# Topics for today

- Number Bases
- Binary and hexadecimal
- Conversion between bases
- Hex colors and rgb colors
- Counting in different bases
- Bit shifting
- Addition of binary numbers

# Base

- Number of digits that the counting system or number system uses
- For example, we (humans) use decimal number system which has 10 digits (0-9), so, the base of decimal number system is 10
- All other numbers are combinations of these 10 digits.
- When we run out of digits in one place, we move to the next place.
- ..., 6, 7, 8, 9, 10, 11, ... 19, 20, 21, ... 29, ... 99, 100, 101, ...

# Binary

- Base 2 number system.
- Only 2 digits - 0 and 1
- Used by electronic system at the most basic level
- 1 - electricity present. 0 - no electricity
- 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, ...
- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

# Place values - decimal

- Result of the multiplication of digit and ‘it’s place’ in the number system.
- In decimal, every place will have some powers of 10.
- 7345 = 7 * 1000 + 3 * 100 + 4 * 10 + 5 * 1
- $= 7 * 10^3 + 3 * 10^2 + 4 * 10^1 + 5 * 10^0$
- So, the place value of ‘3’ in ‘7345’ will be 3 * 100 = 300

# Place values - binary

- Result of the multiplication of digit and ‘it’s place’ in the number system.
- In binary, every place will have some powers of 2.
- 1011
- $1 * 2^3 \quad 0 * 2^2 \quad 1 * 2^1 \quad 1 * 2^0$
- So, the place value of leftmost ‘1’ in ‘1011’ will be $1 * 2^3 = 8$ (in decimal)

# Converting from binary to decimal

- Calculate place value of each digit and add them
- $11001_2 = ?_{10}$
- $= 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0$
- $= 1 * 16 + 1 * 8 + 0 * 4 + 0 * 2 + 1 * 1$
- $= 25_{10}$
- Note: In binary, even numbers end in 0 and odd numbers end in 1

# Converting from decimal to binary

- Divide by base (2) and track the reminders from below
- $23_{10} = ?_2$
- 23 / 2 = 11 R = 1
- 11 / 2 = 5 R = 1
- 5 / 2 = 2 R = 1
- 2 / 2 = 1 R = 0
- So, now from below, starting with the quotient ‘1’
- $= 10111_2$

# Hexadecimal

- Base 16 number system.
- 16 digits- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A,  B,  C,  D,  E,  F
- Decimal- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
- Compact representation for binary numbers
- Group of 4 binary digits to each hex digits
- 0, 1, 2, ... 9, A, ... F, 10, 11, 12, ... 19, 1A, 1B, ... 1F, 20
- 0, 1,  2,    9, 10,...15, 16, 17, 18,... 25, 26, 27,...31, 32

# Converting from hexadecimal to decimal

- Calculate place value of each digit and add them
- Sometimes, prefixed by ‘0x’. ‘0x12’ is hexadecimal 12, not decimal
- $C3F_{16} = 0xC3F = ?_{10}$
- $= C * 16^2 + 3 * 16^1 + F * 16^0$
- $= C * 256 + 3 * 16 + F * 1$
- $= 12 * 256 + 3 * 16 + 15 * 1$
- $= 3135_{10}$

# Converting from decimal to hexadecimal

- Divide by base (16) and track the reminders from below
- $1354_{10} = ?_{16}$
- 1354 / 16 = 84 R = 10 = A
- 84 / 16 = 5 R = 4
- So, now from below, starting with the quotient ‘5’
- $= 54A_{16}$

# Hexadecimal colour

- hash (#) followed by 6 hex digits. #RRGGBB
- Convert to rgb format
- #1F256A
- $1F_{16}$ = decimal = $31_{10}$
- $25_{16}$ = decimal = $37_{10}$
- $6A_{16}$ = decimal = $106_{10}$
- = rgb(31, 37, 106)

# Hex and binary

- groups of 4 binary digits represent one hex digit

| Hex | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Binary | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |

- So, 4 binary digits can be converted directly to 1 hex digit
- And 1 hex digit can be converted directly to 4 binary digits

# Hex and binary

- $1010111001010 1_2 = ?_{16}$
- 0010 1011 1001 0101
- 2 B 9 5
- $= 2B95_{16} = 0x2B95$

- $F12C_{16} = ?_2$
- F 1 2 C
- 1111 0001 0010 1100
- $= 1111\ 0001\ 0010\ 1100_2$

# Convert from one base to another

- Convert to decimal and then into the target base
- $872_9 = ?_{12}$
- $872_9$ to decimal
- $8 * 9^2 + 7 * 9^1 + 2 * 9^0 = 8 * 81 + 7 * 9 + 2 * 1 = 648 + 63 + 2 = 713_{10}$
- $713_{10}$ to base 12
- 713 / 12 = 59 R = 5
- 59 / 12 = 4 R = 11 = B
- $= 4B5_{12}$

# Counting

- Decimal - 0, 1, 2,...9, 10, 11, ...99, 100, 101, ...
- Hexadecimal - 0, 1, 2,...9, A, B,...F, 10, 11,...
- Base 13 - 0, 1, 2,...9,A, B, C, 10, 11,...19, 1A, 1B, 1C, 20
- Base 4 - 0, 1, 2, 3, 10, 11, 12, 13, 20, 21, 22, 23, 30, 31, 32, 33, 100, 101, ...

# Bit Shifting

- Shifting of the digits in binary (used in registers)
- Right Shift: 11010 >> 2 (Drop the right-most 2 digits)
- $= 110_2 = 6_{10}$
- Left Shift: 100 << 1 (Add 1 zero to the end)
- $= 1000_2 = 8_{10}$
- If you have any number in other base, convert to binary and do the shifting
- $45_{10} >> 3$ (Convert to binary)
- $= 101101_2 >> 3$
- $= 101_2 = 5_{10}$

# Addition of binary numbers

- Let's revise the decimal addition

```
   111
   2671
 + 2539
  --------
   5210
```

- Now, turn for binary addition

```
   1111
  100111
 +  1011
  ------
  110010
```

# Demo