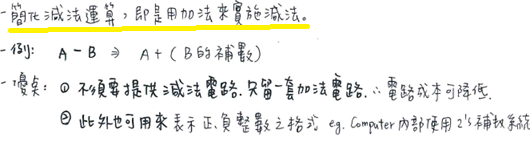
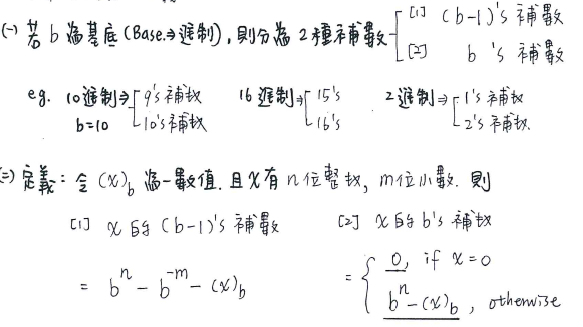
**CH 3 補數系統 Complement**

1. **目的** – 以加法實現減法
2. **種類與定義** – b進位制 🡪 (b - 1)’s補數 + 1 = b’s補數；最大值 – 欲轉換之數 = B – 1 補數
3. **無號數(2進位)**
4. **有號數(2進位)** – 正 : 最左為0，其他不變；負 : sign magnitude : 最左位數判斷0正1負；1’ complement: 對正數(除了最左位以外的數)1換0，反之；2’ complement : 1補數 + 1
5. **四則運算(Sign magnitude)**
6. **1’s Complement** – 值域 : - ( ) ~ 🡪有+0、-0；進位 : 端迴進位(左邊進位給右邊)；沒進位 : 加負號，取1補數(將補數轉換為整數)
7. **2’s Complement –** 值域 : - ( ) ~ ；進位 : 左邊捨去；沒進位 : 加負號，取2補數(將補數轉換為整數)
8. **Excess Code** – x + (存入)；反其值，扣回去
9. **目的**



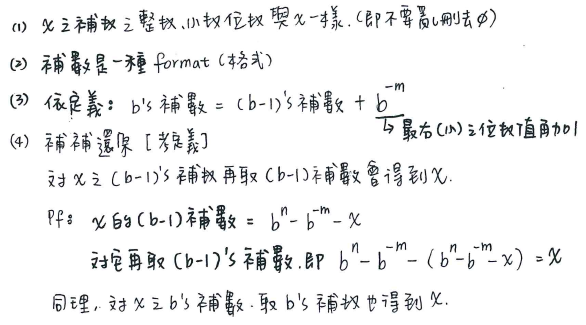
1. **種類與定義**



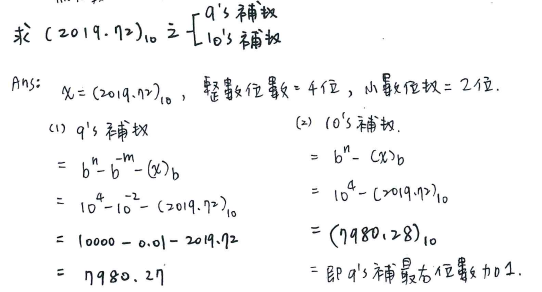
* + - B – 1 補數有正負0所以要再扣最小位數次方之值，B – 1 補數與其原來的數將加不會超過該位數之最大值，看下列例子

