

# CS255

## 數位系統實驗(二)

Introduction to Digital Systems Lab.(II)

上數、下數計數器設計  
&  
非同步清除、同步10模計數器  
、同步60模計數器

## 4. 非同步清除之2位元同步上數計數器

- 計數器利用循序邏輯來計算計時脈波的個數，用正反器的組合可以設計成計數器電路。
- 以下將非同步清除之2位元同步上數計數器，並分別以圖形編輯、Verilog HDL編輯與VHDL編輯3種方式設計。
- 功能要求：
  - 經過所設計之連接方式，以二進位自然數順序，由“00”計數至“11”。
  - 非同步清除：上數計數器的清除與時脈正緣時間是不同步的。

- 腳位：脈波輸入端：CLK
  - 清除控制端：CLR<sub>N</sub>
  - 輸出端：Q1、Q0

上週期輸出	控制線		輸出
(Q1 , Q0)	CLK	CLR <sub>N</sub>	(Q1 , Q0)
X	X	0	00
00	↑	1	01
01	↑	1	10
10	↑	1	11
11	↑	1	00

» 表9 非同步清除2位元同步上數計數器真值表



- 4-1 電路圖編輯

- 表10 迴授電路設計

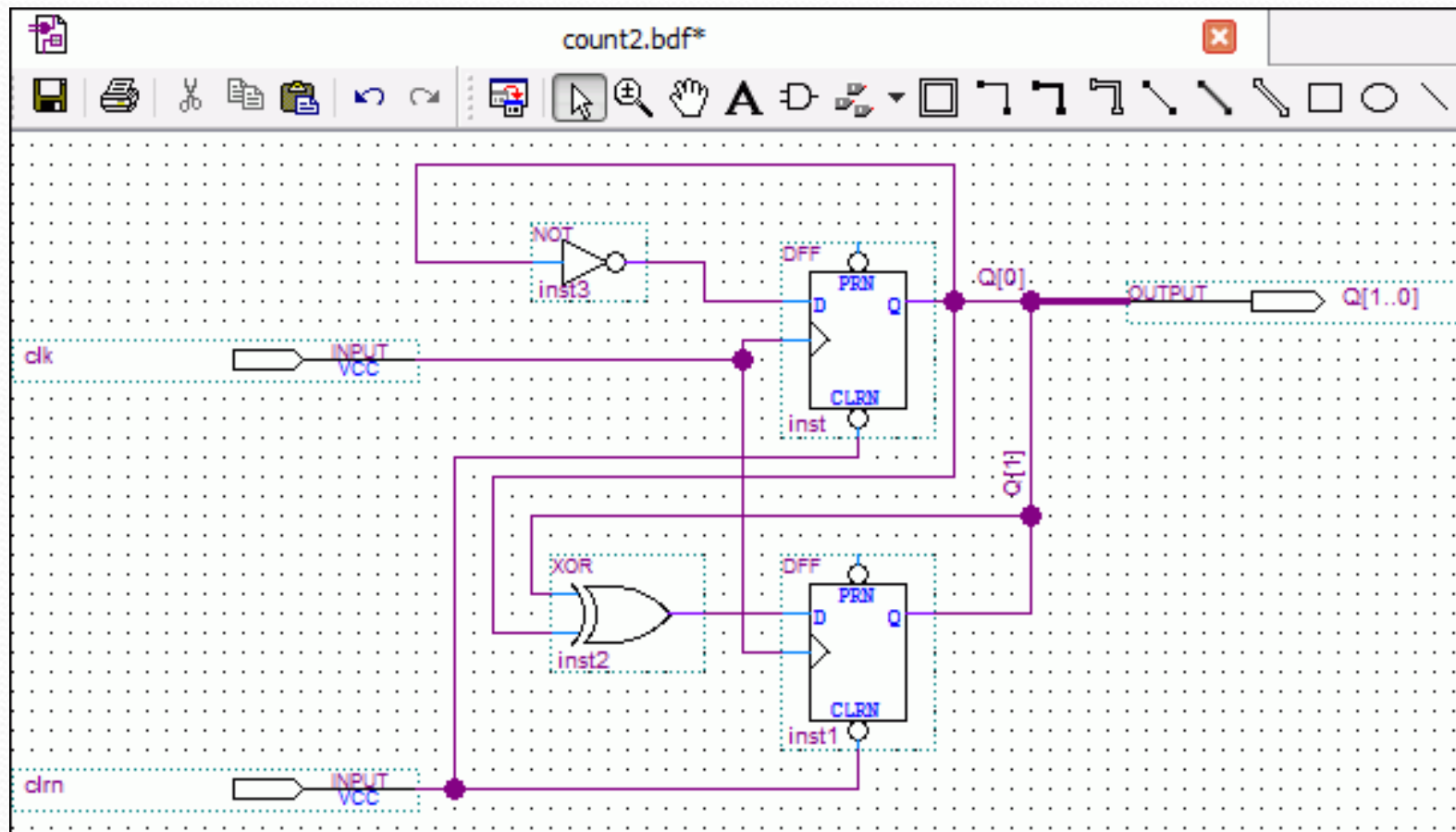
Q1	Q0	D1	D0
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

