

3-3 減法器

有了加法的運算，當然也要有減法運算。二進制的減法運算為 $0-0=0$ 、 $0-1=-1$ 、 $1-0=1$ 、 $1-1=0$ ；其中 $0-1=-1$ 中的 -1 無法用二進制的 0 與 1 兩個數字表示，一般用兩個位元表示，一個是借位(Borrow)，一個是差(Difference)。當輸入端有兩個變數，產生一個差與借位輸出的電路結構，稱為半減器(Half Subtractor; HS)。若在兩個變數相減時，即考慮到上一個位元借位的問題，則輸入端將有三個變數相減(即相減的兩數和上一位元的借位)，同樣的產生一個差與一個借位的輸出，此種電路結構稱為全減器(Full Subtractor, FS)。

3-3-1 半減器一

兩個變數的相減，用半減器即可完成，真值表如下所示，輸入數值有 A、B 值，A 代表被減數，B 代表減數，D 表示差，Bo 表示借位。

真值表

輸入		輸出	
A	B	D	Bo
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

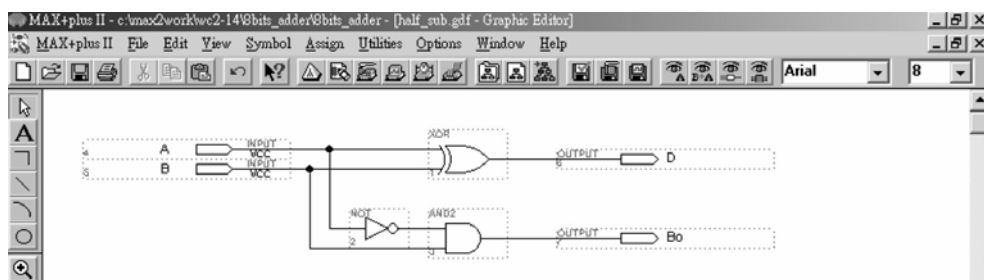


圖 3-3-1 半減器電路圖

如圖 3-3-1 所示，經由步驟：畫電路圖→存檔→設為工作專案→指定 CPLD 晶片→編譯→規劃腳位→編譯→下載燒錄(若是非 ALTERA 的晶片下載前需轉檔)後，即完成電路設計製作，此時可配合外部電路(或實驗器模組，需配合腳位規劃)來驗證。

表 3-3-1 電路圖輸入腳位、CPLD 腳位及實驗器模組腳位對應表

腳位對應關係	輸入		輸出	
名稱	被減數(A)	減數(B)	差(D)	借位(Bo)
CPLD 晶片腳位	PIN 9	PIN 10	PIN 80	PIN 81
實驗器模組對應腳位	DIPB 1	DIPB 2	DG6(LED)	DG7(LED)

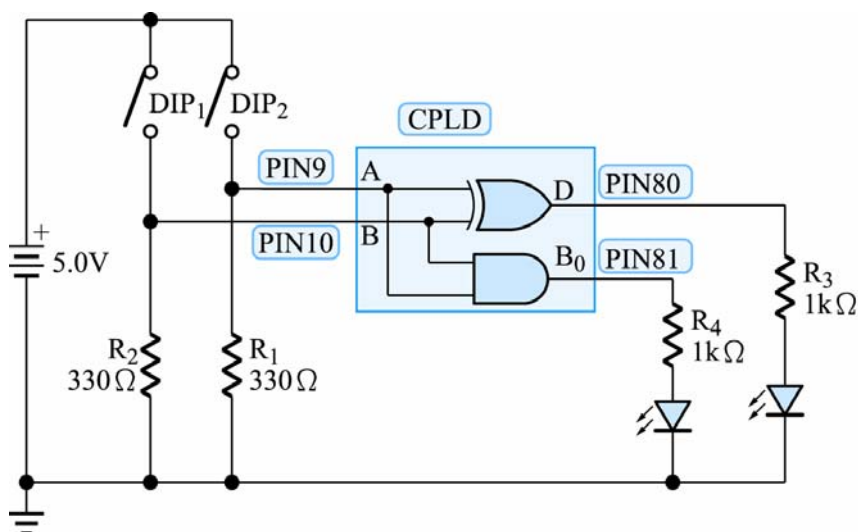


圖 3-3-2 半減器實驗電路圖

3-3-2 半減器二

可以製造半加器的元件，自然也可製造半減器元件，製作步驟如下：

1. 開啟一個新圖形編輯檔，繪製一個半加器的電路圖。
2. 存檔，取檔名為 half_sub.gdf。
3. 編譯。
4. 產生符號檔，點選 File → Create Default Symbol，即可產生新的零件 IC(半減器，名稱 half_sub)。

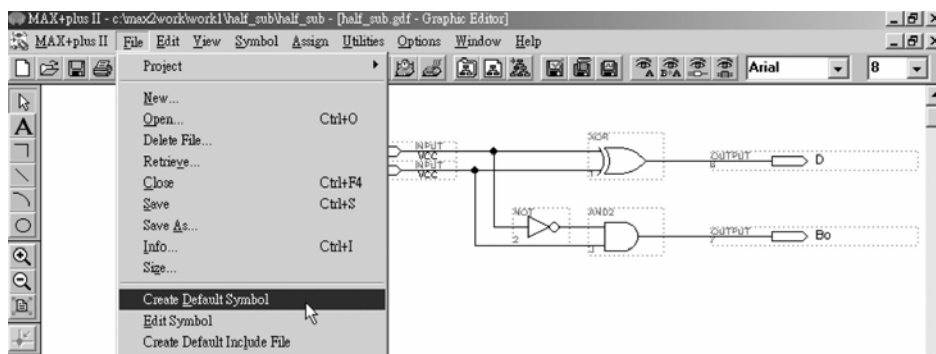
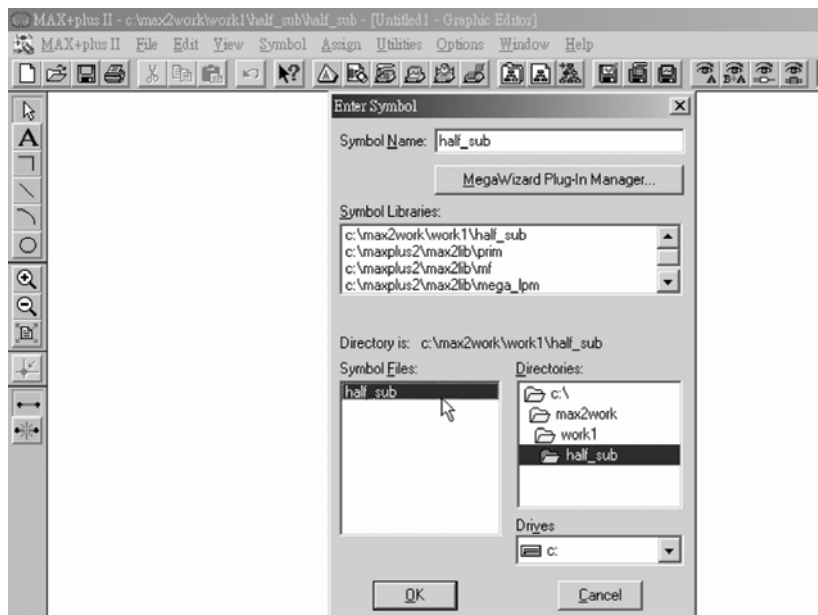


圖 3-3-3 製作自製元件視窗

5. 開啟新的圖形編輯檔，此時即可取用所製作的零件 IC(half_adder)，點選 **Symbol → Enter Symbol**(或在編輯視窗按兩下 **Double Click**)，若出現如下圖之錯誤訊息時不要理它，它只是說尚未定義零件插入點而已，按**確定**後，即出現零件取用視窗。

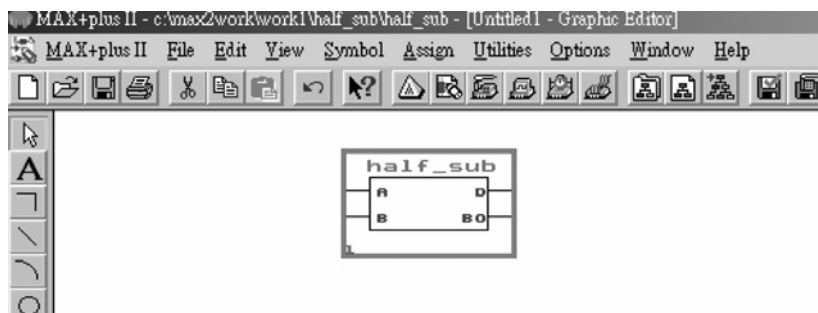


● 圖 3-3-4 插入元件視窗



● 圖 3-3-5 元件取用視窗

6. 取用半減器零件 IC：進入剛才存放半加器的資料夾，點選符號元件檔(Symbol File) half_adder，按 **OK**，在圖形編輯視窗中，即可看到該元件(包裝後的半減器)。可在半減器元件上按兩下，即可開啟其內部電路視窗。



● 圖 3-3-6 自製半減器元件

設計半減器零件 IC 完成後，即可叫出包裝起來的半減器(half_sub)，可用模擬功能測試是否正常。

模擬步驟如下：

1. 在新的編輯檔案中，叫出該元件，並叫出輸入輸出接腳且連接好線路後存檔 (Save As)，在此取名為 half_sub1.gdf，設為指定工作專案(File→Project→Set Project to Current File)，指定 CPLD 晶片 (Assign→Device)，並編譯 (MAX+PLUS II → Compiler)。

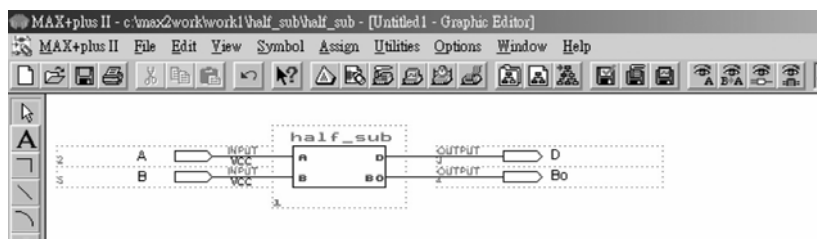


圖 3-3-7 半減器自製元件電路

2. 開啟新的波形編輯檔案，設定功能模擬結束時間(File→End Time)，設定格線間距(Options → Grid Size)，顯示在視窗中適當大小格線(View→Fit in Window)。

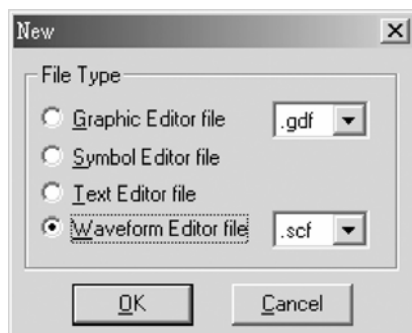


圖 3-3-8 開啟新檔視窗

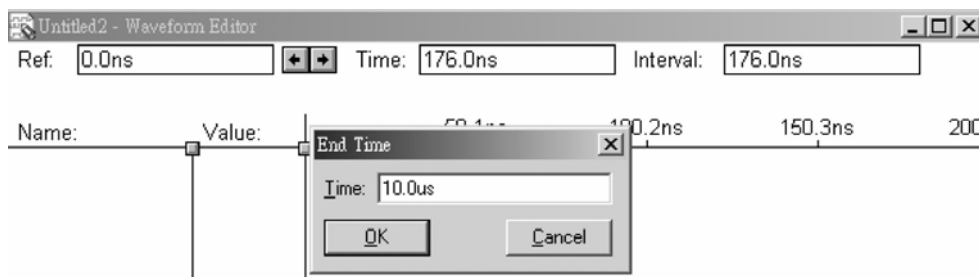
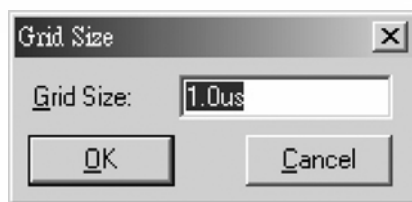
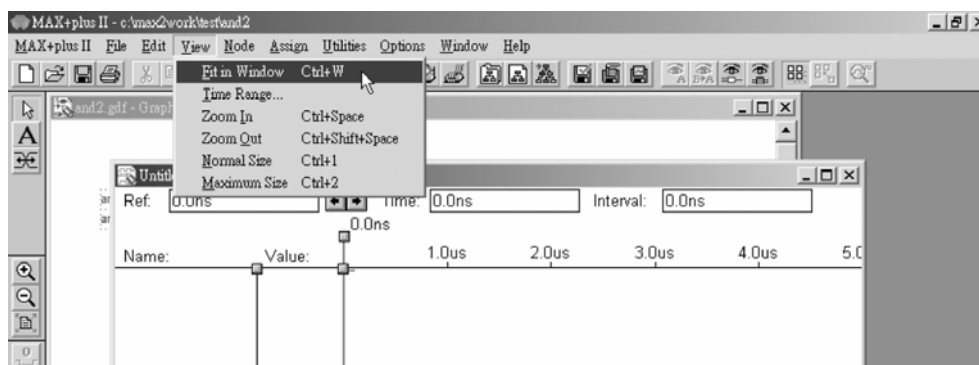


圖 3-3-9 模擬結束時間設定視窗

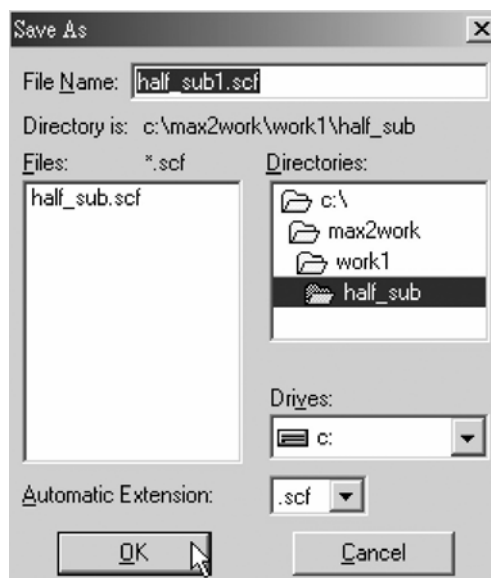


● 圖 3-3-10 模擬單位時間設定視窗

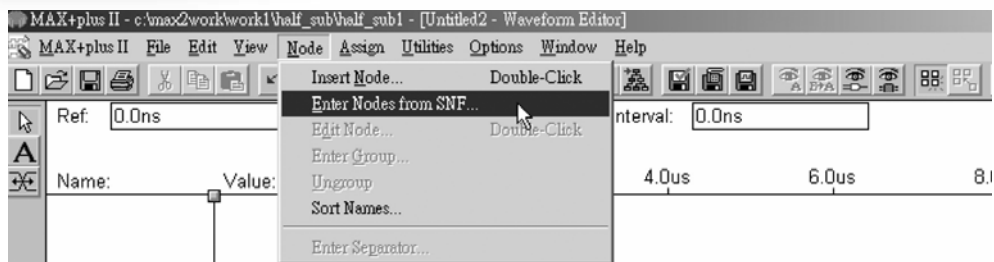


● 圖 3-3-11 視窗調整

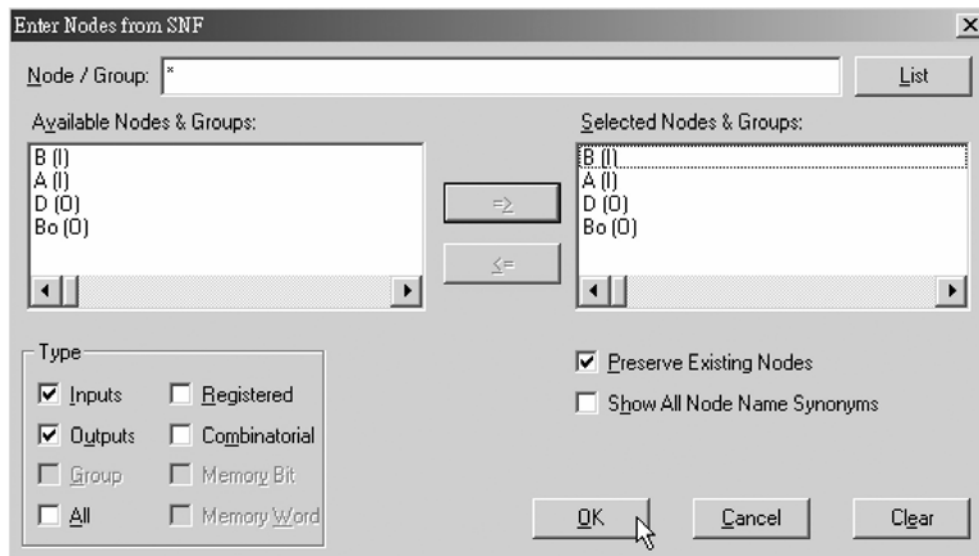
3. 儲存檔案(Save As)，在此取名檔案為 half_sub1.scf，輸入節點(Node → Enter Nodes from SNF，按 List 及 \geq ，OK)，編輯輸入時脈(X)。



