

المعالجات الصغيرة و لغة التجميع

محاضرة 1

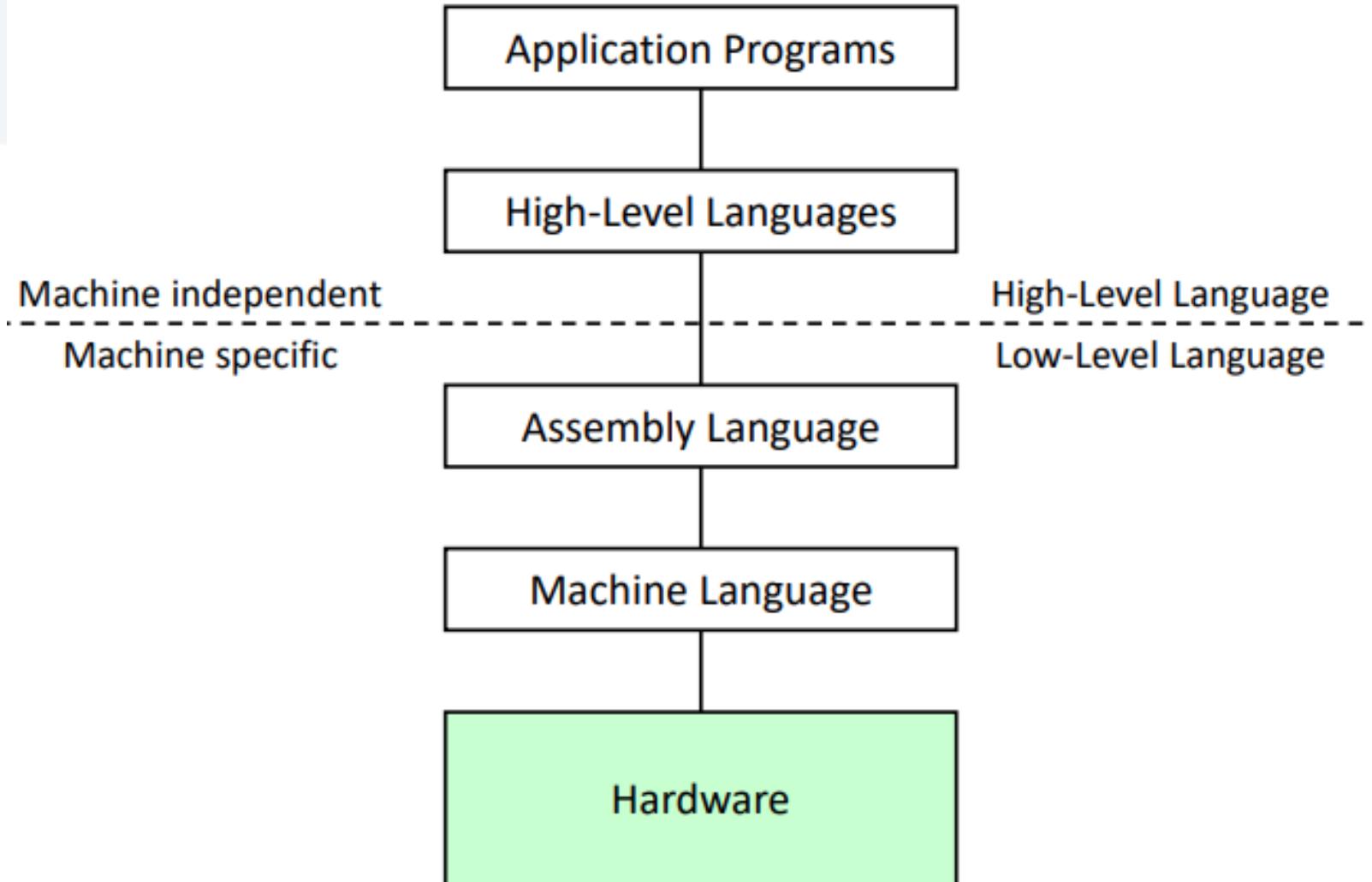
تمثيل المعطيات و مراجعة في نظم العد

د. فادي متوج

# مفردات المقرر

- تمثيل المعطيات و مراجعة في نظم العد
- معمارية المعالج INTEL 8086
- مجموعة تعليمات المعالج 8086
- الذاكرة والتواصل مع وحدات الإدخال والإخراج
- المقاطعات INTERRUPTS
- البرمجة بلغة التجميع
- تصميم الأنظمة بالاعتماد على المعالج 8086

