CSE100
الحاسبات والبرمجة 1

د/ محمد نورعبدالجواد

mnahmed@eng.zu.edu.eg

https://mnourgwad.github.io/CSE100

المحاضرة 3 : تمثيل البيانات داخل الحاسب II

Copyright ©2016 Dr.Ing. Mohammed Nour Abdelgwad Ahmed as part of the course work and learning material. All Rights Reserved.
Where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

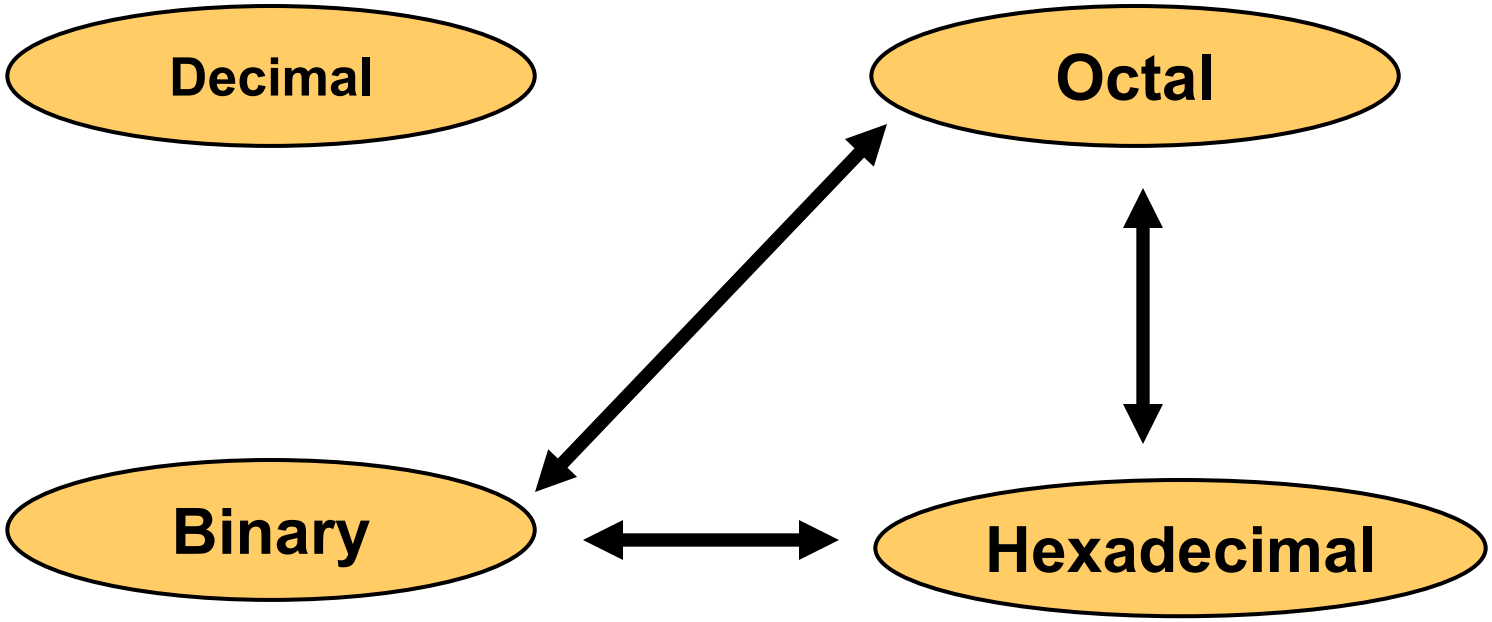
المحاضرة الثالثه

الأهداف لليوم

- التحويل بين الأنظمة العددية عدا العشري.
- تمثيل الارقام السالبة
- العمليات الحسابيه بالنظام الثنائي
- تمثيل الأعداد الكسريه (النقطه المعومه)