Ex. +3(011) + (-3)(100) = 0(111) -> 111為3位數中最大之值

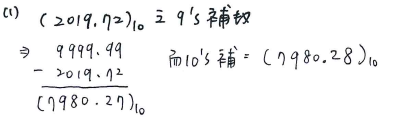
* + - N的B – 1 補數之值最右位數+1 = B補數
  + 補充

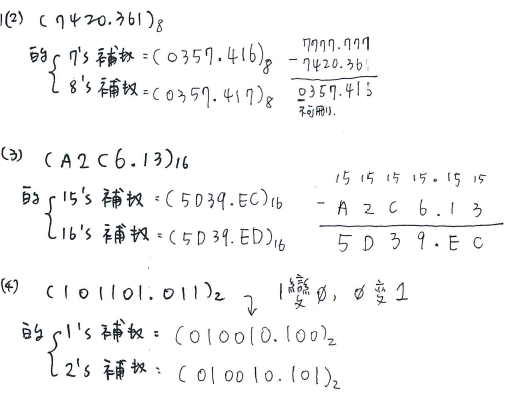


* + 例子

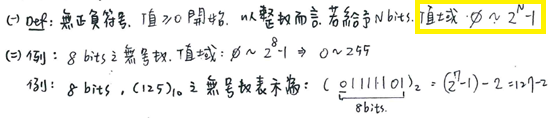


* + 快速做法
    - 最大值 – 欲轉換之數 = B – 1 補數
    - B補數 = B – 1 補數最右位數 + 1