» 表11 D1卡諾圖

	Q0	/Q0
Q1	0	1
/Q1	1	0

» 表12 D0卡諾圖

	Q0	/Q0
Q1	0	1
/Q1	0	1



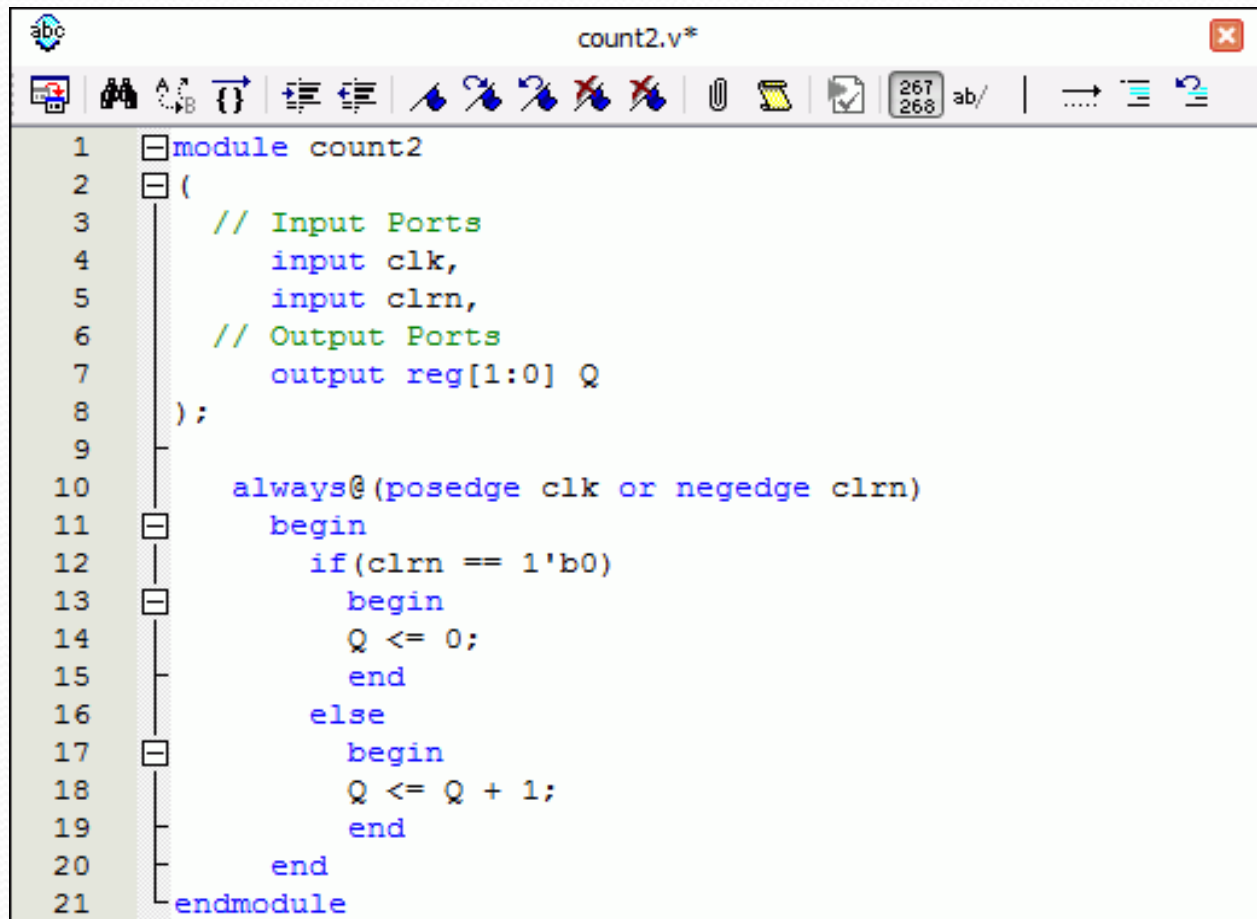
» 圖57 非同步清除2位元同步上數計數器



## ● 4-2 Verilog HDL編輯

- 編輯流程如下：

開啟新增專案精靈：建立專案count2 → 新增檔案 →  
另存新檔：儲存為count2.v → 編輯檔案內容 → 存檔  
並組譯 → 模擬驗證

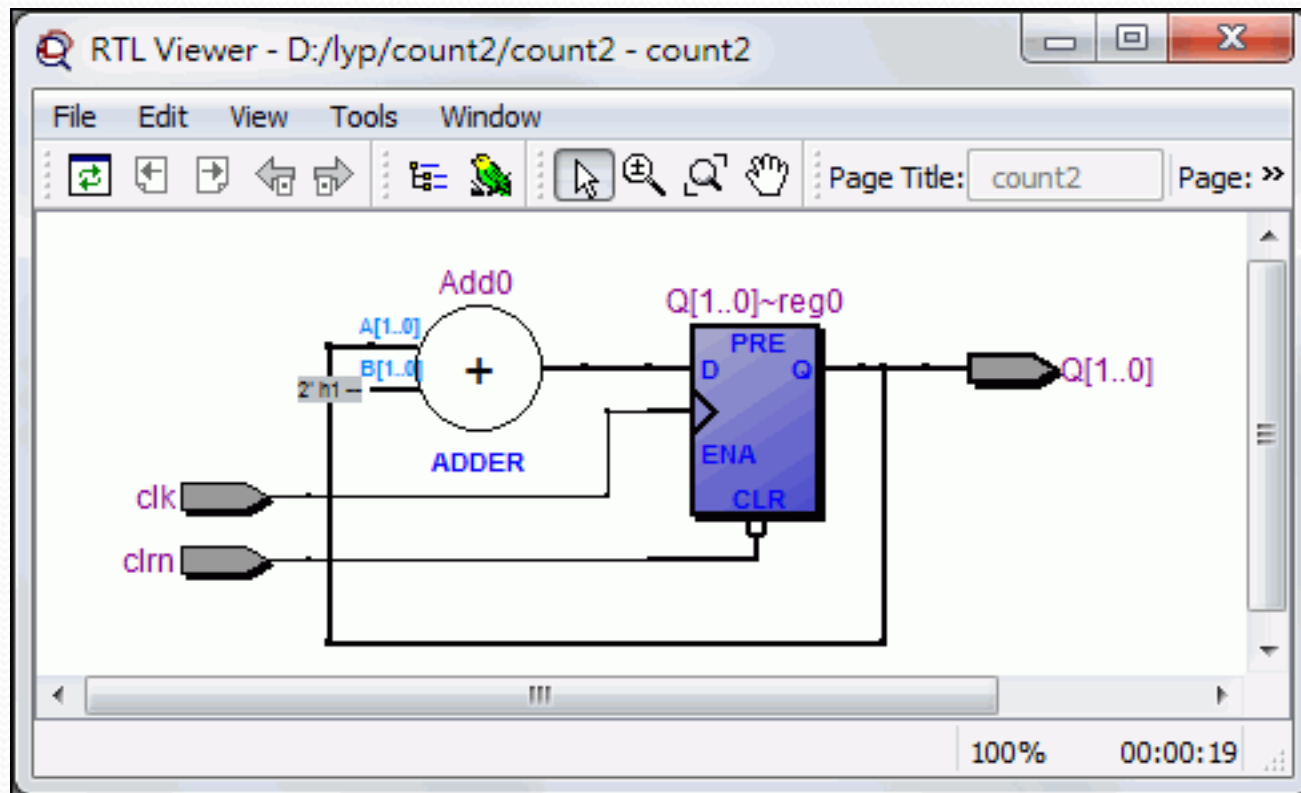


The screenshot shows a software window titled "count2.v\*" with a toolbar at the top. The code is as follows:

```
1  module count2
2  (
3      // Input Ports
4      input clk,
5      input clrn,
6      // Output Ports
7      output reg[1:0] Q
8  );
9
10     always@(posedge clk or negedge clrn)
11     begin
12         if(clrn == 1'b0)
13         begin
14             Q <= 0;
15         end
16     else
17     begin
18         Q <= Q + 1;
19     end
20     end
21 endmodule
```

» 圖58 非同步清除2位元同步上數計數器





» 圖59 RTL Viewer

abc count2.v

```
1 module count2 (clk, clrn, Q);  
2 input clk, clrn;  
3 output [1:0] Q;  
4 dff d0 (.D(~Q[0]), .CLRn(clrn), .CLK(clk), .Q(Q[0]));  
5 dff d1 (.D(^Q), .CLRn(clrn), .CLK(clk), .Q(Q[1]));  
6 endmodule
```