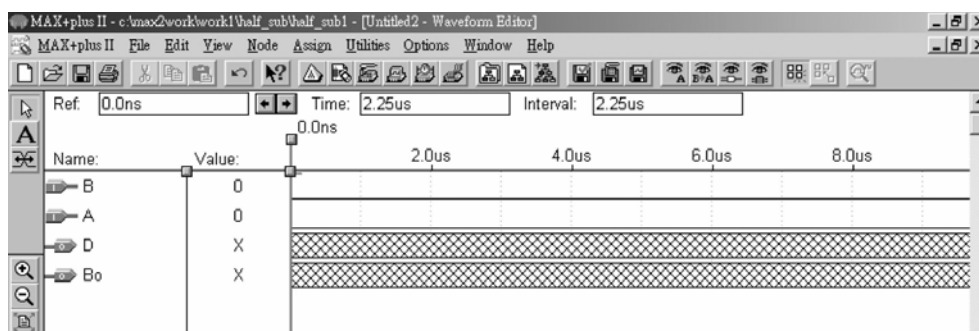
● 圖 3-3-12 儲存檔案視窗





● 圖 3-3-13 執行輸出入節點選擇視窗

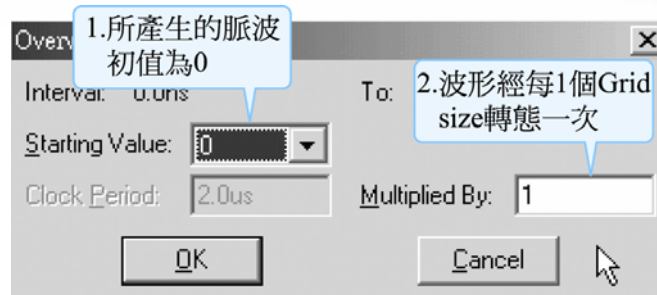


● 圖 3-3-14 輸出入節點選擇視窗

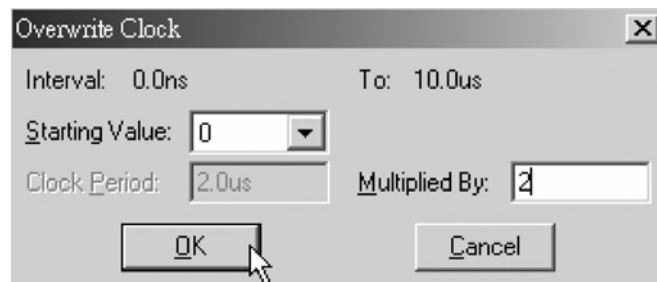


● 圖 3-3-15 模擬波形設定

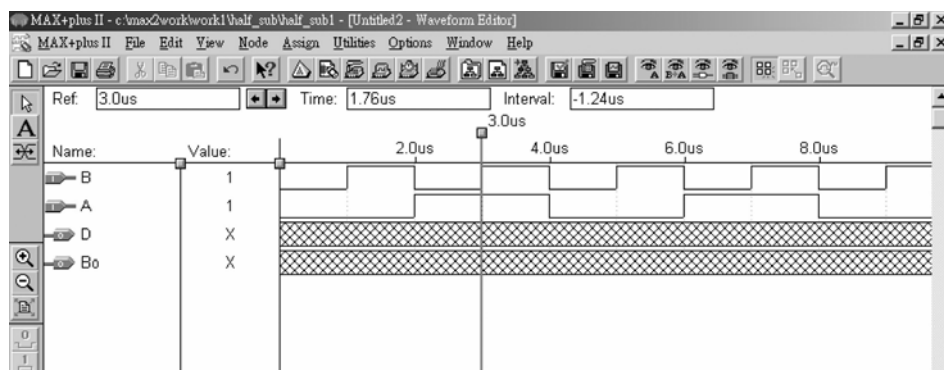
4. 點選 B，再點選，設定波形經每 1 個 Grid Size 轉態一次(Multiplied By = 1)；
點選 A，再點選，設定波形經每 2 個 Grid Size 轉態一次(Multiplied By = 2)。



● 圖 3-3-16 時脈信號設定視窗

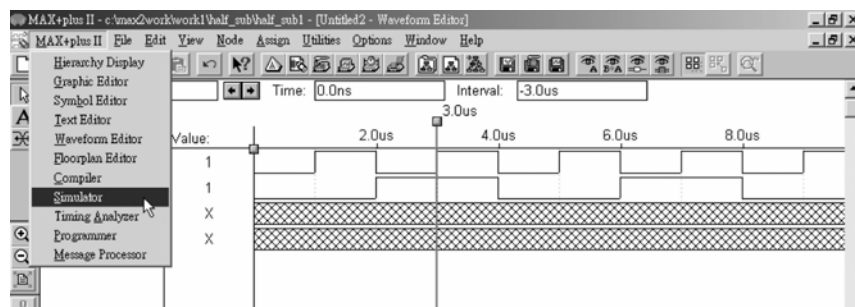


● 圖 3-3-17 時脈信號設定視窗



● 圖 3-3-18 模擬波形設定

5. 執行模擬(MAX+PLUS II → Simulator, Start)如圖 3-3-21 所示，所得波形模擬結果符合半減器元件，代表我們製作的零件 IC 是正確可用的。



● 圖 3-3-19 執行模擬

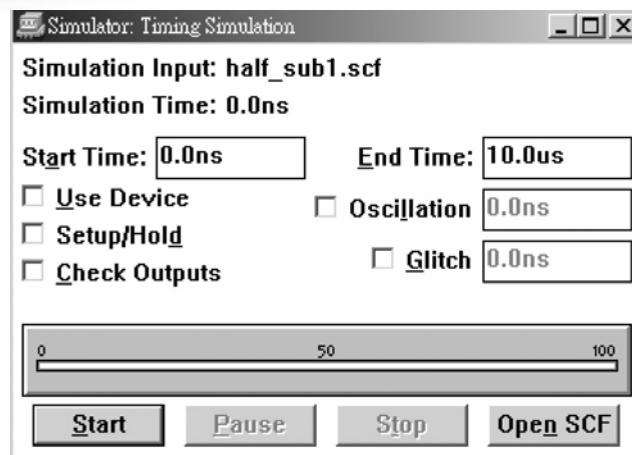


圖 3-3-20 模擬起始視窗

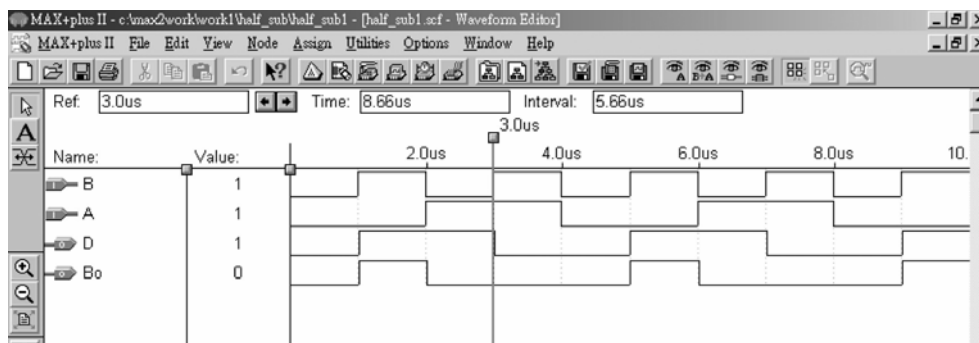


圖 3-3-21 模擬結果視窗

3-3-3 全減器(一)

三個變數的相減，需用全減器才可完成，一般是指相減的兩數之外，尚有上一個位元的借位值。真值表如下所示，輸入方面有被減數 A、減數 B 及 Bi 表示上一個位元的借位數值，輸出方面有 D 表示差，Bo 表示借位輸出。

真值表

輸入			輸出	
A	B	Bi	D	Bo
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

布林函數：

$$D = \overline{A} \overline{B} B_i + \overline{A} B \overline{B}_i + A \overline{B} \overline{B}_i + A B B_i$$

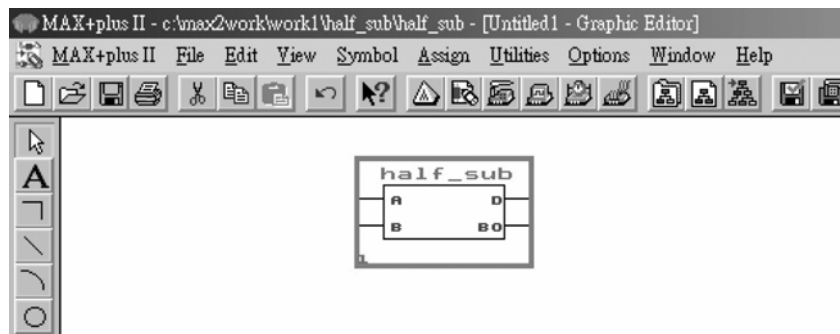
$$B_o = \overline{A} \overline{B} B_i + \overline{A} B \overline{B}_i + \overline{A} B B_i + A B \overline{B}_i$$

依照布林函數繪圖，即可完成全減器，在此列入習題，請讀者自行操作，若有疑問，可參考全加器(一)之作法。

3-3-4 全減器(二)(使用半減器自製元件組成全減器)

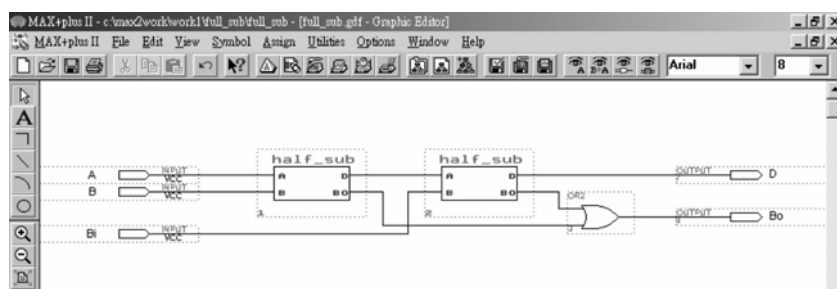
使用兩個半減器的組合亦可完成全減器功能，以下介紹作法。

1. 新增資料夾，命名為全減器(full_sub)。複製半減器元件的繪圖檔(half_sub.gdf)及元件檔(half_sub.sym)儲存於該資料夾。
2. 執行 MAX+PLUS II 軟體，開啟新的圖形編輯檔。
3. 取用半減器元件(symbol)：點選 **Symbol** → **Enter Symbol** 進入剛才存放半減器元件的資料夾 full_sub，點選符號元件檔(Symbol File) half_sub，按 **OK**，在圖形編輯視窗中，即可看到該元件(包裝後的半減器)。可在半減器元件上按兩下，即可開啟其內部電路視窗。



● 圖 3-3-22 自製半減器元件

4. 如圖 3-3-23 繪製全減器電路圖。

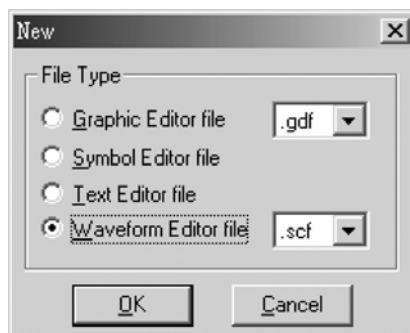


● 圖 3-3-23 自製半減器元件組成全減器電路圖

5. 存檔，取檔名為 full_sub.gdf
6. 設為指定工作專案(File→Project→Set Project to Current File)，指定 CPLD 晶片(Assign→Device)，並編譯(MAX+PLUS II →Compiler)。

如此便完成了全減器的設計，可使用模擬功能測試其是否正確，步驟如下：

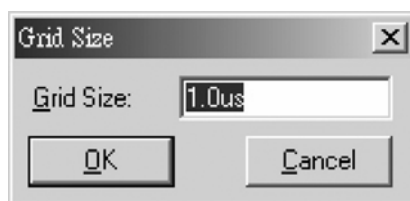
1. 開啟新的波形編輯檔案，設定功能模擬結束時間(File→End Time)，設定格線間距(Options→Grid Size)，顯示在視窗中適當大小格線(View→Fit in Window)。



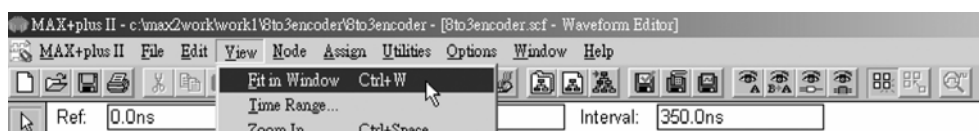
● 圖 3-3-24 開啟新檔視窗



● 圖 3-3-25 模擬結束時間設定視窗



● 圖 3-3-26 模擬單位時間設定視窗



● 圖 3-3-27 視窗調整

2. 儲存檔案(Save As)，檔名 full_sub.scf，輸入節點(Node → Enter Nodes from SNF，按 List 及 \Rightarrow ，OK)，編輯輸入信號。

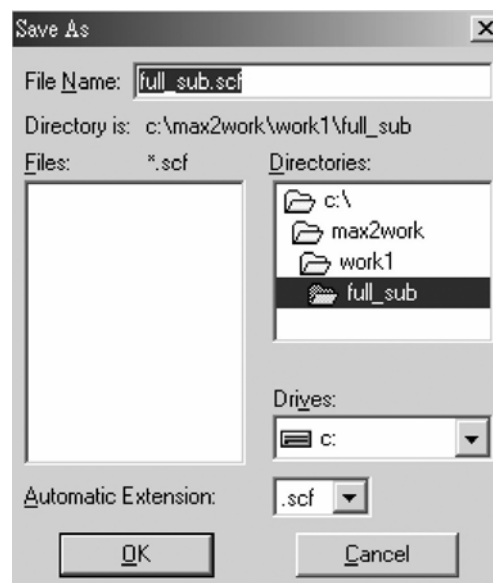


圖 3-3-28 儲存檔案視窗

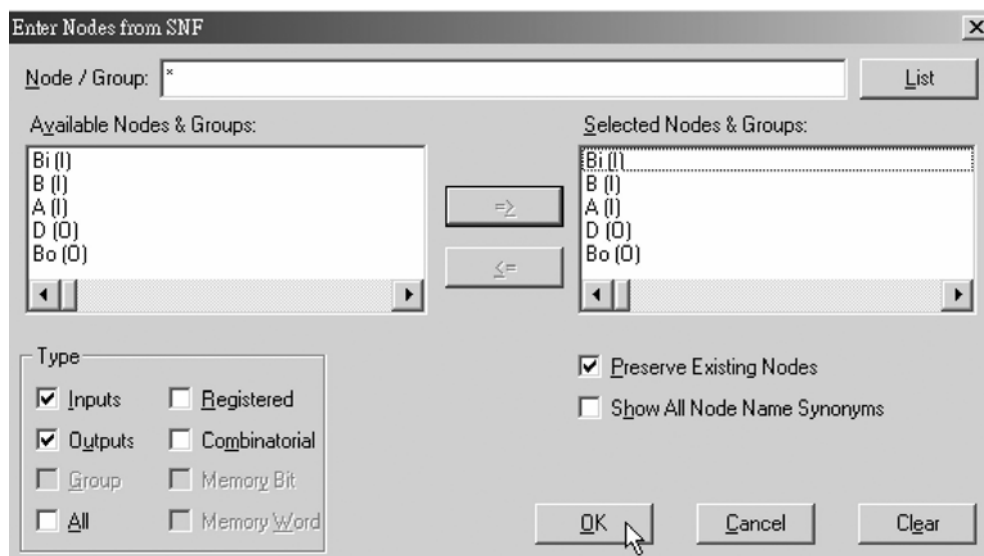


圖 3-3-29 輸出入節點選擇視窗

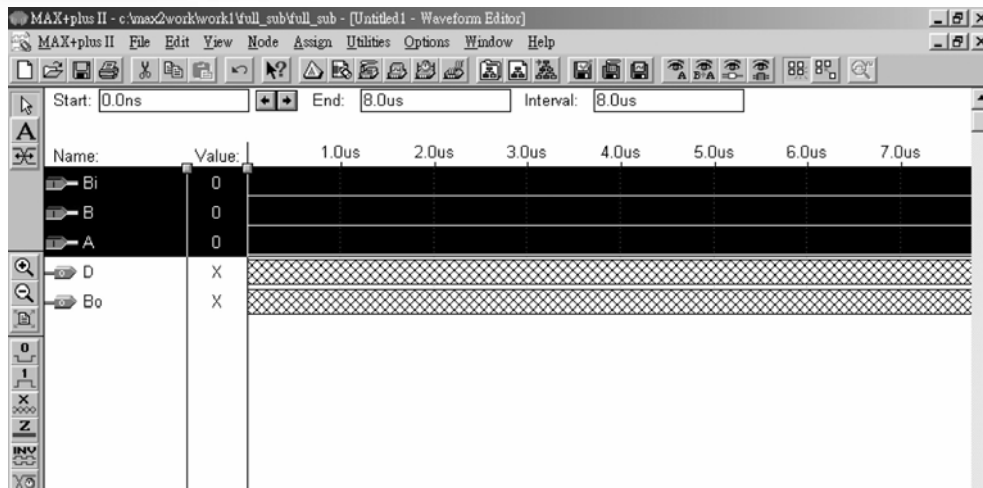


圖 3-3-30 模擬波形設定

- 編輯輸入信號有很多種作法，在此介紹一種群組化編輯法，就是將部份或全部輸出的腳位變成一個群組，直接對該群組編輯輸入訊號，可減少一支一支個別腳位編輯的麻煩。以此為例，輸入腳的部份有三支，要將各種狀況都模擬到的話需有八種組合。而其組合剛好可為(000~111)，所以可以先將三支輸入腳群組化後，再以計數時脈XC編輯。
- 如圖 3-3-31，拖曳滑鼠去選取輸入腳 Bo、B、A，使其反白，在選取區內按下滑鼠右鍵，選擇 Enter Group，此時出現 Enter Group 視窗，輸入群組化後的群組名稱，在此取名 input。