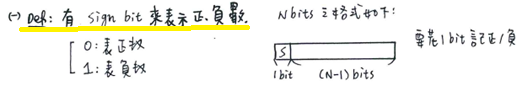




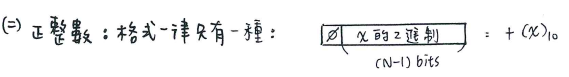
1. **無號數(Unsigned Number)**

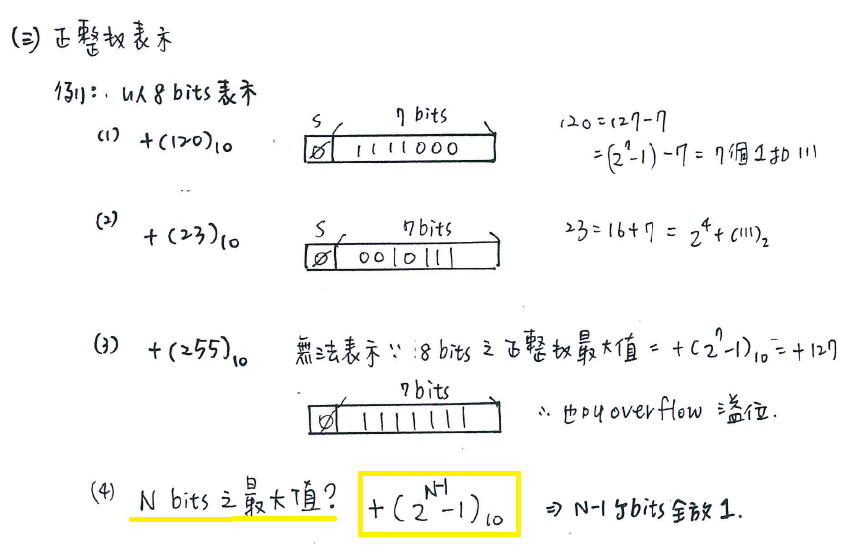


1. **有號數(signed Number)(整數)**



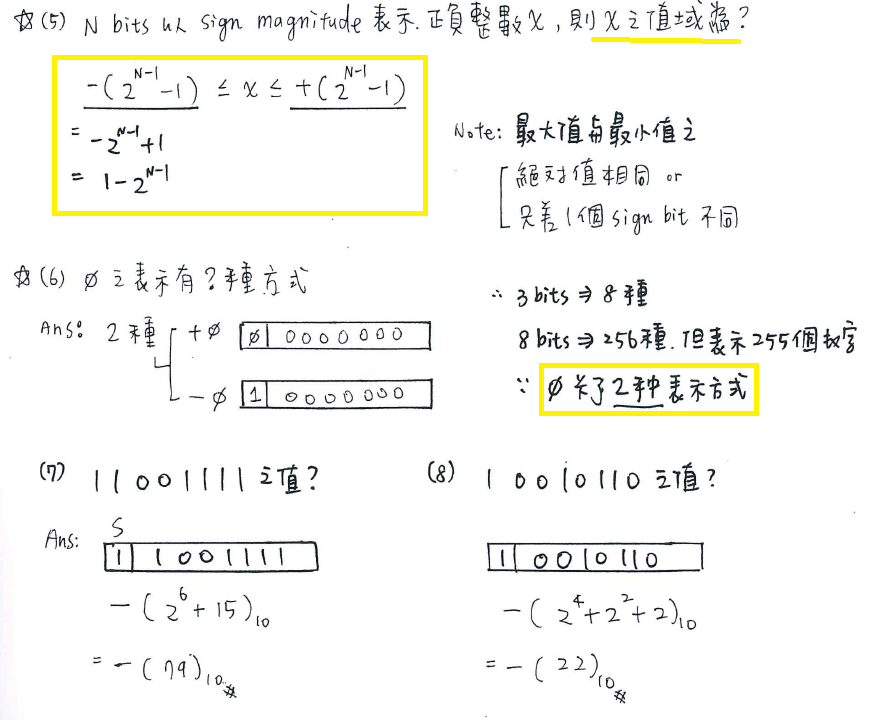
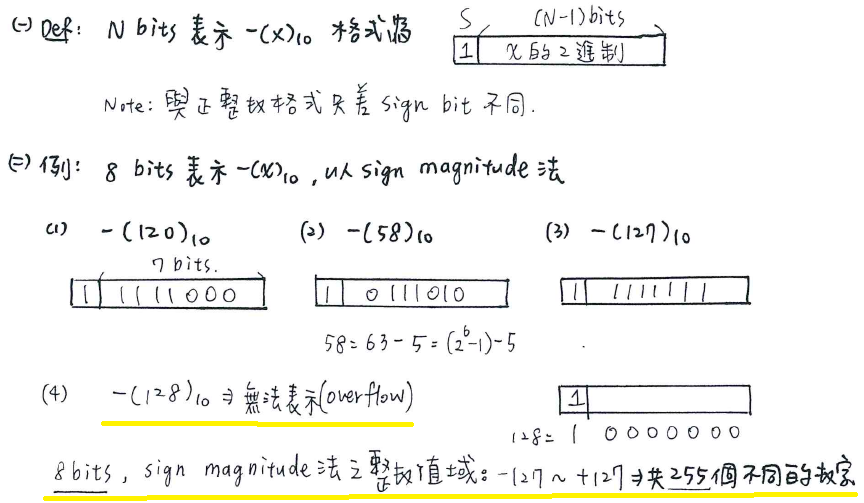
* + **正整數**





