Grade 12 A/L ICT · Lesson 3

3.1 Number Systems and Signed Numbers: Binary, Octal, Hex and 2's Complement

Syllabus 3.1. A computer stores **everything as 0s and 1s**. In this part you learn **why only two states** are used, the four **number systems** (decimal, binary, octal, hexadecimal), how to **convert** between them step by step, and how a computer stores **negative numbers** using **sign-magnitude, 1's complement and 2's complement**, where the **most significant bit (MSB)** is the sign bit.

1. Why computers use two states

Inside a computer, every circuit is like a **switch**: it is either **ON or OFF**. A wire has **high voltage or low voltage**, a disk spot is **magnetised one way or the other**, and a CD spot is a **pit or a land**. We write these two states as **1 and 0**. One 0 or 1 is called a **bit** (binary digit).

- Two states are **easy to make and detect** – the circuit only has to tell high from low.
- They are **reliable**: a small amount of electrical noise does not change a 1 into a 0.
- They match **Boolean logic** (true/false), so circuits can do arithmetic and decisions.
- Groups of bits: **4 bits = nibble**, **8 bits = byte**. With n bits you get **2^n different patterns**.

🧱 **Worked example: How many different patterns can be made with 8 bits? With 16 bits?**
8 bits: 2^8 = **256** patterns (0 to 255). 16 bits: 2^{16} = **65,536** patterns.

2. The four number systems

Every number system has a **base** (radix) – the number of different digits it uses. The value of a digit depends on its **place**: each place is worth **base × the place on its right**.

System	Base	Digits	Place values	Example
Decimal	10	0–9	… 1000, 100, 10, 1	156_{10}
Binary	2	0, 1	… 8, 4, 2, 1	10011100_2
Octal	8	0–7	… 512, 64, 8, 1	234_8
Hexadecimal	16	0–9, A–F (A=10 … F=15)	… 4096, 256, 16, 1	$9C_{16}$

💡 Write the base as a small number after the value, for example **1010_2**, so nobody confuses 10 in binary (two) with 10 in decimal (ten).

Decimal	Binary (4 bits)	Octal	Hex

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

💡 Learn this 0–15 table by heart. It is the key to every fast conversion between binary, octal and hex.



How to convert between number systems

3. Decimal to binary, octal and hex – divide and keep the remainders

Method: divide the decimal number by the new base again and again. Write down each **remainder**. Stop when the quotient is 0. Read the remainders **from the bottom to the top**.

🧮 **Worked example: Convert 156_{10} to binary.**
Divide by 2 (table below). Reading the remainders from bottom to top: 156_{10} = **10011100_2**.
Check: 128 + 16 + 8 + 4 = 156.

Division	Quotient	Remainder

156 ÷ 2	78	0
78 ÷ 2	39	0
39 ÷ 2	19	1
19 ÷ 2	9	1
9 ÷ 2	4	1
4 ÷ 2	2	0
2 ÷ 2	1	0
1 ÷ 2	0	1

🧮 **Worked example: Convert 156_{10} to octal.**
Divide by 8: 156 ÷ 8 = 19 r 4, 19 ÷ 8 = 2 r 3, 2 ÷ 8 = 0 r 2. Bottom to top: **234_8**. Check: 2×64 + 3×8 + 4 = 156.

Division	Quotient	Remainder
156 ÷ 8	19	4
19 ÷ 8	2	3
2 ÷ 8	0	2

🧮 **Worked example: Convert 2026_{10} (this year) to hexadecimal.**
Divide by 16. A remainder of 10 to 15 is written as a letter (14 = E). Bottom to top: 2026_{10} = **$7EA_{16}$**. Check: 7×256 + 14×16 + 10 = 2026.

Division	Quotient	Remainder
2026 ÷ 16	126	10 = A
126 ÷ 16	7	14 = E
7 ÷ 16	0	7

4. Binary, octal and hex to decimal – multiply by place values

Method: write the place value under each digit (1, base, base^2, …from the right). Multiply each digit by its place value and **add** the answers.

