory segment?

Final Address = Starting Address + (Memory Size in bytes) - 1.

Additional Resources

A 2KB Memory Has A Starting Address Of 2000h. Calculate The Final Address. Repeat For 4K And 16K Memories

---

Introduction

In the realm of computer architecture and digital systems, understanding how memory addresses are allocated and calculated is fundamental. Memory addresses serve as unique identifiers for data storage locations within a computer's memory module. When designing or analyzing systems, engineers often need to determine the range of addresses occupied by memory blocks of various sizes. This task becomes particularly straightforward when the starting address is known, and the size of the memory is given.

In this article, we will explore how to calculate the final address for different memory sizes—specifically 2KB, 4KB, and 16KB—when the starting address is fixed at 2000h (which is hexadecimal notation). We will delve into the underlying concepts of memory addressing, conversions between hexadecimal and decimal systems, and the step-by-step process for calculating the ending addresses. This knowledge is crucial for system designers, programmers, and hardware engineers who work with memory mapping and address decoding.

Understanding Memory Size and Addressing

Before diving into calculations, it's essential to clarify some basic concepts:

Memory Size

Memory size refers to the total amount of storage space allocated for a memory block, commonly expressed in bytes. For example:

- 2KB (kilobytes) = 2 1024 bytes = 2048 bytes
- 4KB = 4 1024 bytes = 4096 bytes

- 16KB = 16 1024 bytes = 16384 bytes

Starting Address

The starting address indicates the first memory location of the block. In our case, it is given as 2000h, where 'h' denotes hexadecimal notation. In decimal, this is:

- 2000h = (2 16^3) + (0 16^2) + (0 16^1) + (0 16^0)
- Calculation: (2 4096) + 0 + 0 + 0 = 8192 decimal

Address Range

Memory addresses are sequential. If the starting address is known, and the size of the memory block is specified, the final address (or ending address) can be calculated by adding the size (minus one, because addresses are inclusive) to the starting address.

The Methodology for Calculating Final Addresses

The core idea is:

Final Address = Starting Address + (Memory Size in Bytes) - 1

This formula accounts for the fact that addresses are inclusive, meaning the first byte is at the starting address, and the last byte is at the final address.

Calculations for Different Memory Sizes

1. For 2KB Memory

Step 1: Convert Memory Size to Bytes

- 2KB = 2 1024 bytes = 2048 bytes

Step 2: Convert Starting Address to Decimal

- Starting address: 2000h = 8192 decimal

Step 3: Calculate Final Address in Decimal

- Final address decimal = 8192 + 2048 - 1 = 10239

Step 4: Convert Final Address back to Hexadecimal

- 10239 decimal to hex:

- 10239 / 16 = 639, remainder 15 (F)
- 639 / 16 = 39, remainder 15 (F)

- 39 / 16 = 2, remainder 7
- 2 / 16 = 0, remainder 2

Reading remainders from last to first: 2 7 F F

- Final address: 27FFh

Result

The final address for 2KB memory starting at 2000h is 27FFh.

---

2. For 4KB Memory

Step 1: Convert Memory Size to Bytes

- 4KB = 4 1024 = 4096 bytes

Step 2: Starting Address in Decimal

- 2000h = 8192 decimal (as established)

Step 3: Calculate Final Address in Decimal

- Final address = 8192 + 4096 - 1 = 12287

Step 4: Convert 12287 decimal to hexadecimal

- 12287 / 16 = 768, remainder 15 (F)
- 768 / 16 = 48, remainder 0
- 48 / 16 = 3, remainder 0
- 3 / 16 = 0, remainder 3

Reading remainders from last to first: 3 0 0 F

- Final address: 300Fh

Result

The final address for 4KB memory starting at 2000h is 300Fh.

---

3. For 16KB Memory

Step 1: Convert Memory Size to Bytes

- 16KB = 16 1024 = 16384 bytes

Step 2: Starting Address in Decimal

- 2000h = 8192 decimal

Step 3: Calculate Final Address in Decimal

- Final address = 8192 + 16384 - 1 = 24575

Step 4: Convert 24575 decimal to hexadecimal

- 24575 / 16 = 1534, remainder 1
- 1534 / 16 = 95, remainder 14 (E)
- 95 / 16 = 5, remainder 15 (F)
- 5 / 16 = 0, remainder 5

Reading from last to first: 5 F E 1

- Final address: 5FE1h

Result

The final address for 16KB memory starting at 2000h is 5FE1h.

Summary of Results

Memory Size	Starting Address	Final Address
2KB	2000h	27FFh
4KB	2000h	300Fh
16KB	2000h	5FE1h

Practical Implications and Usage

Understanding these calculations is vital in multiple contexts:

- Memory Mapping: When designing hardware or firmware, engineers need to define memory regions with precise start and end addresses to prevent overlaps and ensure efficient utilization.
- Address Decoding: Properly calculating final addresses helps in setting up address decoding logic for memory chips, buses, or peripherals.
- Programming: Developers working close to hardware, such as embedded systems programmers, often need to calculate memory ranges for data storage, buffers, or registers.
- System Debugging: When diagnosing memory-related issues, knowing the address ranges aids in pinpointing problematic memory accesses.

Additional Considerations

While the above calculations are straightforward, some nuances should be kept in mind:

- Addressing Modes: Some systems may use segmented or paged memory models, impacting how addresses are interpreted.
- Memory Alignment: Certain hardware architectures require memory blocks to be aligned to specific address boundaries.
- Address Wrapping: In some cases, address calculations may wrap around if the address space is limited, such as in 16-bit address buses.

Conclusion

Calculating the final address of a memory block given its size and starting address is a fundamental skill in digital system design and programming. By understanding how to convert between hexadecimal and decimal systems and applying simple addition, engineers can accurately determine memory ranges.

In our example, starting with a fixed address of 2000h, we found that:

- A 2KB memory extends up to 27FFh.
- A 4KB memory extends up to 300Fh.
- A 16KB memory extends up to 5FE1h.

Mastery of these calculations not only ensures correct system design but also facilitates troubleshooting and optimization in complex computing environments. As digital systems continue to grow in complexity, such foundational knowledge remains as relevant as ever.

A 2kb Memory Has A Starting Address Of 2000h Calculate The Final Address Repeat For 4k And 16k Memories

A 2kb Memory Has A Starting Address Of 2000h Calculate The Final Address Repeat For 4k And 16k Memories

Related Articles

- As The Hydraulic Cylinder Elongates, It Raises Pin B Of The Mechanism. When The System Is In The Position
- A Particle Executes Linear Simple Harmonic Motion Of Amplitude A .what Is The Ratio Of The Kinetic Energy
- A Linear Function Goes Through The Points (-3,1) And (1,-3). What Are The Coordinates For The Y-intercept

Back to Home