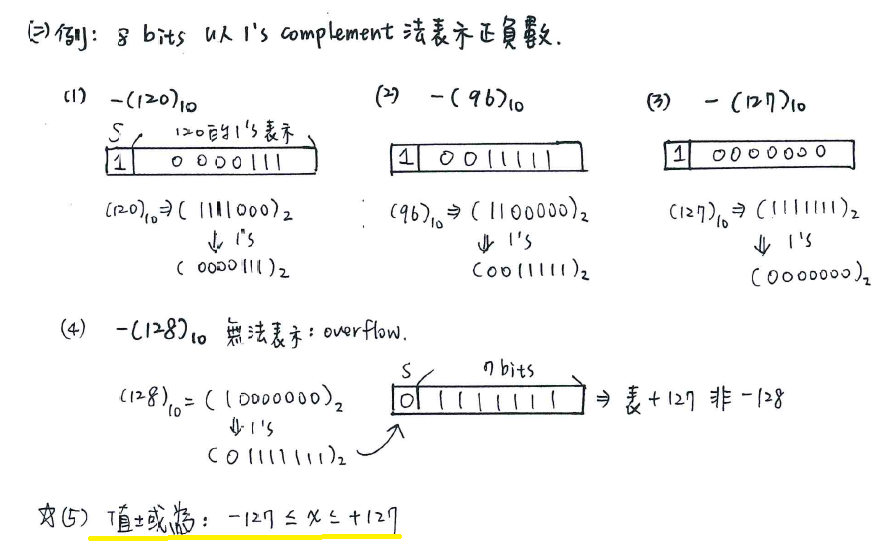
🡪因為全0為0所以扣掉1

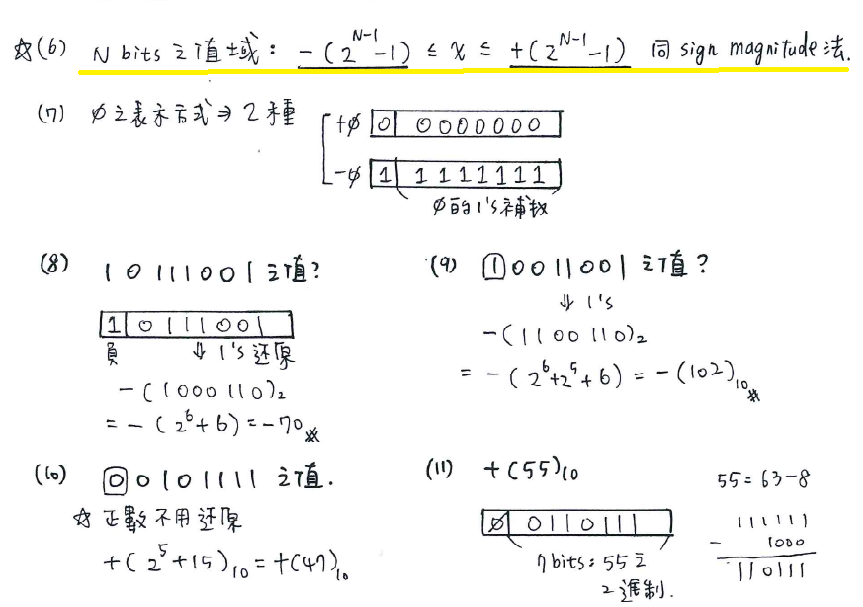
* + **負整數(三種)**
    - 1. Sign magnitude



* + - 1. **1’s complement (1補數)**
    - 2進位的1補數，0換成1，1換成0

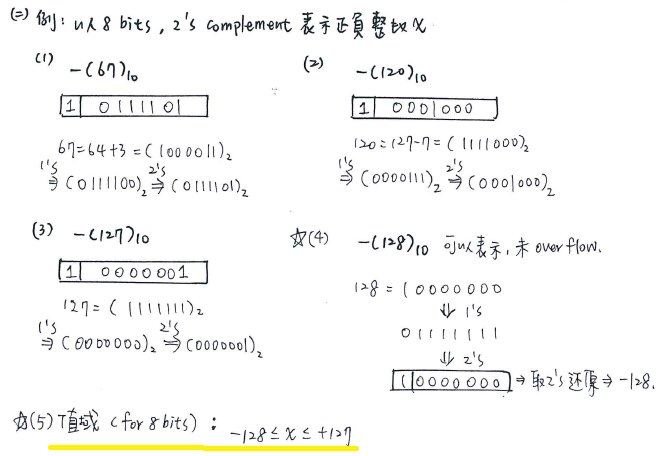


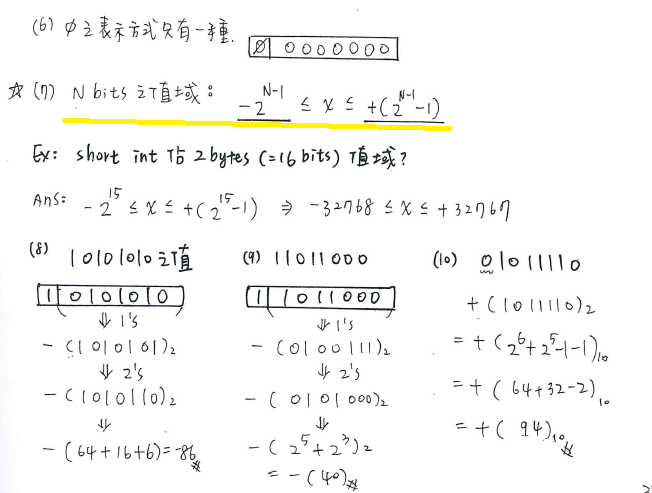


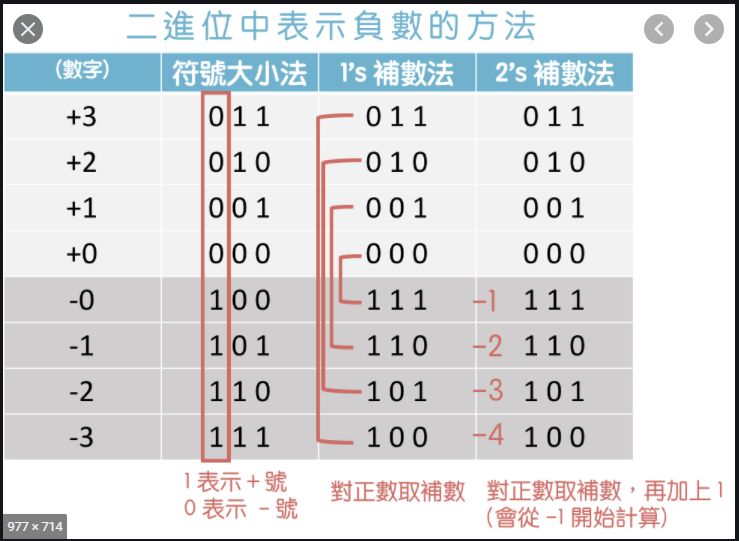


1. **2’s complement (2補數)**