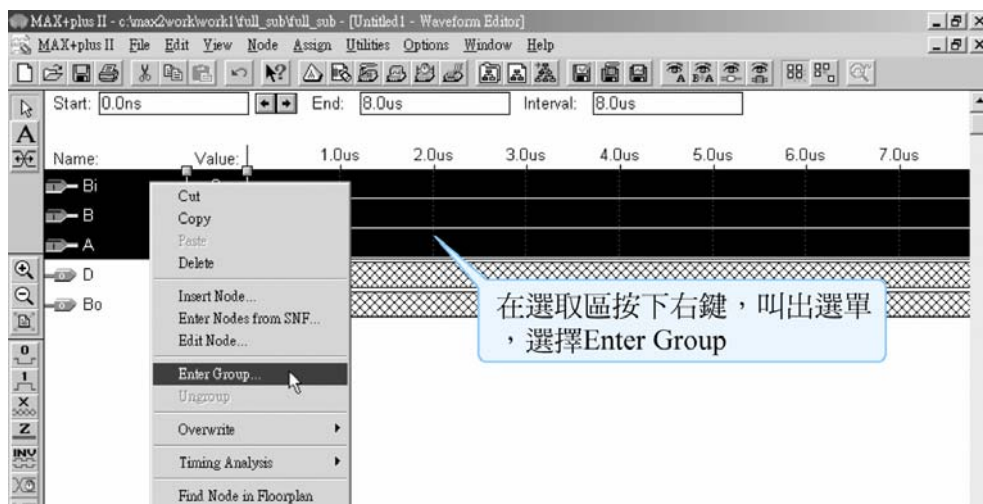
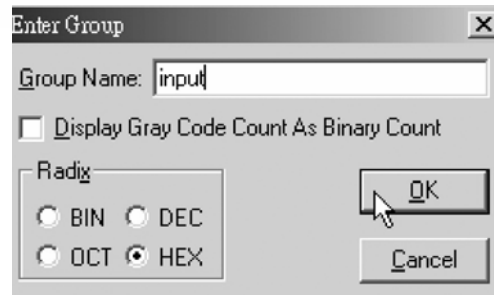
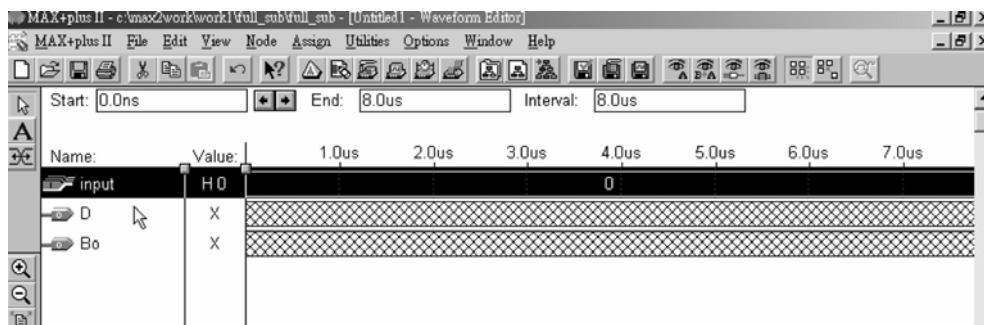


圖 3-3-31 模擬波形設定—輸出入接腳群組化

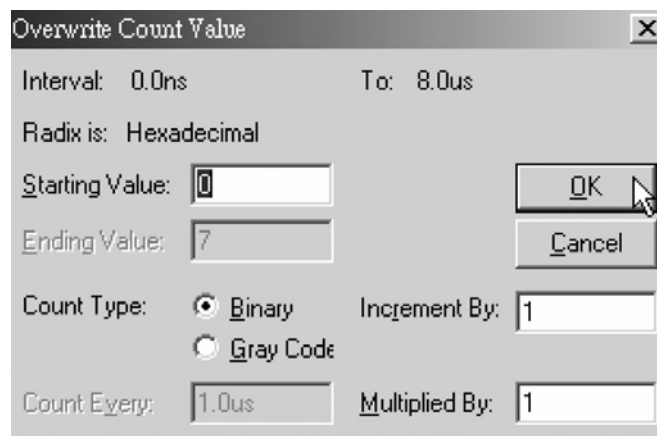


● 圖 3-3-32 模擬波形設定—輸出入接腳群組化輸入名稱



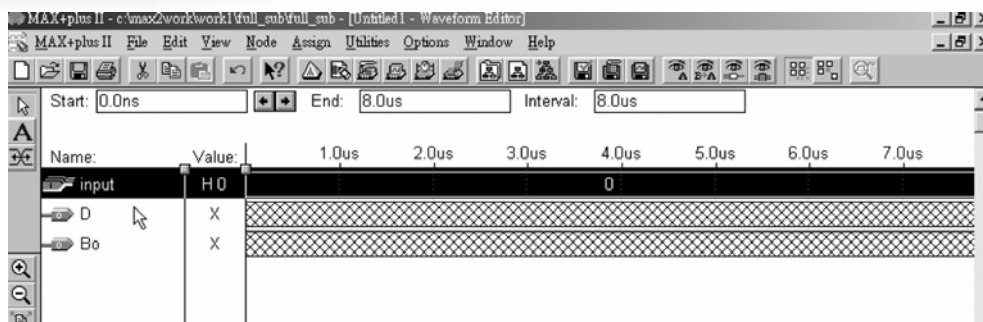
● 圖 3-3-33 模擬波形設定

5. 群組化後，可用計數時脈 XC 編輯，在此設定波形經每 1 個 Grid Size 轉態一次(Multiplied By = 1)，且每次轉態時計數器數值增加 1 (Increment By = 1)。

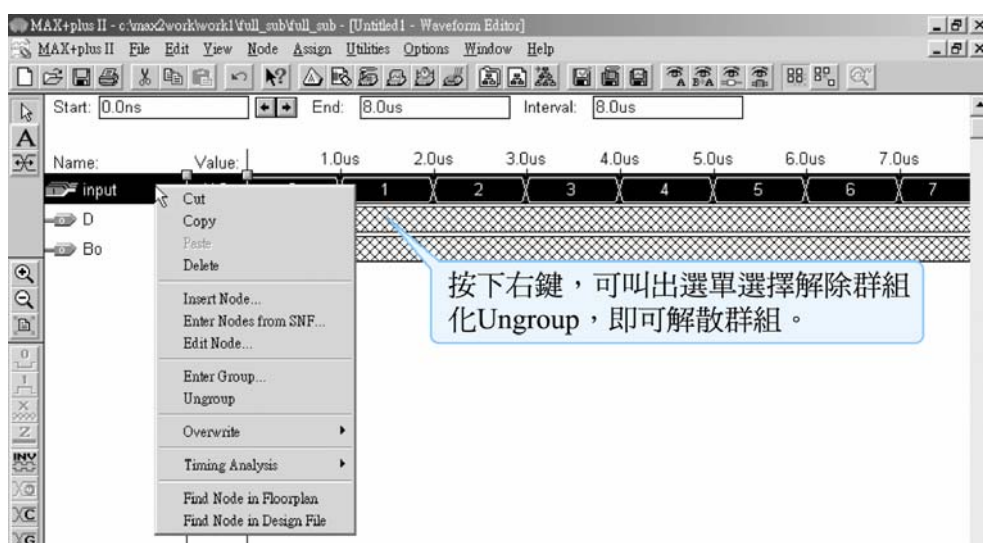


● 圖 3-3-34 計數信號設定視窗

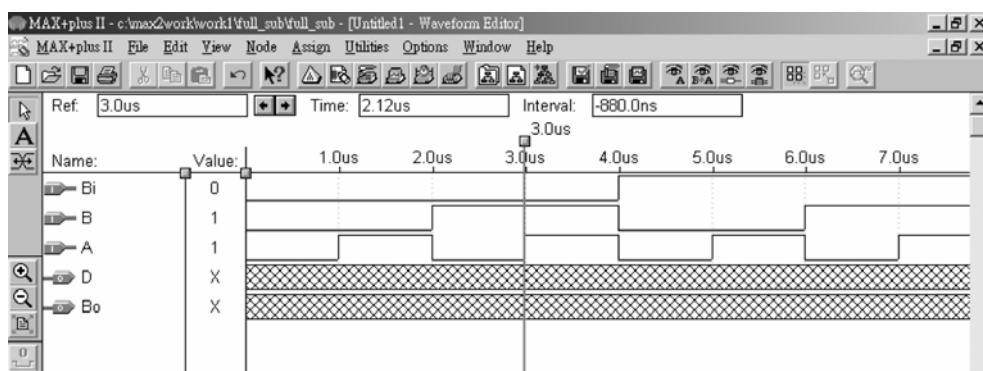
6. 編輯好群組化後輸入腳 input 的輸入時脈後，可如圖在 input 腳按右鍵，叫出選擇視窗後按下 Ungroup 來解除群組化，即可看出已經依照 000~111 編輯好時脈了。(此步驟也可不做就直接進行模擬，在此為示範用，讀者可選擇做與不做，也可模擬結束後再做。)



● 圖 3-3-35 模擬波形設定

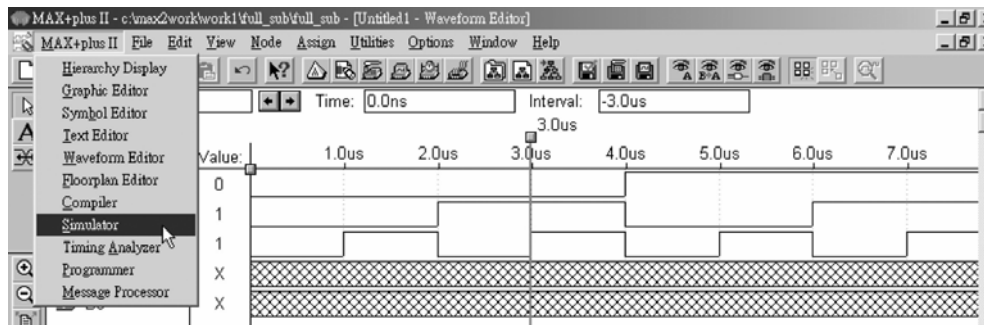


● 圖 3-3-36 模擬波形設定

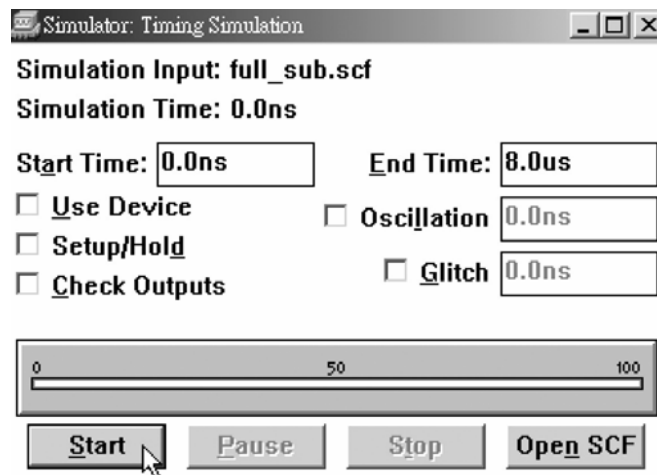


● 圖 3-3-37 模擬波形設定

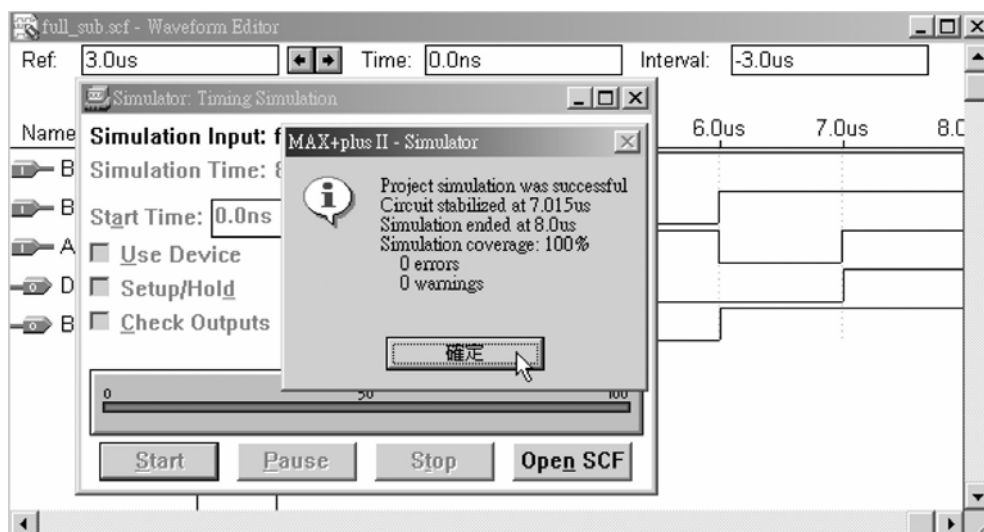
7. 執行模擬(MAX+PLUS II → Simulator, Start)如圖 3-3-40 所示，模擬沒有錯誤及警告，所得波形模擬結果符合全減器，代表我們製作的電路是正確可用的。



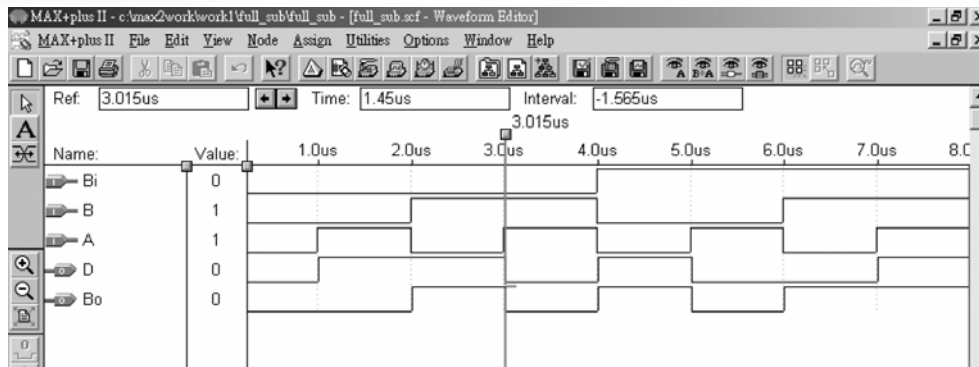
● 圖 3-3-38 執行模擬



● 圖 3-3-39 模擬起始視窗



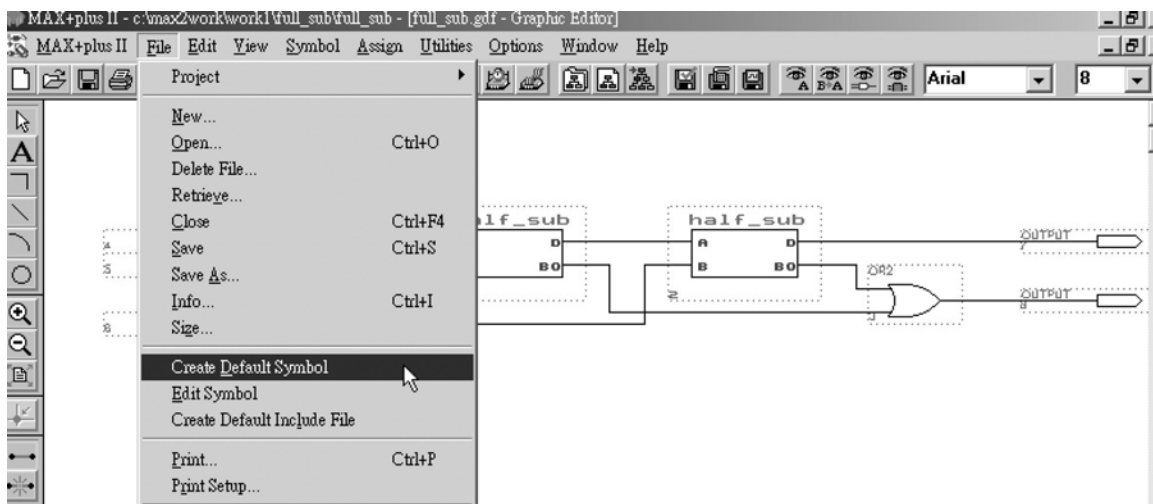
● 圖 3-3-40 模擬訊息視窗



● 圖 3-3-41 模擬結果

3-3-5 全減器(三)(以半減器元件組成全減器元件)

