**Base r**

an-1 … a1a0 x a-1a-2 … a-m 🡪 V(x) = a0 + a1\*r + a2 \* r2 +… an-1 \* rn-1 + a-1\*r-1 + a-2\*r-2 +… a-m\*r-m

V(x) =

**Base 16 (Hex: 0-F)**

A67E = 14 + 7(16) + 6(16)2 + 10(16)3

**Base 5 to base 10:** Multiply by power of original base raised to the power of n.

(14.02)5 = 4 x 50 + 1 x 51 + 0 x 5-1 + 2 x 5-2 = 9.08

**Base 2 to base 10:** Same as last part

(11011.101)2 = 1 x 20 + 1 x 21 + 0 x 22 + 1 x 23 + 1 x 24 + 1 x 2-1 + 0 x 2-2 + 1 x 2-3 = 27.625

**Familiar base to foreign base:** Divide familiar base by unfamiliar base

V(x) = where V(x) is given but x = ?

(125)5 🡪 base 8

125/8 = 15 rem = 5 15/8 = 1 rem = 7 1/8 = 0 rem = 1 (125)5 = (175)8

**Fraction:** Multiply decimal by new base and keep whole number. Remember to keep decimal.

(0.63671875)🡪base 16

(0.63671875) x 16 = 10.1875 10 = A

(0.1875) x 16 = 3.0 3.0 = 3

(0.63671875)10 = (.A3)16

**Base 2 to Octal:** Break into bits of 3 and convert. Keep decimal if fraction.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |

**Base 2 to Hex:** Break into bits of 4 and convert. Keep decimal if fraction

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0000 | 0001 | 0010 | 0011 | 0100 | 101 | 110 | 111 |
| 8 | 9 | A | B | C | D | E | F |
| 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |

**r-1 complement:**

**Given N in radix r:**

**10’s complement:** 9’s complement + 1

**9’s complement:** Subtract each part of your number from 9

12345 = 87654 🡪 10’s complement would be 87655

**1’s complement:** Flip ever bit (all this does is turns a positive number to a negative)

**2’s complement:** Keep all right must bits until the first 1 then flip everything after.

**Unsigned Numbers:** Same as 2’s comp but its not a negative number

Given S, U = ? 🡪 if (s >= 0) then U = S Else 2n + S

Given U, S = ? 🡪 if (U < 2n-1) then S = U Else S = 2n - U

00000001(1) = 11111111(255)

**n-bit:** flip bits of negative number using 1’s comp or 2’s comp

2-3 = 2 + (-3) = 0010 + (1100) = 1110(-1)

2- 3 = 2 + (-3) = 0010 + (1101) = 1111(-1)

**Binary-Coded-Decimal (BCD):** Each number has its own 4-bit binary

236 = (0010 0011 0110)BCD

17 = (0001 0111)BCD = (10001)2 = (11)16 = (21)8

**Excess-3 Codes:** Add 3

|  |  |  |
| --- | --- | --- |
| 0 | 3 | 0011 |
| 1 | 4 | 0100 |
| 2 | 5 | 0101 |
| 3 | 6 | 0110 |
| 4 | 7 | 0111 |
| 5 | 8 | 1000 |
| 6 | 9 | 1001 |
| 7 | 10 | 1010 |

236 = (0101 0110 1001)excess-3 = (763)9’s comp = (1010 1001 0110)1’s comp = (763)excess-3

**Gray Code**

1.) Start with one bit gray code and flip

2.) Mirror those two bits and pad 0’s to the first two and 1’s to the second two

3.) Mirror those four bits and add 0’s to the first four and 1’s to the second four

4.) mirror those eight bits and add 0’s to the first 8 and 1’s to the second eight

**Boolean Algebra**

OR = + AND = \* 1 = TRUE 0 = FALSE

Z = X + Y 🡪 truth table (anything with a 1 is true)

Z = X\*Y 🡪 truth table (anything with both 1’s is true)

Properties of 1 and 0

X+ 0 = X X + 1 = 1 X \* 0 = 0 X \* 1 = X X + not(X) = 1 X \* not(X) = 0

Commutative rule

X + Y = Y + X X \* Y = Y \* X

Associative rule

X + (Y + Z) = (X + Y) + Z X(YZ) = (XY)Z

Idempotency

X + (YZ) = (X + Y)(X + Z) X(Y + Z) = (XY) + (XZ)

Absorption

X + XY = X X + (not(X)Y) = X + Y X(X + Y) = X X(not(X) + Y) = XY

De morgan’s

Not(X + Y) = not(x)not(y)

Not(XY) = not(x) + not(y)

Determine the value of base x if (211)x = (152 )8

(152)8 = 1 x 82 + 5 x 81 + 2 x 80 = (106)10

(211)x = 2 x X2 + 1 x X1 + 1 x X0 = 2X2 + X + 1

2X2 + X +1 = (106)10

Base 7 = 2(7)2 + (7) + 1 = (106)10

**(211)7 = (152)8**