Ch.1 | Digital Systems and binary Numbers

By: Rawan Alfares

Popular Number Systems

1. Binary = Radix 2
   - each digit called bit
   - only two digit 0's and 1's.

2. OCtal = Radix 8
   - only eight digit : 0 to 7
   - Digits 8 and 9 not used.

3. Decimal = Radix 10
   - 10 digits , 0 → 9

4. HexaDecimal = Radix 16
   - 16 digits, 0 → 9 and A → F
   - A = 10 , B = 11 , ... , F = 15.

memorise table.

Decimal Radix 10	Binary Radix 2	Octal Radix 8	Hex Radix 16
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

Bit numbering :

- least Significant Bit [LSB] is Right most. (bit 0)
- Most Significant Bit [MSB] is left most.

most significant number ←

7	6	5	4	3	2	1	0
1	0	0	1	1	1	0	1

→ least significant number

Decimal Value of Binary Number

- each bit represent a power of 2.
- every binary number is represent a power of 2.

to convert from any system to decimal

Radix$^{\text{index}}$ × Value

Binary to Decimal :-

ex :- $(10011101)_2$ ← value

index = 7 6 5 4 3 2 1 0

radix = 2

$$= 2^0 \times 1 + 2^1 \times 0 + 2^2 \times 1 + 2^3 \times 1 + 2^4 \times 1 + 2^5 \times 0 + 2^6 \times 0 + 2^7 \times 1$$

$$= (157)_{10}$$

Octal to Decimal

ex:- $(2107)_8 = (\quad)_{10}$

$\;\;\;\,3\;2\;1\;0$

$8^0 \times 7 + 8^1 \times 0 + 8^2 \times 1 + 8^3 \times 2 = (1095)_{10}$

ex. $(2051)_4$ to decimal

invalid.

Radix = 4, So possible numbers are between $0 \rightarrow 3$

ex. $(3BA4)_{16}$ to decimal $16^0 \times 4 + 16^1 \times 10 + 16^2 \times 11 + 16^4 \times 3$

$= 15268$

NOTE:-

possible number for radix r

is $0 \rightarrow r-1$

ex, $r = 8$, $0 \rightarrow 7$

$r = 2$, $(0, 1)$

$r = 4$, $0 \rightarrow 3$

⋮

Convert from decimal to any system

- Repeatedly divide the decimal integer by the radix of the Sys
- each remainder is a digit in the translated value.
- Stop when quotient is Zero.

ex. Convert $(37)_{10}$ to Binary

radix = 2

37	2
18	1 → LSB
9	0
4	1
2	0
1	0
0	1 → MSB

$(37)_{10} = (100101)_2$

ex. Convert $(422)_{10} \rightarrow (\quad)_{16}$

422	16
26	6 → LSB
1	10
0	1 → MSB

$(422)_{10} = (1A6)_{16}$

* إختصاراً نقسم على Radix حتى يصبح الرقم صفر

* Converting octal and hexadecimal to Binary is easy.

→ $8 = 2^3$ → so each 3 digits equals 1 bit in Binary and vice versa.

→ $16 = 2^4$ → so each 4 digits in hexa equals 1 bit in Binary and vice versa.

7 5 3 0 5 5 2 3 6 2 4	Octal
1 1 1 0 1 0 1 1 0 0 0 1 0 1 1 0 1 0 1 0 0 1 1 1 1 0 0 1 0 1 0 0	32-bit binary
E B 1 6 A 7 9 4	Hexadecimal

* Starting from [LSB] group each 4 Bits into a Hex digit or each 3 bits into one octal digit.

To convert from octal to hexa or vice verse

1. Convert from octal to binary.
2. Convert from binary to hexa.

important properties

* How many possible digits can we have in radix r ?

* what's the result of adding (1) to the largest digit in Radix r ?

ex. $(1)_2 + 1 = (10)_2$ $(7)_8 + 1 = (10)_8$

$(9)_{10} + 1 = (10)_{10}$ $(F)_{16} + 1 = (10)_{16}$

* what's the largest value using 3 digits in Radix r ?

