

### Converting

#### From Hexadecimal to decimal

$$BF = B \cdot 16^1 + F \cdot 16^0 = 11 \cdot 16^1 + 15 \cdot 16^0 = 171 + 15 = 191$$

#### From Binary to Hexadecimal

1) Find the biggest power of 16 that fits in - number of positions

$$3501 - 16, 256, \del{4096}$$

2) Find the biggest multiplier

$$x \cdot 256, \text{ that fits? } x = 13 = D$$

$$3501 - 13 \cdot 256 = 173$$

$$Y \cdot 16 < 173, 10 \cdot 16 = A \cdot 16$$

$$173 - 160 = 13, 13 = D$$

$$3501(2) = DAD(16)$$

#### From Binary to decimal

Just multiplying 1 or 0 by 2 in the power of position

$$10100 = 1 \cdot 2^4 + 1 \cdot 2^2 = 20$$

#### From Decimal to binary

1) find the biggest power of 2 that fits - number of positions

$$63 = 2^5$$

2) Find the biggest multiplier

$$63 - 32 = 31, 2^4$$

$$31 - 16 = 15, 2^3$$

$$15 - 8 = 7, 2^2$$

$$7 - 4 = 3, 2^1$$

$$3 - 2 = 1, 2^0$$

$$63(10) = 111111(2)$$

#### Hexadecimal to binary

1) Each number should represent the four digits in the binary number

BF => 8 digits

B=11, 11=1011

F=15, 15=1111

BF(16)=10111111(2)

#### Decimal to hexadecimal

$$191 : 16 = 11.R \Rightarrow R \cdot 16 = .9375 + 16 = 15$$

if the number before is less than 16, then it is the first digit, if not, then keep counting

11 => B

BF, reading backwards

### Or gate

#### 2 - input OR gate



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

### X() gates

### File Processing

These are XAND and XOR gates, they are exclusive

So for XAND only if both inputs are identical it will give a 1

For XOR only if they are both different, it will give a 1

### And gate

2 - input AND gate



A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

### N() gates

The gates with N, such as NOR, NAND, XNOR, NOR, XNAND, are just gates with a NOT after them

### Not gate



A	Q
0	1
1	0

The screenshot shows a Python tutorial page with several sections:

- Reading Files:** A code block showing how to open a file for reading and return its object, read its contents as a string, and iterate over the object to get the next string.
- File Input Template:** A code block showing a template for reading files in Python, including a function definition and a main block.
- Exercise:** A code block showing a function that accepts a file name as a parameter and returns the contents of the file.
- Exercise Solution:** A code block showing the solution to the exercise, including a function definition and a main block.
- Recall: String Methods:** A table listing various string methods and their descriptions.