التحويل بين الأنظمه الأخري
عدا العشري

التحويل بين تلك الأنظمه العددية



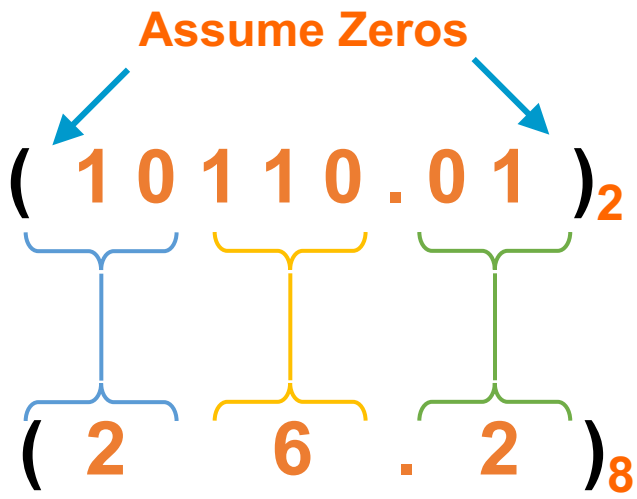
التحويل الثنائي – الثماني

- الثنائي / الثماني: يتم التقسيم في مجموعات من 3

$$(10\ 111\ 011\ 001\ .\ 101\ 110)_2 = (2731.56)_8$$

- $8 = 2^3$
- Each group of **3** bits represents an octal digit

Example:



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

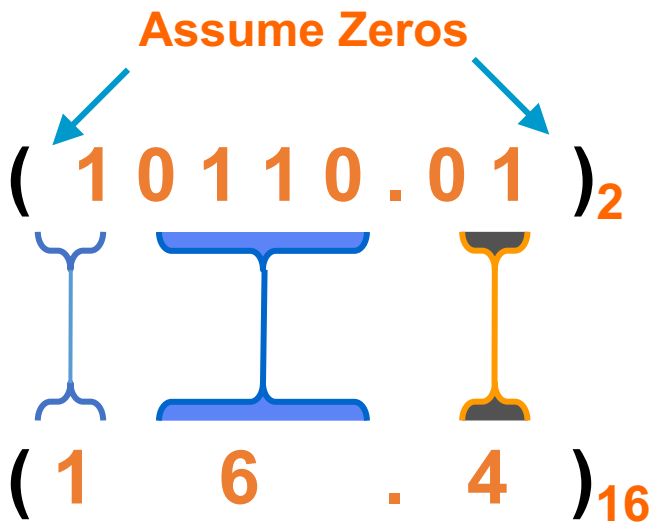
التحويل الثنائي – السداسي عشر

- الثنائي / السداسي عشر: يتم التقسيم في مجموعات من 4

$$(101\ 1101\ 1001\ .\ 1011\ 1000)_2 = (5D9.B8)_{16}$$

- $16 = 2^4$
- Each group of **4** bits represents a hexadecimal digit

Example:

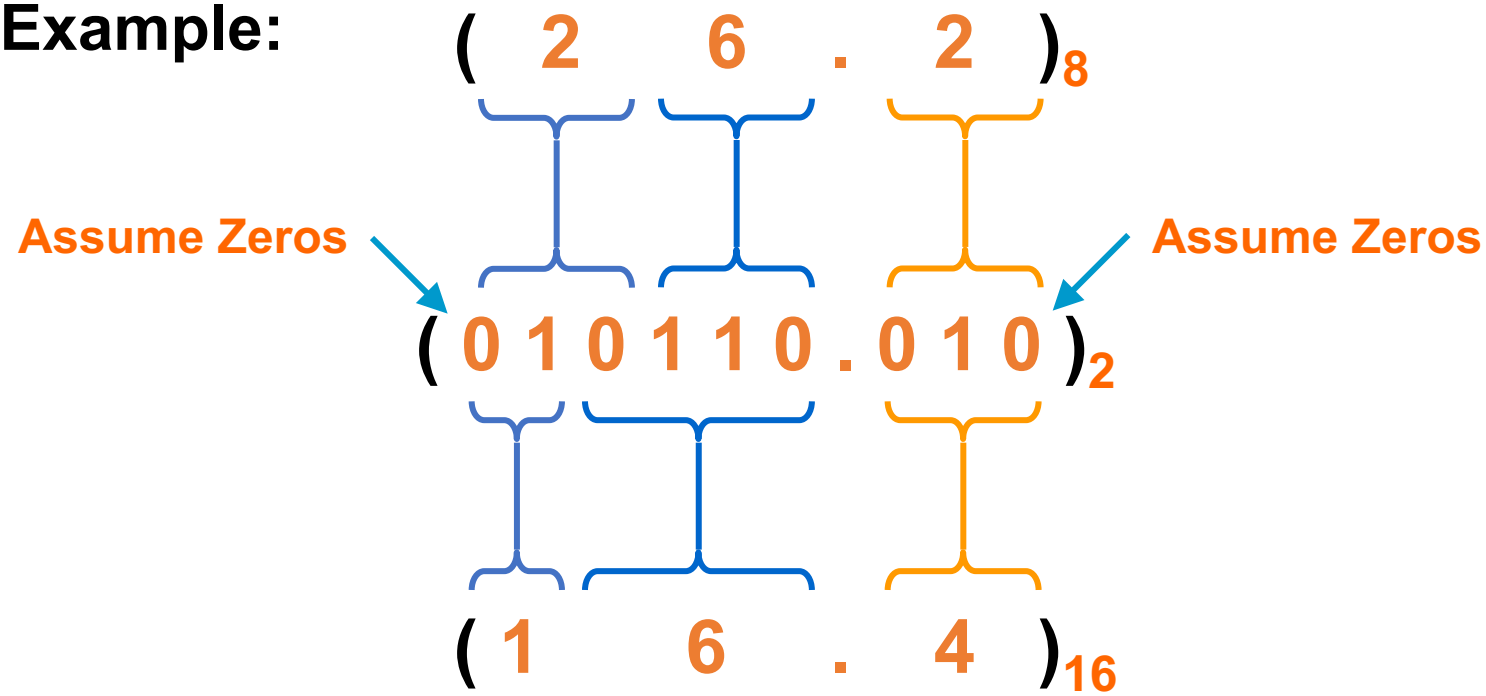


Hex	Binary
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
A	1 0 1 0
B	1 0 1 1
C	1 1 0 0
D	1 1 0 1
E	1 1 1 0
F	1 1 1 1

التحويل الثماني – السداسي عشر

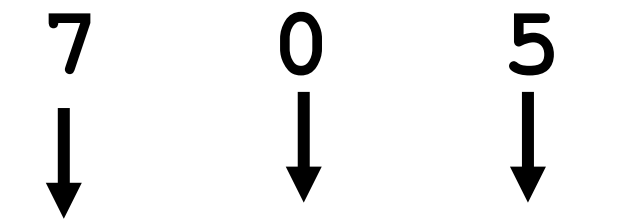
- Convert to **Binary** as an intermediate step

Example:



