

## Number Systems

- الأرقام العشرية (decimal numbers): لأرقام العشرية هي ضمن النطاق التالي [0, 1, 2, ..., 9]

هذه للخانة الواحدة لكن من الممكن أن يتم جمعها مع بعض وتصبح أكبر مثل:

"63638212"

كل عامود من هي الأرقام له وزنه نحن نقرأ من اليمين إلى اليسار لحساب الأعمدة هي :

10

100

1000

etc .. 10000

وهون الاساس أو base هو الرقم "10" لأن الأرقام من 0 < 9

فالأساس العشري هو 10 يعني رح نعرف الأرقام هي

$10^{63638212}$

يمثل الرقم العشري المكون من الرقم n (N-digit decimal number)

يمثل أحد الاحتمالات (0, 1, 2, ...,  $10^{n-1}$ ) وهاد اسمو نطاق الرقم

مثال: الرقم الممثل من 3 أرقام عشرية مثل رقم في احتمال 1000 النطاق الخاص فيه [0, ..., 999].

- الأرقام الثنائية (binary number): هو الرقم المكون من 0 و 1 وبالاخير بيتضمنوا مع بعضن مشان يكونوا رقم مثل:

01001101

أساس الرقم الثنائي هو "2" (N-bit binary)

ومشان نحسبوا بطريقة سهلة هو مضاعفات رقم 2-

1, 2, 4, 8, 16, 32, 64, 128, 256, ...

طبيب، كيف منحول الرقم الثنائي لـ عشري ؟

$10^{10} < (?)^2$

$N\text{-bit} = N\text{-bit} * \text{weight}^{\text{column}}$

منبلش من اليمين:  $10^{10} = 3(2^1) + 2(2^0) + 1(2^1) + 0(2^0)$

- الأرقام الست عشرية (hexadecimal numbers) اختصارا "hex": الرقم السداسي العشري مكون من أربع بتات --  $2^4 =$

16

أرقام الستة عشري تتمثل [F -> A & 9 -> 0]

والأساس الخاص فيه هو 16

DECIMAL	HEX	BINARY
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

طبيب ليش منتعلم صيغ الأرقام ؟ بدون فهم الصيغ يكون صعب فهم كيفية عمل الـ hardware أو البرمجيات منخفضة المستوى الـ low level وهيك ما رح تقدر تطور نظام رقمي أو تقدر تحللو.

### Converting:

\* من ستة عشري إلى ثنائي:

$$(4AF)^{16} \rightarrow (?)^2$$

مناخذن رقم رقم

$$^2(0100) = ^{16}(4)$$

$$(A)^{16} = (1010)^2$$

$$(F)^{16} = (1111)^2$$

$$(4AF)^{16} = (0100\ 1010\ 1111)^2$$

\* من ستة عشري إلى عشري:

$$(4AF)^{16} = (?)^{10}$$

من اليمين يعم الحج:  $(15 \cdot 16)^0 + (10 \cdot 16)^1 + (4 \cdot 16)^2 = 15 + 160 + 1024 = (1199)^{10}$