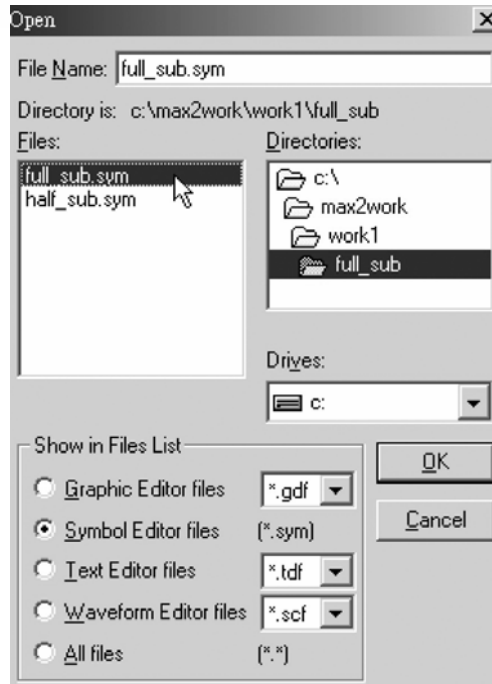
如上一單元完成全減器，也可將其做成一個全減器元件，點選 **File→Create Default Symbol**，即可產生新的零件 IC(全減器，名稱 full_sub.sym)。



● 圖 3-3-42 製作自製元件視窗

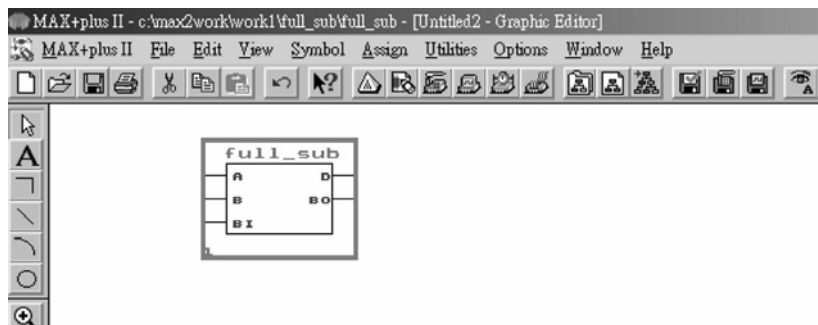
可將全減器叫出來看看是否有製作完成，步驟如下：

1. 開啟新的圖形編輯檔，此時即可取用所製作的元件 IC(full_sub)，點選 **Symbol→Enter Symbol**(或在編輯視窗按兩下 **Double Click**)，按確定後，即出現元件取用視窗。



● 圖 3-3-43 元件取用視窗

2. 取用全減器元件(symbol)：進入剛才存放全減器的資料夾，點選符號元件檔 (Symbol File) full_sub，按 **OK**，在圖形編輯視窗中，即可看到該元件(包裝後的全減器)。可在全減器元件上按兩下，即可開啟其內部電路視窗。
3. 讀者可自行模擬，以測試其功能是否正常。



● 圖 3-3-44 自製全減器元件

3-3-6 四位元減法器

如同四位元加法器可用四個全加器來製作，四位元的減法器，亦可用四個全減器元件來完成，在此我們將介紹上一單元做成的全減器元件完成四位元全減器的方法。

	B ₀₃	B ₀₂	B ₀₁	B ₀₀	
		A ₃	A ₂	A ₁	A ₀
—		B ₃	B ₂	B ₁	B ₀
	B ₀₃	D ₃	D ₂	D ₁	D ₀

四位元減法器運算如上所示，A₀~A₃ 表示被減數四個位元，B₀~B₃ 代表減數的四個位元，其相減產生的借位 B₀₀~B₀₃ 分別在其次一位元相減的運算中，產生的結果差為 D₀~D₃ 以及最後的借位 D₃。依照如此原則，要以全減器元件製作四位元減法器，第一個全減器的被減數及減數分別接 A₀ 及 B₀，其借位輸入端接地(因為第一位元並無上一級的借位，所以其值為 0)。第一個全減器的 D(差)輸出端直接是第一位元 D₀(差)的輸出值，借位輸出端 B₀₀ 則接到第二個減法器的借位輸入端。第二個全減器則接被減數 A₁ 及減數 B₁，產生差(D₁)輸出以及借位(B₀₁)接到第三個減法器，以此類推接完四個減法器即可完成四位元減法器，在此實際電路圖列為習題，請讀者自行練習，若有疑難，可參考四位元全加器之步驟。

3-3-7 八位元減法器

完成四個位元相減，對於八位元減法器，一樣可用同樣的方式來製作，在此我們將以上一單元做成的全減器元件來完成八位元全減器的製作。

	B ₀₇	B ₀₆	B ₀₅	B ₀₄	B ₀₃	B ₀₂	B ₀₁	B ₀₀
		A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁
—		B ₇	B ₆	B ₅	B ₄	B ₃	B ₂	B ₁
	B ₀₇	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁

八位元加減器運算如上所示，A₀~A₇ 表示被減數八個位元，B₀~B₇ 代表減數的八個位元，其相減產生的借位 B₀₀~B₀₇ 分別進入其次一位元的運算中，產生的結果差為 D₀~D₇ 以及最後的借位 B₀₇。依照如此原則，要以全減器元件製作八位元減法器，第一個全減器的被減數及減數分別接 A₀ 及 B₀，其借位輸入端接地(因為第一位元並無上一級的借位，所以其值為 0)。第一個全減器的 D(差)輸出端直接是第一位元 D₀(差)的輸出值，借位輸出端 B₀₀ 則接到第二個加減器的借位輸入端。第二個全減器則接被減數 A₁ 及減數 B₁，產生差(D₁)輸出以及借位(B₀₁)接到第三個減法器，以此類推接完八個減法器，與四位元減法器極為類似。同樣的在此將實際電路圖列為練習題，請讀者自行練習，若有疑難，可參考八位元全加器之步驟。

3-4 編碼器與解碼器

數位邏輯電路只有兩種狀態，一者是 1，一者是 0。若單純要用 1 與 0 來表示自然界的各種現象，則需用多個位元的二進制編碼來表示。將人類看得懂的文字，變成機械看得懂的二進碼需用編碼器；而若要將機械看得懂的二進碼轉換為人類看得懂的文字，則需用解碼器，此單元將介紹幾種編碼器與解碼器。

3-4-1 編碼器

將人類看得懂的文字編成機器看得懂的二進碼的電路，稱為編碼器。一般組合邏輯電路可用來做數字的編碼之用，在此以八對三編碼器為例說明。

真值表

輸入端								輸出端		
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	O ₂	O ₁	O ₀
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

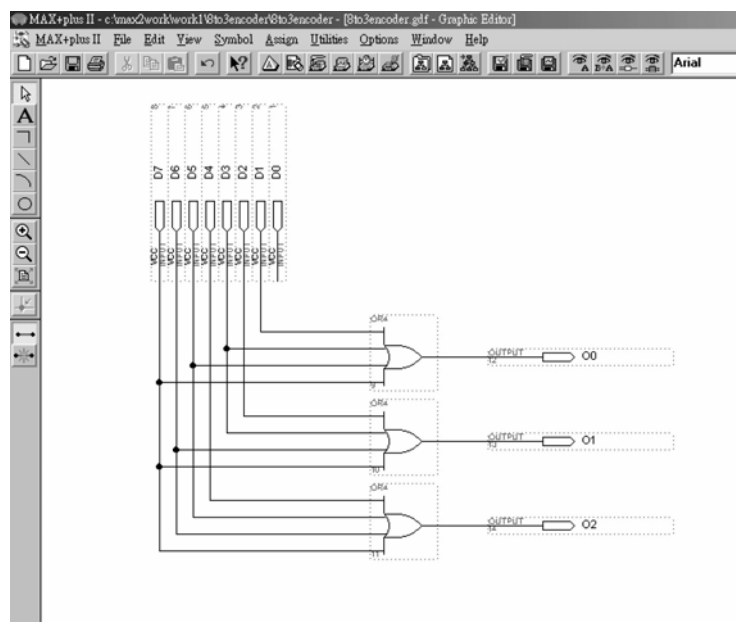
由真值表可知，輸入端部份有八個資料(D₀~D₇)，分別代表十進制的數字(0~7)，輸出端部份有三個位元(O₀~O₂)，代表三位元的二進碼，000~111 分別表示二進制的 0~7。

布林函數：

$$O_2 = D_4 + D_5 + D_6 + D_7$$

$$O_1 = D_2 + D_3 + D_6 + D_7$$

$$O_0 = D_1 + D_3 + D_5 + D_7$$



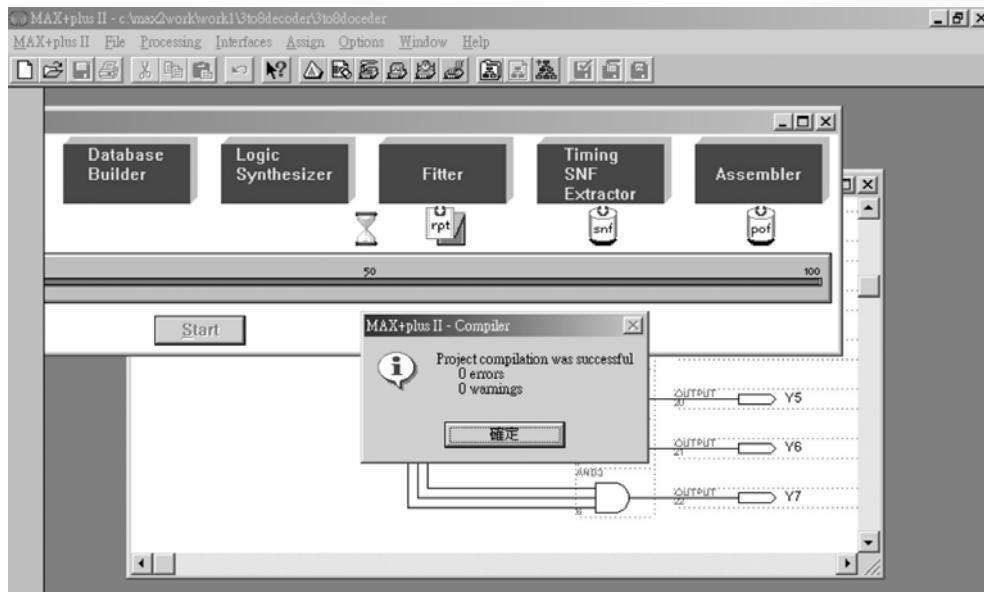
● 圖 3-4-1 八對三編碼器電路圖

如圖 3-4-1 所示，經由步驟：畫電路圖→存檔→設為工作專案→指定 CPLD 晶片→編譯→軟體模擬→規劃腳位→編譯→下載燒錄(若是非 ALTERA 的晶片下載前需轉檔)後，即完成電路設計製作，此時可配合外部電路(或實驗器模組，需配合腳位規劃)來驗證。

步驟如下：

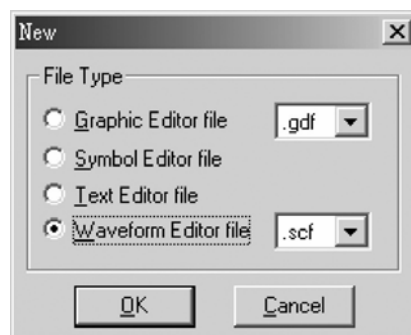
1. 開啟一個新圖形編輯檔，繪製一個八對三編碼器的電路圖。
2. 存檔，取檔名為 8 to 3 encoder.gdf。
3. 設為指定工作專案(File→Project→Set Project to Current File)，指定 CPLD 晶片(Assign→Device)，並編譯(MAX+PLUS II → Compiler)。

如圖所示，編譯時出現一個警告(warning)，指出 D₀ 這個輸入接腳並未使用，實際上是不需要的，但即使有此警告，還是不會影響電路執行結果，在此電路中因為要讓清楚表示八對三編碼器的輸入輸出接腳，所以仍保留 D₀ 接腳。

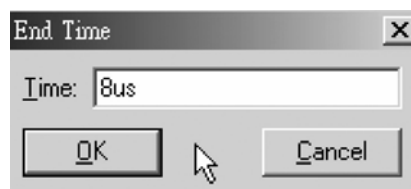


● 圖 3-4-2 執行編譯訊息警告

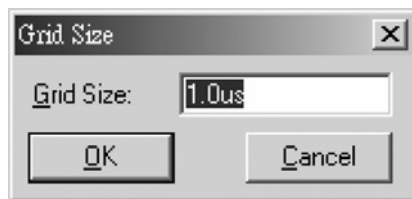
4. 開啟新的波形編輯檔案，設定功能模擬結束時間(File→End Time)，設定格線間距(Options → Grid Size)，顯示在視窗中適當大小格線(View→Fit in Window)。



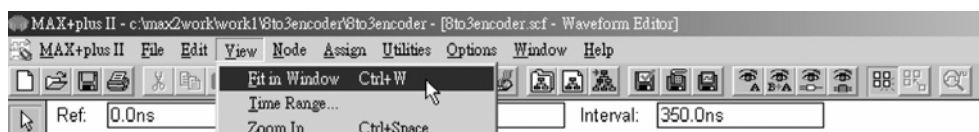
● 圖 3-4-3 開啟新檔視窗



● 圖 3-4-4 模擬結束時間設定視窗

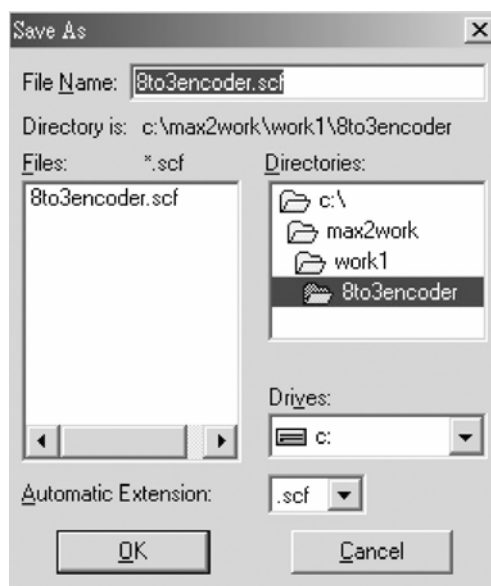


● 圖 3-4-5 模擬單位時間設定視窗

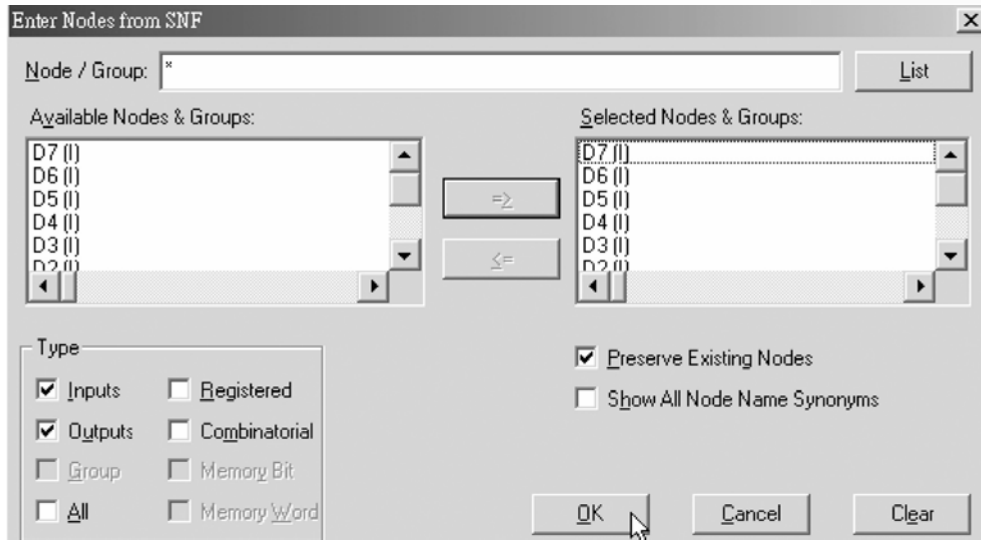


● 圖 3-4-6 視窗調整

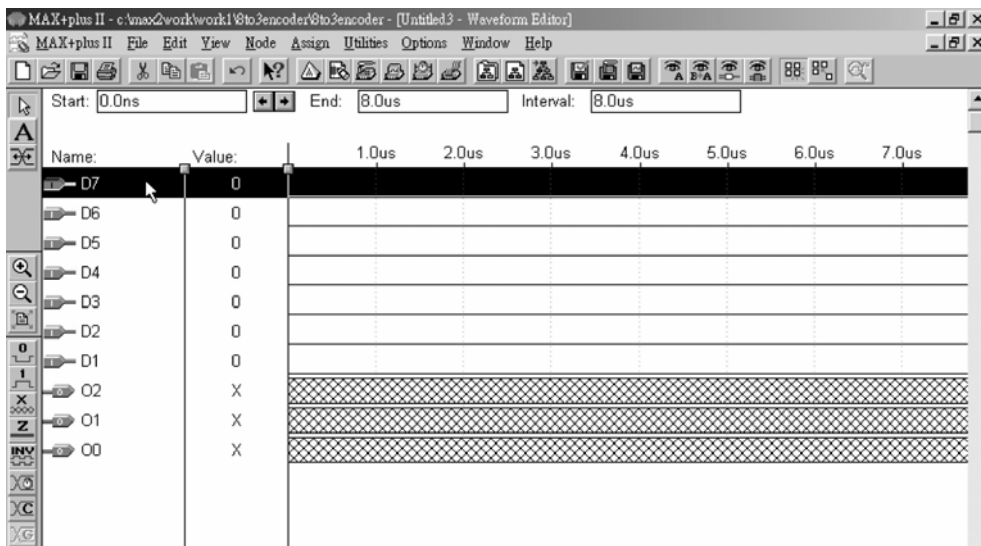
5. 儲存檔案(Save As)，檔名 8to3encoder.scf，輸入節點(Node → Enter Nodes from SNF，按 List 及 \Rightarrow ，OK)，利用 0 1 編輯輸入信號。