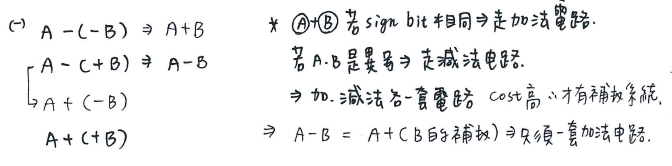




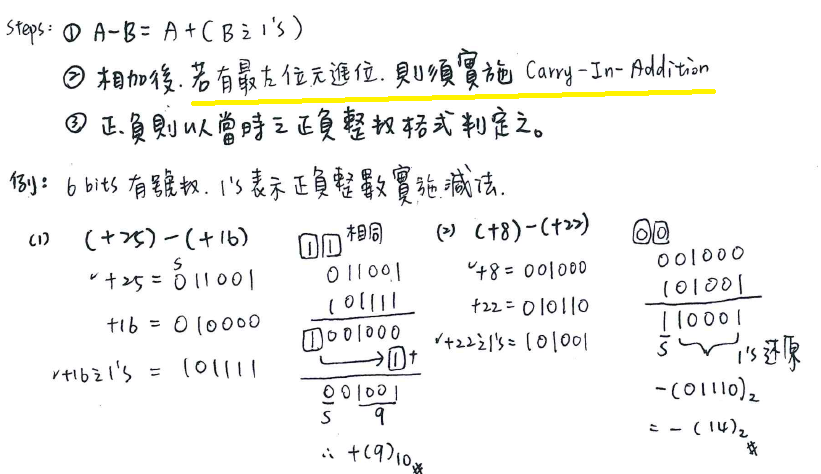




1. Sign magnitude 加減法運算



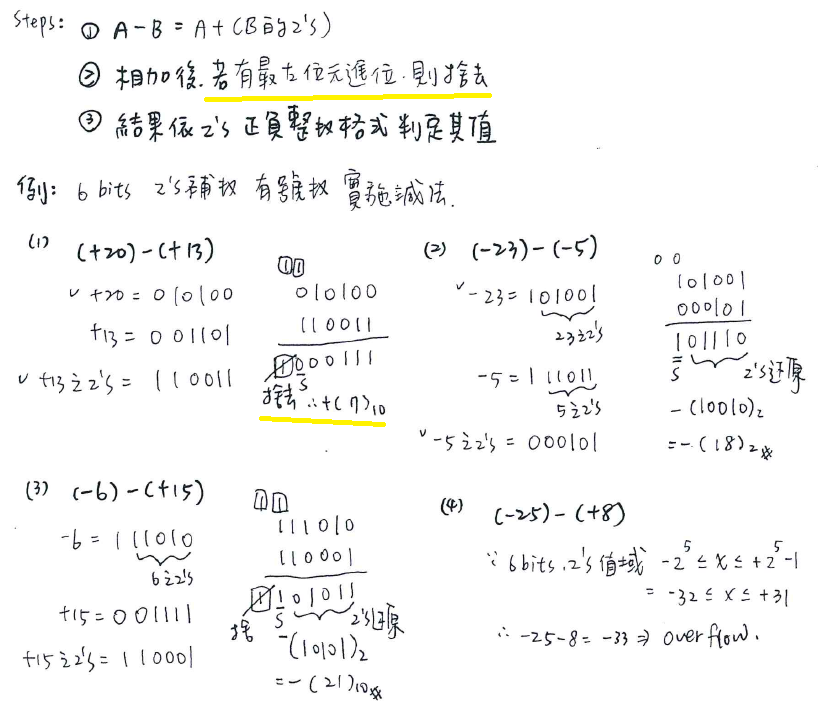
1. 1’ s complement (1補數)
   * 可能有溢位或數值表示錯誤(以3bits為例)
     + 正 + 正 : 011(+3) + 001(+1) = 101， 此數為-2
     + 負 + 負 : 110(-1) + 100(-3) = 1010，最左的bit記不起來被丟掉，變正數



🡪 未還原之值為1補數之結果

🡪 再取補數是因為將1補數還原為整數形式

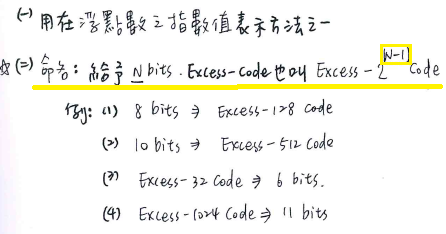
