**期中project、Convolutional Code**

1. **原理介紹**

卷積碼(convolution code)的由來，是因為輸入的原始訊息資料會和編碼器(encoder)的脈衝響應(impulse response)做卷積運算。卷積碼的編碼器是由m個移位暫存器 (shift register)及n個二進位加法器構成，k個輸入位元移入移位暫存器並同時計算出n個輸出位元，編碼率(code rate)為k/n，常表示為(n,k,m) 卷積碼。

編碼方式

原始訊息資料依序由輸入端進入編碼器的暫存器(register，圖內簡稱reg.)，每一個暫存器會儲存一個輸入字元，而它們的起始值都是0。依圖1.1而言，編碼器內有3個二模數加法器(modulo-2 adder，可等於一個Boolean XOR gate，運算方式是0+0 = 0, 0+1 = 1, 1+0 = 1, 1+1 = 0)對儲存的3位元原始資料，做各自的加法運算。接著，暫存器內的字元會移往下一格，(reg1 moves to reg2, reg2 moves to reg3)；然後繼續將資訊傳至輸出端，如此便可以得到要傳輸的內容。

output1=reg.1+reg.3

output2=reg.1+reg.2+reg.3

運算後，輸出端（output）則輸出編碼後的卷積碼資料。

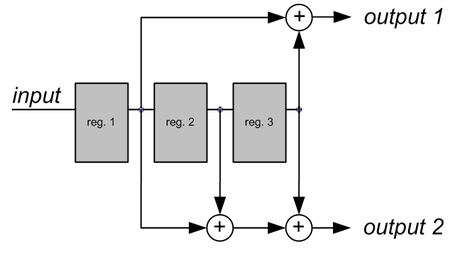


圖1.1

由於原始訊息資料是依序輸入至編碼器，所以3個暫存器儲存的資料是不同時間點的輸入值：reg. 1儲存目前訊息資料，reg. 2儲存前一週期的資料，reg. 3則是前前一週期的資料。因此，每筆卷積碼資料皆與過去的訊息資料有關係，因而保有記憶效應(memory property）。

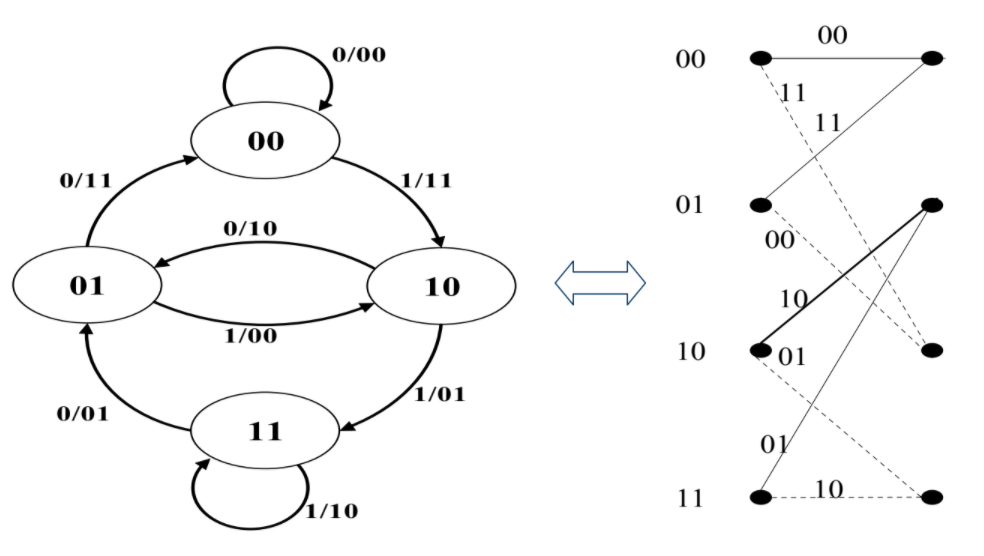
convolution code編碼器輸出是由輸入的資料位元及移位暫存器的內容值所決定，我們把存放在移位暫存器的內容值當作狀態(state)，故假如一個摺積碼編碼器有m個移位暫存器，則編碼器會有2m個不同的狀態。以m=2編碼器為例，它的編碼狀態圖和網格圖會如圖1.2所示。

圖1.2

1. **程式碼**

這次的作業我使用的軟體是MATLAB，程式碼如圖2所示：

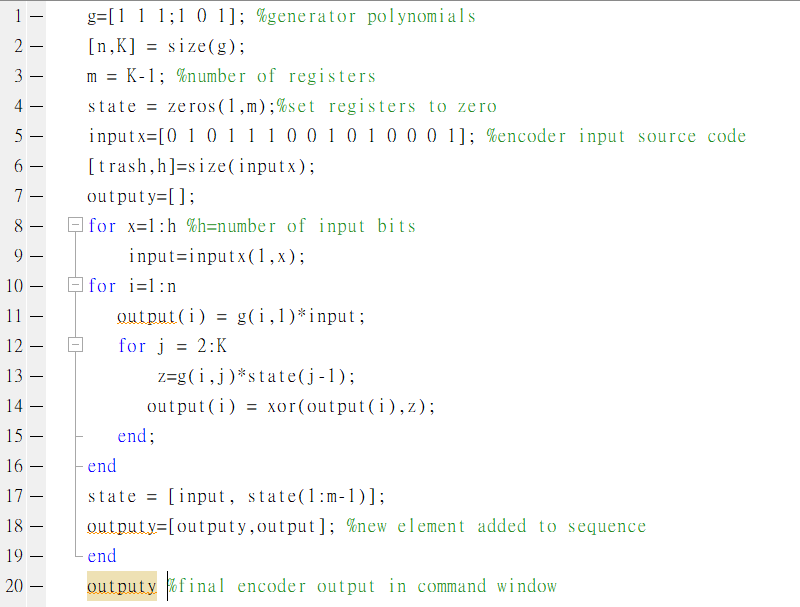


圖2

1. **執行結果**

如圖3所示：

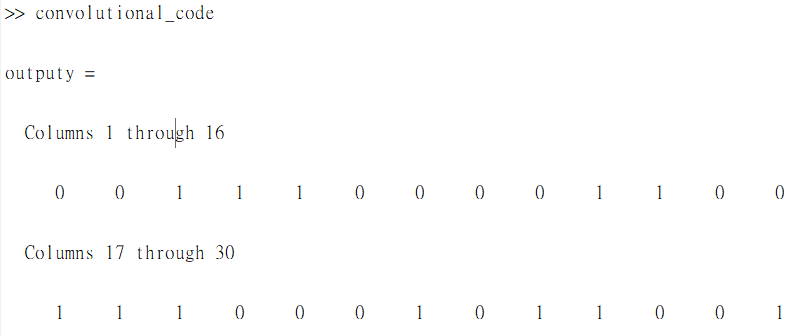


圖3