$r^3 - 1$, ex:- In Binary $2^3 - 1 = 7 \rightarrow (111)_2$

ex. In Octal $8^3 - 1$ $(777)_8$

:

Ingeneral :- the largest value using n digit in Radix r is ..

$\text{radix}^n - 1$ n → number of digit

* How many possible values can be represented
  - using n binary digits 2^n values , $0 \rightarrow 2^n - 1$
  - using n octal digits 8^n values , $0 \rightarrow 8^n - 1$
  - using n decimal digits 10^n values , $0 \rightarrow 10^n - 1$
  - using n Hexa decimal digits 16^n values , $0 \rightarrow 16^n - 1$

Ingeneral :- using n digits in Radix

r^n values , 0 to $r^n - 1$.

representing fractions

* Integer values start with index 0.
* fraction values start with index -1.

Convert Decimal Fraction to Binary.

- by multiplying the fraction by 2 Repeatedly.
- collect integer bits.
- stop when the number on the most Right of the point equals 0.

ex. $(0.6875)_{10} = (.1011)_2$

0.6875 x2	1. 3750
0.3750 x2 =	0. 750
0.75 x2 =	1. 50
0.5 x2 =	1. 0

MSB

* integers عكس .

stop when fraction equals zero.

LSB

* عند التحويل من Decimal إلى أي نظام آخر

integers ⇩ Radix قسمة على

fractions ⇩ Radix ضرب في

ex. $(139.6875)_{10} = (\qquad)_8$

2 1 0 -1 -2 -3 -4

139	8
17	3
2	1
0	2

$(213.54)_8$

* نقسم السؤال إلى جزأين
جزء integer ✓
جزء fractions ✓
ثم نجمعهم مع بعض ✓

$0.6875 \times 8 = 5.5000$

$0.5 \times 8 = 4.0$ stop.

Converting Binary and Hexa to Binary 80

← integer: right to left — — fraction: left to right →

7	2	6	1	3	.	2	4	7	4	5	2	**Octal**
0 1 1 1	0 1 0	1 1 0	0 0 1	0 1 1	.	0 1 0	1 0 0	1 1 1	1 0 0	1 0 1	0 1 0 0	**Binary**
7	5	8	B		.	5	3	C	A	8		**Hexadecima**

0100

important properties of fractions

* How many fractional values exist with m fractional Bits?

2^m In general. r^m value

radix

* whats the largest fraction value if m fraction digits are used in radix r?

$1 - r^{-m}$ → number of digits

r → Radix

ex. Convert $(299.8195)_{10} \rightarrow (20B.9A)_{12}$ at most two fractional digits, if necessary.

299	12
24	(11) = B
2	0
0	2

$0.8195 \times 12 = 9.834$

$0.834 \times 12 = 10.008$

arithmetic operation

Binary Addition :-

```
  1 1 1  1
  0 0 1 1 0 1 1 0          (54)
  0 0 0 1 1 1 0 1 +   +    (29) +
  ---------------          ----
  0 1 0 1 0 0 1 1          (83)
```

Binary Subtraction :-

```
    0 2
  1 1 0 1 1
  1 0 1 1 0 -
  ---------
  0 0 1 0 1
```

* لما بنستلف
بناخد 2.

Octal addition :-

```
  1 1
   5 6 7
   2 4 3 +
  -------
  1 0 3 2
```

$7+3=10 > 8$

$10 = 1\times8 + 2$

Carry, carry

$C_o = A_o \times 8 + B_o$

Carry, sum

Octal Subtraction :-

```
    8+1  8+
  5 6 2 4
    2 6 5 -
  -------
    3 3 7
```

* بنستلف 8.

ex.

```
     7  7
   5 8  8  8
   6 0  0  0
     7  7  7 -
  ----------
   5 0  0  1
```

Hexa Addition :-

ex.

```
   1 1
   A D D
   D A D +
  -------
  1 8 8 A
```

A	B	C	D	E	F
10	11	12	13	14	15

$D+D = 13+13 = 26 = 16\times1 + 10$

Carry, Sum

$A+D+1 = 24 = 16\times1 + 8$

Carry, Sum

Hexa Subtraction :-

```
   16   16
   +6   +3   16+
  8 9  7  4  B
    5  8  7  C -
  -------------
    3  E  A  F
```

complement of numbers

"for unsigned numbers"

[1] Radix Complement :-

A- $r^n - N$

B- $(r-1)$'s Comp $+1$

[2] Diminished Radix Complement :

$(r-1)$'s Comp.