🧮 **Worked example: Convert 101101_2 to decimal.**
32 + 0 + 8 + 4 + 0 + 1 = **45_{10}**. (Shortcut: just add the place values under the 1s.)

Digit	1	0	1	1	0	1
Place value	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Digit × place value	1 × 32 = 32	0 × 16 = 0	1 × 8 = 8	1 × 4 = 4	0 × 2 = 0	1 × 1 = 1

🧮 **Worked example: Convert 347_8 to decimal.**
3×64 + 4×8 + 7×1 = 192 + 32 + 7 = **231_{10}**.

Digit	**3**	**4**	**7**
Place value	$8^2 = 64$	$8^1 = 8$	$8^0 = 1$
Digit × place value	3 × 64 = 192	4 × 8 = 32	7 × 1 = 7

🧮 **Worked example: Convert $2AF_{16}$ to decimal.**
A = 10 and F = 15. 2×256 + 10×16 + 15×1 = 512 + 160 + 15 = **687_{10}**.

Digit	**2**	**A**	**F**
Place value	$16^2 = 256$	$16^1 = 16$	$16^0 = 1$
Digit × place value	2 × 256 = 512	10 × 16 = 160	15 × 1 = 15

5. Binary ↔ octal and hex – the grouping shortcut

Because 8 = 2^3 and 16 = 2^4, **one octal digit = 3 bits** and **one hex digit = 4 bits**. Group the bits **from the right**, add 0s on the left if needed, and change each group to one digit. To go back, change each digit to 3 or 4 bits.

🧮 **Worked example: Convert 11010110_2 to octal and to hexadecimal.**
Groups of 3 (pad to 011 010 110): octal **326_8**. Groups of 4 (1101 0110): 1101 = D, 0110 = 6, so hex **$D6_{16}$**. Both equal 214 in decimal.

Group	**011**	**010**	**110**
Value	3	2	6

Group	**1101**	**0110**
Value	D	6

🧮 **Worked example: Convert 725_8 and $3C9_{16}$ to binary.**
Octal: each digit → 3 bits: 7 = 111, 2 = 010, 5 = 101, so **111010101_2**. Hex: each digit → 4 bits: 3 = 0011, C = 1100, 9 = 1001, so **$0011\ 1100\ 1001_2$** (the leading 00 can be dropped).