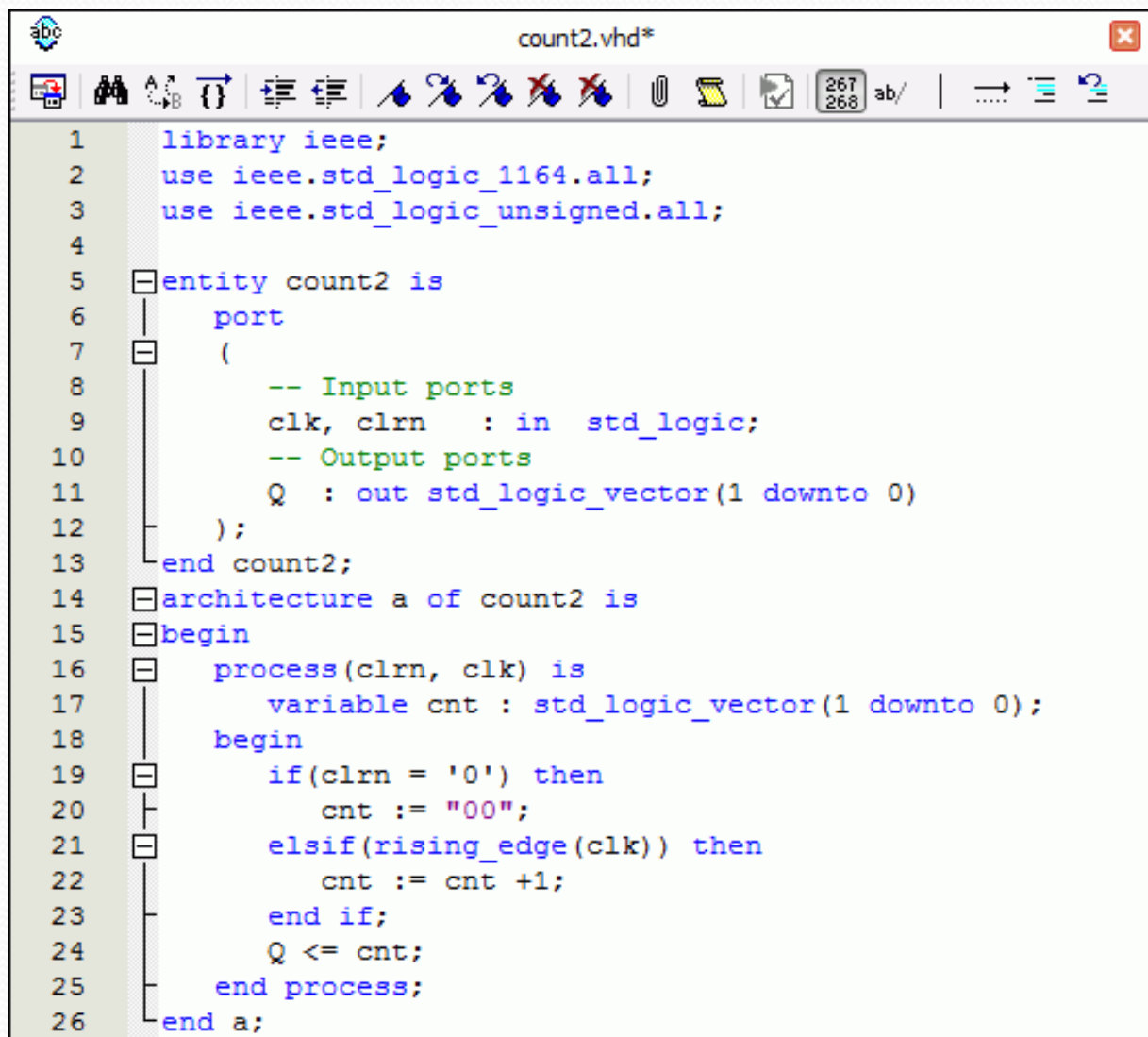
» 圖60 非同步清除2位元同步上數計數器

## ● 4-3 VHDL編輯

- 編輯流程如下：

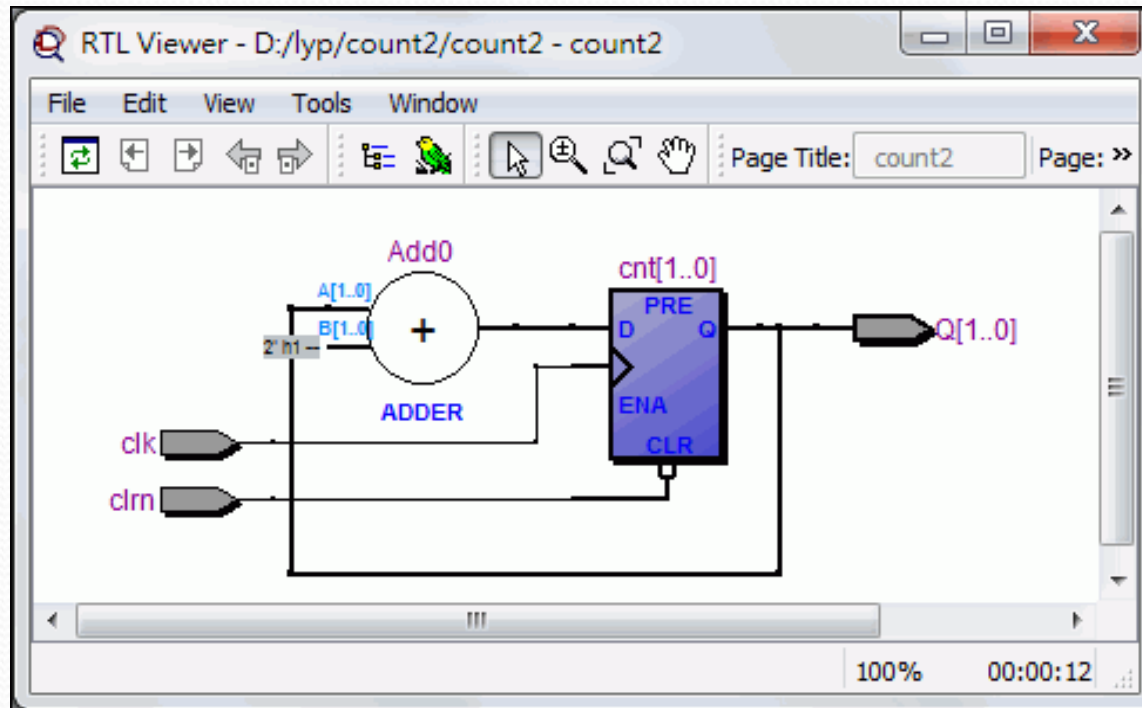
建立count2專案 → 新增檔案 → 另存新檔：儲存為  
count2.vhd → 編輯檔案內容 → 存檔並組譯 → 模擬  
驗證





```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_unsigned.all;
4
5  entity count2 is
6  port
7  (
8      -- Input ports
9      clk, clrn  : in  std_logic;
10     -- Output ports
11     Q  : out std_logic_vector(1 downto 0)
12 );
13 end count2;
14 architecture a of count2 is
15 begin
16     process(clrn, clk) is
17         variable cnt : std_logic_vector(1 downto 0);
18     begin
19         if(clrn = '0') then
20             cnt := "00";
21         elsif(rising_edge(clk)) then
22             cnt := cnt +1;
23         end if;
24         Q <= cnt;
25     end process;
26 end a;
```

» 圖61 非同步清除2位元同步上數計數器



» 圖62 RTL Viewer

abc count2.vhd

```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.ALL;
3  LIBRARY altera;
4  USE altera.maxplus2.ALL;
5  ENTITY count2 IS
6      PORT(  clk, clrn : IN  STD_LOGIC;
7            Q          : OUT STD_LOGIC_VECTOR(1 DOWNTO 0)
8            );
9  END count2;
10 ARCHITECTURE a OF count2 IS
11  SIGNAL temp0, temp1 : STD_LOGIC;
12  SIGNAL cnt : STD_LOGIC_VECTOR(1 DOWNTO 0);
13  BEGIN
14  temp0 <= NOT cnt(0);
15  temp1 <= cnt(0) XOR cnt(1);
16  Q <= cnt;
17  d0:dff
18      PORT MAP (d => temp0, clrn => clrn, prn => '1',
19               clk => clk, q => cnt(0) );
20  d1:dff
21      PORT MAP (d => temp1, clrn => clrn, prn => '1',
22               clk => clk, q => cnt(1) );
23  END a;
```

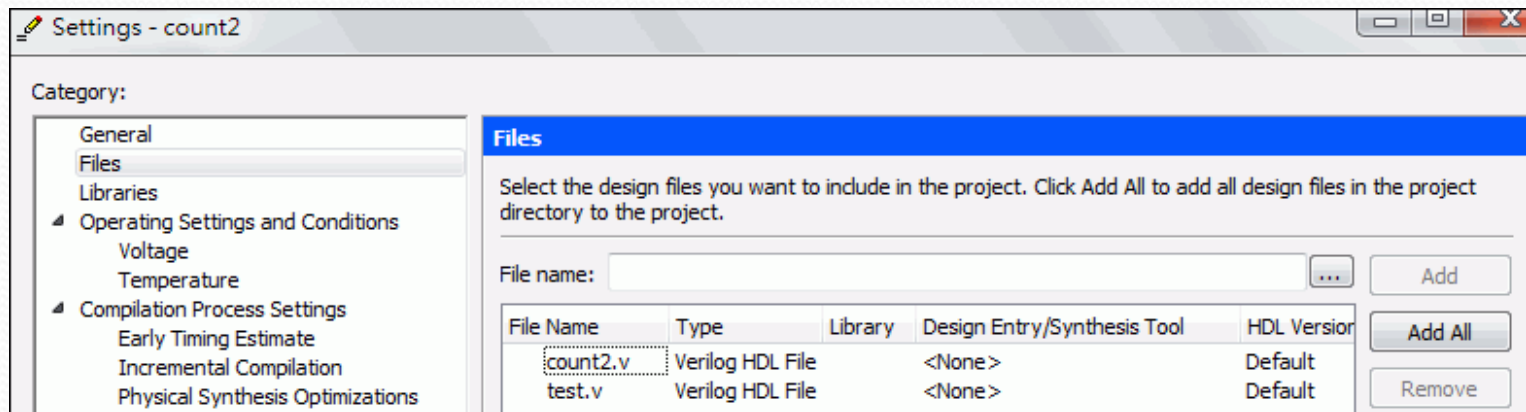
» 圖63 非同步清除2位元同步上數計數器



## ● 4-4 模擬驗證

### ● 編輯流程如下：

開啟專案 → 指定元件 → 建立測試平台(test bench)  
→ 另存新檔 → 建立測試模組名稱 → 加入reg接線  
→ 加入wire接線 → 引入count2的模組 → 設定初始  
值 → 定義時間單位 → 變化波形 → 存檔 → 設定模  
擬工具路徑 → 設定模擬工具 → 組譯並模擬 → 調整  
視窗範圍 → 檢驗模擬結果 → 關閉ModelSim-Altera



