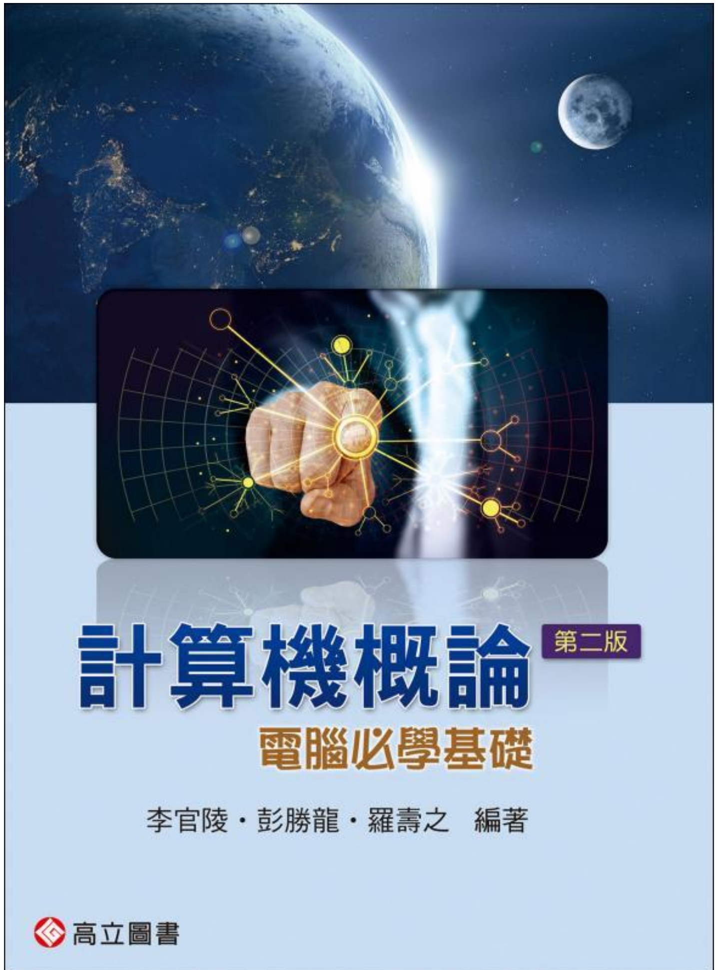


Ch02 數字系統

李官陵 彭勝龍 羅壽之

電腦儲存的單位

- 在電腦的世界裡是使用電子元件來儲存與運算資料
- 電子元件的穩定狀態有兩種
  - 一種是「關」，另一種是「開」
- 用來表示這兩種狀態的單位，我們稱之為**位元 (bit)**。
  - 電腦儲存或傳遞資料的最小單位
  - 可以儲存 0 或 1 兩種狀態

圖 2.1 0 與 1

電腦儲存的單位 (續)

- 電腦中是靠多個位元的組合來計數或描述多種狀態

- 一個位元只能描述兩種狀態變化。
  - 0 與 1
- 2 個位元組合出的變化
  - {00, 01, 10, 11}
  - 2×2 (2 的 2 次方) 共四種變化
- n 個位元呢?
  - 2^n

電腦儲存的單位 (續)

- 早期的電腦中，以 8 個位元為一個存取單位，稱之為一個**位元組** (byte, 1 byte = 8 bits)
  - 最常被使用的表示單位

- 另一種存取資料的單位稱為字組 (word)
  - 可以由 2、4 或 8 個位元組所組成
  - 一個字組包含了幾個位元組則視硬體的結構而定

電腦儲存的單位 (續)

隨堂練習

- 如果一台電腦的字組是由 2 個位元組 (2 bytes) 所組成，那麼一個字組可以描述多少種狀態？

解答：

- 一個字組包含了 2 個位元組，共有 $2 \times 8 = 16$ 個位元
- 可以組合出的變化共有 $2^{16} = 65536$ 種

電腦儲存的單位 (續)

隨堂練習

- 如果我們需要描述的狀態有 16 種，那麼至少需要幾個位元？
- 如果需要描述的狀態有 80 種，又至少需要多少個位元呢？

解答：

- 4 個位元剛剛好就可以描述 16 種狀態。
- 需要靠 7 個位元的組合才能描述80 種狀態

電腦儲存的單位 (續)

資訊容量的單位

- 常見的資訊容量有 KB、MB、GB 、 TB與PB
  - 結尾的 B 指的是位元組 (byte)
  - 開頭的 K、M、G、T 與 P 是數量詞
  - KB 的資料量是 2^{10} bytes
    - 1024 bytes
    - 1024 近似於一千，所以用 Kilo 來表示它
  - MB 的資料量是 $2^{10} \times 2^{10}$ bytes
    - 1024 個 KB
    - 所以用 Mega (百萬) 來表示它
  - 同樣的道理，1 GB = 2^{10} MB、1 TB = 2^{10} GB 而 1 PB = 2^{10} TB。

電腦儲存的單位 (續)

資訊容量的單位 (續)

單位	英文全名	位元組個數
KB	Kilo Bytes	2^{10} bytes
MB	Mega Bytes	2^{20} bytes
GB	Giga Bytes	2^{30} bytes
TB	Tera Bytes	2^{40} bytes
PB	Peta Bytes	2^{50} bytes

二進位表示法

- 電腦世界裡使用的數字系統是二進位制

- 十進位數 327.25
  - 三個 100、二個 10、七個 1、二個 0.1 以及五個 0.01

$$327.25 = 3\times10^{2} + 2\times10^{1} + 7\times10^{0} + 2\times10^{-1} + 5\times10^{-2}$$

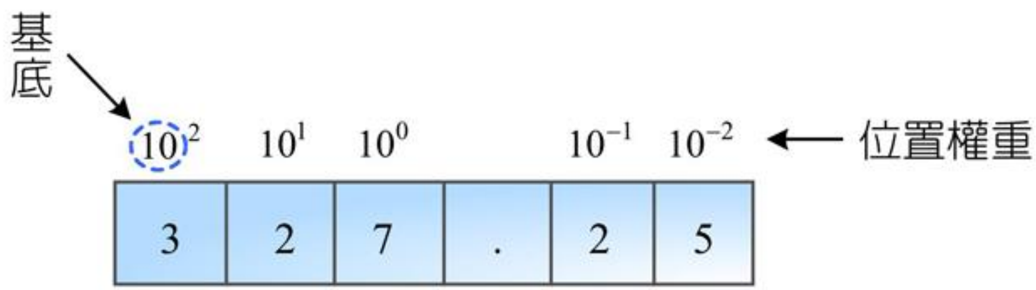


圖 2.3 基底與位置權重

二進位表示法 (續)

- 一個 K 進位 的正數 $N(d_{a-1}d_{a-2}d_{a-3}\cdots d_0.d_{-1}d_{-2}\cdots d_{-b})$

  $\boxed{(N)_k}$

$$(N)_k = d_{a-1}\times K^{a-1} + d_{a-2}\times K^{a-2} + \cdots + d_0\times K^0 + d_{-1}\times K^{-1} + d_{-2}\times K^{-2} + \cdots + d_{-b}\times K^{-b}$$

