# CSIT127 – Binary and Hexadecimal Numbers

## Part A: Binary (base 2)

People prefer to count in lots of ten (base 10)

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, -> 99 100, 101 -> 999 1000

Computers work with 0s and 1s (binary numbers)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Decimal** |  | **Binary** |  |  |  |  |  |  |
| 10 | = | 1 | 0 | 1 | 0 |  | 8+2 = 10 |  |
| 7 | = | 0 | 1 | 1 | 1 |  | 4+2+1=7 |  |
|  |  | **8** | **4** | **2** | **1** | ⇐ | Binary | Weightings |

1. Complete the following 4-bit numbers

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 23 | 22 | 21 | 20 |  |  |  |  |  |
| **8** | **4** | **2** | **1** |  |  |  |  |  |
| 0 | 0 | 0 | 0 |  | 0 |  |  |  |
| 0 | 0 | 0 | 1 |  | 1 |  |  |  |
| 0 | 0 | 1 | 0 |  | 2 |  |  |  |
| 0 | 0 | 1 | 1 |  | 3 |  |  |  |
| 0 | 1 | 0 | 0 |  | 4 |  |  |  |
| \_\_\_0\_\_ | \_\_\_1\_\_ | \_\_0\_\_\_ | \_\_1\_\_\_ |  | 5 |  |  |  |
| \_\_\_0\_\_ | \_\_\_1\_\_ | \_\_1\_\_\_ | \_\_0\_\_\_ |  | 6 |  |  |  |
| \_\_\_0\_\_ | \_\_\_1\_\_ | \_\_1\_\_\_ | \_\_1\_\_\_ |  | 7 |  |  |  |
| \_\_\_\_\_ | \_\_\_\_\_ | \_\_\_\_\_ | \_\_\_\_\_ |  | 8 |  |  |  |

2. The ASCII system uses 7 bit numbers. What is the maximum number of characters it can represent?

0000000 -> 1111111

128

3.The EBCDIC system uses 8 bit numbers. What is the maximum number of characters it can represent?

00000000 -> 11111111

256

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 |
| 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |

## Part B: Hexadecimal (base 16) – A compromise with binary for human readability

If we group every 4 bits together we create a number range 0 – 15. This number range can be represented between 0-9 and then we add A. B. C. D. E. F

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Decimal** |  | **Binary** | |  |  |  | **Hexadecimal** |  |
| 0 | = | 0 | 0 | 0 | 0 | = | 0 |  |
| 1 | = | 0 | 0 | 0 | 1 | = | 1 |  |
| 2 | = | 0 | 0 | 1 | 0 | = | 2 |  |
| 3 | = | 0 | 0 | 1 | 1 | = | 3 |  |
| 4 | = | 0 | 1 | 0 | 0 | = | 4 |  |
| 5 | = | 0 | 1 | 0 | 1 | = | 5 |  |
| 6 | = | 0 | 1 | 1 | 0 | = | 6 |  |
| 7 | = | 0 | 1 | 1 | 1 | = | 7 |  |
| 8 | = | 1 | 0 | 0 | 0 | = | 8 |  |
| 9 | = | 1 | 0 | 0 | 1 | = | 9 |  |
| 10 | = | 1 | 0 | 1 | 0 | = | A |  |
| 11 | = | 1 | 0 | 1 | 1 | = | B |  |
| 12 | = | 1 | 1 | 0 | 0 | = | C |  |
| 13 | = | 1 | 1 | 0 | 1 | = | D |  |
| 14 | = | 1 | 1 | 1 | 0 | = | E |  |
| 15 | = | 1 | 1 | 1 | 1 | = | F |  |
|  |  |  |  |  |  |  |  |  |

So we can represent a typical 8-bit byte as two numbers

e.g.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | = | A5 |

Write the following 8-bit bytes in hexadecimal

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | = 26 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | = 84 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | = B1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | = 8D |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | = FF |