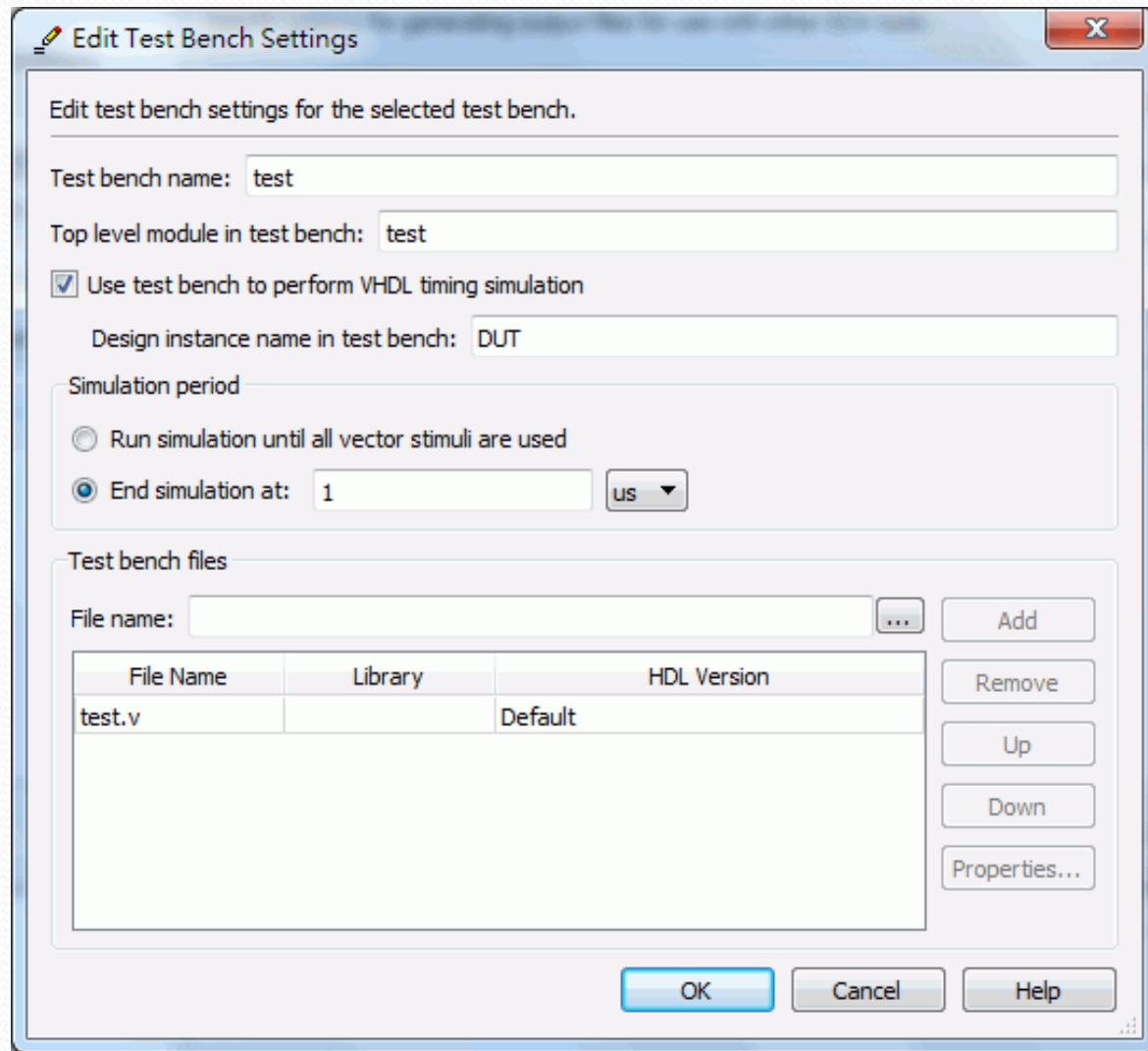
» 圖64 移除與加入檔案



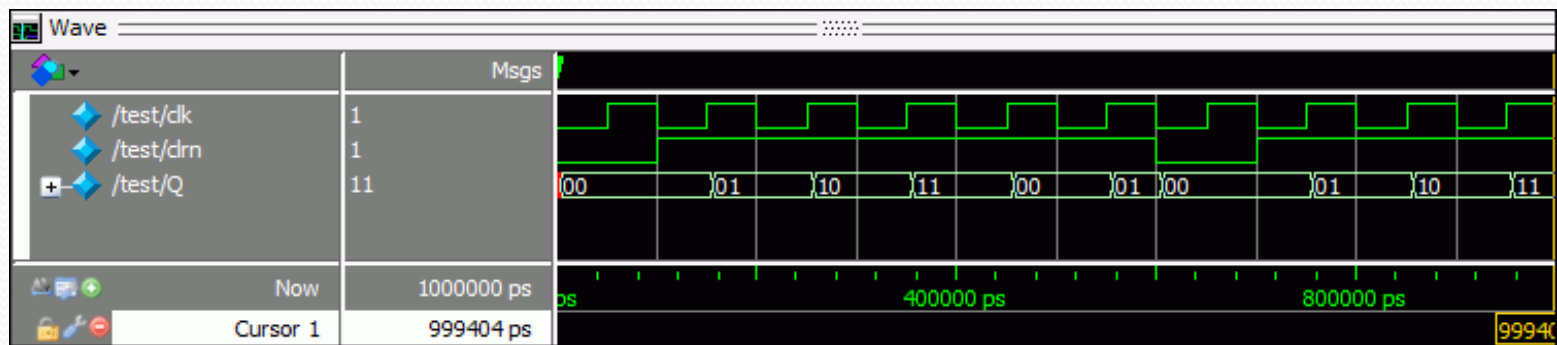
```
`timescale 1ns/10ps
module test;
reg clk;
reg clrn;
wire [1:0] Q;
count2 DUT (.clk(clk), .clrn(clrn), .Q(Q));
initial
begin
  clk <= 1'b0;
  clrn <= 1'b0;
end
always #50 clk <= ~ clk;
initial #100 clrn <= 1'b1;
initial #600 clrn <= 1'b0;
initial #700 clrn <= 1'b1;
endmodule
```

» 表13 Test Bench設定之變化波形





» 圖65 「New Test Benches Settings」視窗



» 圖66 模擬波型

- 檢驗模擬結果：觀察輸出結果，檢查結果如下：
  - 50ns處clk為正緣，clrn=1'b0，輸出清除為0，輸出Q為2'b00。
  - 150ns處clk為正緣，clrn=1'b1，輸出增加1，輸出Q變為2'b01。
  - 250ns處clk為正緣，clrn=1'b1，輸出增加1，輸出Q變為2'b10。
  - 350ns處clk為正緣，clrn=1'b1，輸出增加1，輸出Q變為2'b11。
  - 450ns處clk為正緣，clrn=1'b1，輸出增加1，輸出Q變為2'b00。
  - 550ns處clk為正緣，clrn=1'b1，輸出增加1，輸出Q變為2'b01。
  - 600ns處，clrn=1'b0，輸出清除為0，輸出Q變為2'b00。
  - 750ns處clk為正緣，clrn=1'b1，輸出增加1，輸出Q變為2'b01。
- 與表9之真值表做比較，對照無誤。



## 5. 可預設的同步10模計數器

- 因為每一個正反器均有2個狀態，故若以 $n$ 代表正反器的數目，則其模數為 $2^n$ ，亦即具有 $2^n$ 個自然計數值，
- 因此可設計具有2、4、8、16、32、64等等自然計數值的計數器。
- 然而，在日常生活上，十進位的計數器為最常且最方便使用的計數器，也就是以二進位自然數順序，由“0000”計數至“1001”，
- 但是必需經過設計過的電路連接方式才能達成此項功能。

- 以 IC “74162” 之電路作例子，其具有4位元上數十進位計數和同步預設和同步清除功能。
  - 當載入控制端 “ldn” 為 “0” 時，將輸入資料送至正反器輸入端，
  - 當載入控制端 “ldn” 為 “1” 時，計數器停止載入。
  - 並有一進位輸出端 “RCO”。
- 腳位：控制線5條：clrn、ENT、ENP、load、clk
  - 資料輸入線4條：D3、D2、D1、D0
  - 資料輸出線4條：Q3、Q2、Q1、Q0
  - 串接進位線1條： $Co = Q3 \text{ AND } Q0 \text{ AND } ENT$





clk	clrn	load	Enable		D[3..0]				Q[3..0]			
			Ent	Enp	D[3]	D[2]	D[1]	D[0]	Q[3]	Q[2]	Q[1]	Q[0]
↑	0	X	X	X	X	X	X	X	0	0	0	0
↑	1	0	X	X	A	B	C	D	A	B	C	D
↑	1	1	0	1	X	X	X	X	Q(不變)			
↑	1	1	1	0	X	X	X	X	Q(不變)			
↑	1	1	1	1	X	X	X	X	Q = Q+1(上數) 最高數到 “1001”			