$(r^n - 1) - N$, $r \to$ radix

$n \to$ digits

$N \to$ the number

EX. find the 9's Comp of (134795).

865204

1	8
3	6
4	5
7	2
9	0
5	4

* Comp ← معناها مكملة :- فطريقة سريعة , بشوف كل رقم شو مكملته لـ 9.

أو حسب القانون

$r = 10$, $n = 6$, $N = 134795$

1 2 3 4 5 6

$= r^n - 1 - N$ $n \to 6$ digits

$= 10^6 - 1 - 134795$ $N =$ الرقم نفسه

$= 999999 - 134795$ $r =$ radix

$= 865204$

⟹ to find 10's Comp. 9's Comp +1.

find the 10's comp of (134795).

10's Comp = 9's Comp +1

= 865204 + 1

= 865205

* كطريقة سريعة :-

↳ بترك كل الأصفار , وأول رقم بعد الأصفار بكمله لـ (10) , والباقي بكملهم لـ (9)

EX. find 9's and 10's Comp of (546700).

9's Comp = 453299

10's Comp = 463300

5 4 6 7 0 0

تكملة لـ 9 تكملة لـ 10

453300

* to find the 1's comp :- Convert 0's to 1's and 1's to 0's

2's Comp = 1's Comp + 1.

* Remarks :-

The Comp of comp restors the original value ⟹ Comp(Comp(N)) = N

exampels

1. Consider a binary number : 1011001

   1's Complement :- 0100110

   2's complement :- 0100111 (+1)

2. Consider a binary number :- 00100100

   1's complement :- 11011011

   2's complement :- 11011100 (+1)

00100 100

11011 100

flip same

* طريقة ثانية :-
الأصفار وأول رقم بعدهم بضلهم ,,
واللي بعدهم بقلبهم [1←0, 0←1]

Complement are used to simplify Subtraction operation in digital computers.

Advantages of simplification :-

1. it results in Simpler Circuits.
2. it results in low Cost.

which means (fewer and simpler hardware Components).

* Assume that we need to subtract x from y

then we have two Cases :- $\boxed{1}$ $x \geq y$, $\boxed{2}$ $x < y$

$\boxed{1}$ $x \geq y$

* the final answer will be positive

(r-1)'s Comp

x + (r-1)'s Comp

⇓

if their exist Carry add (1) to the result

r's Comp

x + r's Comp.

⇓

if end Carry occurs will be discarded.

ex. X = 11101 , y = 10111
find x − y, using 1's complement.

X > y , = X − y

= X + 1's (y)

= 11101 + (01000)

= 100110

```
  11101
  01000   +
  ------
 100101
      1   +
  ------
  00110  → the final answer
```

end carry, add it to the result.

ex. X = 11101 , y = 10111
find X − y by 2's comp

= x + 2's (y)

```
= 11101
  01001   +
  ------
1 00110  → the final answer
```

1 will be discarded

NOTE:-

* إذا كانت بدي fraction
بتعامل معها انه رقم
صحيح وبعدها برجع
ال point على نفس
مكانها.

ex:- 10110.00
1's Comp = 01001.11

ex:- 0101.101
2's comp = 101.011

```
0101101
1010010   1's comp
1010011   +1
```

2 X < y → ① the final answer will be negative
② won't produce end carry.
③ after taking the comp of the final carry, put the negative sign.

ex. using 10's comp. to 3250 − 72532

$= 03250 + (27468)$

$$
\begin{array}{r}
{}^{1}03\overset{1}{2}50 \\
+\ 27468 \\
\hline
30718
\end{array}
$$

30718 → no end carry

10's(30718) = −69282

ex. using 9's comp. do 285.31 − 3459.20

$= 285.31 + (6540.79)$

$$
\begin{array}{r}
\overset{1}{2}\overset{1}{8}5.\overset{1}{3}1 \\
+\ 6540.79 \\
\hline
6826.10
\end{array}
$$

6826.10 → no end carry.