مثال

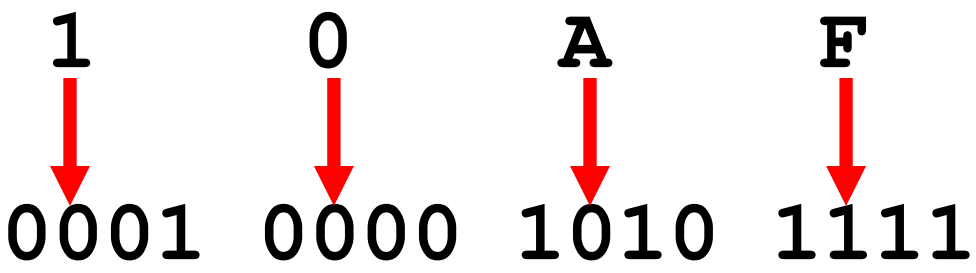
$705_8 = ?_2$



$705_8 = 111000101_2$

مثال

$10AF_{16} = ?_2$



$10AF_{16} = 0001000010101111_2$

مثال

$1076_8 = ?_{16}$

1	0	7	6
001	000	111	110
2	3		E

$1076_8 = 23E_{16}$

مثال

$1F0C_{16} = ?_8$

1	F	0	C
0001	1111	0000	1100
1	7 4	1	4

$1F0C_{16} = 17414_8$

التحويل من الثنائي – الثماني / السداسي عشر

- الثنائي / الثماني: يتم التقسيم في مجموعات من 3

$$(10\ 111\ 011\ 001\ .\ 101\ 110)_2 = (2731.56)_8$$

- الثماني / الثنائي : إعكس الوضع السابق

$$(2731.56)_8 = (10\ 111\ 011\ 001\ .\ 101\ 110)_2$$

- الثنائي / السداسي عشر: يتم التقسيم في مجموعات من 4

$$(101\ 1101\ 1001\ .\ 1011\ 1000)_2 = (5D9.B8)_{16}$$

- السادسي عشر / الثنائي: عكس

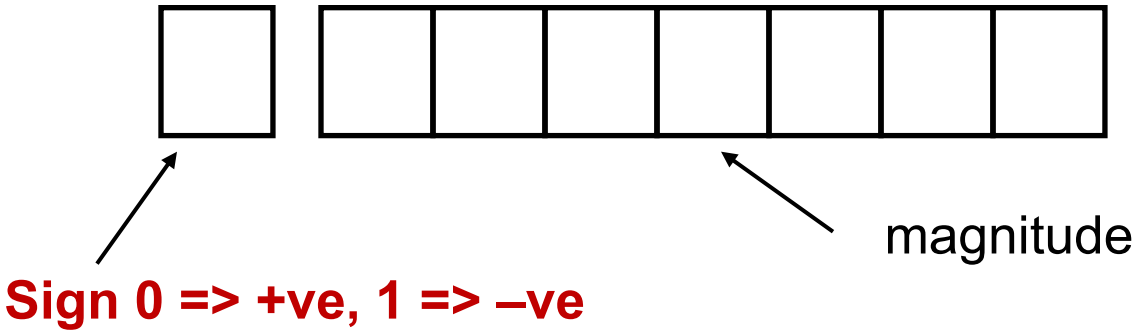
$$(5D9.B8)_{16} = (101\ 1101\ 1001\ .\ 1011\ 1000)_2$$

تمثيل الأعداد
الموجبه والسالبه

تمثيل الأعداد الموجبة والسالبة

طريقة الإشارة والقيمة

8 bits binary number



+7 = (**0** 0 0 0 0 1 1 1)$_2$

–7 = (**1** 0 0 0 0 1 1 1)$_2$

–10 = (**1** 0 0 0 1 0 1 0)$_2$

+10 = (**0** 0 0 0 1 0 1 0)$_2$

الأرقام السالبة: متمم الواحد

متمم الواحد : $(-x = 2^n - x - 1)$

مثال: حول **(-12)** باستخدام متمم الواحد ممثلا بـ **8 bit**

1) حول **+12** ممثله بـ **8bits** <== **00001100**

2) **يتم تغيير كل بت من صفر لواحد والعكس.**

$$-(00001100)_2 = -(12)_{10}$$
$$= (2^8 - 12 - 1)_{10}$$
$$= (243)_{10}$$
$$= \mathbf{(11110011)_{1s}}$$

الأرقام السالبة: متمم الأثنين

متمم الأثنين: $(-x = 2^n - x)$

مثال: حول (12-) باستخدام متمم الأثنين ممثلا بـ 8 bit

الحل

1) حول 12+ ممثله بـ 8bits ==> 00001100

2) **يتم تغيير كل بت من صفر لواحد والعكس.**

3) **يتم جمع 1**

$$-(00001100)_2 = -(12)_{10}$$
$$= (2^8 - 12)_{10}$$
$$= (244)_{10}$$
$$= (11110100)_{2s}$$

في حالة لدينا ***n*-bit**

ما هو المدي من القيم (الأرقام) التي تستطيع تمثيله؟

مدي تمثيل الأرقام الموجيه والسالبه

	أكبر رقم موجب	أكبر رقم سالب
الأشاره و القيمه Sign & Mag.	$(2^{n-1}-1)$	$-(2^{n-1}-1)$
متمم الواحد 1's Comp.	$(2^{n-1}-1)$	$-(2^{n-1}-1)$
متمم الأثنين 2's Comp.	$(2^{n-1}-1)$	**$-(2^{n-1})$**