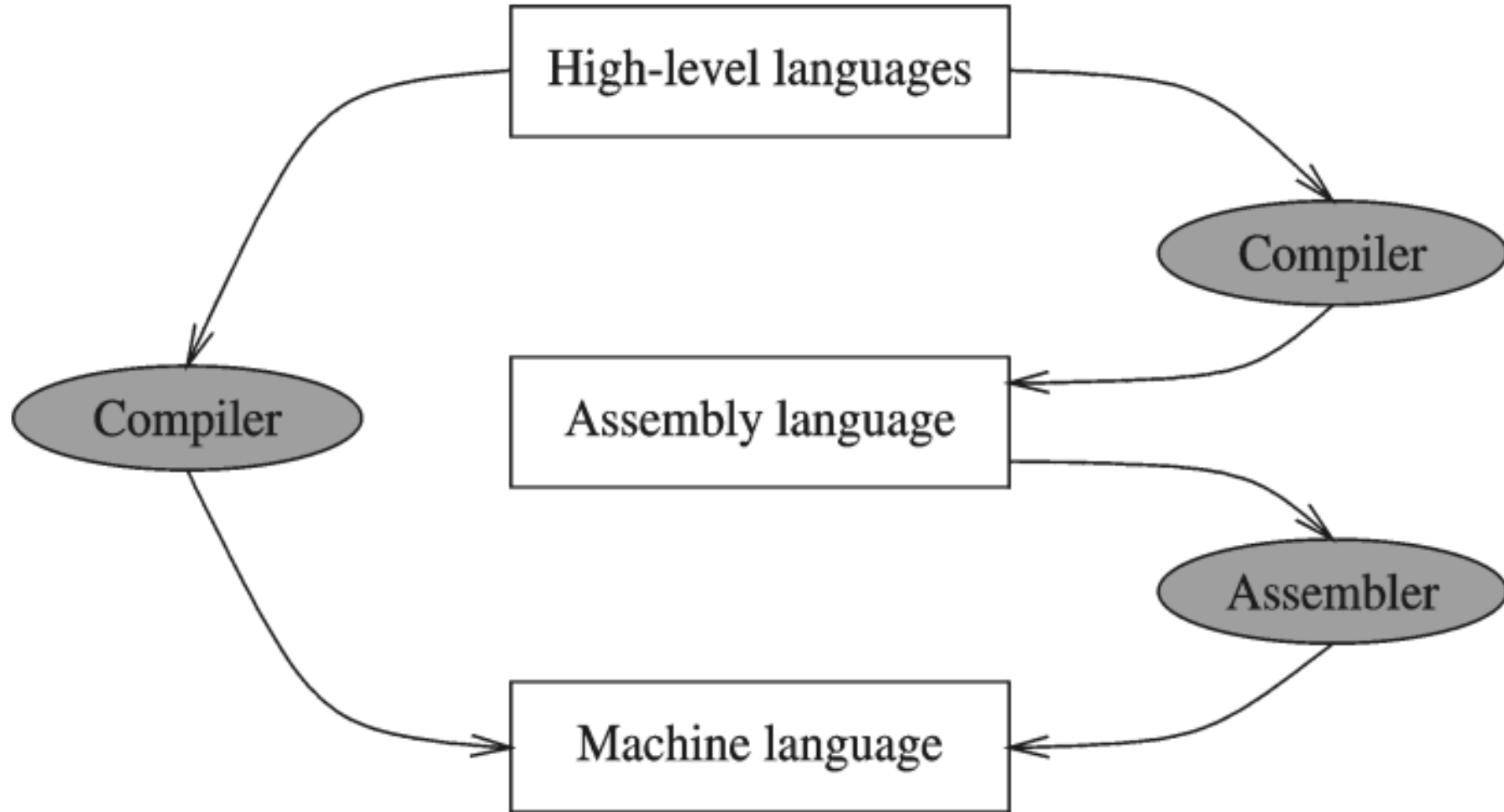
- الهندسة المعمارية للكمبيوتر = بنية مجموعة التعليمات + تنظيم الكمبيوتر
- بنية مجموعة التعليمات : ما يفعله الكمبيوتر (من وجهة نظر منطقية)
- تنظيم الكمبيوتر : كيف يتم تنفيذ (وجهة نظر فيزيائية مرتبطة بالهاردوير)



