ESD - Elettronica dei Sistemi Digitali

Solutions on Data Representation

Prof. Riccardo Berta

2025.10.03

1 Data Representation Exercises

1.1 Exercise 1

What is the largest 32-bit binary number that can be represented with:

1.1.1 (a) Unsigned numbers

Largest value: $1111\ldots1111$ (32 ones)
$= 2^{32} - 1 = 4.294.967.295$
$\approx 2^{30} \times 2^{2} \approx 4GB$

1.1.2 (b) Two's complement numbers

Range: -2^{31} to $2^{31} - 1$
Largest value: $0111\ldots1111$ (31 ones after the leading 0)
$= 2^{31} - 1 = 2.147.483.647$
$\approx 2^{30} \times 2^{1} \approx 2GB$

1.1.3 (c) Sign/magnitude numbers

1 bit for the sign, 31 bits for the magnitude
Largest positive: $0111\ldots1111$
$= 2^{31} - 1 = 2.147.483.647$
$\approx 2^{30} \times 2^{1} \approx 2GB$

1.2 Exercise 2

What is the smallest (most negative) 16-bit binary number that can be represented with:

1.2.1 (a) Unsigned numbers

Unsigned representation cannot encode negative values.
Smallest value: $0000\ldots0000 = 0$

1.2.2 (b) Two's complement numbers

Range: -2^{15} to $+2^{15} - 1$
Smallest value: $1000 \ldots 0000$ (1 followed by 15 zeros)
$= -2^{15} = -32.768$

1.2.3 (c) Sign/magnitude numbers

1 bit for sign, 15 bits for magnitude
Smallest value: $1111 \ldots 1111$ (sign bit = 1, magnitude = max)
$= -(2^{15} - 1) = -32.767$

1.3 Exercise 3

What is the smallest (most negative) 32-bit binary number that can be represented with:

1.3.1 (a) Unsigned numbers

Unsigned representation cannot encode negative values.
Smallest value: $0000 \ldots 0000_2 = 0$

1.3.2 (b) Two's complement numbers

Range: -2^{31} to $+2^{31} - 1$
Smallest value: $1000 \ldots 0000_2$ (1 followed by 31 zeros)
$= -2^{31} = -2.147.483.648$

1.3.3 (c) Sign/magnitude numbers

1 bit for sign, 31 bits for magnitude
Smallest value: $1111 \ldots 1111_2$ (sign bit = 1, magnitude = max)
$= -(2^{31} - 1) = -2.147.483.647$

1.4 Exercise 4

Convert the following unsigned binary numbers to decimal and to hexadecimal:

1.4.1 (a) 1110_2

Decimal: $1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 8 + 4 + 2 = 14$
Hex: $14_{10} = E_{16}$

1.4.2 (b) 100100_2

Decimal: $1 \times 2^5 + 0 + 0 + 1 \times 2^2 = 32 + 4 = 36$
Hex: $36_{10} = 24_{16}$

1.4.3 (c) 11010111_2

Decimal: $128 + 64 + 16 + 4 + 2 + 1 = 215$
Hex: group as $(1101)(0111) = D7_{16}$

1.4.4 (d) $0111010101001 00_2$

Decimal: $= 2^{13} + 2^{12} + 2^{11} + 2^9 + 2^7 + 2^5 + 2^2$
$= 8192 + 4096 + 2048 + 512 + 128 + 32 + 4 = 15012_{10}$ Hex: group as $(0011)(1010)(1010)(0100) = 3AA4_{16}$

1.4.5 (e) 0110_2

Decimal: $4 + 2 = 6$
Hex: $6_{10} = 6_{16}$

1.4.6 (f) 101101_2

Decimal: $32 + 8 + 4 + 1 = 45$
Hex: $45_{10} = 2D_{16}$

1.4.7 (g) 10010101_2

Decimal: $128 + 16 + 4 + 1 = 149$
Hex: group as $(1001)(0101) = 95_{16}$

1.4.8 (h) 110101001001_2

Decimal: $2048 + 1024 + 256 + 64 + 8 + 1 = 3401$
Hex: group as $(1101)(0100)(1001) = D49_{16}$