● 圖 3-4-7 儲存檔案視窗

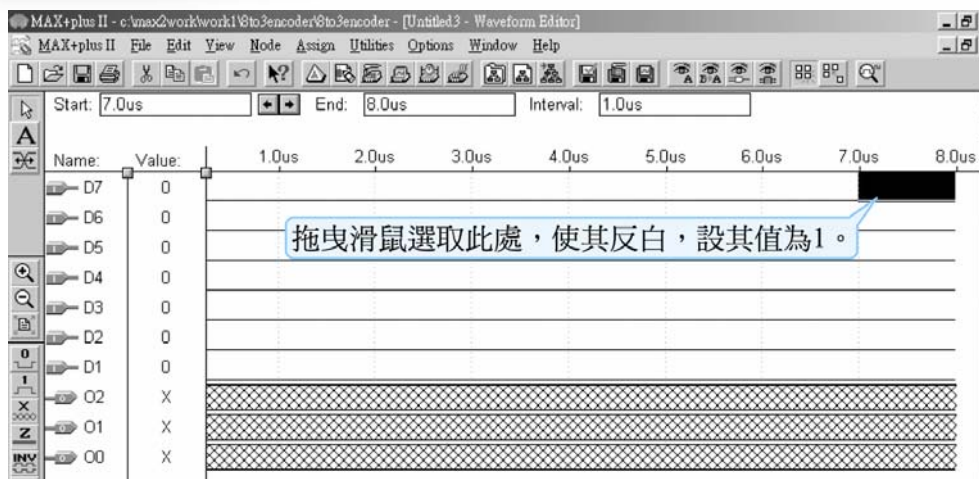


● 圖 3-4-8 輸出入節點選擇視窗

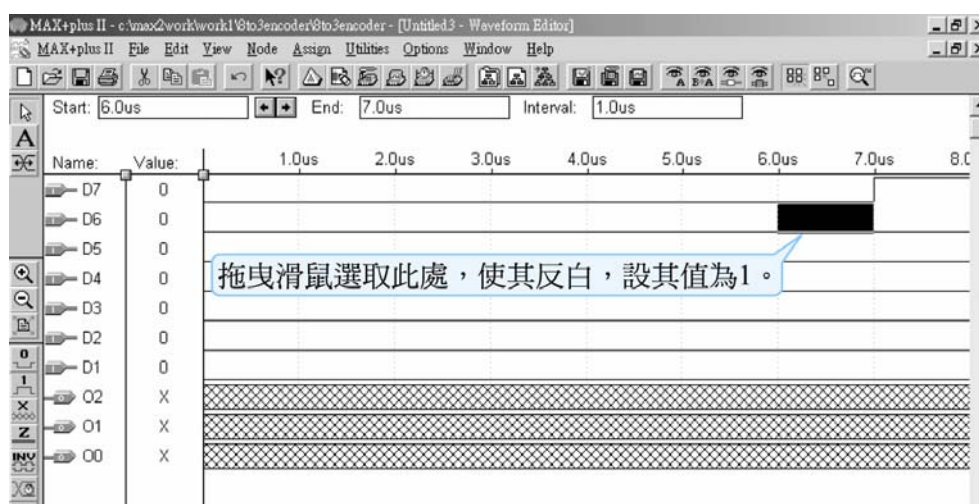


● 圖 3-4-9 模擬波形設定

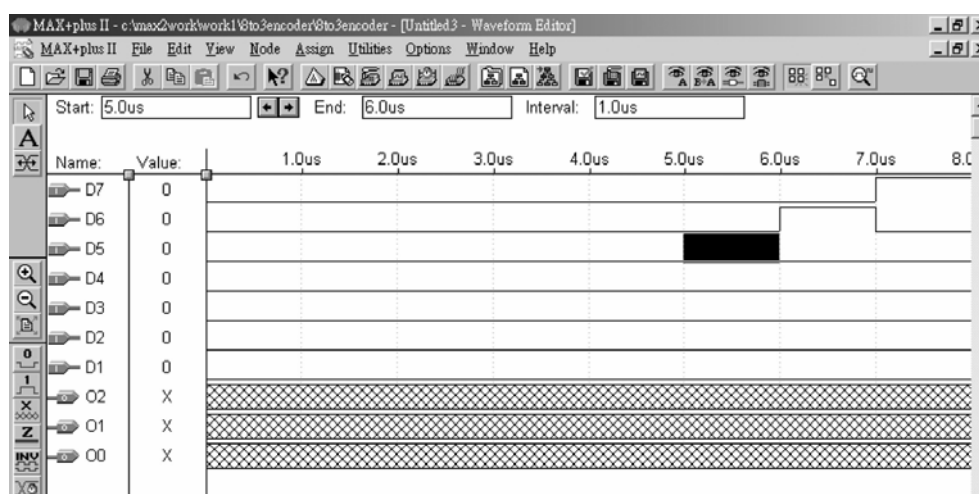
6. 如圖 3-4-10 所示，拖曳滑鼠去選取 D₇ 的 7.0 μ s~8.0 μ s 的區域，使其反白，點選  使該區數值為 1。拖曳滑鼠去選取 D₆ 的 6.0 μ s~7.0 μ s 的區域，使其反白，點選  使該區數值為 1。使用滑鼠去選取 D₅ 的 5.0 μ s~6.0 μ s 的區域，使其反白，點選  使該區數值為 1。重覆類似動作方式，輸入如圖 D₇~D₁ 之數值(D₀ 腳因為未接入電路，所以沒有 D₀ 輸入接腳，也不用輸入。



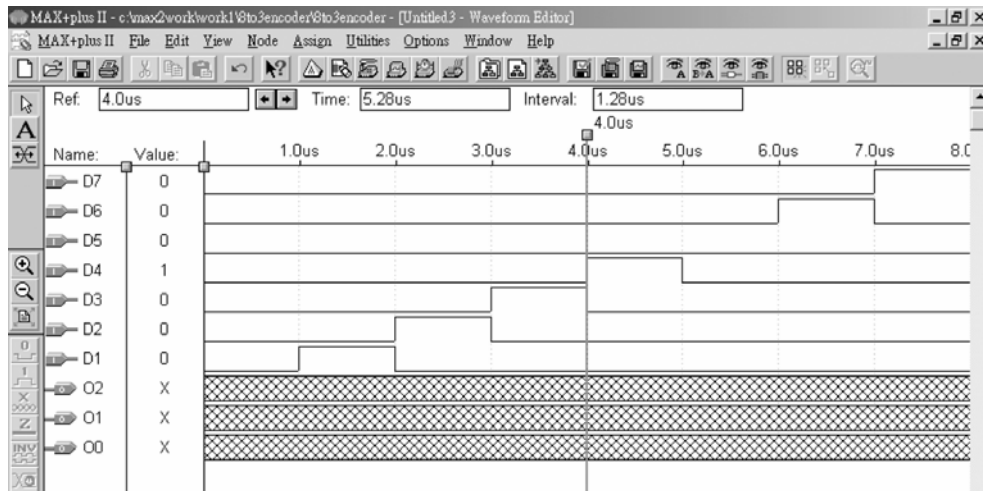
● 圖 3-4-10 模擬波形設定



● 圖 3-4-11 模擬波形設定

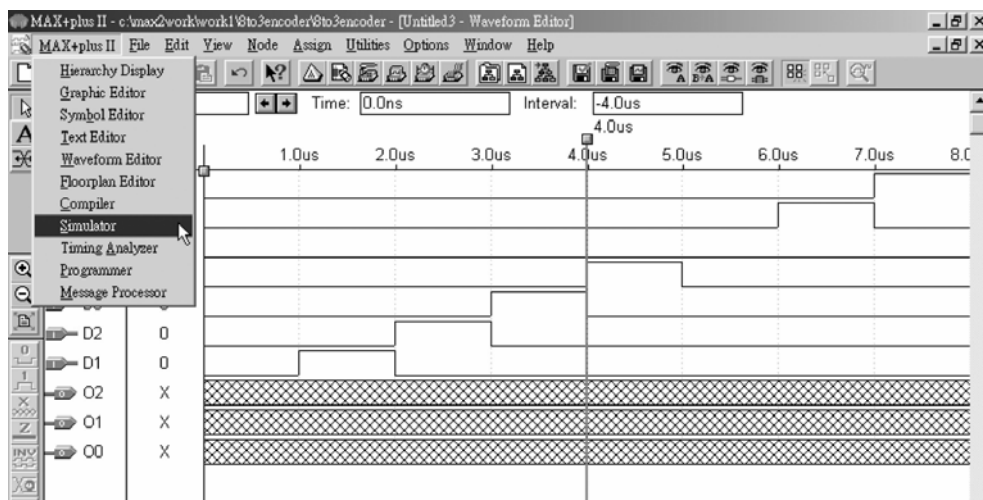


● 圖 3-4-12 模擬波形設定

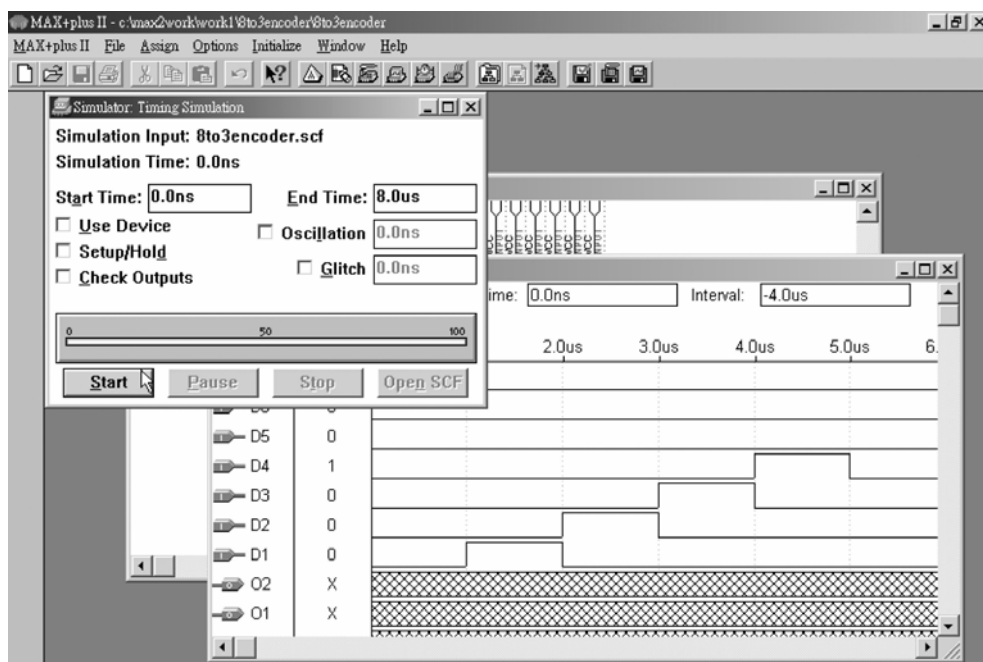


● 圖 3-4-13 模擬波形設定

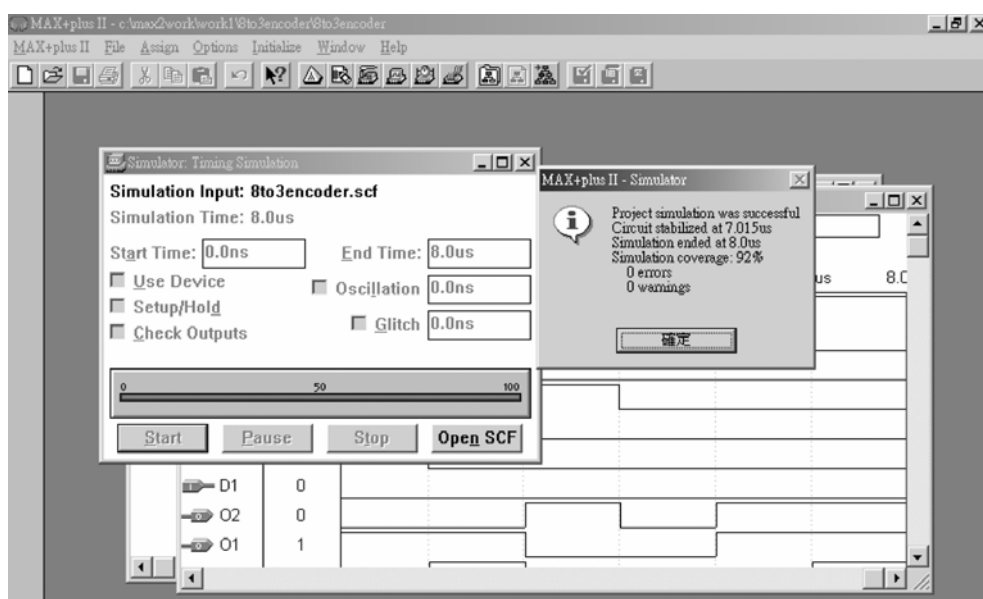
7. 執行模擬(MAX+PLUS II → Simulator, Start)如圖 3-4-16 所示，模擬沒有錯誤及警告，所得波形模擬結果符合八對三編碼器，代表我們製作的電路是正確可用的。



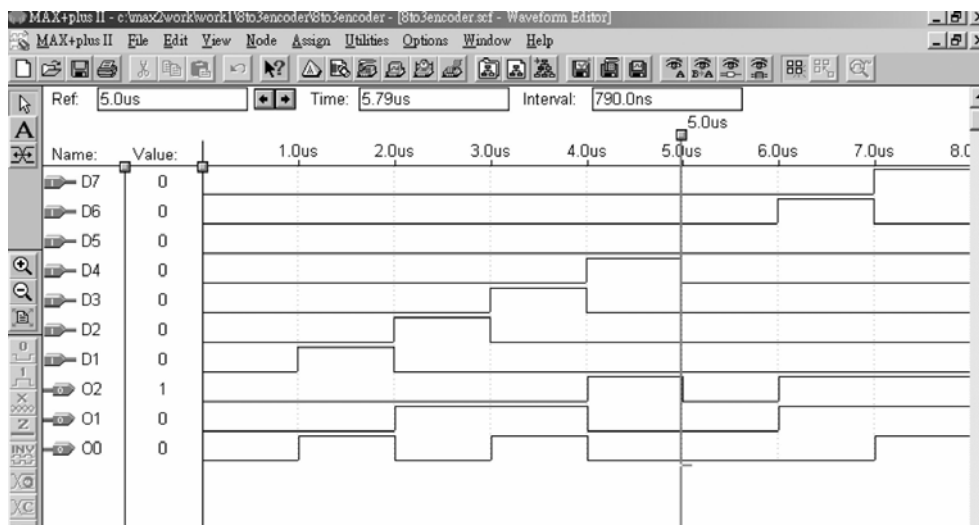
● 圖 3-4-14 執行模擬



● 圖 3-4-15 模擬起始視窗



● 圖 3-4-16 模擬訊息視窗



● 圖 3-4-17 模擬結果

模擬成功之後，可下載(燒錄)到實驗板做實際電路測試，以下為接到尼德公司實驗板的接腳表格，讀者可以按表中接腳設定以完成硬體電路測試。

● 表 3-4-1 電路圖輸出入腳位、CPLD 腳位及實驗器模組腳位對應表

腳位對應關係	輸入								輸出		
名稱	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	O ₂	O ₁	O ₀
CPLD 晶片腳位	PIN 29	PIN 28	PIN 27	PIN 25	PIN 24	PIN 22	PIN 21	PIN 20	PIN 73	PIN 74	PIN 75
實驗器模組對應腳位	DIP A8	DIP A7	DIP A6	DIP A5	DIP A4	DIP A3	DIP A2	DIP A1	LED DG2	LED DG1	LED DG0