Digit	**7**	**2**	**5**
Bits	111	010	101

Digit	**3**	**C**	**9**

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

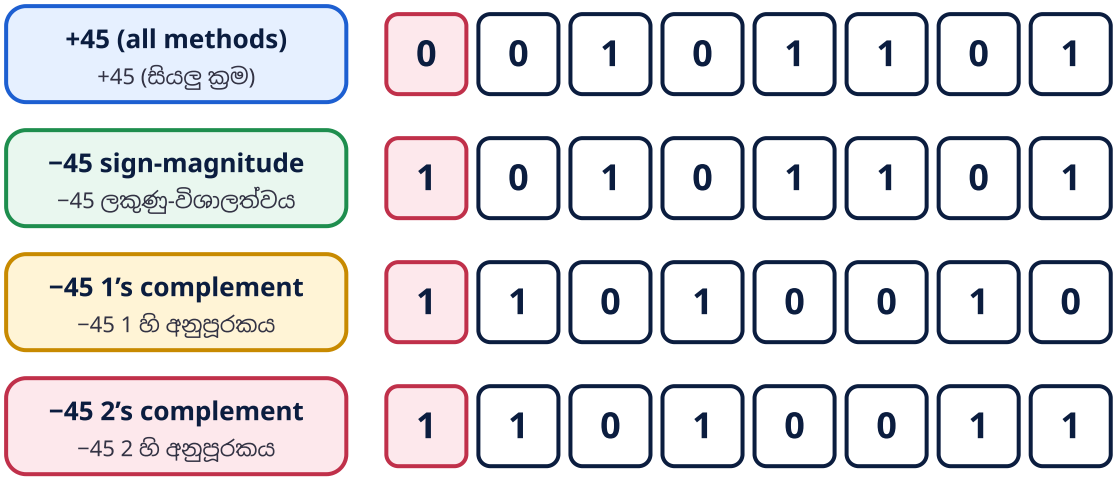
Bits	0011	1100	1001

🧮 **Worked example: Convert 157_8 to hexadecimal, and $2F_{16}$ to octal.**
Go **through binary**. 157_8 → 001 101 111 → regroup in 4s: 0110 1111 → **$6F_{16}$**. $2F_{16}$ → 0010 1111 → regroup in 3s: 000 101 111 → **57_8**.

💡 Never convert octal to hex digit by digit. Always go octal → binary → hex (or hex → binary → octal).

6. Unsigned and signed integers

An **unsigned** integer uses all its bits for the size, so it can only be 0 or positive. 8 bits give 0 to 255. A **signed** integer must also show **+ or −**. The computer uses the **leftmost bit, the most significant bit (MSB)**, as the **sign bit**: **0 = positive, 1 = negative**.



Three ways to store −45 in 8 bits

Sign-magnitude
MSB = sign, the other 7 bits = the size. −45 = **1010 1101**.

1's complement
Write +45, then **flip every bit**. −45 = **1101 0010**.

2's complement
Flip every bit, then **add 1**. −45 = **1101 0011**.

🧮 **Worked example: Represent +45 and −45 in 8-bit sign-magnitude, 1's complement and 2's complement.**
+45 = 0010 1101 in all three (positive numbers look the same). −45: sign-magnitude **1010 1101**, 1's complement **1101 0010**, 2's complement **1101 0011**.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Step	Bits
1. Write +45 in 8 bits	0010 1101
2. Flip every bit (1's complement)	1101 0010
3. Add 1 (2's complement)	**1101 0011**

💡 **2's complement shortcut:** copy the bits from the right up to and including the **first 1**, then flip all the bits to its left. +45 = 00101101 → 11010011.

🧱 **Worked example: What decimal number is the 8-bit 2's complement value 1110 1100?** MSB is 1, so it is negative. Flip: 0001 0011, add 1: 0001 0100 = 20. So the answer is **−20**. (Or: the MSB is worth −128, so −128 + 64 + 32 + 8 + 4 = −20.)

8-bit method	Smallest	Largest	Zeros	Formula (n bits)
Unsigned	0	255	1	0 to $2^n - 1$
Sign-magnitude	−127	+127	two (+0, −0)	$-(2^{n-1} - 1)$ to $2^{n-1} - 1$
1's complement	−127	+127	two (+0, −0)	$-(2^{n-1} - 1)$ to $2^{n-1} - 1$
2's complement	−128	+127	one	-2^{n-1} to $2^{n-1} - 1$

Why 2's complement wins: it has **only one zero**, it gives one extra negative number (−128), and the **same adder circuit** can add and subtract (you will see this in 3.3). That is why almost every processor uses it.

NEW 2026 **Hex in everyday tech:** web colours are hex – **#1D5FD1** means red 1D (29), green 5F (95), blue D1 (209). **IPv6** addresses are 128 bits written as 8 groups of 4 hex digits, e.g. 2001:0db8:0000:0000:0000:0000:0000:0001.

🚀 Beyond the syllabus (2026 update) *Not required for the exam – read for interest.* **Beyond the syllabus – floating point (IEEE 754).** Fractions such as 3.75 are not stored as integers. Computers use the **IEEE 754** standard: a sign bit, an exponent and a mantissa (like scientific notation in binary). A 32-bit float has 1 sign bit, 8 exponent bits and 23 mantissa bits. 3.75 = $11.11_2 = 1.111 \times 2^1$. AI chips in 2025–2026 also use tiny formats such as **FP8 and FP4** to run large models faster.

Key Words

Bit – The smallest unit of data: a single 0 or 1.