1.5 Exercise 5

Convert the following hexadecimal numbers to decimal and to unsigned binary:

1.5.1 (a) $4E_{16}$

$= 4 \times 16^1 + 14 \times 16^0 = 64 + 14 = 78_{10}$
Binary: $4 = 0100,\ E = 1110 \quad \Rightarrow \quad 0100\ 1110_2$

1.5.2 (b) $7C_{16}$

$= 7 \times 16^1 + 12 \times 16^0 = 112 + 12 = 124_{10}$
Binary: $7 = 0111,\ C = 1100 \quad \Rightarrow \quad 0111\ 1100_2$

1.5.3 (c) $ED3A_{16}$

$= 14 \times 16^3 + 13 \times 16^2 + 3 \times 16^1 + 10 \times 16^0$
$= 57.344 + 3.328 + 48 + 10 = 60.730_{10}$
Binary: $E = 1110,\ D = 1101,\ 3 = 0011,\ A = 1010 \quad \Rightarrow \quad 1110\ 1101\ 0011\ 1010_2$

1.5.4 (d) $403FB001_{16}$

$= 4 \times 16^7 + 0 \times 16^6 + 3 \times 16^5 + 15 \times 16^4 + 11 \times 16^3 + 0 \times 16^2 + 0 \times 16^1 + 1$
$= 1.073.741.824 + 3.145.728 + 61.440 + 45.056 + 1 = 1.077.915.649_{10}$
Binary: $4 = 0100,\ 0 = 0000,\ 3 = 0011,\ F = 1111,\ B = 1011,\ 0 = 0000,\ 0 = 0000,\ 1 = 0001$
$\Rightarrow 0100\ 0000\ 0011\ 1111\ 1011\ 0000\ 0000\ 0001_2$

1.5.5 (e) $2B_{16}$

$= 2 \times 16^1 + 11 \times 16^0 = 32 + 11 = 43_{10}$
Binary: $2 = 0010,\ B = 1011 \quad \Rightarrow \quad 0010\ 1011_2$

1.5.6 (f) $9F_{16}$

$= 9 \times 16^1 + 15 \times 16^0 = 144 + 15 = 159_{10}$
Binary: $9 = 1001,\ F = 1111 \quad \Rightarrow \ 1001\ 1111_2$

1.5.7 (g) $42CE_{16}$

$= 4 \times 16^3 + 2 \times 16^2 + 12 \times 16^1 + 14 \times 16^0$
$= 16.384 + 512 + 192 + 14 = 17.102_{10}$
Binary: $4 = 0100,\ 2 = 0010,\ C = 1100,\ E = 1110 \quad \Rightarrow \ 0100\ 0010\ 1100\ 1110_2$

1.5.8 (h) $E34F_{16}$

$= 14 \times 16^3 + 3 \times 16^2 + 4 \times 16^1 + 15 \times 16^0$
$= 57.344 + 768 + 64 + 15 = 58.191_{10}$
Binary: $E = 1110,\ 3 = 0011,\ 4 = 0100,\ F = 1111 \quad \Rightarrow \ 1110\ 0011\ 0100\ 1111_2$

1.6 Exercise 6

Convert the following two's complement binary numbers to decimal:

1.6.1 (a) 1110_2 (4-bit)

MSB = 1 -> negative.
Invert 1110 → 0001, add 1 -> 0010 = 2.
Result = -2_{10}.

1.6.2 (b) 100011_2 (6-bit)

MSB = 1 -> negative.
Invert 100011 → 011100, add 1 -> 011101 = 29.
Result = -29_{10}.

1.6.3 (c) 01001110_2 (8-bit)

MSB = 0 -> positive.
Value = $64 + 8 + 4 + 2 = 78_{10}$.