K^{a-1}	K^{a-2}	……	K^0		K^{-1}	K^{-2}	……	K^{-b}	← 位置權重
d_{a-1}	d_{a-2}	……	d_0	.	d_{-1}	d_{-2}	……	d_{-b}	

圖 2.4 K 進位數字

二進位表示法 (續)

- 二進位數 $(1011.011)_2$，對應的十進位數

$$
\begin{aligned}
(1011.011)_2 &= 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0 + 0\times 2^{-1} + 1\times 2^{-2} + 1\times 2^{-3} \\
&= (11.375)_2
\end{aligned}
$$

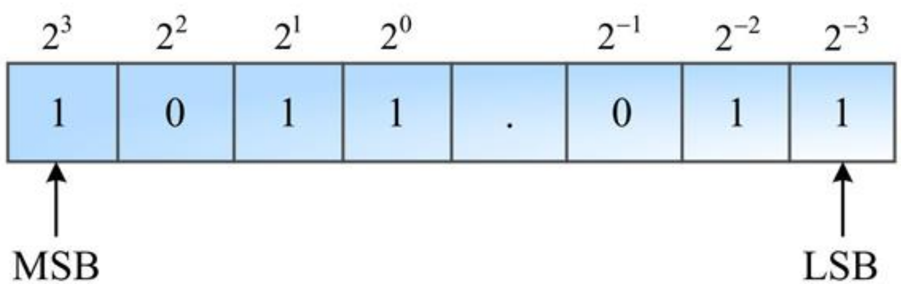


圖 2.5 二進位數

二進位表示法(續)

隨堂練習

- 計算出二進位數 $(10100.101)_2$ 對應的十進位數大小

解答：

$$
\begin{aligned}
(10100.101)_2 &= 1\times 2^4 + 0\times 2^3 + 1\times 2^2 + 0\times 2^1 + 0\times 2^0 + 1\times 2^{-1} \\
&\quad + 0\times 2^{-2} + 1\times 2^{-3} \\
&= (20.625)_{10}
\end{aligned}
$$

給定一個十進位數，要如何算出它對應的二進位數呢？

二進位表示法 (續)

- 將十進位數 $(166)_{10}$ 轉為二進位數
  - 將 166 除以 2，得到商數 83、餘數 0，因此我們知道 $d_0 = 0$
  - 繼續將 83 除以 2，得到商數 41、餘數 1 $\rightarrow$ $d_1 = 1$；
  - 再將 41 除以 2，得到商數 20、餘數 1 $\rightarrow d_2 = 1$；
  - 一直重複這個步驟直到商數為 0

$$(166)_{10} = (10100110)_2$$

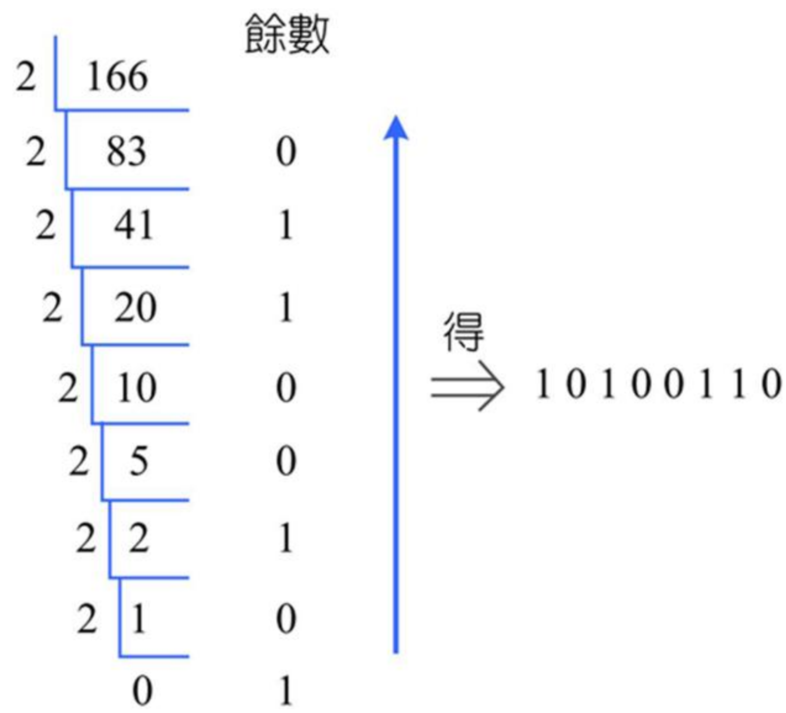


圖 2.6 十進位數轉二進位數範例

二進位表示法 (續)

隨堂練習

- 請計算出十進位數 $(87)_{10}$ 所對應的二進位數。

解答：

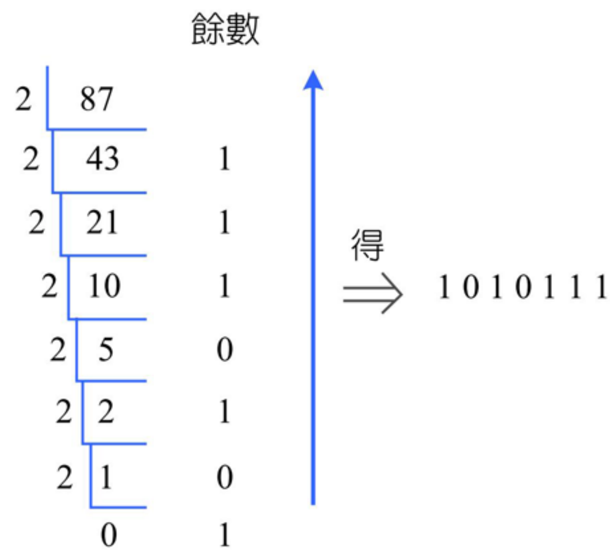


圖 2.7 $(87)_{10}$ 轉二進位數的過程

二進位表示法 (續)

- 給定十進位數 $(0.625)_{10}$
  - 將 0.625 乘以 2，得到 1.25 $\rightarrow d_{-1}=1$；
  - 將整數部位去除，得到 0.25、將 0.25 乘以 2，得到 0.5 $\rightarrow d_{-2}=0$；
  - 因為 0.5 的整數部位為 0，所以直接將 0.5 乘以 2，得到 1 $\rightarrow d_{-3}=1$
  - 將 1 的整數部分去除得到了 0，轉換的過程結束；