9's(6826.10) = −3173.89

signed binary numbers

3 major techniques are used to represent signed numbers :-

1. Signed magnitude.
2. 1's Complement.
3. 2's complement.

[1] Signed magnitude :-

- left most bit is the sign bit :- 0 for $\oplus^{ve}$
  1 for $\ominus^{ve}$
- two represntation for Zero (NOT Good)!
- Range $-2^{(n-1)}-1 \longrightarrow +(2^{n-1}-1)$.
- exampels :-

$-1 \xrightarrow{4\text{ bits}}$ 1001

$+6 \xrightarrow{4\text{ bit}}$ 0110 positive

$+15 \xrightarrow{8\text{ bit}}$ 00001111

$-79 \xrightarrow{8\text{ bit}}$ 11001111

negative

- the only way, that represent $\oplus^{ve}$ and $\ominus^{ve}$.

[2] 1's complement.

- 2 representation for 0.
- Rang $-2^{n-1}-1 \longrightarrow +2^{n-1}-1$

الحل :- ① نكتب الرقم بالموجب

② ناخذ لها 1's Comp له.

ex. -16 $+16$ ①

0 10000

1's comp ↳ 1 01111 -16

(−)

[3] 2's Complement.

- only one representation for 0.
- Range $-2^{n-1} \rightarrow +2^{n-1}-1$

[1] بكتب الرقم بالموجب (آخر bit صفر)

[2] 1's comp

[3] + 1.

- 2's Comp = 1's Comp + 1

بنرجع ناخد 2's Comp عشان نجيب القيمة الأصلية.

- only for negative number.

ex. -16

+16 = 010000

↳ 1's comp = 101111

+1 = 110000

- $2's\,Comp(N) + N = 0$

* negative weight to the sign bit (MSB)

ex. 10110100

-128 64 32 16 8 4 2 1

4 + 16 + 32 + -128 = -76

* Range values for unsigned number $0 \rightarrow +(2^{n}-1)$

* Range values for signed number $-2^{n-1} \rightarrow +2^{n-1}-1$ } according to 2's Comp.

* positive integers $0 \rightarrow 2^{n-1}-1$

* negative integers $-2^{n-1} \rightarrow -1$

8 bits $-128 \rightarrow 127$

* Signed magnitude

$-2^{n-1}-1 \rightarrow +2^{n-1}-1$

ex. A = (00101100) = +44

↳ 2's (11010100) = -44

* Subtraction :

↳ Convert Sub. into addition and get the 2's Comp of the second # with ignoring the carry.

* over flow occurs :-
1. adding two positive numbers and the sum is neg.
2. adding two negative numbers and the sum is pos.

ex.

```
   1111000
 [1]0000111
 + 1111000
 ----------
  00000111
```

1 carry

```
  100110000
   11011010
 + 10011101
 ----------
   01110111
```

جمعت (−) مع (−)
أعطى (+)

* بما إنه رقمين مختلفين يعني يوجد over flow

* اذا طلعوا متشابهين يعني لا يوجد over flow

* Carry is important when
- Adding or Sub, unsigend numbers.
- indicates that the unsigned sum is out of range.

* اذا عندي رقم مخزن 2's comp وبدي أعرف قيمته

ex: 1101 → 2's comp.

```
 -   0010   1's comp
       1+
     0011
```

= -3

ex. 00100100 = +36

```
  11011011
        1+
 ----------
 (11011100) = -36
```

ex: 1001 → 2's comp

```
    0110   1's comp
      +1
 -  0111   = -7
```

ex. (+5) → in 8 bits

```
 00000101   1's comp
 11111010   +1
 11111011
```

* Positive numbers in signed magnitude and 2's comp. are the same.