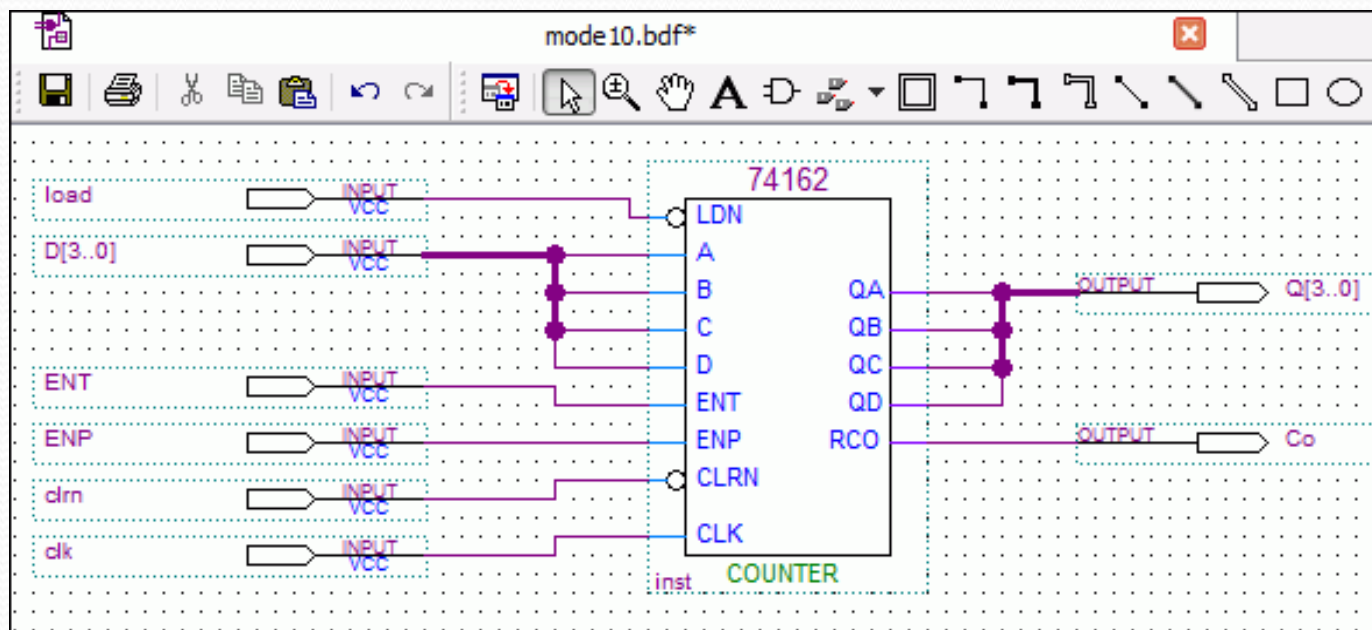
» 表14 可預設的同步10模計數器



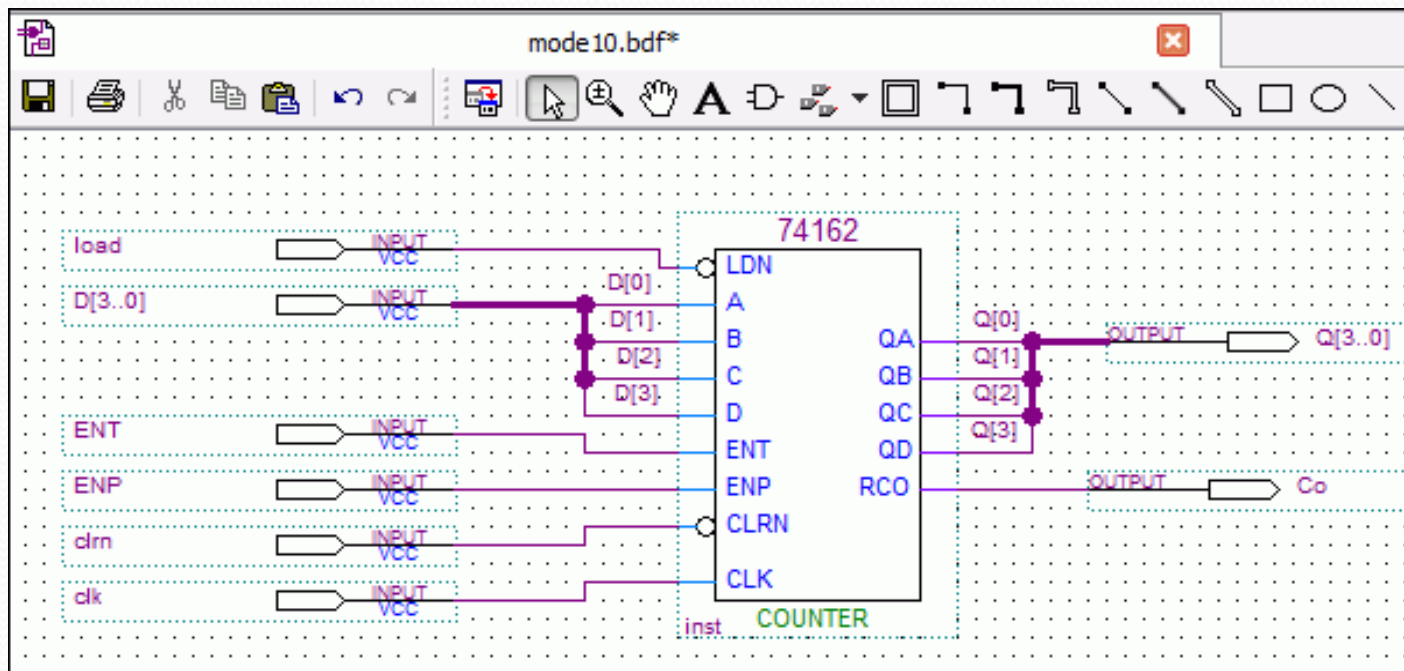
- 5-1 電路圖編輯

- 編輯流程如下：

開啟新增專案精靈：建立專案model0 → 新增檔案  
→ 另存新檔：儲存為model0.bdf → 編輯檔案內容：  
引入四個dfffe符號 → 引入輸入和輸出腳：引入兩個  
input閘，四個output閘 → 更改輸入輸出腳之腳位名  
稱：輸入腳為load、D[3..0]、ENP、ENT、clrn和clk，  
輸出腳為Q[3..0]與Co → 連線：利用畫正交線工  
具與 → 導線命名 → 存檔並組譯 → 模擬驗  
證



» 圖67 連線結果



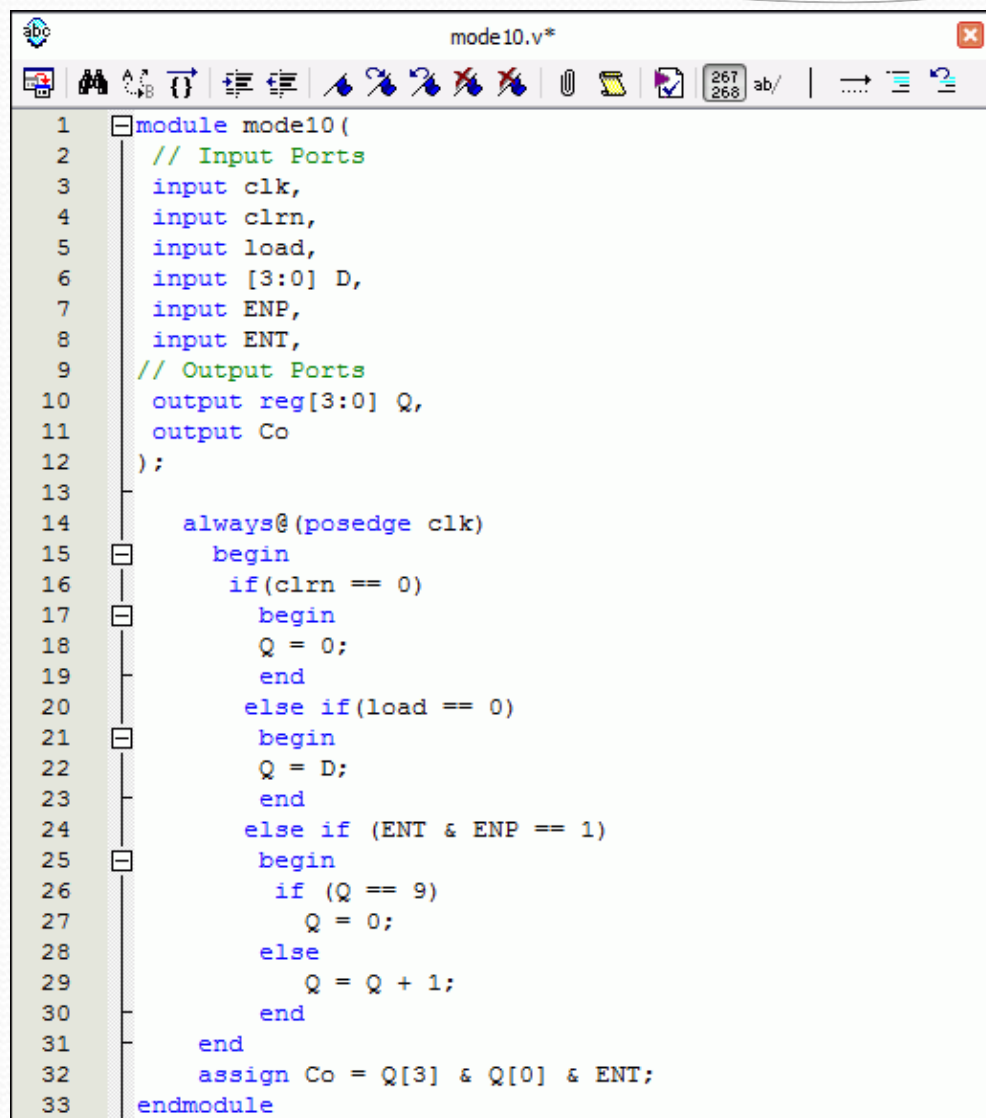
» 圖68 可預設的同步10模計數器



## ● 5-2 Verilog HDL編輯

- 編輯流程如下：

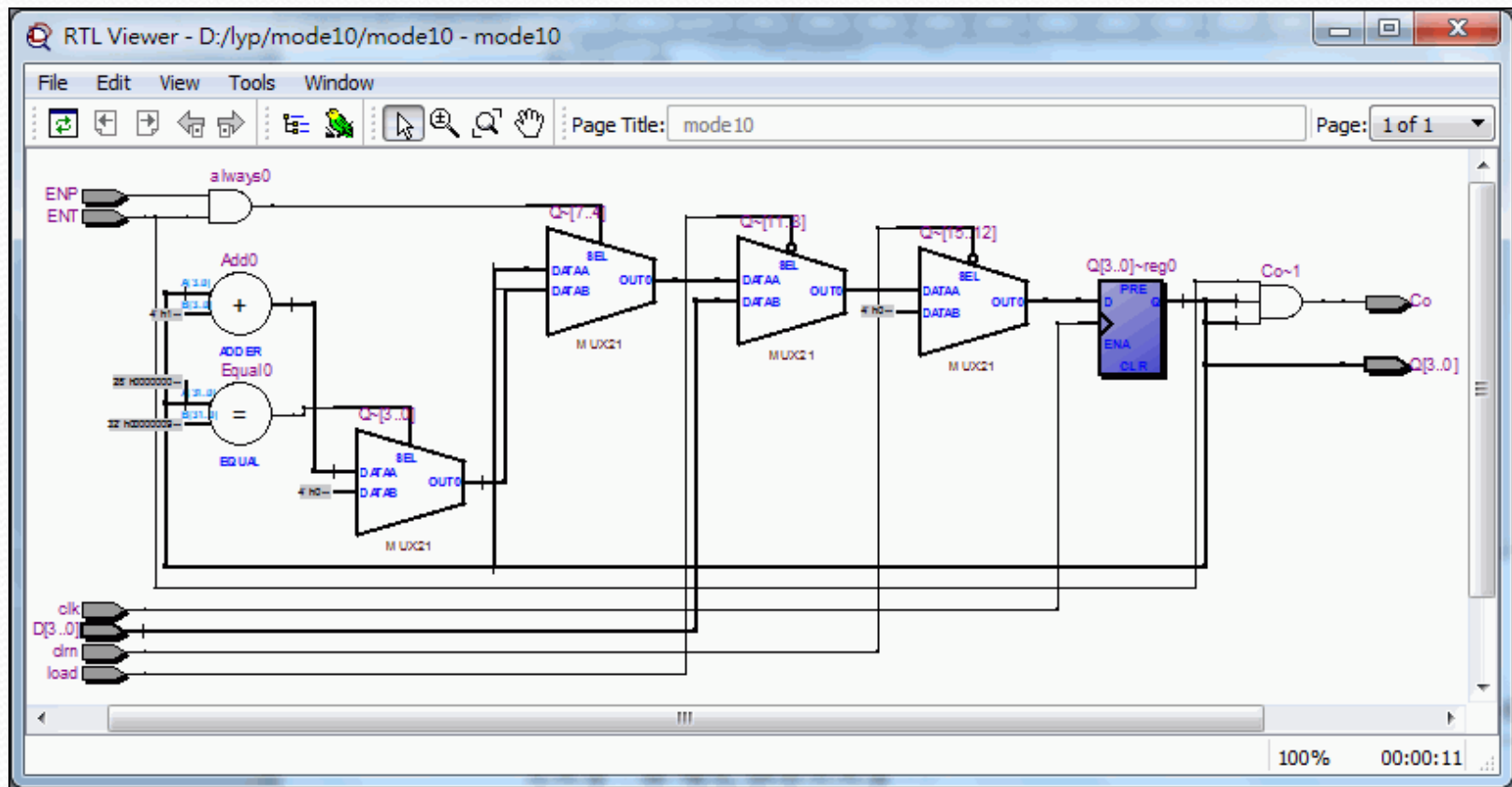
開啟新增專案精靈：建立專案model0 → 新增檔案 →  
另存新檔：儲存為model0.v → 編輯檔案內容 → 存檔  
並組譯 → 模擬驗證