- لغة الآلة هي اللغة الأم للمعالج : يتم تنفيذها مباشرة بواسطة الدارات المكونة للمعالج.
- التعليمات في لغة الآلة تتكون من كود ثنائي: 1 و 0

## لغة التجميع

- لغة ذات مستوى أعلى قليلاً من لغة الآلة
- قابلية فهم و قراءة التعليمات في لغة التجميع أفضل من لغة الآلة
- كل تعليمة في لغة التجميع يقابلها تعليمة في لغة الآلة (تعليمة مقابل تعليمة)
- يقوم المجمع Assembler بترجمة لغة التجميع إلى لغة الآلة
- يقوم المترجم Compiler بترجمة البرامج عالية المستوى إلى لغة الآلة إما بشكل مباشر أو بشكل غير مباشر مروراً بالمجمع.



- يُطلق على كل أمر في البرنامج تعليمة (يوجه الكمبيوتر إلى ما يجب القيام به).
- تتعامل أجهزة الكمبيوتر مع البيانات الثنائية فقط، وبالتالي يجب أن تكون التعليمات بتنسيق ثنائي (0 و 1).
- المجموعة المكونة من جميع التعليمات (في شكل ثنائي) تشكل لغة الآلة للكمبيوتر.



# ترجمة لغات البرمجة عالية المستوى إلى لغة الآلة

Program (C Language):

```
swap(int v[], int k) {  
    int temp;  
    temp = v[k];  
    v[k] = v[k+1];  
    v[k+1] = temp;  
}
```

A statement in a high-level  
language is translated  
typically into several  
machine-level instructions



Compiler

MIPS Assembly Language:

```
sll $2,$5, 2  
add $2,$4,$2  
lw $15,0($2)  
lw $16,4($2)  
sw $16,0($2)  
sw $15,4($2)  
jr $31
```

Assembler

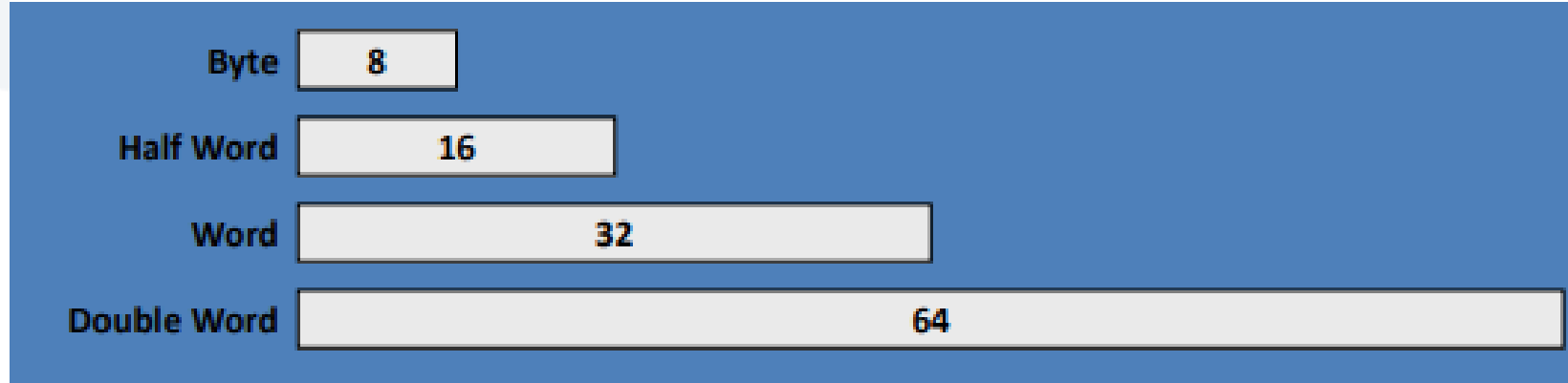


MIPS Machine Language:

```
00051080  
00821020  
8C620000  
8CF20004  
ACF20000  
AC620004  
03E00008
```

# مزايا اللغات عالية المستوى

- تطوير البرنامج أسرع حيث يتطلب البرنامج عدد أقل من التعليمات مقارنة مع لغة التجميع
- صيانة البرنامج أسهل
- البرامج محمولة : تحتوي على تفاصيل قليلة تعتمد على الآلة و يمكن تنفيذها بدون تعديلات على أجهزة مختلفة بينما تعتبر برامج لغة التجميع ليست محمولة.