實驗器輸入接腳有 D₇~D₀ 八支腳，可用指撥開關 DIPA₈~DIPA₁ 來表示，輸出端 O₂~O₀ 三支腳可用綠色發光二極體 DG₂~DG₀ 來表示。

3-4-2 解碼器三對八解碼器

有了編碼器將十進制編成二進碼，自然也需有將二進碼轉成十進制的解碼器。在此以三對八解碼器為例說明。

表 3-4-2 真值表

輸入端			輸出端							
C	B	A	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

由真值表可知，輸入端部份有三位元的二進碼(CBA)，分別代表二進位數字(0~7)，輸出端部份有八條線(Y₇~Y₀)，代表十進制的 0~7。

布林函數：

$$Y_0 = \overline{C} \overline{B} \overline{A}$$

$$Y_1 = \overline{C} \overline{B} A$$

$$Y_2 = \overline{C} B \overline{A}$$

$$Y_3 = \overline{C} B A$$

$$Y_4 = C \overline{B} \overline{A}$$

$$Y_5 = C \overline{B} A$$

$$Y_6 = C B \overline{A}$$

$$Y_7 = C B A$$

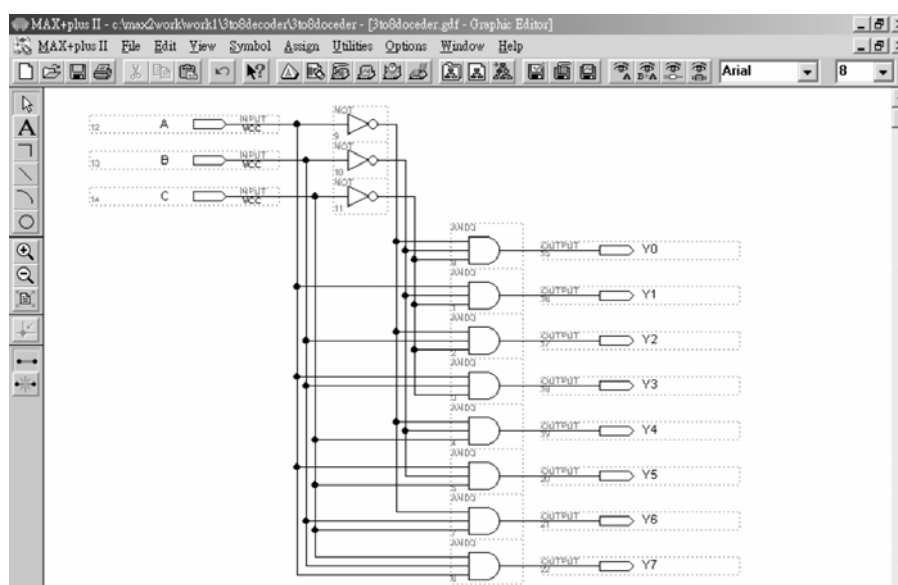


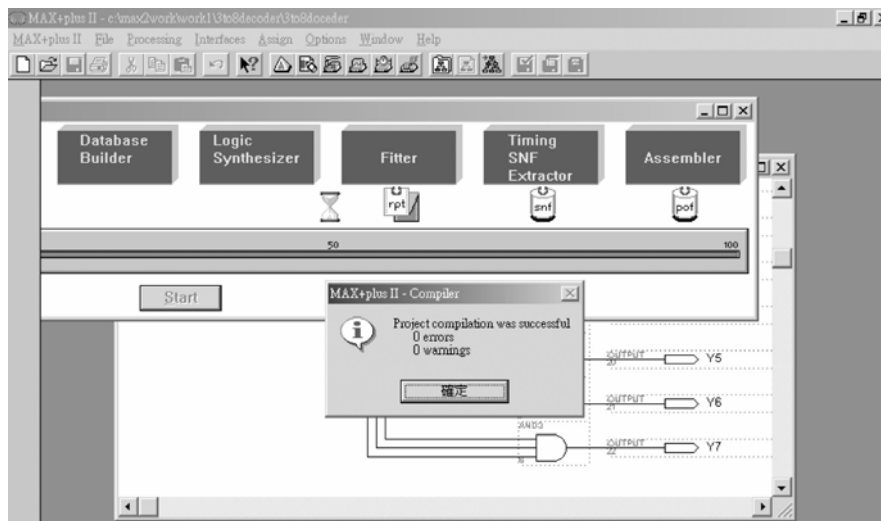
圖 3-4-18 三對八解碼器電路圖

如圖 3-4-18 所示，經由步驟：畫電路圖→存檔→設為工作專案→指定 CPLD 晶片→編譯→軟體模擬→規劃腳位→編譯→下載燒錄(若是非 ALTERA 的晶片下載前需轉檔)後，即完成電路設計製作，此時可配合外部電路(或實驗器模組，需配合腳位規劃)來驗證。

電路製作步驟如下：

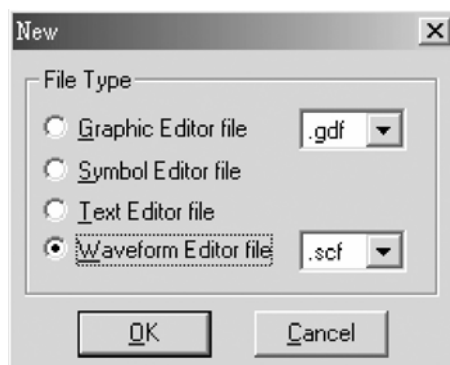
1. 開啟一個新圖形編輯檔，繪製一個三對八解碼器的電路圖。
2. 存檔，取檔名為 3 to 8 decoder.gdf
3. 設為指定工作專案(File→Project→Set Project to Current File)，指定 CPLD 晶片(Assign→Device)，並編譯(MAX+PLUS II → Compiler)。

如圖所示，編譯後沒有錯誤與警告，表示邏輯正確，可進行軟體模擬。

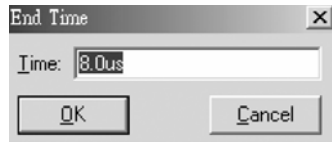


● 圖 3-4-19 編譯訊息視窗

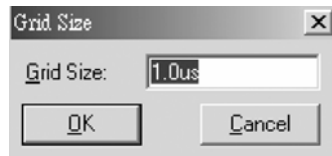
4. 開啟新的波形編輯檔案，設定功能模擬結束時間(File→End Time)，設定格線間距(Options→Grid Size)，顯示在視窗中適當大小格線(View→Fit in Window)。



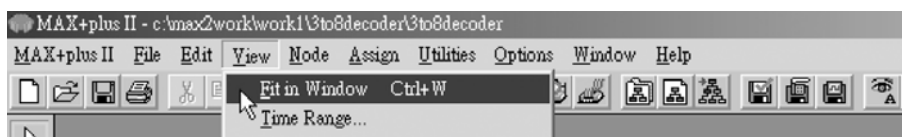
● 圖 3-4-20 開啟新檔視窗



● 圖 3-4-21 模擬結束時間設定視窗

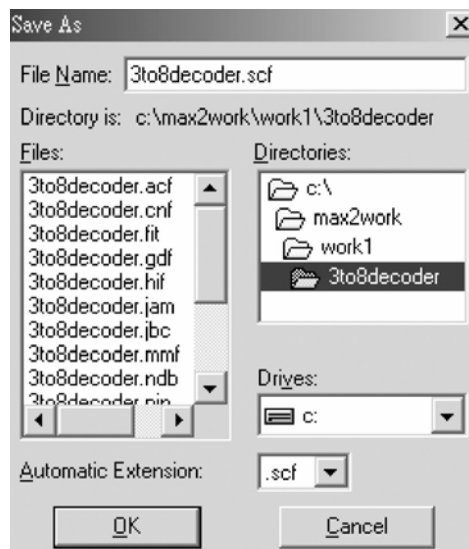


● 圖 3-4-22 模擬單位時間設定視窗

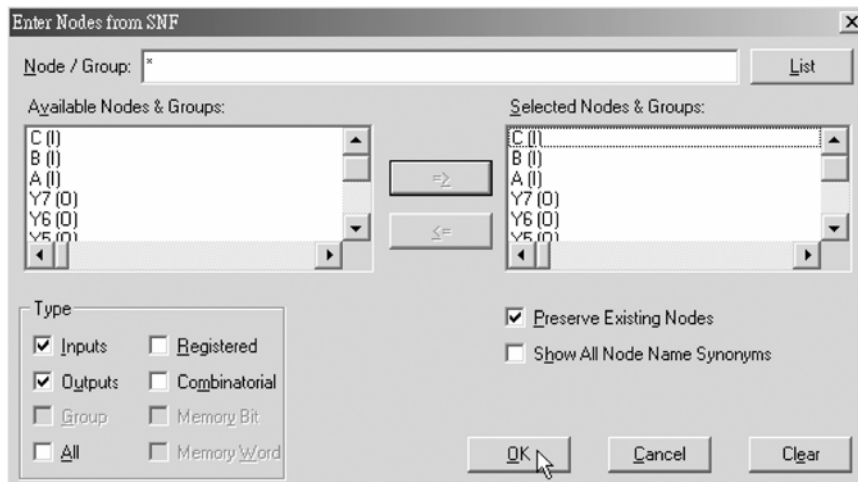


● 圖 3-4-23 視窗調整

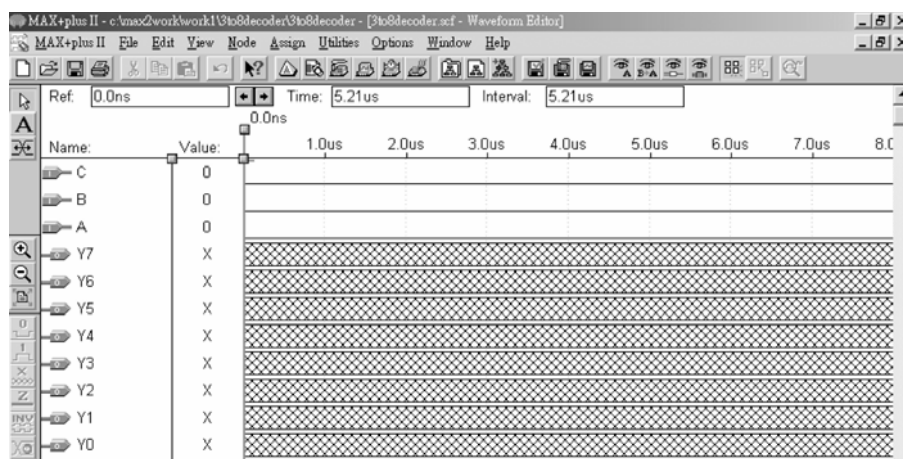
5. 儲存檔案(Save As)，檔名 3 to 8decoder.scf，輸入節點(Node → Enter Nodes from SNF，按 List 及 \Rightarrow ，OK)，利用 編輯輸入信號。






● 圖 3-4-24 儲存檔案視窗



● 圖 3-4-25 輸出入節點選擇視窗



● 圖 3-4-26 模擬波形設定

6. 如圖 3-4-27，拖曳滑鼠去選取 **C** 的 $4.0\mu\text{s} \sim 8.0\mu\text{s}$ 的區域，使其反白，點選  使該區數值為 1。拖曳滑鼠去選取 **B** 的 $2.0\mu\text{s} \sim 4.0\mu\text{s}$ 的區域，使其反白，點選  使該區數值為 1。使用滑鼠去選取 **B** 的 $6.0\mu\text{s} \sim 8.0\mu\text{s}$ 的區域，使其反白，點選  使該區數值為 1。重覆類似動作方式，輸入如圖 CBA 之數值。

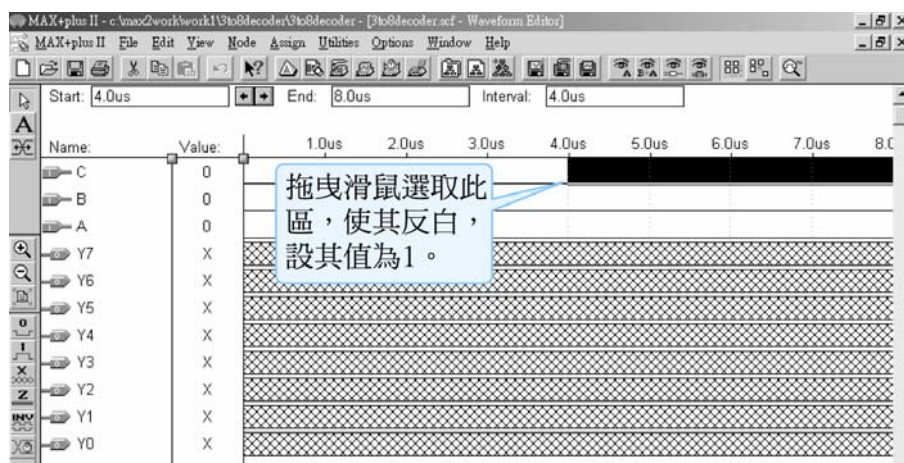


圖 3-4-27 模擬波形設定

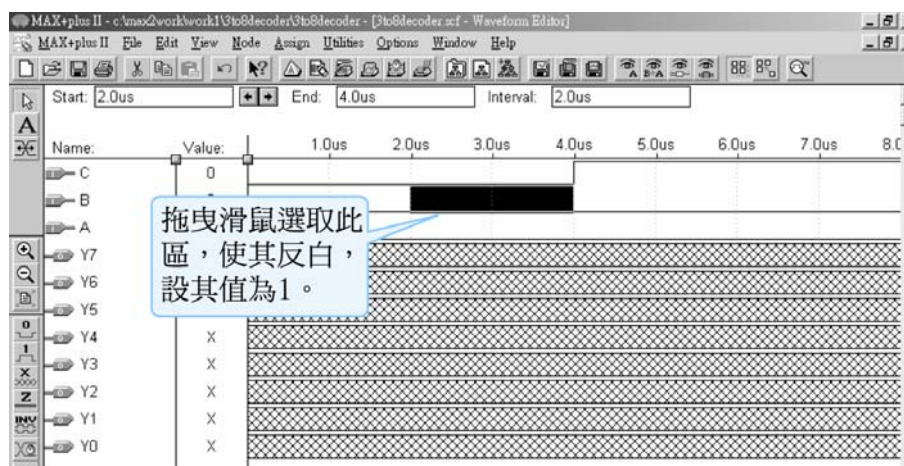


圖 3-4-28 模擬波形設定

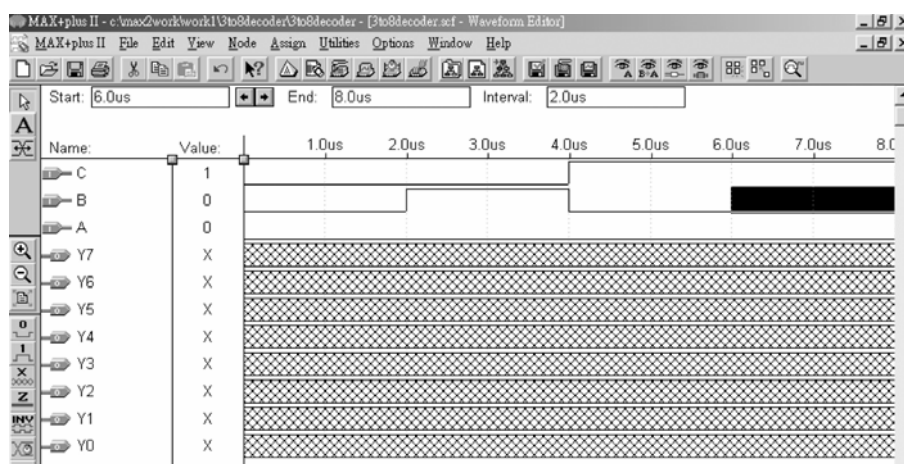
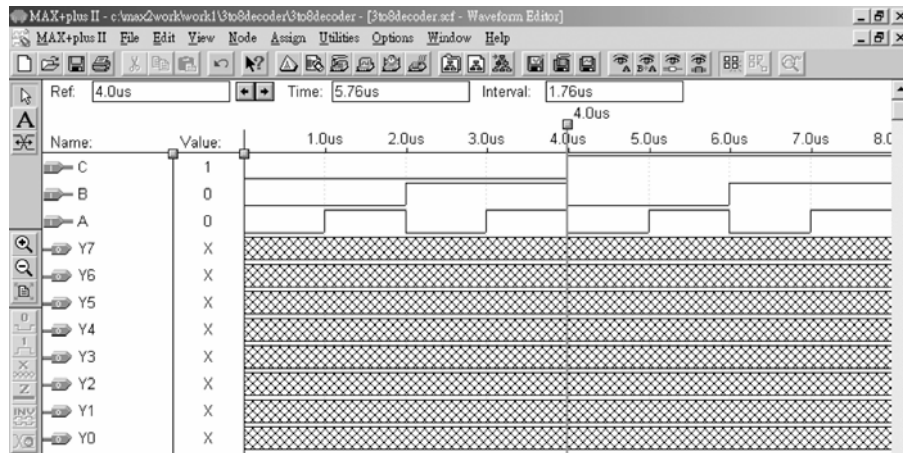
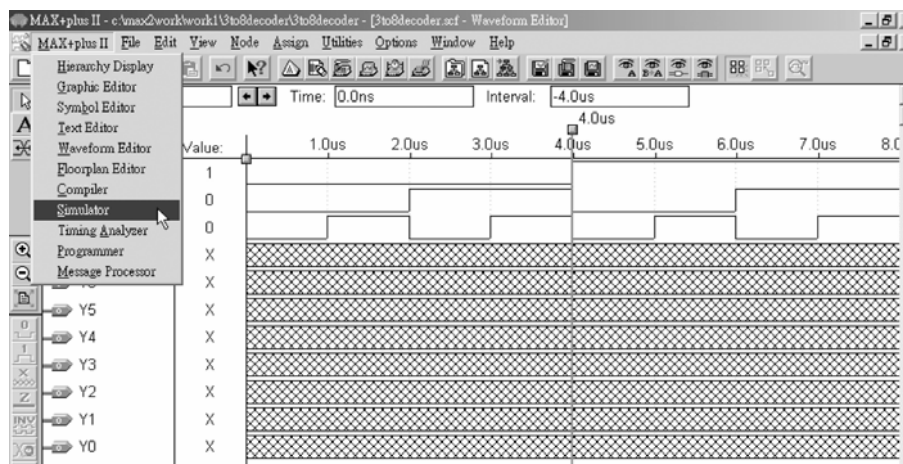