The screenshot shows a window titled "mode10.v\*" with a toolbar at the top. The code is as follows:

```
1 module mode10(  
2     // Input Ports  
3     input clk,  
4     input clrn,  
5     input load,  
6     input [3:0] D,  
7     input ENP,  
8     input ENT,  
9     // Output Ports  
10    output reg[3:0] Q,  
11    output Co  
12 );  
13  
14    always@(posedge clk)  
15    begin  
16        if(clrn == 0)  
17        begin  
18            Q = 0;  
19        end  
20        else if(load == 0)  
21        begin  
22            Q = D;  
23        end  
24        else if (ENT & ENP == 1)  
25        begin  
26            if (Q == 9)  
27                Q = 0;  
28            else  
29                Q = Q + 1;  
30        end  
31    end  
32    assign Co = Q[3] & Q[0] & ENT;  
33 endmodule
```

» 圖69 可預設的同步10模計數器



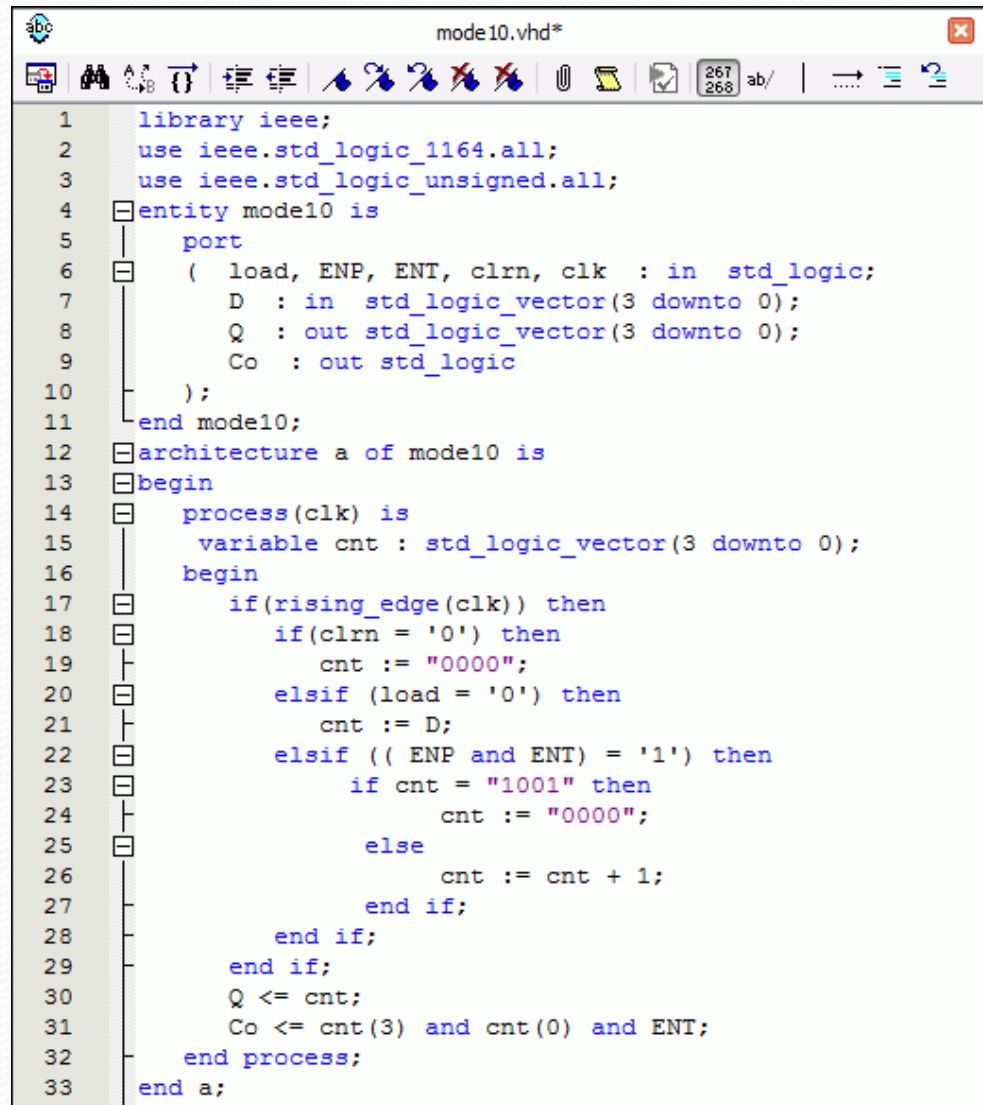
» 圖70 RTL Viewer



## ● 5-3 VHDL編輯

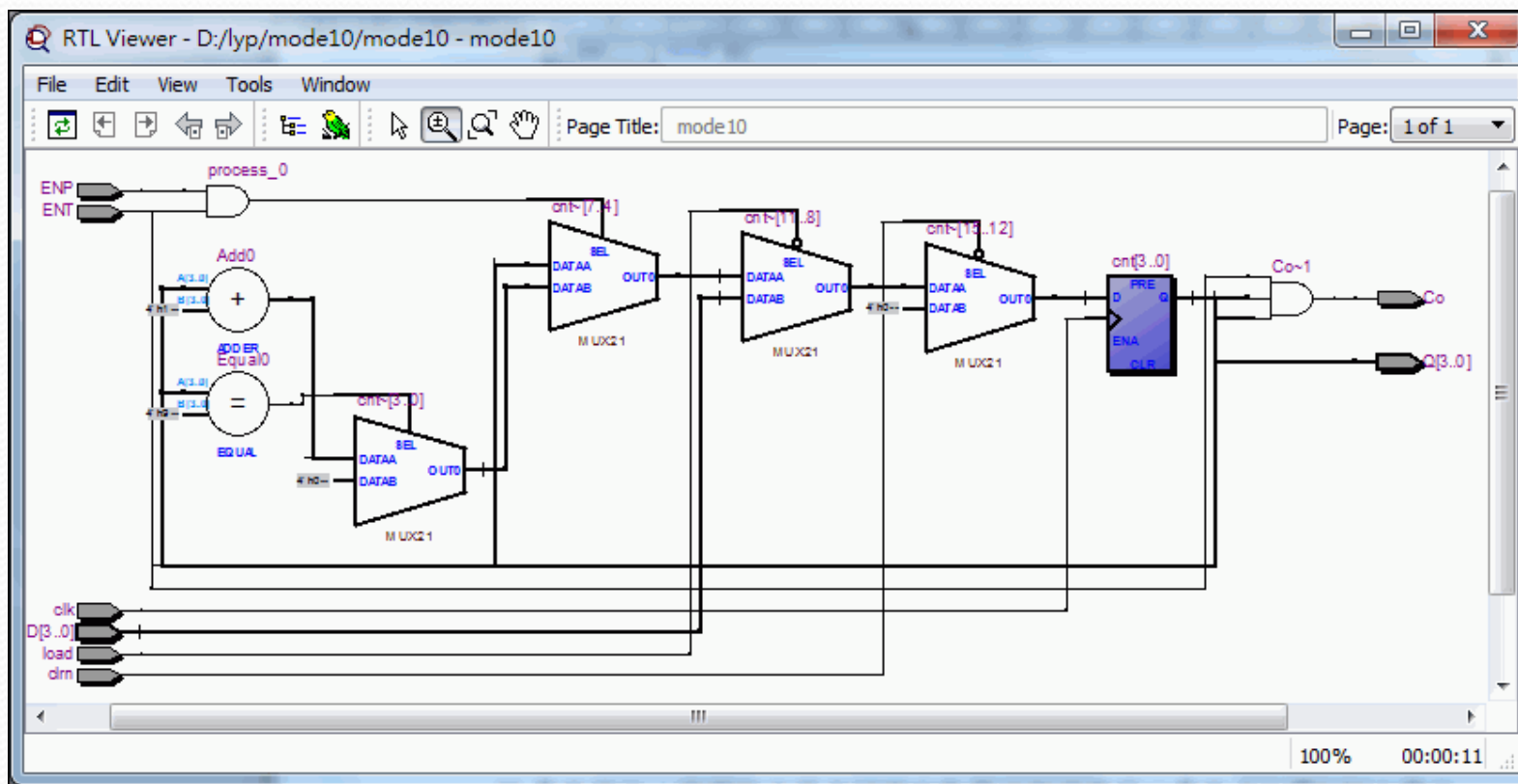
- 編輯流程如下：

建立model0專案 → 新增檔案 → 另存新檔：儲存為  
model0.vhd → 編輯檔案內容 → 存檔並組譯 → 模擬  
驗證



```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_unsigned.all;
4  entity mode10 is
5  port
6  ( load, ENP, ENT, clrn, clk : in std_logic;
7    D : in std_logic_vector(3 downto 0);
8    Q : out std_logic_vector(3 downto 0);
9    Co : out std_logic
10 );
11 end mode10;
12 architecture a of mode10 is
13 begin
14   process(clk) is
15     variable cnt : std_logic_vector(3 downto 0);
16     begin
17       if(rising_edge(clk)) then
18         if(clrn = '0') then
19           cnt := "0000";
20         elsif (load = '0') then
21           cnt := D;
22         elsif (( ENP and ENT) = '1') then
23           if cnt = "1001" then
24             cnt := "0000";
25           else
26             cnt := cnt + 1;
27           end if;
28         end if;
29       end if;
30       Q <= cnt;
31       Co <= cnt(3) and cnt(0) and ENT;
32     end process;
33 end a;
```