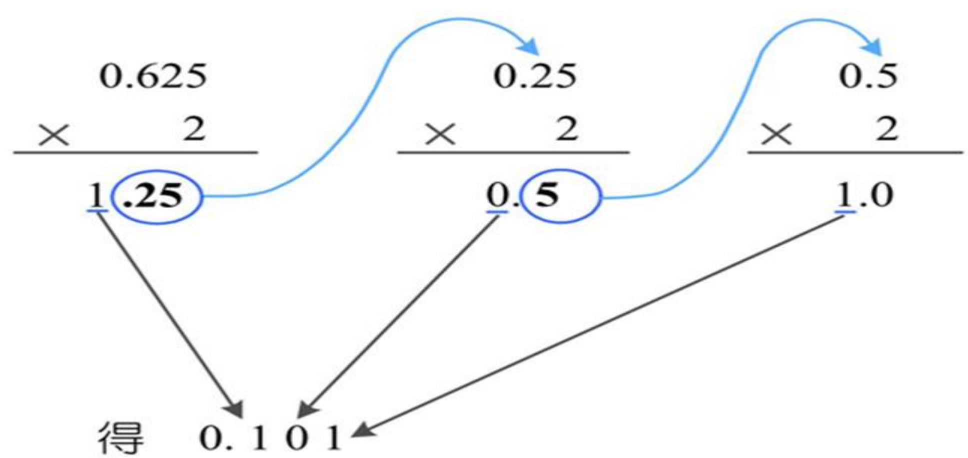


圖 2.8 $(0.625)_{10}$ 轉換為二進位數的過程

二進位表示法(續)

隨堂練習

- 請計算出 $(0.375)_{10}$ 所對應的二進位數。

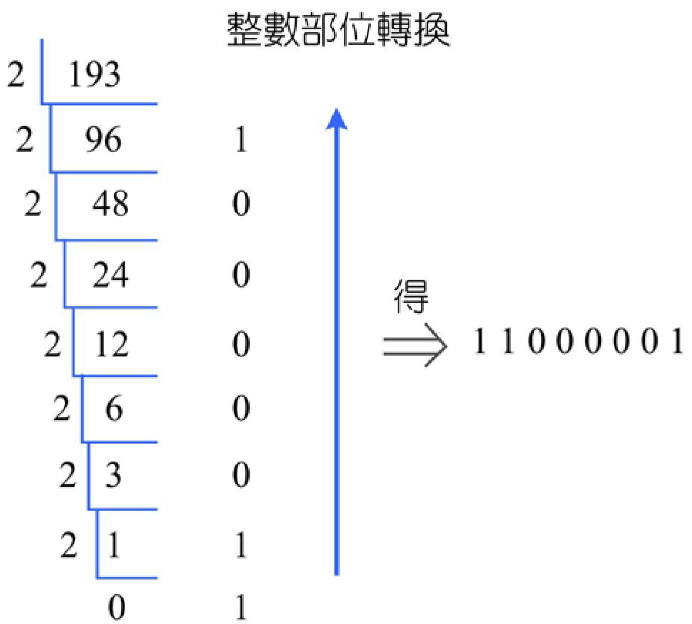
解答：

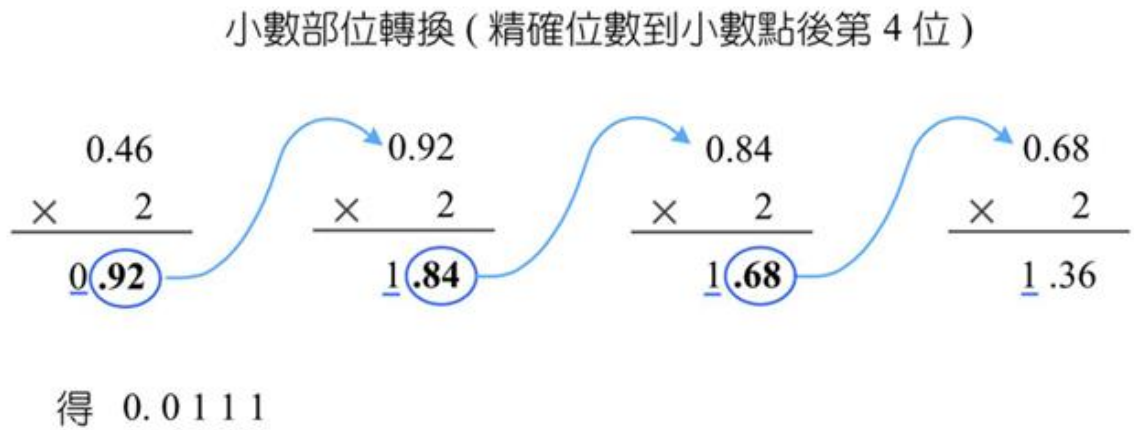
$$0.375\times 2=0.75 \;\rightarrow\; d_{-1}=0$$
$$0.75\times 2=1.5 \;\rightarrow\; d_{-2}=1$$
$$0.5\times 2=1 \;\rightarrow\; d_{-3}=1$$
$$(0.375)_{10}=(0.011)_2$$

不是所有的十進位小數都可以用有限的位元完整的表達，如 $(0.7)_{10}$ 對應的二進位數為 $(0.1011001\cdots)_2$。
我們只須運算到符合轉換時給定的精確位數即可，如精確位數要到小數點後第四位，則我們可以說 $(0.7)_{10} = (0.1011)_2$。

二進位表示法 (續)

- 將 $(193.46)_{10}$ 轉成其對應的二進位數，精確位數到小數點後第四位





因此
$(193.46)_{10} = (11000001.0111)_2$

八進位制與十六進位制

- 十六進位制是逢 16 就進 1 的進位制
  - 用數字 0 到 9 與英文字母 A~F 表示 (A~F 代表 10~15)。
- 八進位制
  - 利用八個符號 0、1、2、...、7 來代表任何數目的數字系統。

表 2.2 二進位、八進位、十進位以及十六進位數字 0~15 的對應關係

十進位	二進位	八進位	十六進位	十進位	二進位	八進位	十六進位
0	0	0	0	8	1000	10	8
1	1	1	1	9	1001	11	9
2	10	2	2	10	1010	12	A
3	11	3	3	11	1011	13	B
4	100	4	4	12	1100	14	C
5	101	5	5	13	1101	15	D
6	110	6	6	14	1110	16	E
7	111	7	7	15	1111	17	F