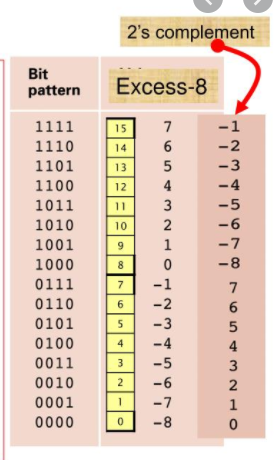
1. **2’s complement (2補數)加減法**

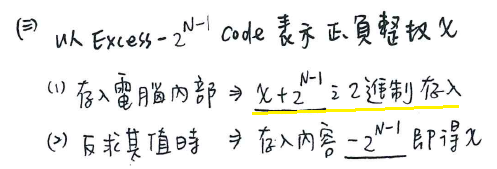
****

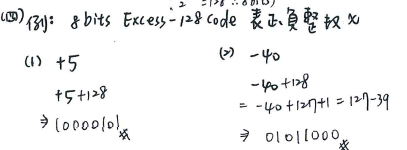
🡪 未還原之值為2補數之結果

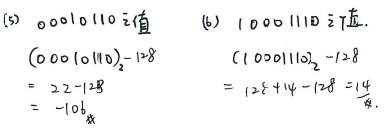
🡪 再取補數是因為將2補數還原為整數形式

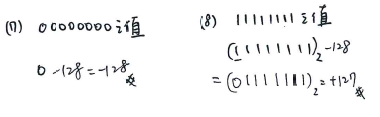
1. **Excess-code**











* + 正 + 正 或 負 – 負 或 正 – 負
    - 可能溢位