العمليات الحسابيه
بالنظام الثنائي

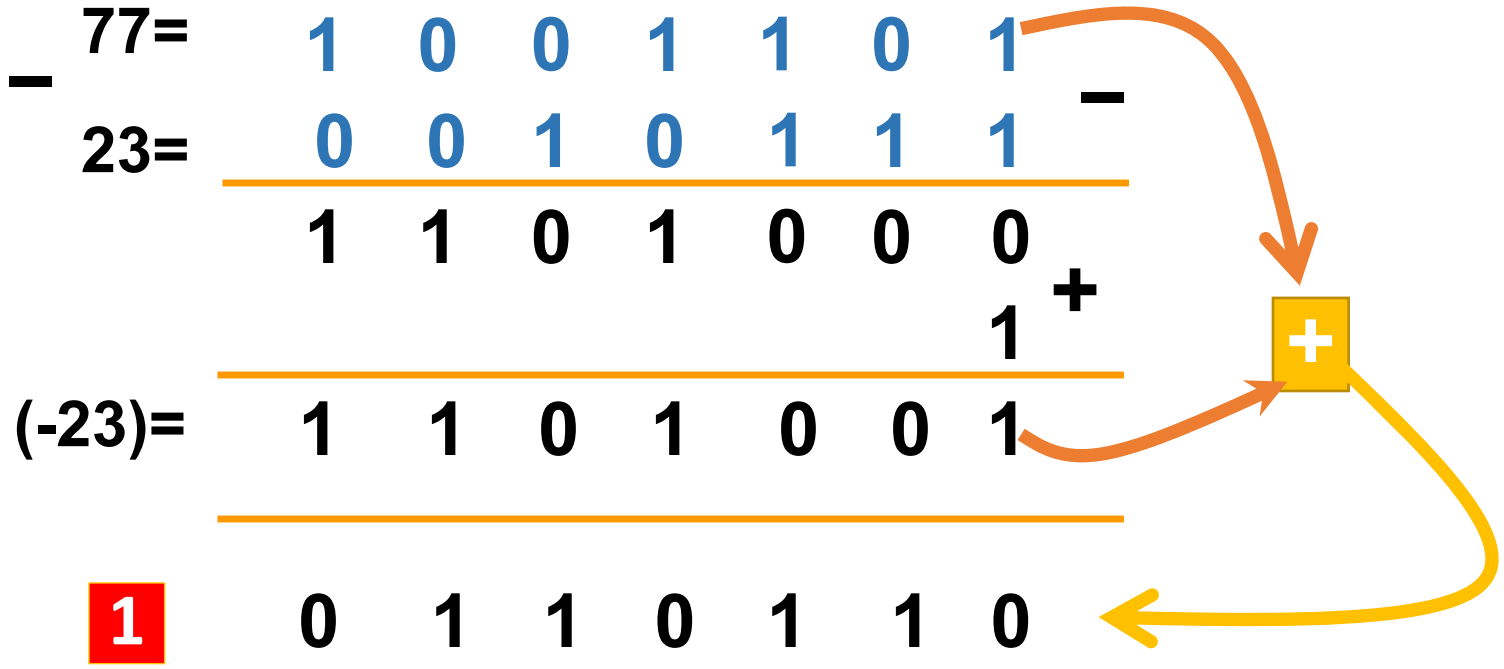
الجمع- الطرح/(متمم الأثنين)

- خطوات الجمع :A + B
  1. قم بالجمع الثنائي للرقمين
  2. يتم إهمال الباقي إذا وجد

- خطوات الطرح: A – B = A + (–B)

1. قم بالحصول علي متمم الاثنين للرقم B من خلال **قلب** قيمة كل خانه ثم إضافة 1
2. قم بجمع متمم الاثنين الخاص بـ B إلي A.

العملية باستخدام 2’s comp.:



الجمع- الطرح/(متمم الأثنين)

- أمثله: قم بإجراء العمليات الحسابيه التاليه ممثله بـ 4 بت

-2	1110
+ -6	+ 1010
----	-------
-8	11000

+3	0011
+ +4	+ 0100
----	------
+7	0111

+4	0100
+ -7	+ 1001
----	-------
-3	1101

+6	0110
+ -3	+ 1101
----	------
+3	10011

الجمع- الطرح/(متمم الواحد)

- خطوات الجمع: A + B

1. قم بالجمع الثنائي للرقمين
2. قم بإضافة الباقي إذا وجد علي الناتج.

- خطوات الطرح A – B = A – B = A + (–B)

1. قم بالحصول علي متمم الواحد لـ B من خلال قلب كل خانه.
2. قم بجمع متمم الواحد الخاص بـ B إلي A.