نوع البيانات	المجال بدون إشارة	المجال معبر عنه بـ 2 مرفوعة للأس
Byte	0 to 255	0 to $(2^8 - 1)$
Half Word	0 to 65,535	0 to $(2^{16} - 1)$
Word	0 to 4,294,967,295	0 to $(2^{32} - 1)$
Double Word	0 to 18,446,744,073,709,551,615	0 to $(2^{64} - 1)$

# الجمع في النظام الثنائي

carry		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)
bit position:	7	6	5	4	3	2	1	0

- نبدأ بالبت الأقل أهمية (بت أقصى اليمين)
- نجمع كل زوج من البتات
- نقوم بتضمين الحمل في عملية الجمع، إن وجد

# الجمع في النظام السداسي عشري

- نبدأ بالأرقام السداسية العشرية الأقل دلالة
- بفرض  $Sum = \text{جمع رقمين سداسي عشري}$
- إذا كان المجموع أكبر من أو يساوي 16
- $Sum = Sum - 16$  و ينتج لدينا حمل = 1
- مثال:

$$\begin{array}{r}
 \text{carry:} \qquad \qquad \qquad 1 \quad 1 \quad \quad 1 \\
 \begin{array}{r}
 1 \ C \ 3 \ 7 \ 2 \ 8 \ 6 \ A \\
 + \ 9 \ 3 \ 9 \ 5 \ E \ 8 \ 4 \ B \\
 \hline
 A \ F \ C \ D \ 1 \ 0 \ B \ 5
 \end{array}
 \end{array}$$

$$\begin{aligned}
 A + B &= 10 + 11 = 21 \\
 21 &\geq 16 \text{ بما أن} \\
 Sum &= 21 - 16 = 5 \\
 Carry &= 1
 \end{aligned}$$

# الأعداد الصحيحة بإشارة

## Signed Integers

توجد عدة طرق لتمثيل الأرقام بإشارة (الأرقام الموجبة و السالبة) :

- المتمم الأحادي 1's complement

- المتمم الثنائي 2's complement

• في الأعداد بإشارة ينقسم المجال إلى جزأين متساويين:

- الجزء الأول يمثل الأعداد الموجبة

- الجزء الثاني يمثل الأعداد السالبة

• سيكون التركيز على تمثيل الأعداد بإشارة من خلال المتمم الثنائي باعتبار أنه هو المستخدم على نطاق واسع في المعالجات

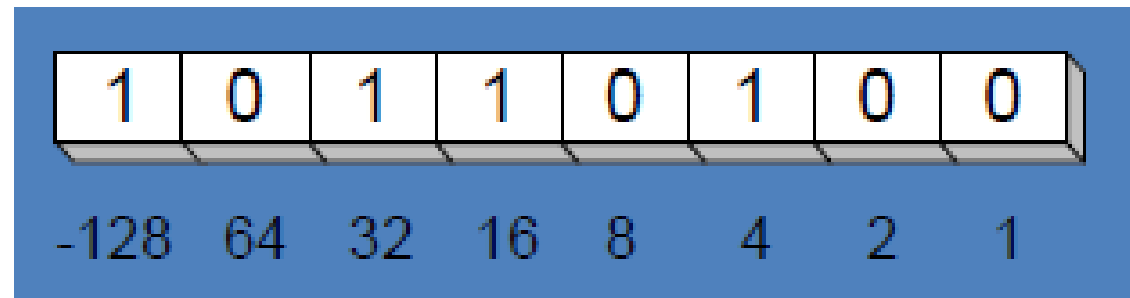
لتمثيل الأعداد الصحيحة بإشارة .

# التمثيل باستخدام المتعم الثنائي

8-bit Binary value	Unsigned value	Signed value
00000000	0	0
00000001	1	+1
00000010	2	+2
...	...	...
01111110	126	+126
01111111	127	+127
10000000	128	-128
10000001	129	-127
...	...	...
11111110	254	-2
11111111	255	-1