* the 2's Comp of N is the negative of N
* the sum of N and the 2's Comp of N is Zero.

BINARY CODES

- do representation
- $\lceil \log_2 \square \rceil$

* لو عندي 20 لون وبدي أميز كل لون عن الآخر بعدد البتات. كم bit بحتاج ؟

$$\lceil \log_2 20 \rceil = 5$$

أو

$$16 < 20 < 32$$

$$2^4 < 20 < 2^5$$

the minimume number of bits.

ex. if you have 3 bits how many numbers Can you do ?

$$2^3 = \underline{8}$$

$2^n \rightarrow n$: number of bits.

Decimal Codes :-

$$\lceil \log_2 10 \rceil = \underline{4}$$

10 char 4bit

* عشان نمثل 10 char بنحتاج 4bit

Decimal	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
Unused	1010 ... 1111

Conversion And Coding

$(13)_{10} = (1101)_2$, this is Conversion. , 4 bit needed

$13 = (0001\ 0011)_{BCD}$, this is coding , 8 bit needed

* In general, Coding requires more bits than Conversion

* كل رقم في BCD بعبّر عنه لحاله

* أكبر عدد في BCD هو 9 (1001)

لكن

إذا طلع الجواب أكبر من 9

بضيف عليه 6

```
  8            BCD
  5 +          1000
 13            0101 +
              11101
               0110 +
          0001 0011
             1    3
```

```
ex. 1897       0001  1000  1001  0111
    2905 +     0010  1001  0000  0101  +
               0100 10010  1010  1100
               0000  0110  0110  0110  +
               0100  1000 10000 10010
                 4     8     0     2
```

Gray Code

↳ successive values differ in only one bit

used :- sequence of binary numbers numbers generated by the hard ware that may cause an error or ambiguity during the transition from one number to the next.

from binary to gray

1 0 0 1 1 0 Binary Code

1 1 0 1 0 1 gray Code

from gray to binary

1 1 0 1 0 1 gray

1 0 0 1 1 0 Binary

1. The Gray code for the binary value (100110) is Answer 1, and this code and the Gray code (110111) Answer 2 (are/are not) successive codes. (Fill in the blank)

Answer 1

Type your answer 110101

Answer 2

Type your answer yes

other decimal codes

Decimal	BCD 8421	5421 code	2421 code	8 4 -2 -1 code	Excess-3 code
0	0000	0000	0000	0000	0011
1	0001	0001	0001	0111	0100
2	0010	0010	0010	0110	0101
3	0011	0011	0011	0101	0110
4	0100	0100	0100	0100	0111
5	0101	1000	1011	1011	1000
6	0110	1001	1100	1010	1001
7	0111	1010	1101	1001	1010
8	1000	1011	1110	1000	1011
9	1001	1100	1111	1111	1100
Unused	···	···	···	···	···

* BCD 5421, 2421, 8 4 -2 -1 are wieghted codes

* Excess-3 is not weighted code.

Excess-3

Decimal → BCD $\xrightarrow[+\,0011]{}$ Excess-3

1. The decimal number 17 can be represented in binary as Answer 1 and in Excess-3 as Answer 2. (Fill in the blank)

Answer 1

Type your answer 10001

Answer 2

Type your answer 0010 0000

17 $\xrightarrow{BCD}$ 0001 0111

```
  0001 0111
+      0011
-----------
  0001 1010
+      0110
-----------
  0010 0000
     2    0
```

ex. $(27)_{10} + (39)_{10}$

```
  0010 0111          0011 1001
+      0011    |   +      0011
-----------    |   -----------
  0010 1010    |     0011 1100
+      0110    |   +      0110
-----------    |   -----------
  0011 0000    |     0100 0010
                        42
```

30 + 42 = 72

ربنا تقبل منا إنك أنت السميع العليم

روان فارس