» 圖71 可預設的同步10模計數器



» 圖72 RTL Viewer



```

abc model10.vhd
3  LIBRARY altera;
4  USE altera.maxplus2.ALL;
5  ENTITY model10 IS
6      PORT(    load,ENP,ENT,clrn,clk    : IN  STD_LOGIC;
7              D    : IN    STD_LOGIC_VECTOR(3 DOWNT0 0);
8              Q    : OUT   STD_LOGIC_VECTOR(3 DOWNT0 0);
9              Co   : OUT   STD_LOGIC      );
10 END model10;
11 ARCHITECTURE a OF model10 IS
12 BEGIN
13 inst : a_74162
14     PORT MAP(CLK => clk, LDN => load, A => D(0),
15             B => D(1), C => D(2), D => D(3),
16             CLRN => clrn, ENP  => ENP,  ENT => ENT,
17             QA => Q(0),QB => Q(1), QC => Q(2),
18             QD => Q(3), RCO => Co );
19 END a;

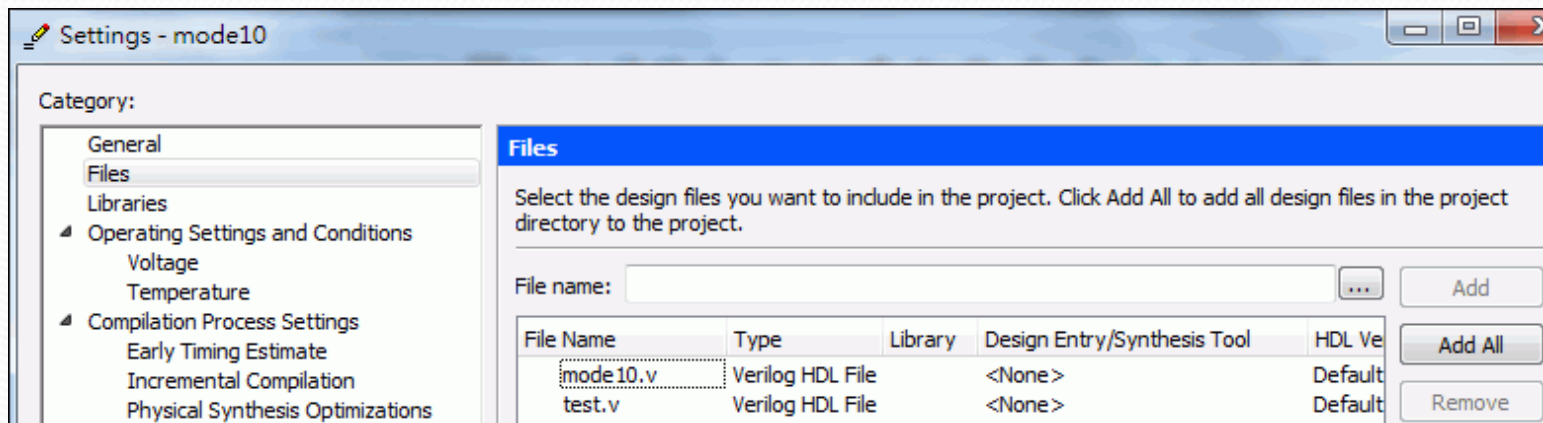
```

» 圖73 可預設的同步10模計數器

## ● 5-4 模擬驗證

### ● 編輯流程如下：

開啟專案 → 指定元件 → 建立測試平台(test bench)  
→ 另存新檔 → 建立測試模組名稱 → 加入reg接線  
→ 加入wire接線 → 引入model0的模組 → 設定初始  
值 → 定義時間單位 → 變化波形 → 存檔 → 設定模  
擬工具路徑 → 設定模擬工具 → 組譯並模擬 → 調整  
視窗範圍 → 檢驗模擬結果 → 關閉ModelSim-Altera



» 圖74 移除與加入檔案



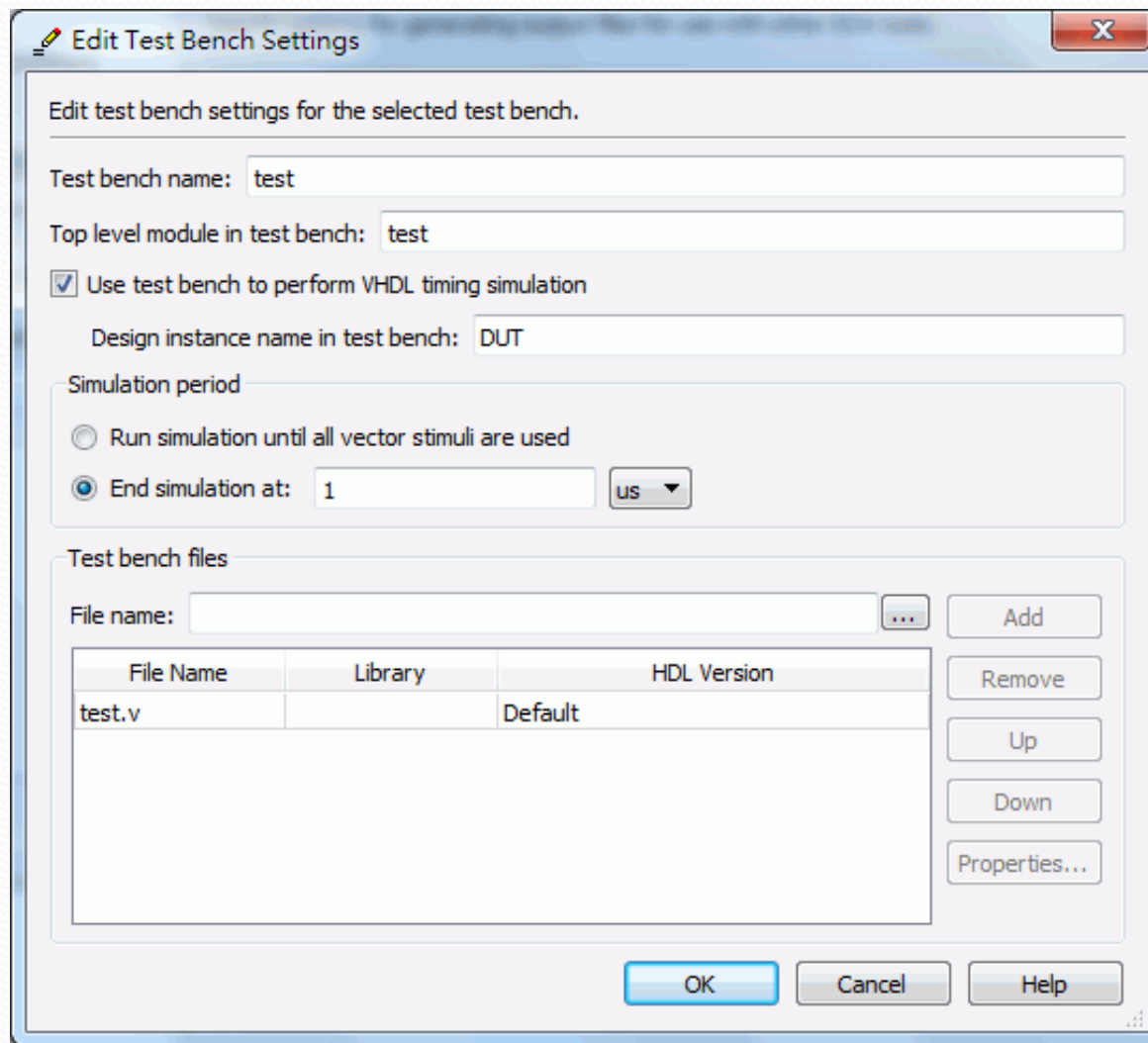
```

`timescale 1ns/10ps
module test;
  // Input Ports
  reg clk;
  reg clrn;
  reg load;
  reg [3:0] D;
  reg ENP;
  reg ENT;
  // Output Ports
  wire[3:0] Q;
  wire Co;
  model0 DUT (.clk(clk), .clrn(clrn), .load(load), .D(D),
              .ENP(ENP), .ENT(ENT), .Q(Q), .Co(Co));

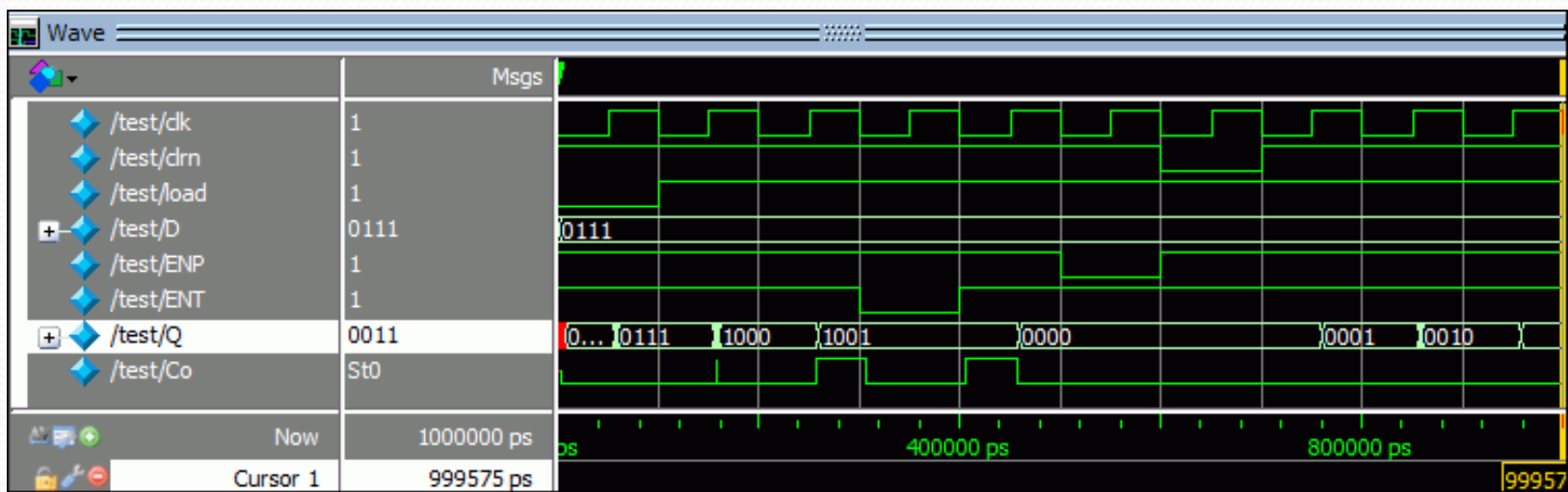
  initial
  begin
    clk <= 1'b0;
    clrn <= 1'b1;
    load <= 1'b0;
    D <= 4'b0111;
    ENP <= 1'b1;
    ENT <= 1'b1;
  end
  always #50 clk <= ~ clk;
  initial #600 clrn <= 1'b0;
  initial #700 clrn <= 1'b1;
  initial #100 load <= 1'b1;
  initial #300 ENT <= 1'b0;
  initial #400 ENT <= 1'b1;
  initial #500 ENP <= 1'b0;
  initial #600 ENP <= 1'b1;
endmodule