الجمع- الطرح/(متمم الواحد)

- أمثله: قم بإجراء العمليات الحسابية التالية ممثله بـ 4 بت

```
  +5        0101
+ -5      + 1010
----      -------
  -0        1111
```

```
  +3        0011
+ +4      + 0100
----      -------
  +7        0111
```

```
  -3        1100
+ -7      + 1000
----      -------
 -10       10100
----      +    1
                -------
                  0101
```

```
  -2        1101
+ -5    +   1010
----        ------
  -7        10111
----      +    1
                ------
                 1000
```

تمثيل الأعداد الكسريه
(النقطه المعومه)

تمثيل الأعداد الكسرية

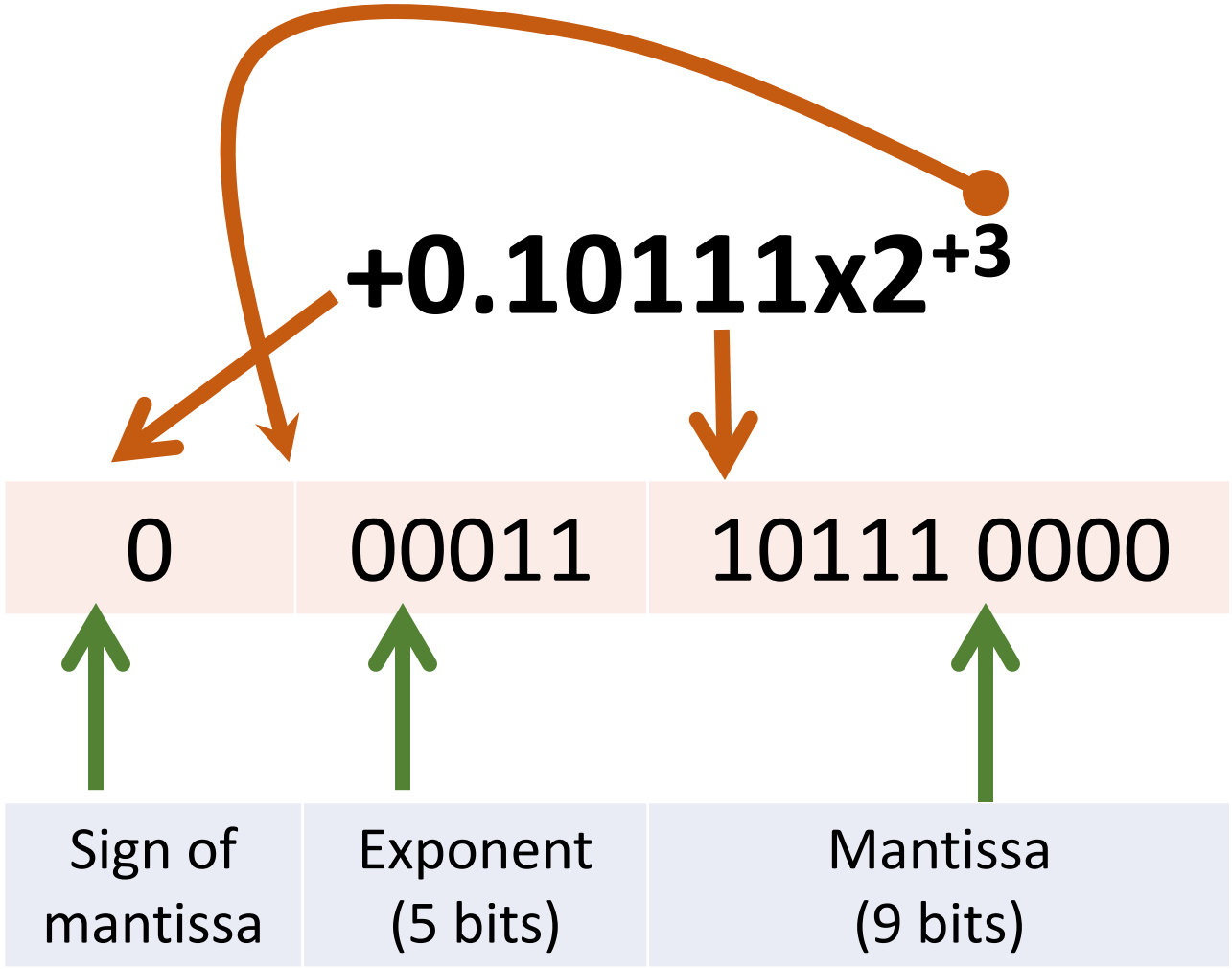
مثال:

$$1256.3 = 125.63 \times 10^{1}$$
$$= 12.563 \times 10^{2}$$
$$= 1.2563 \times 10^{3}$$
$$= 0.12563 \times 10^{4}$$