圖 3-4-29 模擬波形設定

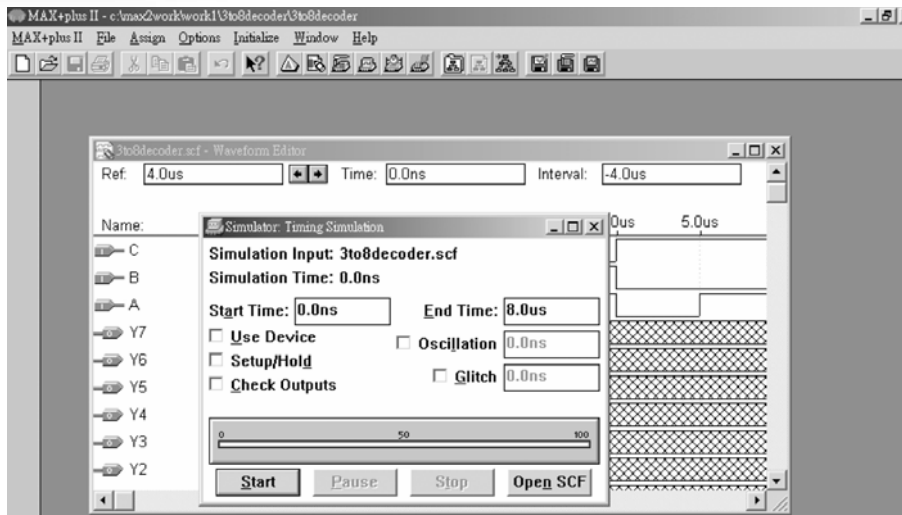


● 圖 3-4-30 模擬波形設定

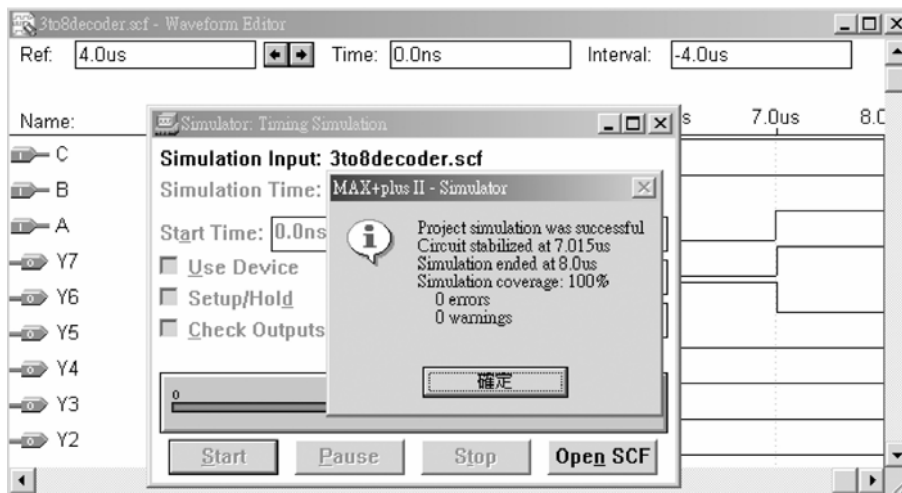
7. 執行模擬(MAX+PLUS II → Simulator, Start)如圖 3-4-33 所示，模擬沒有錯誤及警告，所得波形模擬結果符合三對八解碼器，代表我們製作的電路是正確可用的。



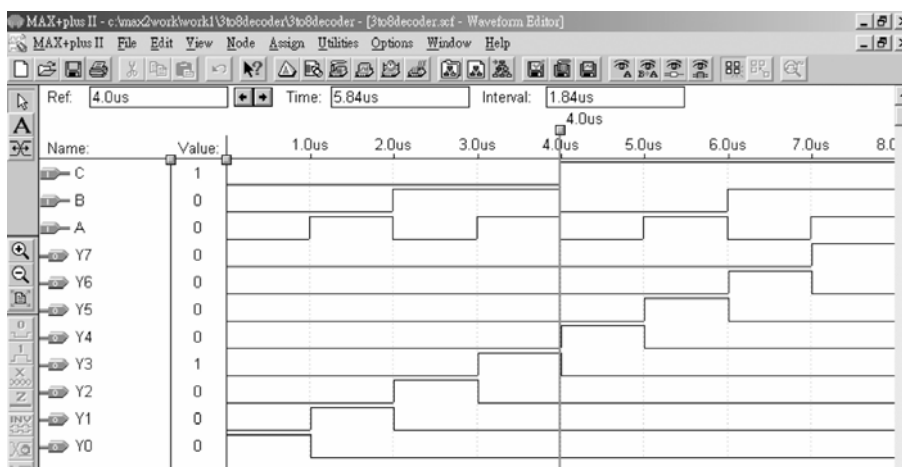
● 圖 3-4-31 執行模擬視窗



● 圖 3-4-32 模擬起始視窗



● 圖 3-4-33 模擬訊息視窗



● 圖 3-4-34 模擬結果

模擬成功之後，可下載(燒錄)到實驗板做實際電路測試，以下為接到尼德公司實驗板的接腳表格，讀者可以按表中接腳設定以完成硬體電路測試。

表 3-4-3 電路圖輸入腳位、CPLD 腳位及實驗器模組腳位對應表

腳位對應關係	輸入			輸出							
名稱	C	B	A	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
CPLD 晶片腳位	PIN 22	PIN 21	PIN 20	PIN 81	PIN 80	PIN 79	PIN 77	PIN 76	PIN 75	PIN 74	PIN 73
實驗器模組對應腳位	DIP A3	DIP A2	DIP A1	LED DG7	LED DG6	LED DG5	LED DG4	LED DG3	LED DG2	LED DG1	LED DG0

實驗器輸入接腳有 C，B，A 三支腳，可用指撥開關 DIPA₃~DIPA₁ 來表示，輸出端 Y₇~Y₀ 八支腳可用綠色發光二極體 DG₇~DG₀ 來表示。

3-4-3 共陽極七段顯示解碼器

要將二進碼解碼，除了一般解碼器外，尚可設計成由七段顯示器顯示數值的七段顯示解碼器，七段顯示器(Seven-segment indicator)的內部結構如圖 3-4-35 所示，分為共陽極(Common anode)與共陰極(Common cathode)，七段顯示器的七個段內有發光二極體(LED)，旁邊小數點的部份也有發光二極體，若是將這些發光二極體的陽極全部接在一起，稱為共陽極七段顯示器。當其接正電壓時，七段顯示器的接腳(a,b,c,d,e,f,g,dot)若接低電位，則該接腳的發光二極體將會亮起來。而共陰極的七段顯示器則是發光二極體的陰極全部接在一起後接地，七段顯示器的接腳(a,b,c,d,e,f,g,dot)若接高電位，則該接腳的發光二極體將會亮起來。因此可用組合邏輯設計來讓七段顯示器有不同顯示。

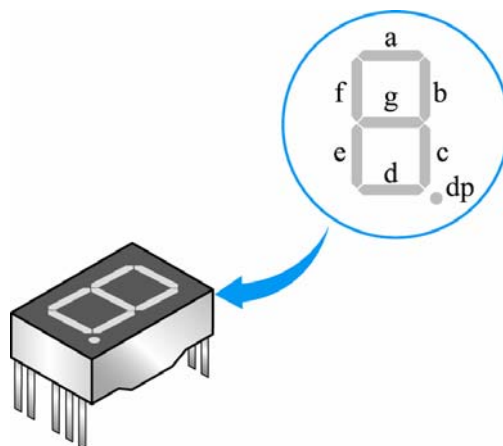


圖 3-4-35(a) 七段顯示器結構圖

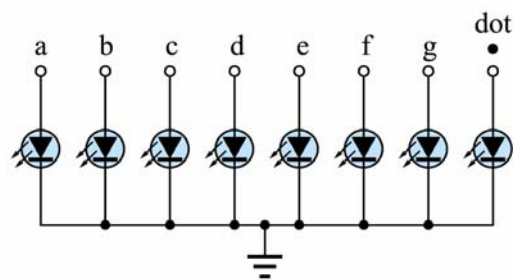
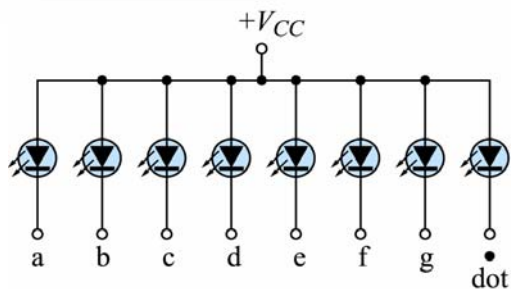
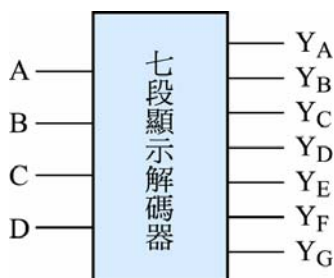


圖 3-4-35 (b) 共陽極七段顯示器結構圖

圖 3-4-35 (c) 共陰極七段顯示器結構圖

● 圖 3-4-35 七段顯示器結構圖

若要設計使七段顯示器顯示 0~F 十六種變化(0~15)的解碼器，輸入端需有四支腳，ABCD 可代表二進碼 0000~1111(0~15)，輸出七支接腳(a~g)，在此以 CPLD 設計，命名 $Y_A, Y_B, Y_C, Y_D, Y_E, Y_F, Y_G$ 分別接到七段顯示器的 a,b,c,d,e,f,g，真值表如下所示。



● 圖 3-4-36 七段顯示解碼器示意圖

共陽極七段顯示解碼器真值表

顯示 字型	輸入端				輸出端						
	D	C	B	A	Y_G	Y_F	Y_E	Y_D	Y_C	Y_B	Y_A
0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	0	1	1	1	1	1	0	0	1
2	0	0	1	0	0	1	0	0	1	0	0
3	0	0	1	1	0	1	1	0	0	0	0
4	0	1	0	0	0	0	1	1	0	0	1
5	0	1	0	1	0	0	1	0	0	1	0
6	0	1	1	0	0	0	0	0	0	1	1
7	0	1	1	1	1	1	1	1	0	0	0
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	1	1	0	0	0
A	1	0	1	0	0	0	0	1	0	0	0
b	1	0	1	1	0	0	0	0	0	1	1
C	1	1	0	0	1	0	0	0	1	1	0
d	1	1	0	1	0	1	0	0	0	0	1
E	1	1	1	0	0	0	0	0	1	1	0
F	1	1	1	1	0	1	1	1	0	0	0

布林函數：

$$Y_G = \overline{D} \overline{C} \overline{B} \overline{A} + \overline{D} \overline{C} \overline{B} A + \overline{D} C B A + D C \overline{B} \overline{A}$$

$$Y_F = \overline{D} \overline{C} \overline{B} A + \overline{D} \overline{C} B \overline{A} + \overline{D} \overline{C} B A + \overline{D} C B A + D C \overline{B} A + D C B A$$

$$Y_E = \overline{D} \overline{C} \overline{B} A + \overline{D} \overline{C} B A + \overline{D} C \overline{B} \overline{A} + \overline{D} C \overline{B} A + \overline{D} C B A + D \overline{C} \overline{B} A + D C B A$$

$$Y_D = \overline{D} \overline{C} \overline{B} A + \overline{D} C \overline{B} \overline{A} + \overline{D} C B A + D \overline{C} \overline{B} A + D \overline{C} B \overline{A} + D C B A$$

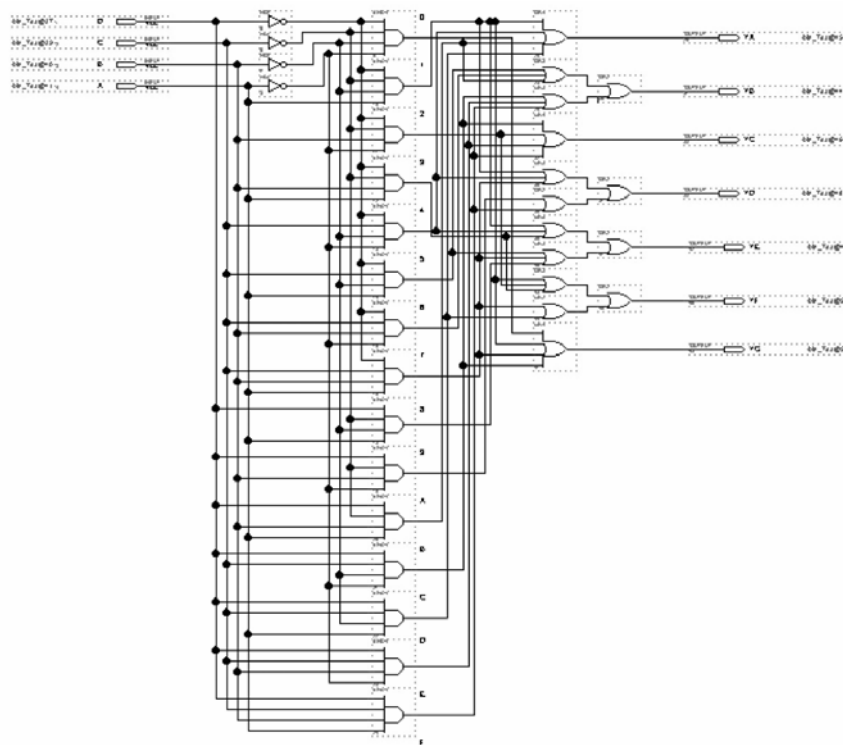
$$Y_C = \overline{D} \overline{C} B \overline{A} + D C \overline{B} \overline{A} + D C B \overline{A}$$

$$Y_B = \overline{D} C \overline{B} A + \overline{D} C B \overline{A} + D \overline{C} B A + D C \overline{B} \overline{A} + D C B \overline{A}$$

$$Y_A = \overline{D} \overline{C} \overline{B} A + \overline{D} C \overline{B} \overline{A} + \overline{D} C B \overline{A} + \overline{D} C \overline{B} A + D C \overline{B} A$$

設計方法如下：

1. 開啟新圖形編輯檔，依照真值表繪製邏輯電路圖，如圖 3-4-37 所示。(因為電路圖很大，若書本看不清楚，讀者可於本書附贈光碟中的 bin7_sa 資料夾中找到此檔參考)



● 圖 3-4-37 共陽極七段顯示解碼器電路圖

2. 如圖 3-4-37 所示，經由步驟：畫電路圖→存檔→設為工作專案→指定 CPLD 晶片→編譯→軟體模擬→規劃腳位→編譯→下載燒錄(若是 Atmel 的晶片下載前需轉檔)後，即完成電路設計製作，此時可配合外部電路(或實驗器模組，需配合腳位規劃)來驗證。

執行步驟如下：

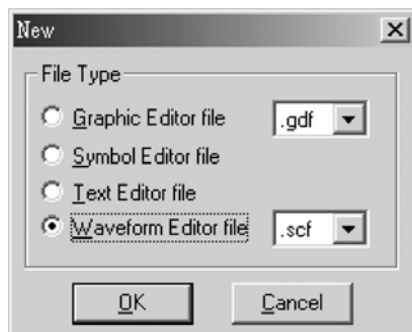
1. 開啟一個新圖形編輯檔，繪製一個共陽極七段顯示解碼器的電路圖。
2. 存檔，取檔名為 bin_7sa.gdf
3. 設為指定工作專案(**File**→**Project**→**Set Project to Current File**)，指定 CPLD 晶片(**Assign**→**Device**)，並編譯(**MAX+PLUS II** → **Compiler**)。