طبعاً هون عوضنا الـ F بـ 15 و الـ A بـ 10

## Bits, Nibbles and Bytes - Binary

مثل ما حكينا من قبل الـ bit هو عبارة عن 0 أو 1

الـ Byte = 8bits

ويمكن ان يمثل  $2^8 = 256$  احتمال

$$1111\ 1111 = 1 - 2^8$$

الـ Nibble = 4bits يعني نص Byte

ويمكن أن يمثل  $2^4 = 16$  احتمال

$$1111 = 1 - 2^4$$

ملاحظة: لتخزين حرف بالـ hex يحتاج 4bits يعني nibble

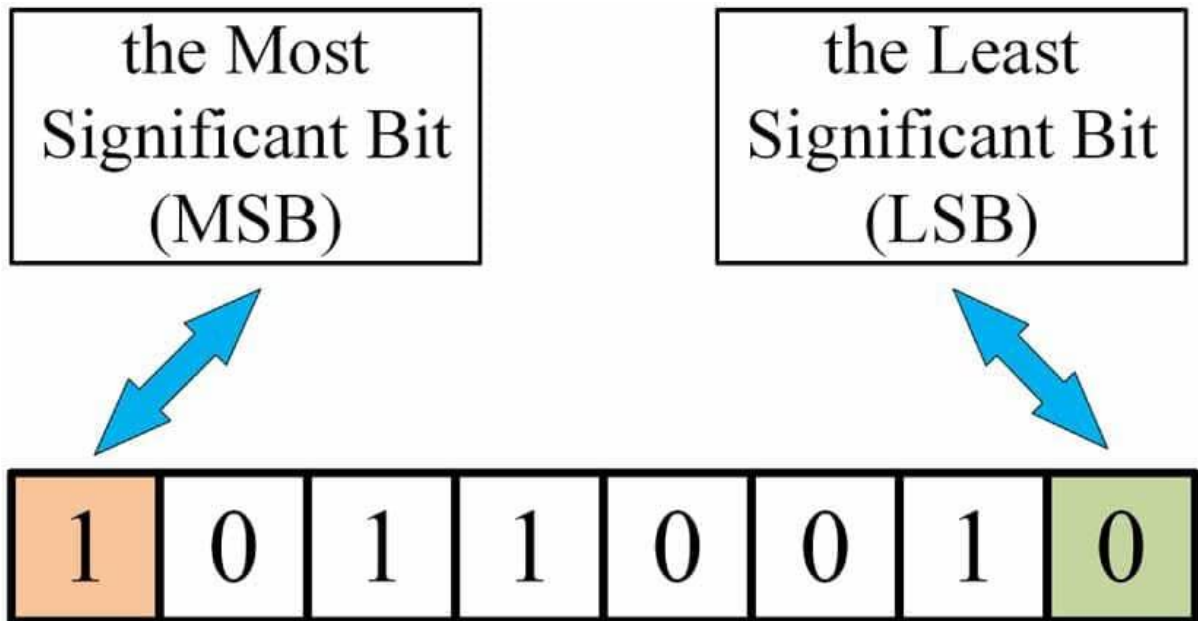
الـ **Micro processors** يتعامل مع البيانات بأجزاء اسمها "word"

حجم الـ **word** يعتمد على بنية (معمارية) المعالج مثلاً: إذا كان processor-64 سيكون  $\text{word} = 8\text{bytes} = 64\text{bits}$

## LSb & MSb

- البت الموجود بالعامود الأول اسمه (LSb): Least Significant bit

- البت الموجود بالعامود الأخير اسمه (MSb): Most Significant bit



وحدات القياس (Unit Prefix)

-كيلو بايت | kilo :

$$Kb1 = 1024 = 2^{10}$$

-ميغا بايت | mega:

$$Mb1 = 1,048,576 = 2^{20}$$

-غيغا بايت | giga:

$$Gb1 = 1,073,741,824 = 2^{30}$$

-تيرا بايت | tera:

$$Tb1 = 1,099,511,627,776 = 2^{40}$$

## Binary Addition

الجمع في الثنائي ببشبه الجمع بالعشري

$$+1011$$

$$=0011$$

$$1110$$

$$\text{الـ } 1+1 = \text{بنحط باليد واحد}$$

$$0 = 0+0$$

$$1 = 0+1$$

يمكن نتعامل بعدد ثابت ومساوي للمطروح أو المجموع لكن إذا كان غير مساوي أو اكبر هون بصير شي اسمه overflow إذا صار بعد الجمع زيادة بالأرقام.

## Signed-Binary Numbers

نحننا كنا بنتعامل مع الأرقام الموجبة من الصفر وما فوق ، طيب بالعمليات الحسابية الطبيعية في سالب كمان؛

في أكثر من طريقة بالباينري الأنظمة المختلفة بتستخدمها مشان تمثل الأعداد السالبة مثل: complement, sign-magnitude

## Sign-magnitude Numbers

الـ MSb سيكون هو إلى بيدل عالاشارة والبتات الأخرى هي إلي بنمثل الحجم والرقم.

إذا كان الـ MSb = 0 معناها الإشارة -

وإذا كان 1 يعني الإشارة +

مثال:

$$(0)011 = -3$$

$$(1)011 = +3$$

بس هي الطريقة فيها سلبيات منها عدم القدرة على الجمع والطرح.

## Complement Numbers\*

كمان الـ MSb هو إلى بيدل عالاشارة

إذا كان 0 : +

وإذا كان 1 : -

نتحقق من الموجب والسالب من خلال طرحهما والسلبى يكون '1'

الخطوات:

-أول الشي الصفار كلها بتنزل حتى نشوف رقم 1 مثال:

$$^2(100100)$$

$$(xxxx00)^2$$

-بينزلو بس صفرين

$$^2(00)$$

-وبعدها منزل الواحد مثل ما هو

$$(xxx100)^2$$

-آخر الشيء منعكس الأرقام البقية وهيكل بصير الرقم:

$$^2(011100)$$

طريقة ثانية:

اسمها taking the tow's Complement

أول الشيء بنعكس الأرقام وبعدين منضيف 1 على الـ LSB

مثال:

$$^2(0010)$$

هيكل بصير:

$$^2(1101)$$

منضفלו واحد:

$$^2(1110) = 1 + ^2(1101)$$

هاد أهم شي بيلزمنا من الـ Numbers System