1.6.4 (d) 10110101_2 (8-bit)

MSB = 1 -> negative.
Invert 10110101 → 01001010, add 1 -> 01001011 = 75.
Result = -75_{10}.

1.6.5 (e) 1001_2 (4-bit)

MSB = 1 -> negative.
Invert 1001 → 0110, add 1 -> 0111 = 7.
Result = -7_{10}.

1.6.6 (f) 110101_2 (6-bit)

MSB = 1 -> negative.
Invert 110101 → 001010, add 1 -> 001011 = 11.
Result = -11_{10}.

1.6.7 (g) 01100010_2 (8-bit)

MSB = 0 -> positive.
Value = $64 + 32 + 2 = 98_{10}$.

1.6.8 (h) 10111000_2 (8-bit)

MSB = 1 -> negative.
Invert 10111000 → 01000111, add 1 -> 01001000 = 72.
Result = -72_{10}.

1.7 Exercise 7

Convert the following decimal numbers to unsigned binary and to hexadecimal

1.7.1 (a) 42_{10}

$42 \div 2 = 21$ remainder 0
$21 \div 2 = 10$ remainder 1

$10 \div 2 = 5$ remainder 0
$5 \div 2 = 2$ remainder 1
$2 \div 2 = 1$ remainder 0
$1 \div 2 = 0$ remainder 1
Reading upwards -> 101010_2
Group: $0010\ 1010_2 = 2A_{16}$

1.7.2 (b) 63_{10}

$63 \div 2 = 31$ r 1
$31 \div 2 = 15$ r 1
$15 \div 2 = 7$ r 1
$7 \div 2 = 3$ r 1
$3 \div 2 = 1$ r 1
$1 \div 2 = 0$ r 1
-> 111111_2
Group: $0011\ 1111_2 = 3F_{16}$

1.7.3 (c) 229_{10}

$229 \div 2 = 114$ r 1
$114 \div 2 = 57$ r 0
$57 \div 2 = 28$ r 1
$28 \div 2 = 14$ r 0
$14 \div 2 = 7$ r 0
$7 \div 2 = 3$ r 1
$3 \div 2 = 1$ r 1
$1 \div 2 = 0$ r 1
-> 11100101_2
Group: $1110\ 0101_2 = E5_{16}$

1.7.4 (d) 845_{10}

$845 \div 2 = 422$ r 1
$422 \div 2 = 211$ r 0
$211 \div 2 = 105$ r 1
$105 \div 2 = 52$ r 1

$52 \div 2 = 26$ r 0
$26 \div 2 = 13$ r 0
$13 \div 2 = 6$ r 1
$6 \div 2 = 3$ r 0
$3 \div 2 = 1$ r 1
$1 \div 2 = 0$ r 1
-> 1101001101_2
Group: $0011\ 0100\ 1101_2 = 34D_{16}$

1.7.5 (e) 56_{10}

$56 \div 2 = 28$ r 0
$28 \div 2 = 14$ r 0
$14 \div 2 = 7$ r 0
$7 \div 2 = 3$ r 1
$3 \div 2 = 1$ r 1
$1 \div 2 = 0$ r 1
→ 111000_2
Group: $0011\ 1000_2 = 38_{16}$

1.7.6 (f) 75_{10}

$75 \div 2 = 37$ r 1
$37 \div 2 = 18$ r 1
$18 \div 2 = 9$ r 0
$9 \div 2 = 4$ r 1
$4 \div 2 = 2$ r 0
$2 \div 2 = 1$ r 0
$1 \div 2 = 0$ r 1
-> 1001011_2
Group: $0100\ 1011_2 = 4B_{16}$

1.7.7 (g) 183_{10}

$183 \div 2 = 91$ r 1
$91 \div 2 = 45$ r 1
$45 \div 2 = 22$ r 1

$22 \div 2 = 11$ r 0
$11 \div 2 = 5$ r 1
$5 \div 2 = 2$ r 1
$2 \div 2 = 1$ r 0
$1 \div 2 = 0$ r 1
-> 10110111_2
Group: $1011\ 0111_2 = B7_{16}$

