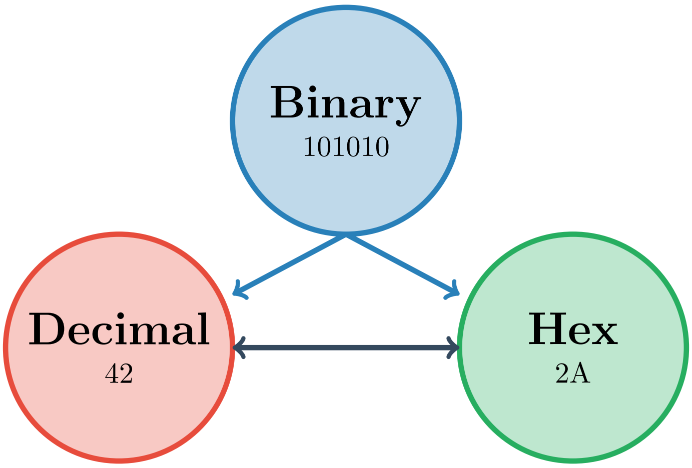
Number Systems

Understanding Binary, Decimal, & Hexadecimal

A Visual Guide for Computer Scientists

APCSA - Lesson 7



"There are only 10 types of people in the world:
those who understand binary, and those who don't."

Contents

1 Introduction to Number Systems **3**
1.1 Why Do We Need Different Number Systems? 3

2 Decimal (Base 10) **4**
2.1 Place Value in Decimal 4

3 Binary (Base 2) **5**
3.1 Why Binary? 5
3.2 Binary Place Values - MEMORIZE THIS! 5
3.3 Binary Example 5

4 Hexadecimal (Base 16) **6**
4.1 Hexadecimal Digits 6
4.2 Why Hexadecimal? 6
4.3 The Magic Relationship 6
4.4 Hexadecimal Example 6

5 Conversion Methods **7**
5.1 Decimal to Binary 7
5.2 Binary to Decimal 8
5.3 Binary to Hexadecimal (THE EASY WAY!) 9
5.4 4-Bit to Hex Conversion Table 10

6 Quick Reference Guide **11**
6.1 Powers to Memorize 11

7 Practice Problems **12**
7.1 Warm-up Exercises 12
7.2 Challenge Problems 12

8 Real-World Applications **13**
8.1 RGB Colors 13
8.2 Computer Memory 13
8.3 IP Addresses 13

9 Test Preparation **14**
9.1 Checklist Before the Exam 14
9.2 Common Mistakes to Avoid 14
9.3 Speed Tips for the Exam 14

10 Summary **15**

1 Introduction to Number Systems

A **number system** is a way of representing numbers using a specific set of symbols. The **base** (or radix) of a number system determines how many unique digits are available.

1.1 Why Do We Need Different Number Systems?

- **Humans** naturally use **decimal** (base 10) - we have 10 fingers!
- **Computers** use **binary** (base 2) - electricity is ON or OFF
- **Programmers** use **hexadecimal** (base 16) - compact representation of binary

System	Base	Digits	Example
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	42_{10}
Binary	2	0, 1	101010_2
Hexadecimal	16	0-9, A-F	$2A_{16}$

Table 1: Common Number Systems in Computing

2 Decimal (Base 10)

Base 10 - Decimal

The number system we use every day! Each position represents a power of 10.

2.1 Place Value in Decimal

Thousands	Hundreds	Tens	Ones
10^3	10^2	10^1	10^0
1000	100	10	1
2	**4**	**7**	**3**

Example

Calculate 2473_{10}:

$$\begin{aligned}
2473_{10} &= (2 \times 10^3) + (4 \times 10^2) + (7 \times 10^1) + (3 \times 10^0) \\
&= (2 \times 1000) + (4 \times 100) + (7 \times 10) + (3 \times 1) \\
&= 2000 + 400 + 70 + 3 \\
&= \boxed{2473}
\end{aligned}$$

3 Binary (Base 2)

Base 2 - Binary

The language of computers! Only uses 0 and 1. Each position represents a power of 2.