二進位數與十六進位數的互換

- 如何將 $(110101101.101101)_2$ 以十六進位的方式表示呢？

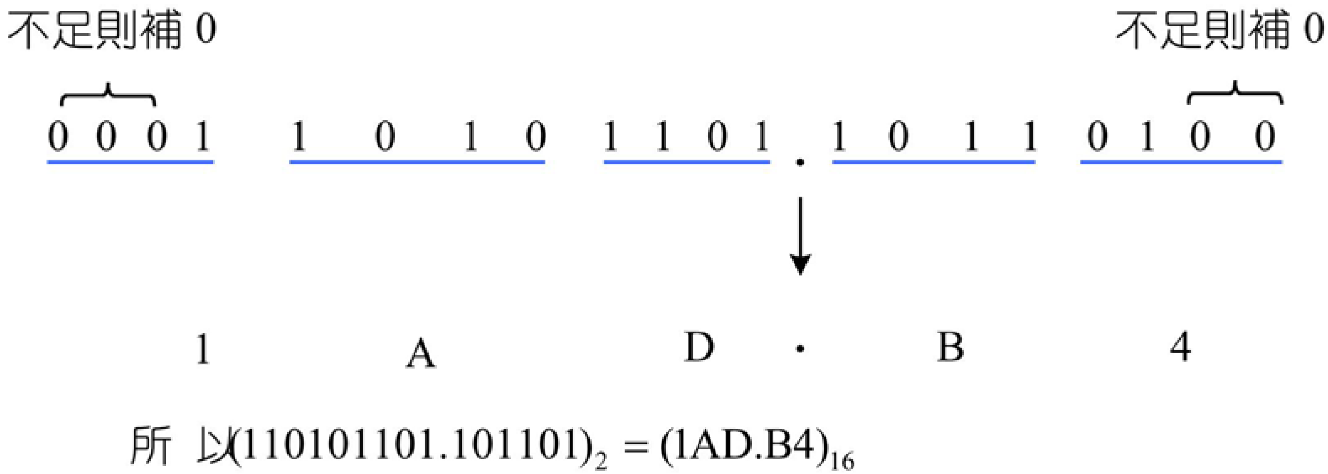


圖 2.10　二進位數轉換為十六進位數的過程

二進位數與十六進位數的互換

隨堂練習

- $(1011011.01011)_2=(?)_{16}$

解答

5B.58

二進位數與十六進位數的互換 (續)