Byte – A group of 8 bits.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Nibble – A group of 4 bits, equal to one hex digit.

Base (radix) – The number of different digits a number system uses.

Binary – The base 2 number system using only 0 and 1.

Octal – The base 8 number system using digits 0 to 7.

Hexadecimal – The base 16 number system using 0 to 9 and A to F.

Place value – The value of a position, a power of the base.

MSB – The leftmost bit; the sign bit in a signed number.

Unsigned integer – A whole number stored with no sign, so it is zero or positive.

Sign-magnitude – A sign bit followed by the size of the number in binary.

1's complement – A negative number made by flipping every bit of the positive number.

2's complement – A negative number made by flipping every bit and adding 1.

Sample Q&A

1. Why do computers use the binary system?

> **Answer:** Electronic circuits have two stable states (on/off, high/low voltage). Two states are easy to make, easy to detect, reliable against noise, and match Boolean logic.

2. Convert 156_{10} to binary, octal and hexadecimal.

> **Answer:** Binary 10011100_2, octal 234_8, hexadecimal $9C_{16}$.

3. Why is hexadecimal used by programmers?

> **Answer:** One hex digit stands for exactly 4 bits, so long binary values become short and easy to read, and converting back to binary is quick. Examples: memory addresses, colour codes, MAC and IPv6 addresses.

4. What is the MSB and what does it show in a signed number?

> **Answer:** The MSB is the most significant (leftmost) bit. In a signed number it is the sign bit: 0 means positive and 1 means negative.

5. Write −20 in 8-bit 2's complement.

> **Answer:** +20 = 0001 0100; flip = 1110 1011; add 1 = 1110 1100.

6. Give two disadvantages of sign-magnitude.

> **Answer:** It has two zeros (00000000 and 10000000), and normal binary addition gives wrong answers when the signs are different, so extra circuits are needed.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

MCQ

1. Why do computers use binary?

A. It uses ten digits
B. Hex cannot be stored
C. Circuits have two reliable states
D. Binary numbers are shorter than decimal
E. Humans read binary easily