3.1 Why Binary?

- **Simple:** Only two states - ON (1) or OFF (0)
- **Reliable:** Easy to distinguish between two states
- **Fast:** Electronic circuits can switch quickly
- **Foundation:** All digital computing is built on binary

3.2 Binary Place Values - MEMORIZE THIS!

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	**2**	**4**	**8**	**16**	**32**	**64**	**128**

Pro Tip

Pattern: Each position is *double* the previous!
$1 \to 2 \to 4 \to 8 \to 16 \to 32 \to 64 \to 128 \to 256 \to 512 \to 1024$

3.3 Binary Example

Example

Convert 10101010_2 **to decimal:**

1	**0**	**1**	**0**	**1**	**0**	**1**	**0**
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	0	32	0	8	0	2	0

$$\begin{aligned} 10101010_2 &= (1 \times 128) + (0 \times 64) + (1 \times 32) + (0 \times 16) \\ &\quad + (1 \times 8) + (0 \times 4) + (1 \times 2) + (0 \times 1) \\ &= 128 + 0 + 32 + 0 + 8 + 0 + 2 + 0 \\ &= \boxed{170_{10}} \end{aligned}$$

4 Hexadecimal (Base 16)

Base 16 - Hexadecimal

A programmer's shortcut! Uses 16 digits: 0-9 and A-F. Provides compact representation of binary.

4.1 Hexadecimal Digits

Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Dec	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Pro Tip

Mnemonic: "A Boy Can Dance Every Friday"
A=10, B=11, C=12, D=13, E=14, F=15

4.2 Why Hexadecimal?

- **Compact:** One hex digit = four binary digits
- **Readable:** Easier for humans than long binary strings
- **Common:** Used in colors (`#FF0000`), memory addresses, etc.

4.3 The Magic Relationship

4 Binary Digits = 1 Hex Digit

This is the KEY to easy conversions!

4.4 Hexadecimal Example

Example

Convert $2AF3_{16}$ to decimal:

First, convert letters: $A = 10, \quad F = 15$

$$\begin{aligned} 2AF3_{16} &= (2 \times 16^3) + (10 \times 16^2) + (15 \times 16^1) + (3 \times 16^0) \\ &= (2 \times 4096) + (10 \times 256) + (15 \times 16) + (3 \times 1) \\ &= 8192 + 2560 + 240 + 3 \\ &= \boxed{10995_{10}} \end{aligned}$$

5 Conversion Methods

5.1 Decimal to Binary

Base 2 - Binary

Algorithm: Repeated Division by 2

1. Divide the number by 2
2. Write down the remainder (0 or 1)
3. Divide the quotient by 2
4. Repeat until quotient is 0
5. Read remainders from **bottom to top**

Example

Convert 42_{10} to binary:

$$
\begin{aligned}
42 \div 2 &= 21 \quad \text{remainder } \mathbf{0}\\
21 \div 2 &= 10 \quad \text{remainder } \mathbf{1}\\
10 \div 2 &= 5 \quad \text{remainder } \mathbf{0}\\
5 \div 2 &= 2 \quad \text{remainder } \mathbf{1}\\
2 \div 2 &= 1 \quad \text{remainder } \mathbf{0}\\
1 \div 2 &= 0 \quad \text{remainder } \mathbf{1}
\end{aligned}
$$

Read upward!

$$101010_2$$

Quick Check: $(1\times32)+(0\times16)+(1\times8)+(0\times4)+(1\times2)+(0\times1) = 32+8+2 = 42$

5.2 Binary to Decimal

Base 2 - Binary

Algorithm: Sum of Powers

1. Write the place values above each digit
2. Multiply each binary digit by its place value
3. Add all the results

Example

Convert 11010_2 **to decimal:**

Place values:	16	8	4	2	1
Binary digits:	**0**	**1**	**0**	**1**	**1**
Values:	16	8		2	

$$16 + 8 + 0 + 2 + 0 = \boxed{26_{10}}$$

5.3 Binary to Hexadecimal (THE EASY WAY!)

Base 16 - Hexadecimal

Algorithm: Group by 4s

1. Group binary digits in sets of 4 (from right to left)
2. Convert each group to its hex equivalent
3. Combine the hex digits

Example

Convert 11100110_2 **to hexadecimal:**

Binary: 1 1 1 0 0 1 1 0

1110_2 ↓ $= 14_{10} =$ **D**

0110_2 ↓ $= 6_{10} =$ **6**

$$\boxed{D6_{16}}$$

5.4 4-Bit to Hex Conversion Table

MEMORIZE THIS TABLE!