- 將 $(25B.7C)_{16}$ 轉成二進位數

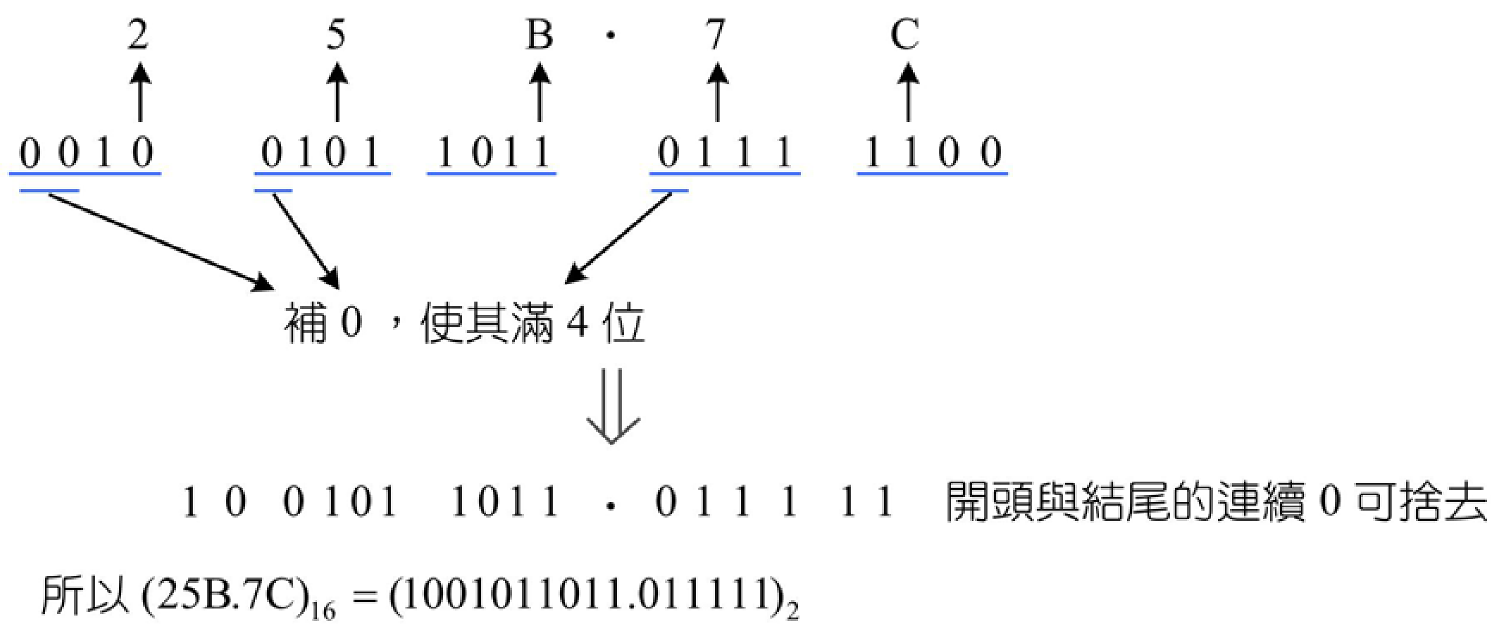


圖 2.11 十六進位數轉換為二進位數的過程

二進位數與十六進位數的互換

隨堂練習

- $(A3.C37)_{16}=(?)_2$

解答

10100011.110000110111

二進位數與八進位數的互換

- 如何將 $(1110101101.10110)_2$ 以八進位的方式表示呢？

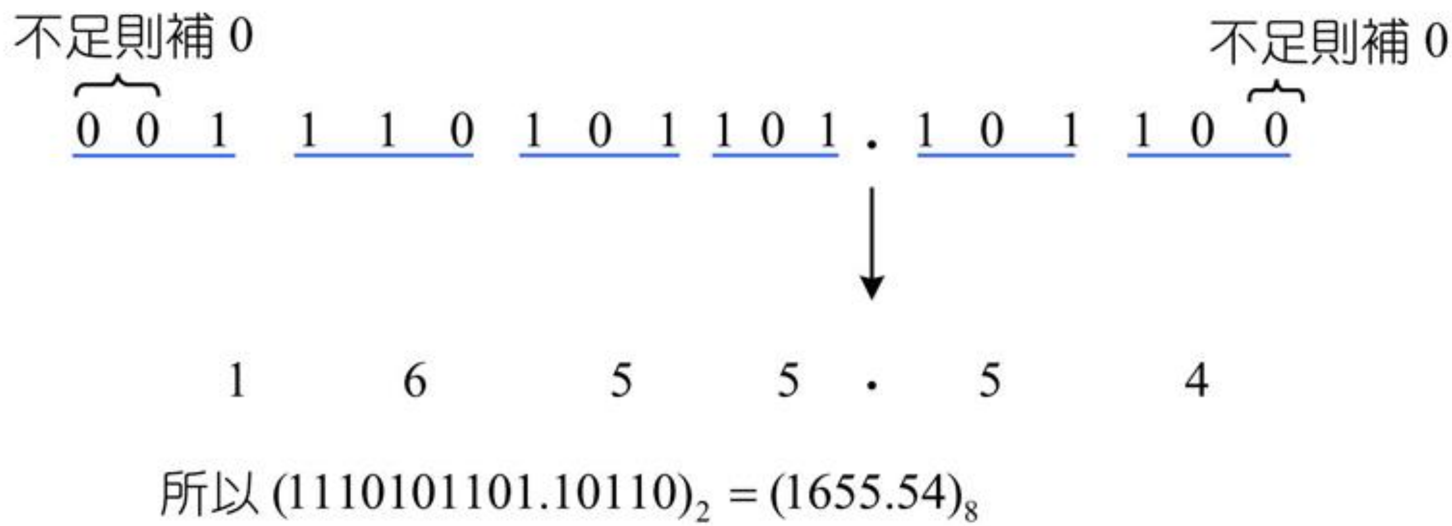
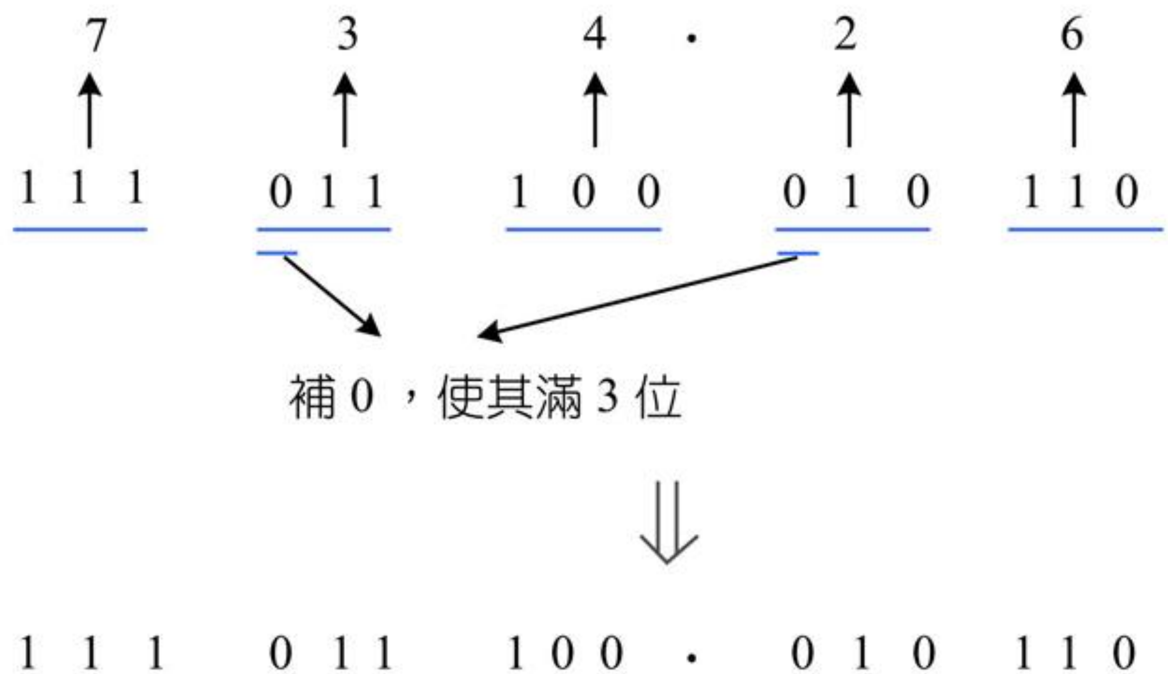


圖 2.12 二進位數轉換為八進位數的過程

二進位數與八進位數的互換

- 將 $(734.26)_8$ 轉成二進位數



所以 $(734.26)_8 = (111011100.010110)_2$

圖 2.13 八進位數轉換為二進位數的過程

整數表示法與補數概念

- 表示正整數
  - 假設我們利用 8 個位元 (1 個位元組) 表示一個正整數，可以表示的正整數範圍為 $0 \sim 2^8 - 1$，也就是從 0 到 255 的整數
  - 這樣不考慮正負數的整數，稱之為「無正負符號的整數」(unsigned integer)

整數表示法與補數概念 (續)

- 當我們需要表示負數時該怎麼做呢？
  - 最簡單的作法是將最左邊的位元切割出來，當作正負符號使用，稱之為符號位元 (sign bit)
    - 符號位元為 0 時即代表正數，1 時則代表負數
    - 剩下來的位元則用來表示數的大小，這樣的表示法我們稱之為「帶正負符號數字表示法」(sign-magnitude representation)。

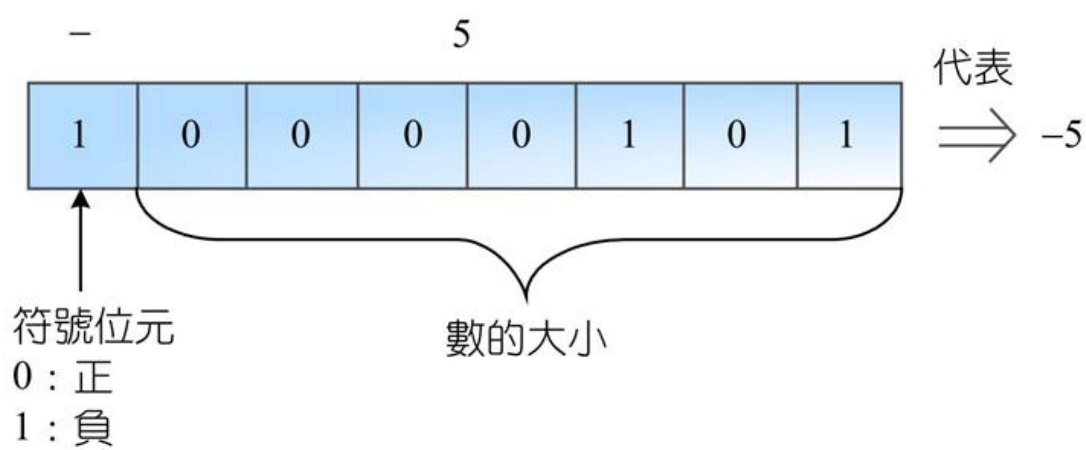


圖 2.14 用帶正負符號數字表示法描述 $(-5)_{10}$ 範例

整數表示法與補數概念 (續)

- 帶正負符號數字表示法
  - 8個位元可以描述的數字範圍為 $-(2^7-1) \sim 2^7-1$
    - $-127 \sim 127$。
  - 目前的電腦並不採用這種方式表示整數
    - 0 存在兩種表示法
    - 正數負數的運算 (如加法、減法) 並不直接

表 2.3 用 8 位元帶正負符號數字表示法描述的十進位數

十進位數	帶正負符號數字	十進位數	帶正負符號數字
–127	11111111	127	01111111
–126	11111110	126	01111110
…	…	…	…
–1	10000001	1	00000001
–0	10000000	0	00000000

一的補數表示法

- 以位元字串最左邊的位元表示正負符號
- 正數的表示方式與帶正負符號數字表示法相同
  - 假設我們用 1 個位元組來表示整數
  - 則 $(5)_{10}$ 的一的補數表示法為 $(00000101)_2$
- 利用 8 位元儲存整數能表示的正數範圍
  - $(00000000)_2 \sim (01111111)_2$，也就是 0~127。

補數的觀念

若兩數的和為 1，則此兩數互為 1 的補數

一的補數表示法 (續)

- 步驟 1：忽略其符號，將該數轉成二進位表示法，並利用 $n-1$ 位元表示，若該數超過了 $n-1$ 位元，則稱為溢位 (overflow)，意指該數無法用 n 位元一的補數表示法描述，轉換停止
- 觀察該數的符號
  - 若為正數：在 $n-1$ 位元的左邊補上 1 個位元，並設為 0，完成轉換
  - 若為負數：將每個位元做一補數的轉換，即將 1 轉成 0、0 轉成 1，完成後在此 $n-1$ 位元的左邊補上 1 個位元，並設成 1，完成轉換

