**Homework 1**

**Question 1: Construct a table for a 5-3-2-1 weighted code for decimal digits, and write (9731)10  using this code.**

|  |  |
| --- | --- |
| Decimal Digits | 5-3-2-1 |
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0100 |
| 4 | 0101 |
| 5 | 1000 |
| 6 | 1001 |
| 7 | 1010 |
| 8 | 1100 |
| 9 | 1101 |

(9731)10 –> ( )5-3-2-1

=(1101 1010 0100 0001)5-3-2-1

**Question 2: The solutions to a quadratic equation *x*2 *−* 13*x* + 32 = 0 are *x* = 5 and *x* = 4 What is the base of the numbers? To get the mark of this question, Show the details of your solution.**

Largest value= 5, so r=5 at least.

The equation in decimal is (x-5)(x-4)=x^2-13x+32

Expanding the LHS gives the equation in unknown base r:

(x-5)(x-4)=x^2-9x+20

Comparing the coefficients of the first expression with the second:

(-9)r=(-13)10

(20)r=(32)10

d1d2 in base r = d1\*r + d2\*1 in base 10

2\*r + 0\*1=32

2r=32

r=16

**Question 3: Convert the following binary numbers to decimal**

|  |  |
| --- | --- |
| **(a)** | **100010110** |
| **(b)** | **10101011** |
| **(c)** | **101011011011** |
| **(d)** | **10000000000** |

1. r=2,m=9,n=0

(1 × 2⁸) + (0 × 2⁷) + (0 × 2⁶) + (0 × 2⁵) + (1 × 2⁴) + (0 × 2³) + (1 × 2²) + (1 × 2¹) + (0 × 2⁰)

= (278)₁₀

1. r=2,m=8,n=0

(1 × 2⁷) + (0 × 2⁶) + (1 × 2⁵) + (0 × 2⁴) + (1 × 2³) + (0 × 2²) + (1 × 2¹) + (1 × 2⁰)

= (171)₁₀

1. r=2,m=12,n=0

(1 × 2¹¹) + (0 × 2¹⁰) + (1 × 2⁹) + (0 × 2⁸) + (1 × 2⁷) + (1 × 2⁶) + (0 × 2⁵) + (1 × 2⁴) + (1 × 2³) + (0 × 2²) + (1 × 2¹) + (1 × 2⁰)

= (2779)₁₀

1. r=2,m=11,n=0

(1 × 2¹⁰) + (0 × 2⁹) + (0 × 2⁸) + (0 × 2⁷) + (0 × 2⁶) + (0 × 2⁵) + (0 × 2⁴) + (0 × 2³) + (0 × 2²) + (0 × 2¹) + (0 × 2⁰)

= (1024)₁₀

**Question 4: Convert the following decimal values to binary**

|  |  |
| --- | --- |
| **(a)** | **217** |
| **(b)** | **186** |
| **(c)** | **265** |
| **(d)** | **957** |

a)

|  |  |  |
| --- | --- | --- |
| Division by 2 | Quotient | Remainder |
| (217)/2 | 108 | 1 |
| (108)/2 | 54 | 0 |
| (54)/2 | 27 | 0 |
| (27)/2 | 13 | 1 |
| (13)/2 | 6 | 1 |
| (6)/2 | 3 | 0 |
| (3)/2 | 1 | 1 |
| (1)/2 | 0 | 1 |

=(11011001)2

b)

|  |  |  |
| --- | --- | --- |
| Division by 2 | Quotient | Remainder |
| (186)/2 | 93 | 0 |
| (93)/2 | 46 | 1 |
| (46)/2 | 23 | 0 |
| (23)/2 | 11 | 1 |
| (11)/2 | 5 | 1 |
| (5)/2 | 2 | 1 |
| (2)/2 | 1 | 0 |
| (1)/2 | 0 | 1 |

=(10111010)2

c)

|  |  |  |
| --- | --- | --- |
| Division by 2 | Quotient | Remainder |
| (265)/2 | 132 | 1 |
| (132)/2 | 66 | 0 |
| (66)/2 | 33 | 0 |
| (33)/2 | 16 | 1 |
| (16)/2 | 8 | 0 |
| (8)/2 | 4 | 0 |
| (4)/2 | 2 | 0 |
| (2)/2 | 1 | 0 |
| (1)/2 | 0 | 1 |

=(100001001)2

d)

|  |  |  |
| --- | --- | --- |
| Division by 2 | Quotient | Remainder |
| (957)/2 | 478 | 1 |
| (478)/2 | 239 | 0 |
| (239)/2 | 119 | 1 |
| (119)/2 | 59 | 1 |
| (59)/2 | 29 | 1 |
| (29)/2 | 14 | 1 |
| (14)/2 | 7 | 0 |
| (7)/2 | 3 | 1 |
| (3)/2 | 1 | 1 |
| (1)/2 | 0 | 1 |

=(1110111101)2

**Question 5: Convert each of the following hexadecimal numbers to its equivalent decimal value**

1. **36**
2. **ABCD**
3. **89 (d)2000**
4. r=16,m=2,n=0

(3 × 16¹) + (6 × 16⁰)

= (54)₁₀

1. r=16,m=4,n=0

(10 × 16³) + (11 × 16²) + (12 × 16¹) + (13 × 16⁰)

= (43981)₁₀

1. r=16,m=2,n=0

(8 × 16¹) + (9 × 16⁰)

= (137)₁₀

1. r=16,m=4,n=0

(2 × 16³) + (0 × 16²) + (0 × 16¹) + (0 × 16⁰)