إشارة الرقم	تمثيل الأس بإشارته	قيمة الرقم
+	+4	12563
Sign of mantissa	Exponent	Mantissa

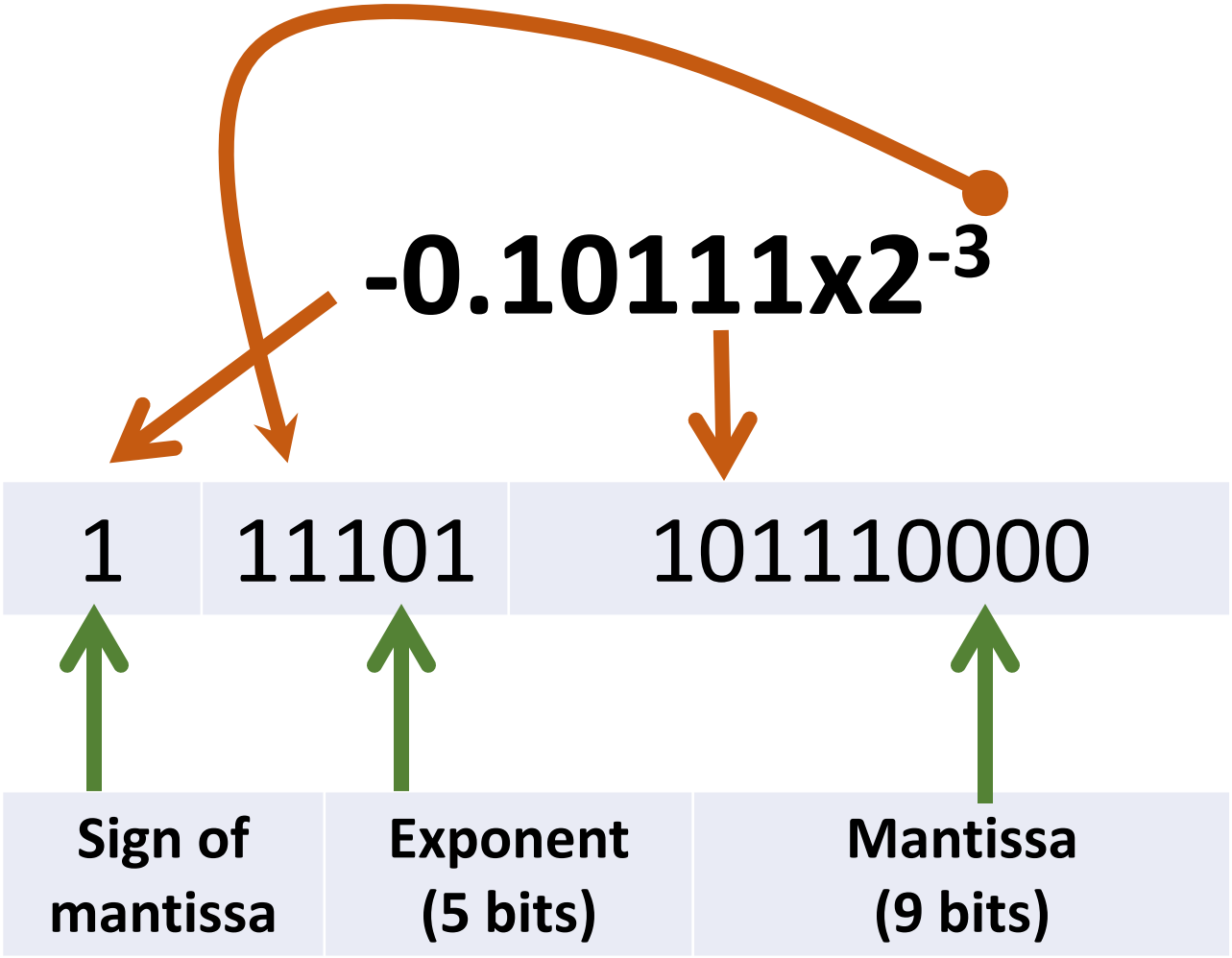
تمثيل الأعداد الكسريه:
طريقة النقطه المعومه (Floating Point)

مثال:



تمثيل الأعداد الكسريه:
طريقة النقطه المعومه (Floating Point)

مثال:



تمثيل الأعداد الكسريه:

طريقة النقطه المعومه (Floating Point)

مثل العدد 100.375 (بالنظام العشري) إلى مثيله بالنظام الثنائي و ذلك بطريقة النقطة المعومة في كلمة طولها 2 حرف (2 bytes) على أن تحجز ست خانات لتمثيل الأس . إحسب قيمة الخطأ المطلق الحادث ! اقترح اختيار أفضل لعدد خانات تمثيل الأس ! و ذلك للحصول على تمثيل أدق . ثم احسب قيمة الخطأ المطلق في هذه الحالة .