Binary	Hex	Decimal	Binary	Hex	Decimal
0000	0	0	1000	8	8
0001	1	1	1001	9	9
0010	2	2	1010	A	10
0011	3	3	1011	B	11
0100	4	4	1100	C	12
0101	5	5	1101	D	13
0110	6	6	1110	E	14
0111	7	7	1111	F	15

Pro Tip

Key Patterns:

- $1111_2 = F_{16} = 15_{10}$ (all ones in 4 bits)
- $1010_2 = A_{16} = 10_{10}$ (alternating pattern)
- $10000000_2 = 80_{16} = 128_{10}$ (highest bit only)
- $11111111_2 = FF_{16} = 255_{10}$ (max value in 8 bits)

6 Quick Reference Guide

6.1 Powers to Memorize

Powers of 2:

$$\begin{aligned}
2^0 &= 1 \\
2^1 &= 2 \\
2^2 &= 4 \\
2^3 &= 8 \\
2^4 &= 16 \\
2^5 &= 32 \\
2^6 &= 64 \\
2^7 &= 128 \\
2^8 &= 256 \\
2^9 &= 512 \\
2^{10} &= 1024
\end{aligned}$$

Powers of 16:

$$\begin{aligned}
16^0 &= 1 \\
16^1 &= 16 \\
16^2 &= 256 \\
16^3 &= 4096 \\
16^4 &= 65536
\end{aligned}$$

Common Values:

$$\begin{aligned}
FF_{16} &= 255_{10} \\
100_{16} &= 256_{10} \\
FFF_{16} &= 4095_{10}
\end{aligned}$$

7 Practice Problems

7.1 Warm-up Exercises

1. Convert 25_{10} to binary

2. Convert 11001_2 to decimal

3. Convert $3F_{16}$ to decimal

4. Convert 11110000_2 to hexadecimal

5. Convert $A5_{16}$ to binary

7.2 Challenge Problems

1. Convert 156_{10} to binary, then to hexadecimal

2. Convert 10110111_2 to decimal, then to hexadecimal

3. What is the largest decimal number you can represent with 8 bits?

4. How many bits do you need to represent the decimal number 300?

5. Convert the color `#FF6B35` (hex) to binary for each component (R, G, B)

8 Real-World Applications

8.1 RGB Colors

Web colors use hexadecimal notation: `#RRGGBB`

Color	Hex Code	RGB (Decimal)	Binary
Red	#FF0000	(255, 0, 0)	11111111 00000000 00000000
Green	#00FF00	(0, 255, 0)	00000000 11111111 00000000
Blue	#0000FF	(0, 0, 255)	00000000 00000000 11111111
Purple	#800080	(128, 0, 128)	10000000 00000000 10000000

8.2 Computer Memory

- 1 bit = 1 binary digit (0 or 1)
- 1 byte = 8 bits = 2 hex digits
- 1 kilobyte (KB) = 1024 bytes = 2^{10} bytes
- 1 megabyte (MB) = 1024 KB = 2^{20} bytes
- 1 gigabyte (GB) = 1024 MB = 2^{30} bytes
- 1 terabyte (TB) = 1024 GB = 2^{40} bytes

Pro Tip

Why 1024 instead of 1000?
Computers use powers of 2, and $2^{10} = 1024$ is the closest power of 2 to 1000!

8.3 IP Addresses

IPv4 addresses are four 8-bit numbers:

Example

Example: 192.168.1.1

$$\begin{aligned}
192_{10} &= 11000000_2 = C0_{16} \\
168_{10} &= 10101000_2 = A8_{16} \\
1_{10} &= 00000001_2 = 01_{16} \\
1_{10} &= 00000001_2 = 01_{16}
\end{aligned}$$

Full address in hex: `C0.A8.01.01`

9 Test Preparation

9.1 Checklist Before the Exam

Can you confidently:

☐ Recite the binary place values (1, 2, 4, 8, 16, 32, 64, 128)?