1.7.8 (h) 754_{10}

$754 \div 2 = 377$ r 0
$377 \div 2 = 188$ r 1
$188 \div 2 = 94$ r 0
$94 \div 2 = 47$ r 0
$47 \div 2 = 23$ r 1
$23 \div 2 = 11$ r 1
$11 \div 2 = 5$ r 1
$5 \div 2 = 2$ r 1
$2 \div 2 = 1$ r 0
$1 \div 2 = 0$ r 1
-> 1011110010_2
Group: $0010\ 1111\ 0010_2 = 2F2_{16}$

1.8 Exercise 8

Convert the following decimal numbers to 8-bit two's complement numbers or indicate overflow. Range of 8-bit two's complement: $-128 \leq N \leq +127$.

1.8.1 (a) 24

128:0,
64:0,
32:0,
16:1 (remainder 8),
8:1 (0),
4:0,
2:0,

1:0
00011000

1.8.2 (b) -59

$256 - 59 = 197$
128:1 (69),
64:1 (5),
32:0,
16:0,
8:0,
4:1 (1),
2:0,
1:1 (0)
11000101

1.8.3 (c) 128

Outside interval $[-128, 127]$
overflow

1.8.4 (d) -150

$-150 < -128$
overflow

1.8.5 (e) 127

128:0,
64:1 (63),
32:1 (31),
16:1 (15),
8:1 (7),
4:1 (3),
2:1 (1),
1:1 (0)
01111111

1.8.6 (f) 48

128:0,
64:0,
32:1 (16),
16:1 (0),
8:0, 4:0,
2:0,
1:0
00110000

1.8.7 (g) -34

$256 - 34 = 222$
128:1 (94),
64:1 (30),
32:0,
16:1 (14),
8:1 (6),
4:1 (2),
2:1 (0),
1:0
11011110

1.8.8 (h) 133

$133 > 127$
overflow

1.8.9 (i) -129

$-129 < -128$
overflow

1.9 Exercise 9

How many bytes are in a 32-bit word? How many nibbles are in the 32-bit word? How many bytes are in a 64-bit word? How many nibbles are in the 64-bit word? How many bits are in 2 bytes? How many bits are in 6 bytes?

A **word** of 32 bits = 32/8 = 4 bytes.
Each byte = 2 nibbles -> $4 \times 2 = 8$ nibbles in a 32-bit word.

A **word** of 64 bits = 64/8 = 8 bytes.
Each byte = 2 nibbles -> $8 \times 2 = 16$ nibbles in a 64-bit word.

In **2 bytes**: $2 \times 8 = 16$ bits.
In **6 bytes**: $6 \times 8 = 48$ bits.

1.10 Exercise 10

Convert the following decimal numbers to IEEE 754 single-precision format:

1.10.1 (a) 45.375_{10}

Sign: positive -> $sign = 0$
Integer 45 -> 101101_2
Fraction .375 = 3/8 -> $.011_2$
Combined -> 101101.011_2
Normalize: $1.01101011_2 \times 2^5$ -> unbiased exponent $E = 5$
Exponent (bias 127): $E + 127 = 132 = 10000010_2$
Mantissa (drop leading 1): 01101011 then pad -> 01101011000000000000000
-> 0 10000010 01101011000000000000000

1.10.2 (b) -13.25_{10}

Sign: negative -> $sign = 1$
Integer 13 -> 1101_2
Fraction .25 = 1/4 -> $.01_2$
Combined -> 1101.01_2
Normalize: $1.10101_2 \times 2^3$ -> unbiased exponent $E = 3$
Exponent (bias 127): $E + 127 = 130 = 10000010_2$
Mantissa: drop leading 1 -> 10101 then pad -> 10101000000000000000000
-> 1 10000010 10101000000000000000000

