**NUMBER SYSTEMS - CONVERSIONS**

**Base – 2 (Binary) numbers**

* uses digits **0 and 1**. There is **no single digit for the value two**. It **requires two symbols**.

**Base – 10 (Decimal) numbers**

* uses digits **0 – 9**. There is **no single digit for the value ten**. It **requires two symbols**.

**Base – 16 (Hexadecimal) numbers**

* the same pattern is true for hex. We need to use a **single character** for the values from **0 – 15**.
* uses digits **0 – 9** as with Base-10, but…what **single character** will we use for 10, 11, 12, 13, 14, and 15?
* the simple solution was to use the alphabet letters **A, B, C, D, E, and F** (lower case letters are also used)
* A = 10, B = 11, C = 12, D = 13, E = 14, and F = 15.

So here is how you count in Base -10 and Base -16:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | **31** | **32** | **33** | **34** | **DEC** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **A** | **B** | **C** | **D** | **E** | **F** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **1A** | **1B** | **1C** | **1D** | **1E** | **1F** | **20** | **21** | **22** | **HEX** |

**Converting a Base- 16 number to our familiar Base -10** (We will use only **two-digit hex numbers** in this course.)

**Example 1: Convert 4D to base – 10**.

Method:

* Multiply the first hex digit by 16 and then add on the second hex digit.
* 4 x 16 = 64
* D = 13
* 64 + 13 = **77** in base – 10

**Example 2: Convert CA to base – 10**.

Method:

* Multiply the first hex digit by 16 and then add on the second hex digit.
* C = 12. 12 x 16 = 192
* A = 10
* 192 + 10 = **202** in base – 10.

**Example 3: Convert FF to base – 10**.

Method:

* Multiply the first hex digit by 16 and then add on the second hex digit.
* F = 15. 15 x 16 = 240
* F = 15
* 240 + 15 = **255** in base – 10. **NOTE**: This is the highest number possible with a two-digit hex number.

**Converting a Base – 10 number to Base – 16** (Limited to numbers **up to 255** as explained above.)

**Example 1: Convert 77 in base – 10 to hexadecimal**.

Method:

* Divide the base-10 number by 16 to get the **first hex digit**. The **remainder** is the second hex digit.
* 77 divided by 16 equals **4**, with remainder **13**.
* But **13 is D** in hex, so the answer is **4D**.

**Example 2: Convert 203 in base – 10 to hexadecimal**.

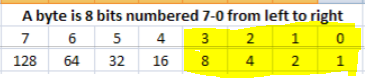
Method:

* Divide the base-10 number by 16 to get the **first hex digit**. The **remainder** is the second hex digit.
* 203 divided by 16 equals **12**, with remainder **11**.
* But **12 is C** in hex, and **11 is B** in hex, so the answer is **CB**.

**Converting a Base – 16 number to Base – 2, Binary**

**Example 1: Convert 4D to binary**.

Method:

* Each of the two hex digits can be replaced by an **equivalent** **4-bit binary number (a nybble)**.
* Refer to the yellow part of the graphic shown here to determine the equivalent 4-bit numbers.
* Putting the nybbles together the answer is **0100 : 1101**

|  |  |  |
| --- | --- | --- |
| **Hexadecimal** | **4** | **D** = 13 |
| **Binary** | **0100** | **1101** |

* NOTE: any number of **leading zeroes can be dropped**, so **1001101** is also correct.

**Example 2: Convert hex E8 to Base – 2, Binary**

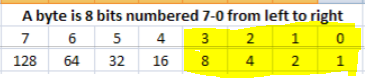
* Putting the nybbles together the answer is **1110 : 1000**

|  |  |  |
| --- | --- | --- |
| **Hexadecimal** | **E** = 14 | **8** |
| **Binary** | **1110** | **1000** |

**Converting from Binary to Hexadecimal**

**Example 1: Convert 101110 to Hexadecimal**

Method:

* Break the binary number into **4-bit nybbles starting from the right end**.
* 101110 becomes 10 **:** 1110, or 0010 : 1110
* Determine the base – 10 value of each nybble, then replace each with the equivalent hex digit.

|  |  |  |
| --- | --- | --- |
| **Binary** | **0010** | **1110** = 14 |
| **Hexadecimal** | **2** | **E** |

* So the answer is **2E**.

**Converting from Base-10 to Binary**

**Example 1: Convert 101 in Base-10 to Binary**

Method:

* **Starting from Bit 7**, turn on bits until the **running total is 101**.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit Number** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |  |
| **Bit Value If ON** | **128** | **64** | **32** | **16** | **8** | **4** | **2** | **1** |  |
|  | **0** | **1** | **1** | **0** | **0** | **1** | **0** | **1** | **64 + 32 + 4 + 1 = 101** |

So the answer is **01100101**, or **1100101** since leading zeros can be dropped.

**Example 2: Convert 221 in Base-10 to Binary**

Method:

* **Starting from Bit 7**, turn on bits until the **running total is 221**.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit Number** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |  |
| **Bit Value If ON** | **128** | **64** | **32** | **16** | **8** | **4** | **2** | **1** |  |
|  | **1** | **1** | **0** | **1** | **1** | **1** | **0** | **1** | **128 + 64 + 16 + 8 + 4 + 1 = 221** |

So the answer is **01100101**, or **1100101** since leading zeros can be dropped.

**Converting from Binary to Base-10**

**Example 1: Convert 1010101 in Binary to Base-10**

Method:

* Add up the bit values of the ON bits. NOTE: if less than 8 bits, **fill from the right end**.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bit Number** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |  |
| **Bit Value If ON** | **128** | **64** | **32** | **16** | **8** | **4** | **2** | **1** | **64 + 16 + 4 + 1 = 85** |
|  |  | **1** | **0** | **1** | **0** | **1** | **0** | **1** |  |

So the answer is **85** in Base-10