• لإيجاد مطال القيمة بإشارة ممثلة باستخدام المتعم

الثنائي نقوم بإسناد وزن سلبى للبت الأكثر أهمية MSB



$$-128 + 32 + 16 + 4 = -76$$

## مجال الأعداد بإشارة

- من أجل الأعداد بـ  $n$  بت يكون المجال :

$$-2^{n-1} \text{ to } (2^{n-1} - 1)$$

Byte	-128 to +127	$-2^7 \text{ to } (2^7 - 1)$
Half Word	-32,768 to +32,767	$-2^{15} \text{ to } (2^{15} - 1)$
Word	-2,147,483,648 to +2,147,483,647	$-2^{31} \text{ to } (2^{31} - 1)$
Double Word	-9,223,372,036,854,775,808 to +9,223,372,036,854,775,807	$-2^{63} \text{ to } (2^{63} - 1)$



# تشكيل المتمم الثنائي

- مثال: مثل العدد (-36) بحسب المتمم الثنائي؟

للحصول على العدد -36 ننتقل في البداية من العدد +36	$00100100 = +36$
الخطوة 1: نعكس الخانات ( إيجاد المتمم الأحادي)	$11011011$
الخطوة 2: نجمع 1 إلى القيمة الناتجة من الخطوة 1	$+ \quad 1$
حاصل الجمع = التمثيل بحسب المتمم الثنائي	$11011100 = -36$

Binary Value

=  $00100\boxed{1}00$

2's Complement

=  $11011\boxed{1}00$

- توجد طريقة أخرى للحصول على المتمم الثنائي:

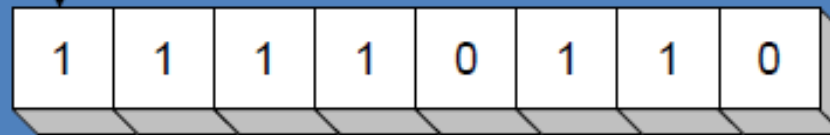
- ✓ نبدأ بالبحث من الخانة الأقل أهمية عن أول ورود 1
- ✓ نترك كل الأصفار على يمينه بدون تغيير
- ✓ نقرب كل الخانات على يساره ( نجعل الأصفار واحداً و بالعكس)

# خانة الإشارة Sign Bit

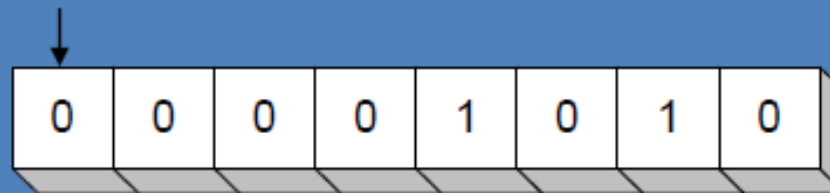
1 = negative

0 = positive

Sign bit



Negative



Positive

## توسعة خانة الإشارة

• يمكن توسعة العدد بإشارة من خلال تكرار خانة الإشارة في أقصى اليسار بعدد خانات التوسعة