☐ Convert any decimal number (0-255) to binary?

☐ Convert any 8-bit binary number to decimal?

☐ Know that A=10, B=11, C=12, D=13, E=14, F=15?

☐ Convert decimal to hexadecimal by repeated division?

☐ Convert hexadecimal to decimal using powers of 16?

☐ Convert binary to hex by grouping in 4s?

☐ Convert hex to binary by expanding each digit to 4 bits?

☐ Explain why computers use binary?

☐ Explain why programmers use hexadecimal?

9.2 Common Mistakes to Avoid

1. **Reading remainders in wrong order:** When converting decimal to binary/hex, read remainders from *bottom to top*!
2. **Forgetting leading zeros:** When converting hex to binary, each hex digit must become *exactly 4 bits*. $A_{16} = 1010_2$ (not just 1010, but needs to maintain 4-bit groups)
3. **Confusing A-F values:** Remember A=10, not 1! Use the mnemonic.
4. **Wrong grouping direction:** When converting binary to hex, group from *right to left*, not left to right!
5. **Off-by-one errors:** $2^8 = 256$, but the largest 8-bit number is 255 (because we start counting at 0)!

9.3 Speed Tips for the Exam

Pro Tip

- **Binary to Decimal:** Start with the rightmost bit and double as you go left
- **Decimal to Binary:** Use the place value chart - cross off values as you go
- **Binary Hex:** This should be your FASTEST conversion - just memorize the 4-bit table!
- **Check your work:** Always do a quick reverse conversion to verify

10 Summary

Key Takeaways

1. Three Number Systems:

- **Decimal (Base 10):** Our everyday system - powers of 10
- **Binary (Base 2):** Computer's language - powers of 2
- **Hexadecimal (Base 16):** Programmer's shortcut - powers of 16

2. The Magic Relationship:

$$\text{4 Binary Digits} = \text{1 Hex Digit}$$

3. Conversion Strategies:

- Decimal → Binary/Hex: Repeated division, read up
- Binary/Hex → Decimal: Sum of (digit × power)
- Binary Hex: Group/expand by 4 bits (EASIEST!)

4. Must Memorize:

- Binary place values: 1, 2, 4, 8, 16, 32, 64, 128...
- Hex letters: A=10, B=11, C=12, D=13, E=14, F=15
- 4-bit to hex conversion table

Answer Key - Practice Problems

Warm-up Exercises

1. $25_{10} = 11001_2$
   - $25 \div 2 = 12$ R 1
   - $12 \div 2 = 6$ R 0
   - $6 \div 2 = 3$ R 0
   - $3 \div 2 = 1$ R 1
   - $1 \div 2 = 0$ R 1
   - Read up: 11001
2. $11001_2 = 25_{10}$
   - $(1 \times 16) + (1 \times 8) + (0 \times 4) + (0 \times 2) + (1 \times 1)$
   - $= 16 + 8 + 0 + 0 + 1 = 25$
3. $3F_{16} = 63_{10}$
   - $3 = 3$, $F = 15$
   - $(3 \times 16) + (15 \times 1) = 48 + 15 = 63$
4. $11110000_2 = F0_{16}$
   - Group by 4s: 1111 0000
   - $1111_2 = F_{16}$, $0000_2 = 0_{16}$
   - Answer: $F0_{16}$
5. $A5_{16} = 10100101_2$
   - $A = 10_{10} = 1010_2$
   - $5 = 5_{10} = 0101_2$
   - Answer: 10100101_2

Challenge Problems

1. $156_{10} = 10011100_2 = 9C_{16}$
2. $10110111_2 = 183_{10} = B7_{16}$
3. 255 (all bits are 1: $11111111_2 = FF_{16} = 255_{10}$)
4. 9 bits (since $2^8 = 256 < 300 < 512 = 2^9$)
5. `#FF6B35`:
   - R: $FF_{16} = 11111111_2$
   - G: $6B_{16} = 01101011_2$
   - B: $35_{16} = 00110101_2$