一的補數表示法 (續)

- 請將 $(-25)_{10}$ 用 8 位元一的補數表示法表現。

解答：

- 步驟 1：忽略符號，將 25 轉成二進位數 11001，並用 7 個位元表示，得 0011001
- 步驟 2：因為是負數，所以將 0011001 轉成其一的補數 1100110
- 接著左邊再補上 1 個位元並設為 1 代表負數，得到 11100110。
- 11100110 即為答案。

一的補數表示法 (續)

隨堂練習

- 請將 $(-15)_{10}$ 用 8 位元一的補數表示法表現

解答：

- 步驟 1：忽略符號，將 15 轉成二進位數 1111，並用 7 個位元表示，得 0001111。
- 步驟 2：因為是負數，所以將 0001111 轉成其一的補數 1110000
- 接著左邊再補上 1 個位元並設為 1 代表負數，得到 11110000
- 11110000 即為答案。

一的補數表示法 (續)

- 若 $(11001101)_2$ 是利用一的補數表示法儲存的整數，請問它代表的是哪一個十進位數呢？

解答：

- 最左邊的位元為 1，所以代表的是負數
- 右邊的 7個位元 (1001101) 取出，並將它還原為原來的數
- 原數為 0110010，將0110010 轉成十進位數得到 50
- 最後將負號補上，故得到其代表的十進位數為－50。

一的補數表示法 (續)

隨堂練習

- 若 $(11101001)_2$ 是利用一的補數表示法儲存的整數，請問它代表的是哪一個十進位數呢？

解答：

- 最左邊的位元為 1，所以代表的是負數
- 右邊的 7個位元 (1101001) 取出，並將它還原為原來的數
- 原數為 0010110，將0010110 轉成十進位數得到 22
- 最後將負號補上，故得到其代表的十進位數為－22

一的補數表示法 (續)

- 一的補數表示法同樣有兩個0 的問題。
- 正負數運算 (如加法、減法) 並不那麼直接。

表 2.4 用 8 位元一的補數表示法描述的十進位數

十進位數	一的補數表示法	十進位數	一的補數表示法
–127	10000000	127	01111111
–126	10000001	126	01111110
–125	10000010	125	01111101
…	…	…	…
–2	11111101	2	00000010
–1	11111110	1	00000001
–0	11111111	0	00000000

二的補數表示法

- 十補數
  - 2 的十補數為 8 (2 + 8 = 10)
  - 7 的十補數為 3 (7 + 3 = 10)

在二進位系統中，若兩數互為對方的二的補數，則代表兩數的和使得原來的每一個位數均為 0，並產生溢位。

二的補數表示法 (續)

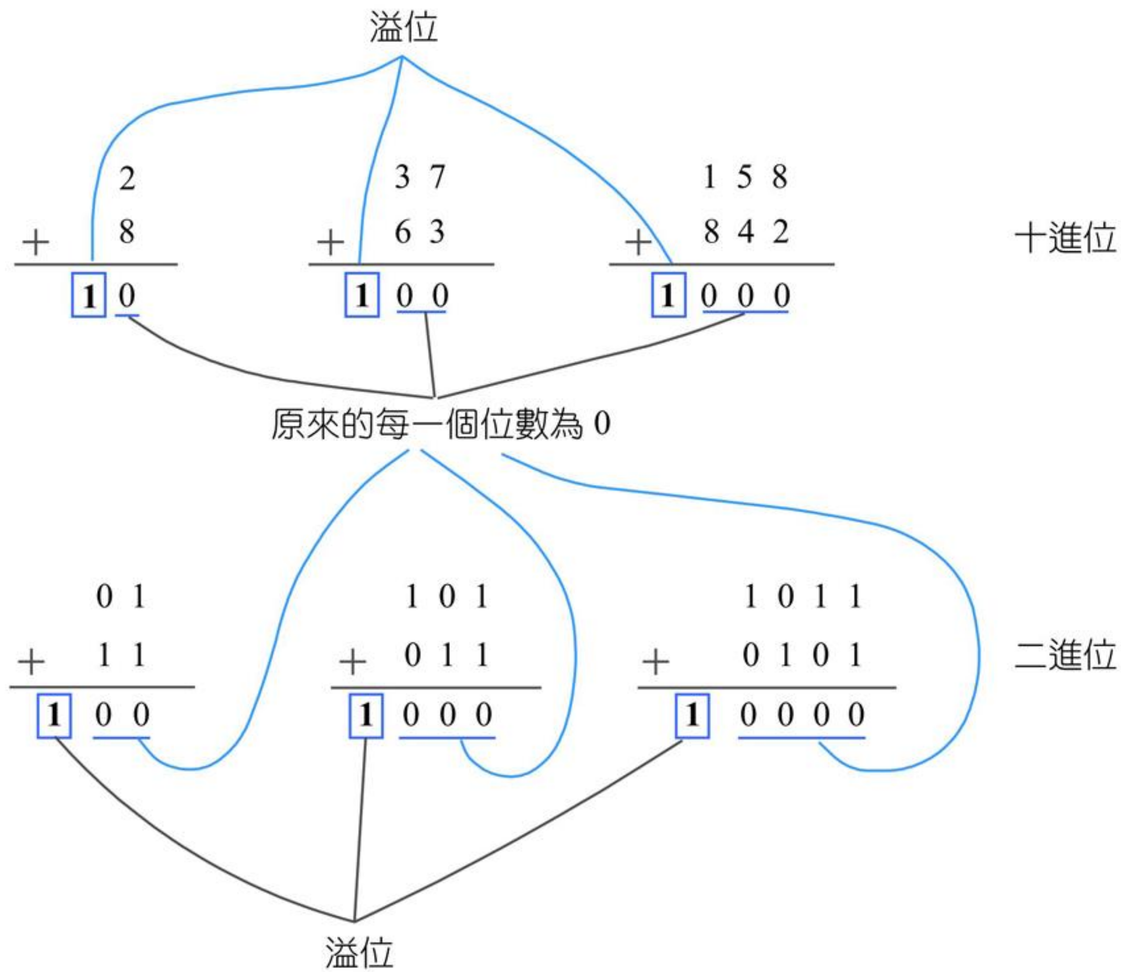


圖 2.15 進位中十的補數與二進位中二的補數

二的補數表示法 (續)

- 給定一個二進位數，有沒有方法快速的將它的二補數算出來呢？
  - 找出它的一補數 (即 1 變 0、0 變 1)，接著再將此補數加上 1，即可得到此二進位數的二的補數
  - =>「1 變 0、0 變 1，變完之後再加 1」

二的補數表示法 (續)

- 二進位數 001100 的二的補數
  - 先將 1 變 0、0 變 1，得到110011
  - 再將此數加上 1，即可得到 001100 的二補數為 110100