المطلوبة دون أن يؤثر ذلك على إشارة و مطال العدد

• مثال: وسع العدد 10110011 بحيث يصبح طوله 16 بت؟

$$10110011 = -77 \rightarrow \boxed{11111111}10110011 = -77$$

• مثال: وسع العدد 01100010 بحيث يصبح طوله 16 بت؟

$$01100010 = +98 \rightarrow \boxed{00000000}01100010 = +98$$



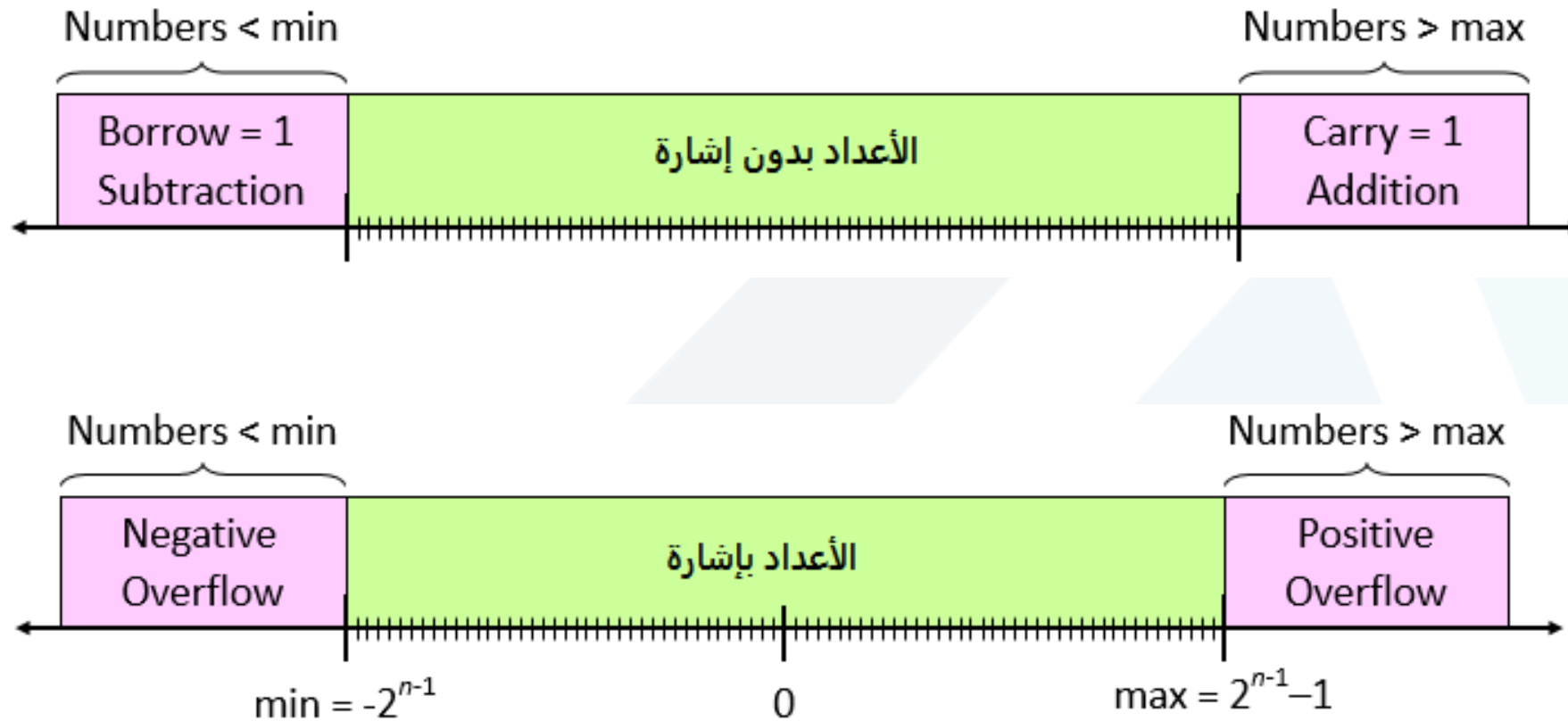
# الحمل والطفحان

## Carry and Overflow

- الحمل مهم عندما :
  - ✓ جمع أو طرح الأعداد الصحيحة بدون إشارة
  - ✓ يشير إلى أن المجموع خارج المجال
  - ✓ إما المجموع أصغر من 0 أو المجموع أكبر من الحد الأقصى لقيمة  $n$  بت بدون إشارة
- الطفحان مهم عندما :
  - ✓ جمع أو طرح الأعداد الصحيحة بإشارة
  - ✓ يشير إلى أن المبلغ الموقع خارج المجال
- يحدث الطفحان عند:
  - ✓ جمع رقمين موجبين ويكون المجموع سالب
  - ✓ جمع رقمين سالبين ويكون المجموع موجباً
- يمكن التخلص من حالة الطفحان بزيادة عدد الخانات الممثلة للأعداد بالمتعمم الثنائي

# الحمل والطفحان

## Carry and Overflow



# Examples

				1					
	0	0	0	0	1	1	1	1	15
+	0	0	0	0	1	0	0	0	8
<hr/>									
	0	0	0	1	0	1	1	1	23
Carry = 0    Overflow = 0									

1	1	1	1	1					
	0	0	0	0	1	1	1	1	15
+	1	1	1	1	1	0	0	0	248 (-8)
<hr/>									
	0	0	0	0	0	1	1	1	7
Carry = 1    Overflow = 0									

				1					
	0	1	0	0	1	1	1	1	79
+	0	1	0	0	0	0	0	0	64
<hr/>									
	1	0	0	0	1	1	1	1	143 (-113)
Carry = 0    Overflow = 1									

1			1	1					
	1	1	0	1	1	0	1	0	218 (-38)
+	1	0	0	1	1	1	0	1	157 (-99)
<hr/>									
	0	1	1	1	0	1	1	1	119
Carry = 1    Overflow = 1									

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	space	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

❖ أمثلة:

❖ رمز ASCII لحرف L هو 4C

❖ رمز ASCII لحرف a هو 61