تمثيل الأعداد الكسريه:

$100.375 = 1\ 100\ 100\ .\ 011$

$0.110\ 010\ 001\ 1 \times 2^{7}$

	خانة الاس						خانة الـ Mantessa								
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	1	1	1	1	1	0	0	1	0	0	0	1

قيمة العدد الممثل هي **100.25** وبناء عليه فإن الخطأ المطلق = **0.125**

لتحسين ذلك الخطأ:

- نقترح تخفيض عدد خانات الأس لتصبح **5** بدلا من **6**
- ثم زيادة عدد خانات تمثيل الرقم لتصبح **10** بدلا من **9**

BCD AND ASCII CODE

Binary Codes

- "An *n*-bit *binary code* is a group of *n* bits that assume up to 2^n distinct combinations of 1s and 0s, with each combination representing one element of the set being coded"

- For the 10 digits need a 4 bit code. One code is called Binary Coded Decimal (BCD)

Binary Coded Decimal (BCD)

- The BCD is simply the 4 bit representation of the decimal digit.
- For multiple digit base 10 numbers, each symbol is represented by its BCD digit
- What happened to 6 digits not used?

Decimal Symbol	BCD Digit
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Binary Coded Decimal (BCD)

- BCD for 417 to 195

$$(417)_{10} = (0\ 1\ 0\ 0 \quad 0\ 0\ 0\ 1 \quad 0\ 1\ 1\ 1)_{BCD}$$

$$(195)_{10} = (0\ 0\ 0\ 1 \quad 1\ 0\ 0\ 1 \quad 0\ 1\ 0\ 1)_{BCD}$$

Alphanumeric Codes

How do you handle alphanumeric data?

- Formulate a binary code to represent characters! ☺
- For the 26 letter of the alphabet would need 5 bit for representation.
- But what about the upper case and lower case, and the digits, and special characters

A code called ASCII

- ASCII stands for American **Standard Code for Information Interchange**

- The code uses 7 bits to encode 128 unique characters

As a note, formally, work to create this code began in 1960. 1st standard in 1963. Last updated in 1986.

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	Start of Header	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	Start of Text	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	End of Text	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	End of Transmission	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	Enquiry	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	Acknowledgment	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	Bell	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	Backspace	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	Horizontal Tab	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	Line feed	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	Vertical Tab	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	Form feed	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	Carriage return	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	Shift Out	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	Shift In	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	Data Link Escape	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	Device Control 1	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	Device Control 2	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	Device Control 3	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	Device Control 4	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	Negative Ack.	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	Synchronous idle	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	End of Trans. Block	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	Cancel	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	End of Medium	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	Substitute	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	Escape	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	File Separator	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	Group Separator	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	Record Separator	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	Unit Separator	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	Del

asciitbl.com

ASCII Code

- Represents the **numbers**
  - All start 011 xxxx and the xxxx is the BCD for the digit
- Represent the characters of the **alphabet**
  - Start with either 100, 101, 110, or 111
- A few **special** characters are in this area
  - Start with 010 – space and !”#$%&’()*+.-,/
  - Start with 000 or 001 – control char like ESC

ASCII Example

- Encoding of 123

  011 0001 011 0010 011 0011

- Encoding of Joanne

  100 1010 110 1111 110 0001

  110 1110 110 1110 110 0101

- Note that these are 7 bit codes

What to do with the 8th Bit?

- In digital systems data is usually organized as bytes or 8 bit of data.
- How about using the 8th bit for an error coding. This would help during data transmission, etc.
- Parity bit – the extra bit included to make the total number of 1s in the byte either even or odd – called even parity and odd parity

Example of Parity

- Consider data 100 0001
  - Even Parity 0100 0001
  - Odd Parity 1100 0001

- Consider data 101 0100
  - Even Parity 1101 0100
  - Odd Parity 0101 0100

- A parity code can be used for ASCII characters and any binary data.