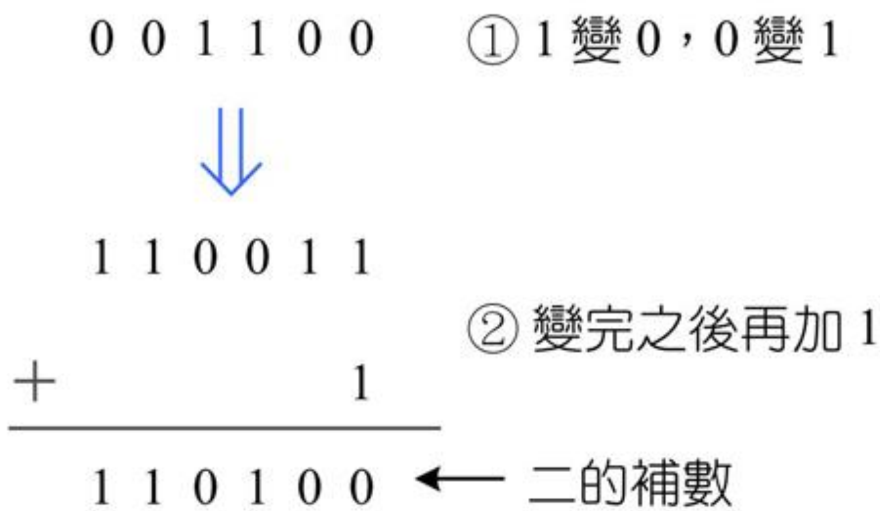


圖 2.16 範例八運算步驟

二的補數表示法 (續)

- 以位元字串最左邊的位元表示正負符號，剩下來的位元則用來表示數的大小。
- 正數的表示方式與帶正負符號數字表示法相同
  - 用 1 個位元組來表示整數，則 $(5)_{10}$ 的二的補數表示法為 $(00000101)_2$
  - 利用 8 位元儲存整數時，能表示的正整數範圍為 $(00000000)_2 \sim (01111111)_2$，也就是 $0 \sim 127$
- 負數表示 (以 8 位元為例)
  - 先忽略其符號，並將該數轉成 7 位元的二進位數
  - 將此二進位數換成其二的補數，最後在左邊加上 1 個位元並設為 1 (代表負數) 即完成轉換。

二的補數表示法 (續)

- 以 8 位元表示 $-(5)_{10}$ 為例
  - 將 5 轉成一個 7 位元的二進位數，得到 0000101
  - 將 0000101 轉成其二的補數，得到 1111011
  - 最後在最左邊補上 1 個位元並設成 1 (代表負數)，即得到 $(11111011)_2$ 為 -5 的二補數表示法

二的補數表示法 (續)

- 若 $(00010101)_2$ 是利用二的補數法表示的整數，那麼它代表的十進位數為何？
  - 最左邊的位元為 0，此數為一個正數
  - 取出右邊 7 個位元 (代表著數的大小)，算出其表示的十進位數為 21，即得到答案

二的補數表示法 (續)

- 若 $(10010101)_2$ 是利用二的補數法表示的整數，那麼它代表的十進位數為何？
  - 最左邊的位元為 1，此數為一個負數
  - 取出右邊 7 個位元，並將其轉成原數
    - 在補數系統中任何數的 *N* 補數的 *N* 補數會是它自己
    - 原數 為1101011
  - 算出原數表示的十進位數為 107，因此得知 10010101 表示的是 − 107

二的補數表示法 (續)

隨堂練習

- 若 $(11001100)_2$ 是利用二的補數表示法儲存的整數，請問它代表的是哪一個十進位數呢？

解答：

- 最左邊的位元為 1，代表的是負數
- 將右邊的 7 個位元 (1001100) 取出，將其轉成二的補數得到原數為 0110100
- 將原數轉成十進位數得到 52
- 最後將負號補上，故得到其代表的十進位數為 − 52。

二的補數表示法 (續)

- 二的補數表示法能描述的負數範圍
  - 以 8 位元為例，10000000 是其能表示的最大負數
  - 最左邊的 1 代表負數，而 0000000 則為原數的二的補數，故可得到原數為 $(10000000)_2 = 2^7$。
- 利用 8 位元儲存數字，二的補數表示法可表示的負整數範圍為 $-1 \sim -128$；將其擴充到 n 個位元，n 位元的二的補數表示法能表示的整數範 $-2^{n-1} \sim 2^{n-1} - 1$。

二的補數表示法 (續)

- 步驟 1：判斷該數大小
  - 若該數為 -2^{n-1}：則 1000…000 (1 後面跟著 $n-1$ 個 0) 即為所求，轉換結束
  - 若該數不在 $-2^{n-1} \sim -2^{n-1}-1$ 範圍內：該數無法用 n 位元二的補數表示法描述，轉換停止。
  - 其它：繼續步驟 2
- 步驟 2：忽略其符號，將該數轉成 $n-1$ 位元的二進位表示法
- 步驟 3：觀察該數的符號
  - 若為正數：在 $n-1$ 位元的左邊補上 1 個位元，並設為 0，完成轉換
  - 若為負數：將每個位元做二補數的轉換，完成後在此 $n-1$ 位元的左邊補上 1 個位元，並設成 1，完成轉換

二的補數表示法 (續)

- 請將 $(-38)_{10}$ 用 8 位元二的補數表示法表現

解答：

- －38 在表示的範圍內
- 先忽略符號，將 38 轉成 7 位元的二進位數得 0100110
- 因為是負數，所以將 0100110 轉成其二的補數 1011010
- 左邊再補上 1 個位元並設為 1 代表負數，得到 11011010
- 11011010 即為答案

二的補數表示法 (續)

隨堂練習

- 請將 $(-21)_{10}$ 用 8 位元二的補數表示法表現

解答：

- −21 在表示的範圍內
- 先忽略符號，將 21 轉成 7 位元的二進位數 0010101
- 因為是負數，所以將 0010101 轉成其二的補數 1101011
- 左邊再補上 1 個位元並設為 1 代表負數，得到 11101011
- 11101011即為答案

二的補數表示法 (續)

表 2.5 用 8 位元二的補數表示法描述的十進位數

十進位數	二的補數表示法	十進位數	二的補數表示法
–128	10000000	—	—
–127	10000001	127	01111111
–126	10000010	126	01111110
–125	10000011	125	01111101
…	…	…	…
–2	11111110	2	00000010
–1	11111111	1	00000001
—	—	0	00000000

0的表示法唯一!!