1.10.3 (c) 0.1_{10}

Sign: positive -> $sign = 0$
Fraction (repeating): $0.0001100110011..._2$
Normalize -> $1.1001100110011..._2 \times 2^{-4}$ -> $E = -4$
Exponent (bias 127): $E + 127 = 123 = 01111011_2$
Mantissa: take 23 bits after the leading 1, with rounding: 10011001100110011001101
-> 0 01111011 10011001100110011001101
(*Note: 0.1 is not exactly representable in binary; this is the rounded IEEE single-precision value.*)

1.10.4 (d) -0.125_{10}

Sign: negative -> $sign = 1$
Binary: $0.125 = 1/8 = 2^{-3}$ Normalize: $1.0_2 \times 2^{-3}$ -> $E = -3$
Exponent (bias 127): $E + 127 = 124 = 01111100_2$
Mantissa: exactly zero (since significand is 1.000...)
-> 1 01111100 00000000000000000000000

1.11 Exercise 11

Convert the following IEEE 754 single-precision numbers into decimal values:

1.11.1 (a) 0 10000010 01100000000000000000000

Sign = 0 6. -> positive
Exponent = $10000010_2 = 130$ 6. -> unbiased $E = 130 - 127 = 3$
Mantissa = $1.011_2 = 1 + 0.25 + 0.125 = 1.375$
Value = $1.375 \times 2^3 = 11_{10}$

1.11.2 (b) 1 10000001 01000000000000000000000

Sign = 1 -> negative
Exponent = $10000001_2 = 129$ -> unbiased $E = 129 - 127 = 2$
Mantissa = $1.01_2 = 1 + 0.25 = 1.25$
Value = $-(1.25 \times 2^2) = -5_{10}$

1.11.3 (c) 0 01111101 10000000000000000000000

Sign = 0 -> positive
Exponent = $01111101_2 = 125$ -> unbiased $E = 125 - 127 = -2$
Mantissa = $1.1_2 = 1.5$
Value = $1.5 \times 2^{-2} = 1.5 \times 0.25 = 0.375_{10}$

1.11.4 (d) 1 01111100 00000000000000000000000

Sign = 1 -> negative
Exponent = $01111100_2 = 124$ -> unbiased $E = 124 - 127 = -3$
Mantissa = $1.0_2 = 1.0$
Value = $-(1.0 \times 2^{-3}) = -0.125_{10}$

1.12 Exercise 12

A particular modem operates at 768 Kb/sec. How many bytes can it receive in 1 minute?

Case 1: K = 1000 (decimal kilo, common in communications)**

- Data rate: $768 \times 1000 = 768.000$ bits/sec
- Time: 60 seconds
  $768.000 \times 60 = 46.080.000$ bits
- Convert to bytes: $\frac{46.080.000}{8} = 5.760.000$ bytes (≈ 5.76 MB

Case 2: K = 1024 (binary kilo, kibi)**

- Data rate: $768 \times 1024 = 786.432$ bits/sec
- Time: 60 seconds
  $786.432 \times 60 = 47.185.920$ bits
- Convert to bytes: $\frac{47.185.920}{8} = 5.898.240$ bytes ≈ 5.63 MiB

1.13 Exercise 13

USB 3.0 can send data at 5 Gb/sec. How many bytes can it send in 1 minute?

- Step 1 – Convert Gb/sec to bits/sec
  $5\,\text{Gb/sec} = 5 \times 10^9 = 5.000.000.000$ bits/sec
- Step 2 – Multiply by time (60 seconds)
  $5.000.000.000 \times 60 = 300.000.000.000$ bits
- Step 3 – Convert bits to bytes (8 bits = 1 byte) $\frac{300.000.000.000}{8} = 37.500.000.000$ bytes
- In 1 minute, USB 3.0 can send **37.500.000.000 bytes**
  ≈ **37.5 GB (decimal)**