= (8192)₁₀

**Question 6: Convert each of the following decimal numbers to hexadecimal**

**(a) 372 (b)2313**

**(c) 33 (d)1024**

|  |  |  |
| --- | --- | --- |
| Division by 16 | Quotient | Remainder |
| (372)/16 | 23 | 4 |
| (23)/16 | 1 | 7 |
| (1)/16 | 0 | 1 |

a)

=(174)16

b)

|  |  |  |
| --- | --- | --- |
| Division by 16 | Quotient | Remainder |
| (2313)/16 | 144 | 9 |
| (144)/16 | 9 | 0 |
| (9)/16 | 0 | 9 |

=(909)16

c)

|  |  |  |
| --- | --- | --- |
| Division by 16 | Quotient | Remainder |
| (33)/16 | 2 | 1 |
| (2)/16 | 0 | 2 |

=(21)16  
 d)

|  |  |  |
| --- | --- | --- |
| Division by 16 | Quotient | Remainder |
| (1024)/16 | 64 | 0 |
| (64)/16 | 4 | 0 |
| (4)/16 | 0 | 4 |

=(400)16

**Question 7: The numbers below are expressed in 6-3-1-1 code. Convert them to decimal**

|  |  |
| --- | --- |
| **(a)** | **010101010101** |
| **(b)** | **000110000100** |
| **(c)** | **10111** |
| **(d)** | **1110101** |

|  |  |
| --- | --- |
| Decimal Digits | 6-3-1-1 |
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0011 |
| 3 | 0100 |
| 4 | 0101 |
| 5 | 0111 |
| 6 | 1000 |
| 7 | 1001 |
| 8 | 1011 |
| 9 | 1100 |

1. (0101 0101 0101)6-3-1-1 = (444)10
2. (0001 1000 0100)6-3-1-1 = (163)10
3. (0001 0111)6-3-1-1 = (15)10
4. (0111 0101)6-3-1-1 = (54)10

**Question 8: Encode the following decimal numbers in Excess-3 code and then attach an even parity bit.**

**(a) 38**

**(b) 275 (c)9201**

**(d) 51**

|  |  |
| --- | --- |
| Decimal Digits | Excess-3 |
| 0 | 0011 |
| 1 | 0100 |
| 2 | 0101 |
| 3 | 0110 |
| 4 | 0111 |
| 5 | 1000 |
| 6 | 1001 |
| 7 | 1010 |
| 8 | 1011 |
| 9 | 1100 |

1. (38)10 = (01101011)excess-3 = 101101011
2. (275)10 = (010110101000)excess-3 = 1010110101000
3. (9201)10 = (1100010100110100)excess-3 = 11100010100110100
4. (51)10 = (10000100)excess-3 = 010000100

**Question 9: Each of the following numbers represents a signed decimal number in the 2*js*-complement system. Determine the decimal value in each case.**

|  |  |
| --- | --- |
| **(a)** | **01101** |
| **(b)** | **11101** |
| **(c)** | **01111011** |
| **(d)** | **11111111** |
| **(e)** | **01111111** |
| **(f)** | **10000000** |

1. sign bit 0 indicates positive

(01101)2’s comp -> (01101)2 -> (13)10

1. sign bit 1 indicates negative

(11101)2’s comp -> (00011)2 -> (-3)10

1. sign bit 0 indicates positive

(01111011)2’s comp -> (01111011)2 -> (123)10

1. sign bit 1 indicates negative

(11111111)2’s comp -> (00000001)2 -> (-1)10

1. sign bit 0 indicates positive

(01111111)2’s comp -> (01111111)2 -> (127)10

1. sign bit 1 indicates negative

(10000000)2’s comp -> (-128)10

**Question 10: Perform the following operations in the 2*js*-complement system.**

1. **Add -87 to +256**
2. **Add -35 to +65**
3. **Add +490 to +22**

**(d) Add -255 to -230**

**(e) Add -129 to+128 (f) Add +986 to+123**

**For each of the above operations:**

**Determine the minimum number of bits required to represent both summands. You might need to sign-extend one of the summands, since for proper summation, both summands must have the same number of bits. Perform the binary addition in 2*js*-complement arithmetic. The result must have the same number of bits as the summands. Determine whether there is overflow or no. To avoid overflow, determine the minimum number of bits required to represent both the summands and the result.**

1. Extended to 10 bits to represent and match (+256)10 properly

(-87)10 -> (1110101001)2’s comp

(+256)10 -> (0100000000)2’s comp

1110101001+0100000000=0010101001

=(169)10

1. (-35)10 -> (11011101)2’s comp

(+65)10 -> (01000001)2’s comp

11011101+01000001=00011110

=(30)10

1. Extended to 10 bits to represent and match (+490)10 properly

(+490)10 -> (0111101010)2’s comp

(+22)10 -> (0000010110)2’s comp

0111101010+0000010110=1000000000

=(512)10

1. Extended to 10 bits to represent sum properly

(-255)10 -> (1100000001)2’s comp

(-230)10 -> (1100011010)2’s comp

1100000001+1100011010=1000011011

=(-485)10

1. Extended to 9 bits to represent both summands properly

(-129)10 -> (101111111)2’s comp

(+128)10 -> (010000000)2’s comp

101111111+010000000=111111111

=(-1)10

1. Extended to 12 bits to represent sum properly

(+986)10 -> (001111011010)2’s comp

(+123)10 -> (000001111011)2’s comp

001111011010+000001111011=010001010101

=(1109)10