Answer

Data Representation

1.2 Converting between Number Bases

Convert 01101001 to decimal = 105	Convert 10 001101 (binary) to hexadecimal = 14D
Convert 151 (octal) to decimal = 105	Convert 123 (octal) to binary = 001010011
Convert 15B (hexadecimal) to decimal = 347	Convert 123 (octal) to hexadecimal = 53
Convert 156 (decimal) to binary = 10011100	Convert 1F3 (hex) to binary = 00011111 0011
Convert 1234 (decimal) to octal = 2322	Convert 1F3 (hexadecimal) to Octal = 763
Convert 689 (decimal) to hexadecimal = 2B1	
Convert 101101001 (binary) to octal = 551	

Exercise 1 Number in Binary: 101110 Number in Octal: 56 Number in Hexadecimal: 2e	Exercise 2: 10 8 16

1.3 Arithmetic Operations

Base 2: 01011111 + 00010001 00111110 01110000	Base 8: 363 + 247 110 carries 632	Base 16: F3 + a7 100 19A
Base 2: 01011111 - 00010001 00000000 01001110	Base 8: 363 - 247 010 borrows 114	Base 16: F3 - a7 10 4C

1.4 Negative Numbers

What is the range of:
a) 4-bit unsigned number: 0 <= value <= 15
b) Excess (biased) : -8 <= value <=7
b) Two's complement: -8 <= value <= 7

(1) Convert 10110011 to Decimal if the number is represented as signed, 8-bit in Excess representation: 179-128=51
(2) Convert 10110011 to Decimal if the number is represented as signed, 8-bit in Two's complement representation: 1 = Negative number, complement it => 0100 1100 = 76, add 1 = 77, answer = -77
(3) Convert 10110011 to Decimal if the number is represented as unsigned, 8-bit: = 179
(4) Convert -17 to 8-bit binary using 8-bits, Excess representation 17 = 00010001, complement the last seven bits + 1=>01101110 + 1 = 01101111
(5) Convert –17 to binary using 8-bits, two's complement representation 17 = 00010001, one's complement = 11101110, two's complement = 11101111

1.4.5 Overflow

Exercise 0111 + 0101 1110 carries 1100 invalid	Exercise 1001 + 1100 10000 carries 10101 invalid	Exercise 1100 + 0111 11000 carries 10011 = 0011 valid	Exercise 1001 + 0011 0110 carries 1100 valid

1.5 Subtraction by complement addition

1) 01101011 – 00010110. One's complement of 00010110 is 11101001, Two's complement of 00010110 is 11101010 11101010 + 01101011 111010100 carries 101010101 Answer = 01010101, checking: 107_{10} – (22_{10})= 85_{10}
2) 00101100 – 01101001. One's complement of 01101001is 10010110, Two's complement of 01101001 is 10010111 10010111 + 00101100 01111000 carries 11000011 Answer = 11000011, checking: 44_{10} – (105_{10})= -61_{10}

1.6 Bitwise Logical Operations

Exercise	Exercise	Exercise
1010 0001 &0101 1111 0000 0001	1010 0001 |0101 1111 1111 1111	1010 0001 ^0101 1111 1111 1110

1.7 Shift Operations

A) What is the result of 00001100 << 3? = 01100000 (96) (Note: 12 * 8 = 96)	B) 32-bit binary signed number: int x = 31 << 2;= 124
A) 8-bit binary signed number: 10100001 >> 3 = 11110100	B) 32-bit binary signed number: int x = 21 >> 2; Answer = 5
A) 8-bit binary signed number: 10100001 >>> 3 = 00010100	B) 32-bit binary signed number: int x = -1 >>> 2; Hex: 3FFFFFFF (Note: Decimal : 1073741823)

x=21
x=10101
y=56, y=111000
x & y=16
x | y=61
x << 2=84
x >> 2=5
x >>> 2=5
x ^ y=45

2.5 Conversion

Convert 0062 0073 0042 662E 0020 5DF6 from UCS-2 to UTF-8 coding.

0062 => 62
0073 => 73
0042 => 42
662E => 0110 0110 0010 1110
⇨ **1110** 0110 **10** 0110 00 **10** 10 1110 = E6 98 AE
0020 => 20
5DF6 => 0101 1101 1111 0110
⇨ **1110** 0101 **10** 1101 11 **10** 11 0110 = E5 B7 B6

Convert 21 E8 A2 93 C7 8E from UTF-8 to UCS-2 coding.

21=> 0021
E8 A2 93 => 1110 **1000** 10**10 0010** 10**01 0011**
⇨ 1000 10 0010 01 0011 = 88 93
C7 8E => 110**0 0111** 10**00 1110**
⇨ 0000 0001 1100 1110 = 01 CE