如圖 3-4-38 所示，編譯後沒有錯誤與警告，表示邏輯正確，可進行軟體模擬。

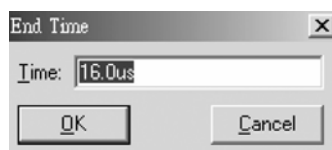


● 圖 3-4-38 編譯訊息視窗

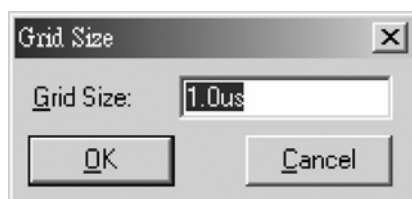
4. 開啟新的波形編輯檔案，設定功能模擬結束時間(**File**→**End Time**)設定 $16\ \mu\text{s}$ ，設定格線間距(**Options**→**Grid Size**)設定 $1\ \mu\text{s}$ ，顯示在視窗中適當大小格線(**View**→**Fit in Window**)。



● 圖 3-4-39 開啟新檔視窗

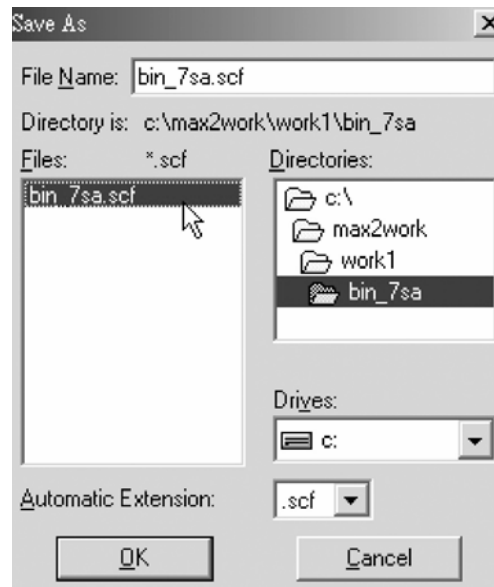


● 圖 3-4-40 模擬結束時間設定視窗

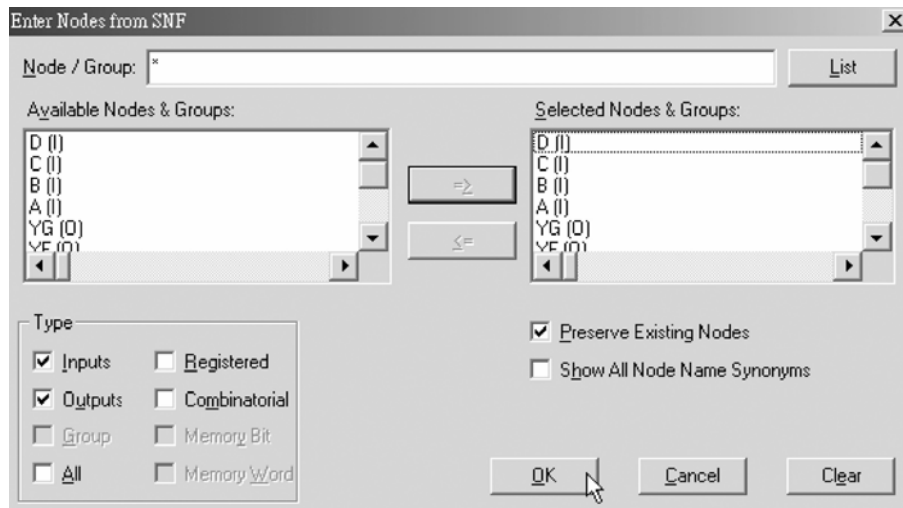


● 圖 3-4-41 模擬單位時間設定視窗

5. 儲存檔案(Save As)，檔名 bin_7sa.scf，輸入節點(Node → Enter Nodes from SNF，按 List 及 \Rightarrow ，OK)，編輯輸入信號，輸入端有四支腳，總共有十六種組合，剛好 $16\mu s$ 可模擬完，可將輸入端四支腳群組化，用計數時脈編輯 0000~1111 即可完成輸入信號編輯。



● 圖 3-4-42 儲存檔案視窗



● 圖 3-4-43 輸出入節點選擇視窗

6. 執行模擬(MAX+PLUS II → Simulator, Start)如圖 3-4-44 所示, 所得波形模擬結果符合共陽極七段顯示解碼器, 代表我們製作的電路是正確可用的。

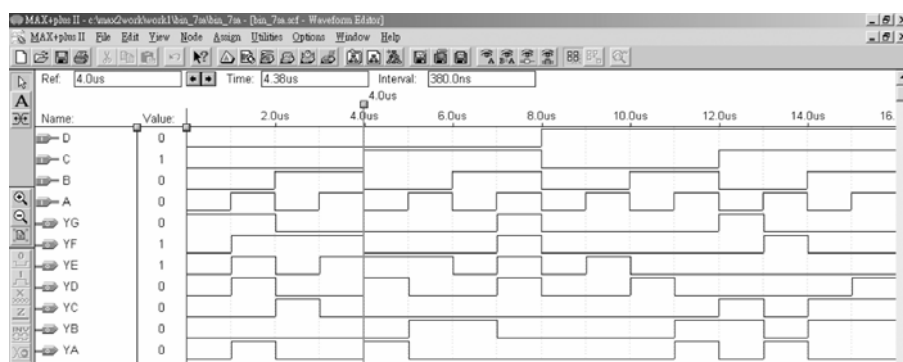


圖 3-4-44 模擬結果

模擬成功之後, 可下載(燒錄)到實驗板做實際電路測試, 以下為接到尼德公司實驗板的接腳表格, 讀者可以按表中接腳設定以完成硬體電路測試。

表 3-4-4 電路圖輸出入腳位、CPLD 腳位及實驗器模組腳位對應表

腳位對應關係	輸入				輸出						
名稱	D	C	B	A	YA	YB	YC	YD	YE	YF	YG
CPLD 晶片腳位	PIN 24	PIN 22	PIN 21	PIN 20	PIN 30	PIN 31	PIN 33	PIN 34	PIN 35	PIN 36	PIN 37
實驗器模組對應腳位	DIP A ₄	DIP A ₃	DIP A ₂	DIP A ₁	A ₀	B ₀	C ₀	D ₀	E ₀	F ₀	G ₀

實驗器使用第 A 個指撥開關的 DIPA₁~DIPA₄ 接腳當輸入端, 第 0 個七段顯示器 A₀~G₀ 接腳當輸出端。

3-4-4 共陰極七段顯示器解碼器

共陰極七段顯示器與共陽極相反, 當其輸入接腳(a,b,c,d,e,f,g,dot)接高態時, 則相對應的發光二極體會亮。以下為共陰極七段顯示解碼器真值表, 其輸出端與共陽極七段顯示解碼器相反, 所以電路製作時, 可在共陽極電路的各輸出端直接接上反閘, 即變成共陰極七段顯示解碼電路。電路圖如圖 3-4-45 所示, 因電路圖很大, 讀者若在書上看不清楚, 可參考本書所附光碟 BIN_7SK 資料夾內), 讀者可以自行操作及模擬。

表 3-4-5 共陰極七段顯示解碼器真值表

顯示 字型	輸入端				輸出端						
	D	C	B	A	Y _G	Y _F	Y _E	Y _D	Y _C	Y _B	Y _A
0	0	0	0	0	0	1	1	1	1	1	1
1	0	0	0	1	0	0	0	0	1	1	0
2	0	0	1	0	1	0	1	1	0	1	1
3	0	0	1	1	1	0	0	1	1	1	1
4	0	1	0	0	1	1	0	0	1	1	0
5	0	1	0	1	1	1	0	1	1	0	1
6	0	1	1	0	1	1	1	1	1	0	0
7	0	1	1	1	0	0	0	0	1	1	1
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	0	0	1	1	1
A	1	0	1	0	1	1	1	0	1	1	1
b	1	0	1	1	1	1	1	1	1	0	0
C	1	1	0	0	0	1	1	1	0	0	1
d	1	1	0	1	1	0	1	1	1	1	0
E	1	1	1	0	1	1	1	1	0	0	1
F	1	1	1	1	1	0	0	0	1	1	1

布林函數：

$$Y_G = \overline{D}\overline{C}\overline{B}\overline{A} + \overline{D}\overline{C}B\overline{A} + \overline{D}C\overline{B}\overline{A} + \overline{D}C\overline{B}A + \overline{D}CBA + D\overline{C}\overline{B}\overline{A} \\ + D\overline{C}\overline{B}A + D\overline{C}B\overline{A} + D\overline{C}BA + DC\overline{B}\overline{A} + DC\overline{B}A + DCB\overline{A} + DCBA$$

$$Y_F = \overline{D}\overline{C}\overline{B}\overline{A} + \overline{D}C\overline{B}\overline{A} + \overline{D}C\overline{B}A + \overline{D}CBA + D\overline{C}\overline{B}\overline{A} + D\overline{C}\overline{B}A \\ + D\overline{C}B\overline{A} + D\overline{C}BA + DC\overline{B}\overline{A} + DC\overline{B}A + DCB\overline{A} + DCBA$$

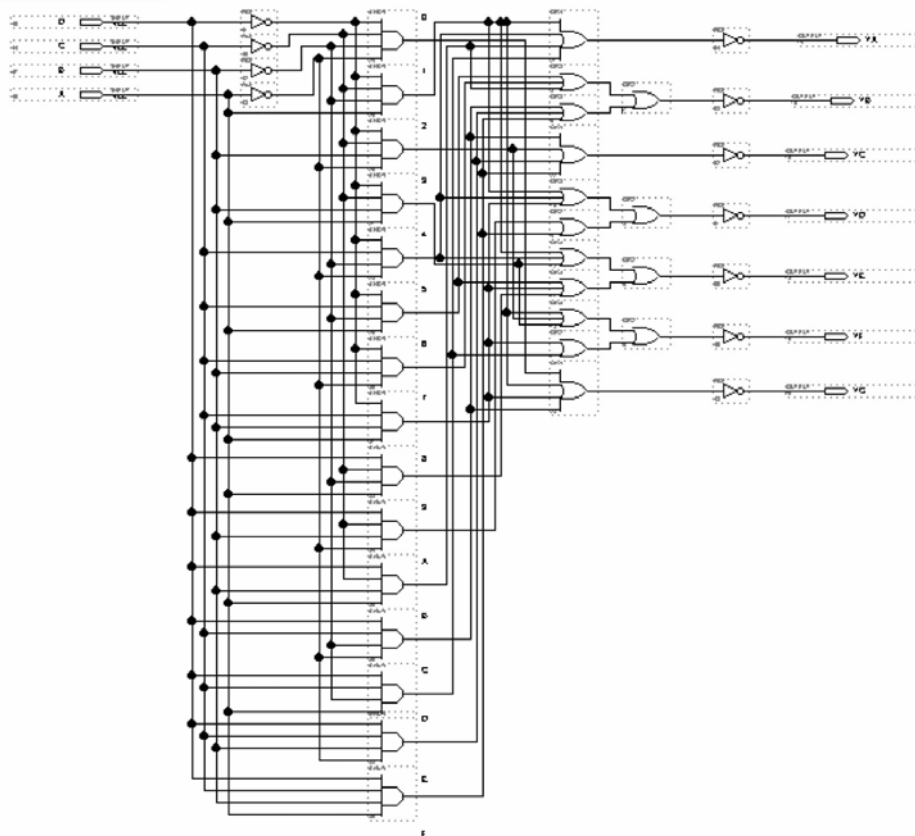
$$Y_E = \overline{D}\overline{C}\overline{B}\overline{A} + \overline{D}\overline{C}B\overline{A} + \overline{D}C\overline{B}\overline{A} + \overline{D}C\overline{B}A + \overline{D}CBA + D\overline{C}\overline{B}\overline{A} \\ + D\overline{C}\overline{B}A + DC\overline{B}\overline{A} + DC\overline{B}A + DCB\overline{A} + DCBA$$

$$Y_D = \overline{D}\overline{C}\overline{B}\overline{A} + \overline{D}\overline{C}B\overline{A} + \overline{D}C\overline{B}\overline{A} + \overline{D}C\overline{B}A + \overline{D}CBA + D\overline{C}\overline{B}\overline{A} \\ + D\overline{C}\overline{B}A + DC\overline{B}\overline{A} + DC\overline{B}A + DCB\overline{A} + DCBA$$

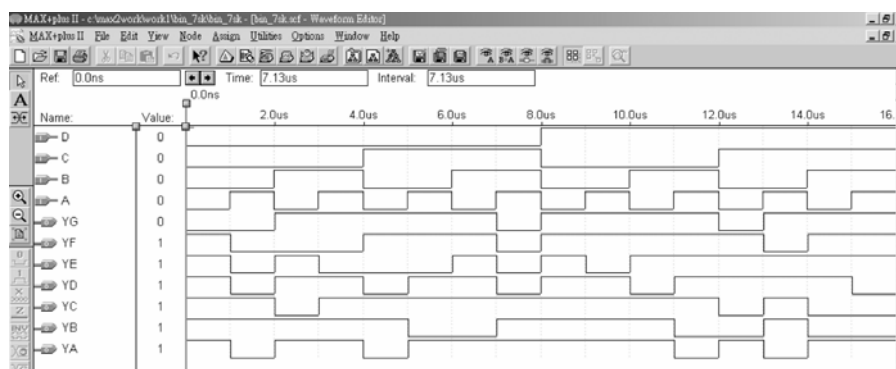
$$Y_C = \overline{D}\overline{C}\overline{B}\overline{A} + \overline{D}\overline{C}B\overline{A} + \overline{D}C\overline{B}\overline{A} + \overline{D}C\overline{B}A + \overline{D}CBA + D\overline{C}\overline{B}\overline{A} \\ + D\overline{C}\overline{B}A + DC\overline{B}\overline{A} + DC\overline{B}A + DCB\overline{A} + DCBA$$

$$Y_B = \overline{D}\overline{C}\overline{B}\overline{A} + \overline{D}\overline{C}B\overline{A} + \overline{D}C\overline{B}\overline{A} + \overline{D}C\overline{B}A + \overline{D}CBA + D\overline{C}\overline{B}\overline{A} \\ + D\overline{C}\overline{B}A + DC\overline{B}\overline{A} + DC\overline{B}A + DCB\overline{A} + DCBA$$

$$Y_A = \overline{D}\overline{C}\overline{B}\overline{A} + \overline{D}\overline{C}B\overline{A} + \overline{D}C\overline{B}\overline{A} + \overline{D}C\overline{B}A + \overline{D}CBA + D\overline{C}\overline{B}\overline{A} \\ + D\overline{C}\overline{B}A + DC\overline{B}\overline{A} + DC\overline{B}A + DCB\overline{A} + DCBA$$



● 圖 3-4-45 共陰極七段顯示解碼器電路圖



● 圖 3-4-46 模擬結果

模擬成功之後，可下載(燒錄)到實驗板做實際電路測試，以下為接到尼德公司實驗板的接腳表格，讀者可以按表中接腳設定以完成硬體電路測試。

表 3-4-6 電路圖輸入腳位、CPLD 腳位及實驗器模組腳位對應表

腳位對應關係	輸入				輸出						
名 稱	D	C	B	A	YA	YB	YC	YD	YE	YF	YG
CPLD 晶片腳位	PIN 24	PIN 22	PIN 21	PIN 20	PIN 30	PIN 31	PIN 33	PIN 34	PIN 35	PIN 36	PIN 37
實驗器模組對應腳位	DIP A4	DIP A3	DIP A2	DIP A1	A0	B0	C0	D0	E0	F0	G0

實驗器使用第 A 個指撥開關的 DIPA₁~DIPA₄ 接腳當輸入端，第 0 個七段顯示器 A₀~G₀ 接腳當輸出端。

3-5 多工器與解多工器

資料傳輸時，若是將由許多位置的資料點傳輸至許多位置的資料點時，則連接的線路將錯縱複雜。如果傳輸距離遠時，線路連接及維護更是一大問題。這種多端點資料傳送情形，一般都用多工器(Multiplexer)與解多工器(Demultiplexer)來解決。在資料發送端安裝多線變一線的多工器。多工器上有選擇線路，用來決定該送出哪一線路上的資料。多工器用於發送資料端資料選擇之用，又稱為資料選擇器(Data selector)。當資料接收端有多點位置可接收資料時，則需安裝一線變多線的解多工器，解多工器上亦有選擇線路，用來決定該由哪一條線路接收資料。解多工器用於接收資料端分配資料之用，又稱資料分配器(Data distributor)。如圖 3-5-1 所示，甲地資料欲傳送到乙地，甲地資料傳輸點數量有 2^n 個，則需有 n 條選擇線，用以決定送出哪一個端點之資料，乙地接收點數量有 2^m 個，需有 m 條選擇線用以決定將資料分配給哪一點。



圖 3-5-1 多工器與解多工器示意圖