二補數系統的加法與減法

- 二補數系統數字的加法和一般加法相同
- 在運算完成後就可以看出結果的正負號

需要注意兩個正數相加以及兩個負數相加時是否有溢位產生

- 最左邊代表符號的位元是否由 0 變成了 1 (兩正數相加時須注意)，或是由 1 變成了 0 (兩負數相加時須注意)
- 發生溢位，代表目前使用的位元數不夠儲存相加後的結果，故得到的結果將被忽略，視為無效

二補數系統的加法與減法

	十進位	二的補數表示法
	23	00010111
+	18	00010010
	41	00101001

沒有溢位產生，所以 00101001 即為答案，也是 41 的二補數表示法

二補數系統的加法與減法 (續)

十進位	二的補數表示法
123	01111011
+ 88	01011000
211	$\boxed{1}$1010011
	溢位

兩正數相加，有溢位產生 (符號位元從 0 變成 1)，代表目前使用的位元數不夠儲存相加後的結果，故結果視為無效

二補數系統的加法與減法 (續)

	十進位	二的補數表示法
	−23	11101001
+	−18	11101110
	−41	$\boxed{\mathbf{1}}$11010111
		進位

- 兩負數相加，沒有溢位產生 (符號位元沒有從 1 變成 0)
- 最後產生的 1 超出了最左邊的位元，稱之為進位
- 進位在二的補數加法上是可以被忽略的

二補數系統的加法與減法 (續)

	十進位	二的補數表示法
	–123	10000101
+	–88	10101000
	–211	$\boxed{1}\,\boxed{0}0101101$

進位 溢位

- 兩負數相加，有溢位產生 (符號位元從 1 變成 0)，代表目前使用的位元數不夠儲存相加後的結果，故結果視為無效

二補數系統的加法與減法 (續)

	十進位	二的補數表示法
	–23	11101001
+	18	00010010
	–5	11111011

- 正負數相加，故得到的結果即為所求

浮點數表示法

- IEEE 二進位浮點數算數標準 IEEE 754是電腦最常用的實數表示法

- 科學記號
  - 可以將它視為是一種標準化
  - $5732.889 \Rightarrow 5.732889 \times 10^3$
  - $0.032546 \Rightarrow 3.2546 \times 10^{-2}$

- 用於二進位數字系統
  - 將二進位數改寫成 $a \times 2^n$ 形式，其中 $1 \leq a < 2$，且 n 為整數
    - $(1011.01101)_2$ 標準化後會記為 1.01101101×2^3
    - $(0.001011)_2$ 標準化後會記為 1.011×2^{-3}

浮點數表示法 (續)

- 小數點右邊的數，稱為尾數 (mantissa，以 1.01101101×2^3 為例，其尾數為01101101)
- 2 的指數次方，稱為指數 (exponent，以 1.01101101×2^3 為例，其指數為 3)

- 如某數的尾數為 0100110 且指數為 − 8
  - => 某數等於 1.0100110×2^{-8}

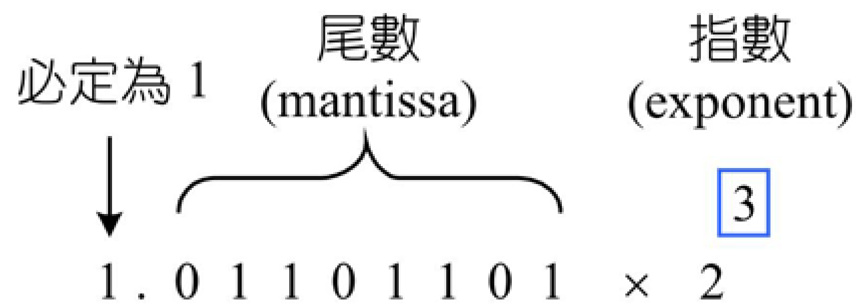


圖 2.17 尾數與指數

浮點數表示法 (續)

- IEEE 754 標準，它將儲存的空間分成三個部分：符號位元 (sign bit)、指數部分及尾數部分

圖 2.18 IEEE 754 標準 —— (a) 單倍精確度；(b) 雙倍精確度

浮點數表示法 (續)

$$\underset{\downarrow}{\overset{\text{必定為 1}}{1}}.\overbrace{0\ 1\ 1\ 0\ 1\ 1\ 0\ 1}^{\text{尾數 (mantissa)}} \times 2^{\overset{\text{指數 (exponent)}}{\boxed{3}}}$$

- 以單倍精確度為例：

1. 符號位元：以 0 代表正數，1 代表負數
2. 指數部分：以過剩 127 (Excess 127) 的方式表現，其中 127 即為單倍精確度的**指數偏移值 (exponent bias)**
   - 將標準化後的數字的指數部分加上 127 後，再儲存下來
3. 尾數部分：儲存標準化後的小數部分，不夠的位元部分則補 0

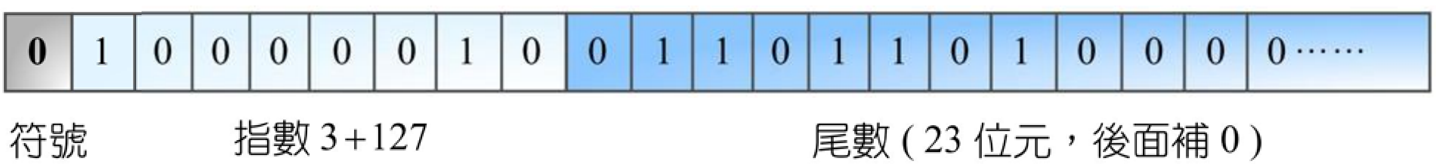


圖 2.19 以單倍精確度儲存 1.01101101×2^3

- 在實務儲存上，有些值域是被保留下來用以表示一些特殊值

浮點數表示法 (續)

表 2.6 單倍精確度浮點數部分極值的情況

類別	正負號	實際指數	指數部分 (實際指數 +127)	尾數部分
零	0	–127	0	000 0000 0000 0000 0000 0000
負零	1	–127	0	000 0000 0000 0000 0000 0000
1	0	0	127	000 0000 0000 0000 0000 0000
–1	1	0	127	000 0000 0000 0000 0000 0000
正無限大	0	128	255	000 0000 0000 0000 0000 0000
負無限大	1	128	255	000 0000 0000 0000 0000 0000

浮點數表示法 (續)

請將二進位實數 0.00001101001 用單倍數精確度方式儲存

解答：

- 先將 0.00001101001 正規化 $\rightarrow 1.101001 \times 2^{-5}$
- 符號為正
- 實際指數為 -5，經偏移後得指數部分 $-5+127=122$
- 尾數部分為 101001

0	0	1	1	1	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0……

共 23 位元

浮點數表示法 (續)

隨堂練習

- 請將二進位實數 − 110100.101 用單倍數精確度方式儲存

解答：

- 先將 − 110100.101 正規化 → − 1.10100101 × 2^5
- 符號為負
- 實際指數為 5，經偏移後得指數部分 5 + 127 = 132
- 尾數部分為 10100101

1	1	0	0	0	0	1	0	0	1	0	1	0	0	1	0	1	0	0	0	0……

共 23 位元

浮點數表示法 (續)

隨堂練習

- 11000110001101010000000000000000 是一個用單倍數精確度方式儲存的實數，請問它代表的數字為何？

解答：

- 符號位元為 1，故代表一個負數
- 指數部分為 $(10001100)_2 = (140)_{10}$，將偏移數減去後，即可得實際指數為 $140 - 127 = 13$
- 尾數部分為 01101010000000000000000
- 故實際代表的二進位數為 -1.0110101×2^{13}

學習重點

- 二進位表示法
- 進位轉換
  - 二進位與十進位
  - 二進位與十六進位
  - 二進位與八進位
- 二的補數
- 浮點數表示法
  - 標準化
  - 尾數與指數
  - 指數偏移