2. What is 156_{10} in binary?

A. 10011010
B. 11001100
C. 10101100
D. 10011101
E. 10011100

3. What is 101101_2 in decimal?

A. 45
B. 46
C. 37
D. 44
E. 54

4. What is $2AF_{16}$ in decimal?

A. 527
B. 671
C. 703
D. 687
E. 2715

5. What is 11010110_2 in hexadecimal?

A. B6
B. C6
C. D6
D. D3
E. 6D

6. What is 11010110_2 in octal?

A. 332
B. 326
C. 623
D. 316
E. 362

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

7. What is $3C9_{16}$ in binary?

A. 1111001001
B. 1111001100
C. 1011001001
D. 1111000110
E. 1100111001

8. What is 157_8 in hexadecimal?

A. 9D
B. F6
C. 6F
D. 6E
E. 157

9. How many bits does one hexadecimal digit stand for?

A. 16
B. 3
C. 8
D. 2
E. 4

10. What is −45 in 8-bit 2’s complement?

A. 00101101
B. 11010011
C. 10101101
D. 11010010
E. 11010100

11. What is −45 in 8-bit sign-magnitude?

A. 11010011
B. 00101101
C. 10101100
D. 11010010
E. 10101101

12. The 8-bit 2’s complement value 11101100 equals:

A. 20
B. 236
C. −108
D. −20
E. −19

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

13. What is the range of an 8-bit 2’s complement integer?

A. −127 to +127
B. −255 to +255
C. 0 to 255
D. −128 to +127
E. −128 to +128

14. Which method has only one representation of zero?

A. 2’s complement
B. Sign-magnitude
C. None of them
D. 1’s complement
E. Both sign-magnitude and 1’s complement

15. What is the largest unsigned number in 8 bits?

A. 255
B. 127
C. 511
D. 128
E. 256

16. In a signed binary number the MSB with value 1 means:

A. There is an overflow
B. The number is positive
C. The number is zero
D. The number is odd
E. The number is negative

17. Which term means: “The smallest unit of data: a single 0 or 1”?

A. Octal
B. Bit
C. Nibble
D. Place value
E. Sign-magnitude

18. Which term means: “A group of 8 bits”?

A. MSB
B. 1’s complement
C. Byte
D. Hexadecimal
E. Base (radix)

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

19. Which term means: "A group of 4 bits, equal to one hex digit"?

A. Place value
B. Nibble
C. Binary
D. 2's complement
E. Unsigned integer

20. Which term means: "The number of different digits a number system uses"?

A. Octal
B. Bit
C. Sign-magnitude
D. Base (radix)
E. MSB

Structured Questions

1. (a) Convert 2026_{10} to binary showing the division. (b) Convert your answer to octal and hexadecimal by grouping. *(3 marks)*

> (a) repeated ÷2, remainders bottom to top = 11111101010_2; (b) groups of 3 = 3752_8; groups of 4 = $7EA_{16}$.

2. Convert to decimal showing place values: (a) 101101_2 (b) 347_8 (c) $2AF_{16}$. *(3 marks)*

> (a) 32+8+4+1 = 45; (b) 3×64+4×8+7 = 231; (c) 2×256+10×16+15 = 687.

3. Represent −45 in 8 bits using (a) sign-magnitude (b) 1's complement (c) 2's complement. *(3 marks)*

> (a) 10101101; (b) 11010010; (c) 11010011.

4. (a) Give the range of 8-bit unsigned and 8-bit 2's complement integers. (b) Give two reasons why 2's complement is preferred. *(4 marks)*

> (a) unsigned 0 to 255; 2's complement −128 to +127; (b) only one zero; the same adder does addition and subtraction.

5. (a) Find the decimal value of the 8-bit 2's complement number 11101100. (b) Convert $2F_{16}$ to octal. *(4 marks)*

> (a) MSB 1 so negative; flip and add 1 gives 20; value −20; (b) 0010 1111 → 000 101 111 = 57_8.

Summary

- Computers use two states (0 and 1) because they are easy, reliable and match logic.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

- Bases: decimal 10, binary 2, octal 8, hexadecimal 16 (A–F = 10–15).
- Decimal → base: divide and read remainders bottom to top. Base → decimal: multiply by place values and add.
- Binary ↔ octal: groups of 3 bits; binary ↔ hex: groups of 4 bits, from the right.
- Signed numbers use the MSB as the sign bit: sign-magnitude, 1's complement (flip), 2's complement (flip + 1).
- 8-bit 2's complement range is −128 to +127 with only one zero, so CPUs use it.

MCQ answer key – 3.1

1. **C** – On/off states are easy to make and detect.
2. **E** – 156 = 128 + 16 + 8 + 4.
3. **A** – 32 + 8 + 4 + 1 = 45.
4. **D** – 2×256 + 10×16 + 15 = 687.
5. **C** – 1101 = D, 0110 = 6.
6. **B** – 011 010 110 = 3 2 6.
7. **A** – 3 = 0011, C = 1100, 9 = 1001.
8. **C** – 001 101 111 → 0110 1111 = 6F.
9. **E** – 16 = 2^4.
10. **B** – Flip 00101101 → 11010010, add 1 → 11010011.
11. **E** – Sign 1 then 45 in 7 bits.
12. **D** – −128 + 64 + 32 + 8 + 4 = −20.
13. **D** – -2^7 to $2^7 - 1$.
14. **A** – SM and 1's complement have +0 and −0.
15. **A** – $2^8 - 1 = 255$.
16. **E** – MSB is the sign bit.
17. **B** – Bit – The smallest unit of data: a single 0 or 1.
18. **C** – Byte – A group of 8 bits.
19. **B** – Nibble – A group of 4 bits, equal to one hex digit.
20. **D** – Base (radix) – The number of different digits a number system uses.

© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.