```

» 表15 Test Bench設定之變化波形



» 圖75 「New Test Benches Settings」視窗



» 圖76 模擬波型



- 檢驗模擬結果：觀察輸出結果，檢查結果如下：
  - 50ns處clk為正緣，clrn=1，load=0，執行載入D的值至輸出，因為D=4'b0111，所以輸出Q=D=4'b0111。
  - 150ns處clk為正緣，clrn=1，load=1，ENP=1，ENT=1，執行計數，輸出增加1，所以輸出Q變為4'b1000。
  - 250ns處clk為正緣，clrn=1，load=1，ENP=1，ENT=1，執行計數，輸出增加1，所以輸出Q變為4'b 1001。輸出Co=  $Q(3) \& Q(0) \& ENT \& ENP = 1$ 。
  - 300ns至350ns處ENT=0，故輸出Co=  $Q(3) \text{ AND } Q(0) \text{ AND } ENT=0$ 。
  - 350ns處clk為正緣，clrn=1，load=1，clrn=1，load=0，ENP=1，ENT=0，輸出Q保持不變依舊為4'b 1001，輸出Co=  $Q(3) \text{ AND } Q(0) \text{ AND } ENT=0$ 。
  - 400ns至450ns處，ENT=1，Co= $Q(3) \text{ AND } Q(0) \text{ AND } ENT=1$ 。
  - 450ns處clk為正緣，clrn=1，load=1，ENP=1，ENT=1，輸出增加1，輸出Q變為4'b 0001。
  - 550ns處clk為正緣，clrn=1，load=1，ENP=1，ENT=0，輸出Q保持不變依舊為4'b 0001。
  - 650ns處clk為正緣，clrn=0，執行清除，所以輸出Q變清除為4'b0000。
- 與表14之真值表做比較，對照無誤。

## 6. 非同步清除同步載入60模計數器

- 數字鐘的秒針部分由60模計數器組成，分針部分亦由60模計數器所組成。
- 本範例分別設計6模計數器與十模計數器組合成60模計數器。



- 腳位：脈波輸入端：clk
  - 同步載入控制端：load
  - 非同步清除端：clrn
  - 致能端：en
  - 資料載入端：Da[3..0]、Db[2..0]
  - 輸出端：Qa[3..0]、Qb[2..0]
  - 進位輸出端：Co=en AND Qa3 AND Qa0 AND Qb2 AND Qb0



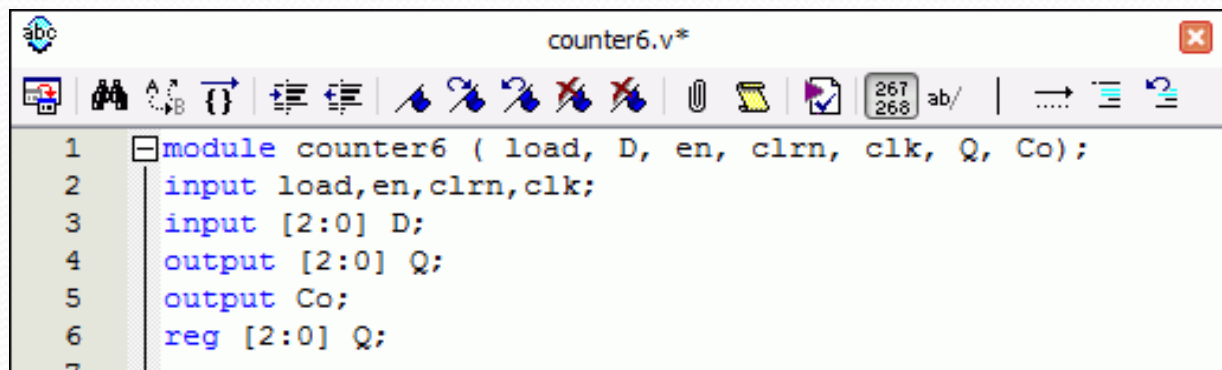
控制端				十位載入	個位載入	十位輸出	個位輸出
clk	clrn	load	en	Db[2..0]	Da[3..0]	Qb[2..0]	Qa[3..0]
X	1	X	X	X	X	0	0
↑	0	1	X	B	A	B	A
↑	0	0	0	X	X	Q (不變)	
↑	0	0	1	X	X	Q=Q+1 最高數到59	

» 表14 60模計數器真值表

## ● 6-2 Verilog HDL編輯

- 編輯流程如下：

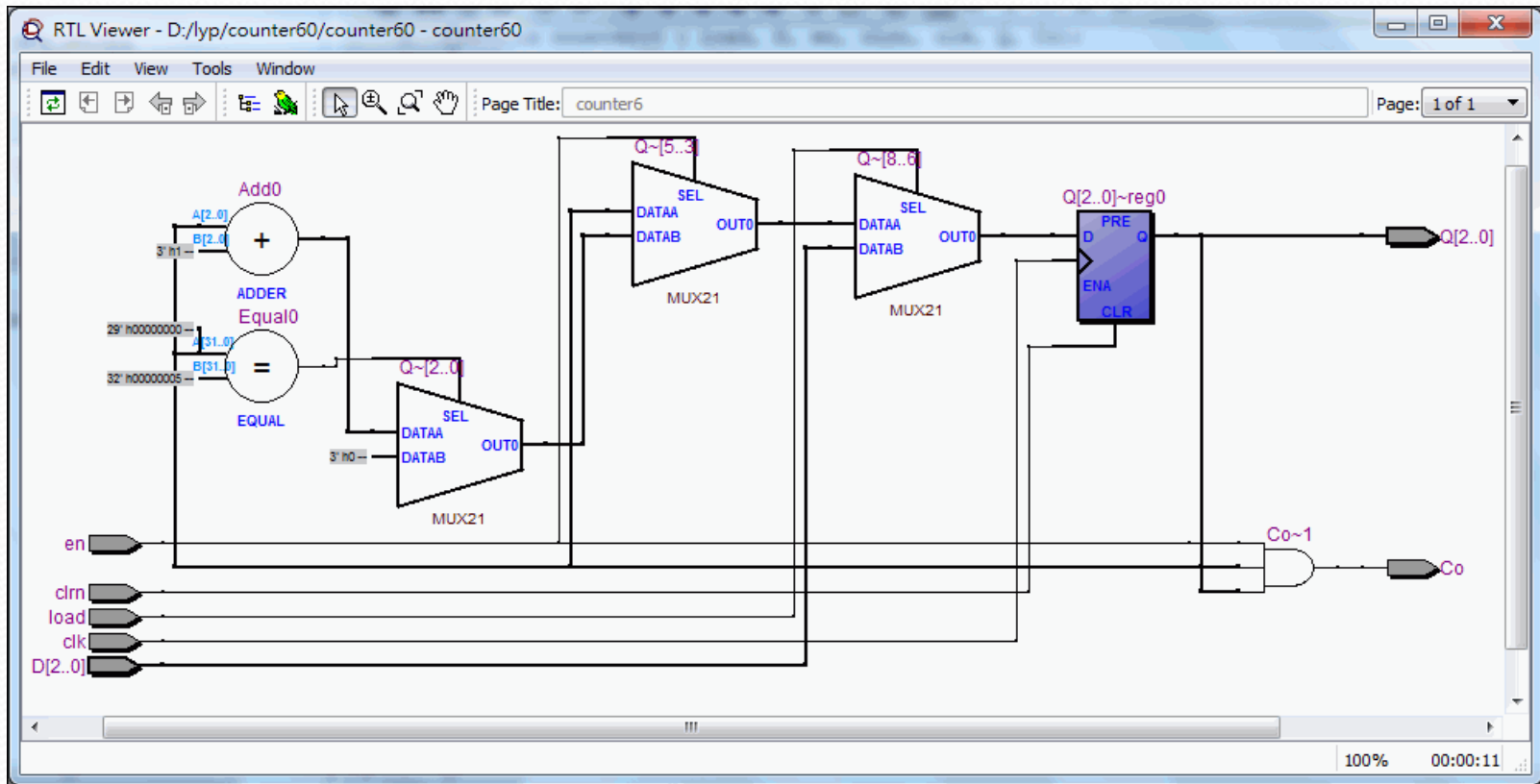
開啟新增專案精靈：建立counter60專案 → 建立6模計數器檔案 → 另存新檔：儲存為counter6.v → 建立10模計數器 → 另存新檔：儲存為counter10.v → 建立60模計數器 → 另存新檔：儲存為counter60.v → 存檔並組譯 → 模擬驗證



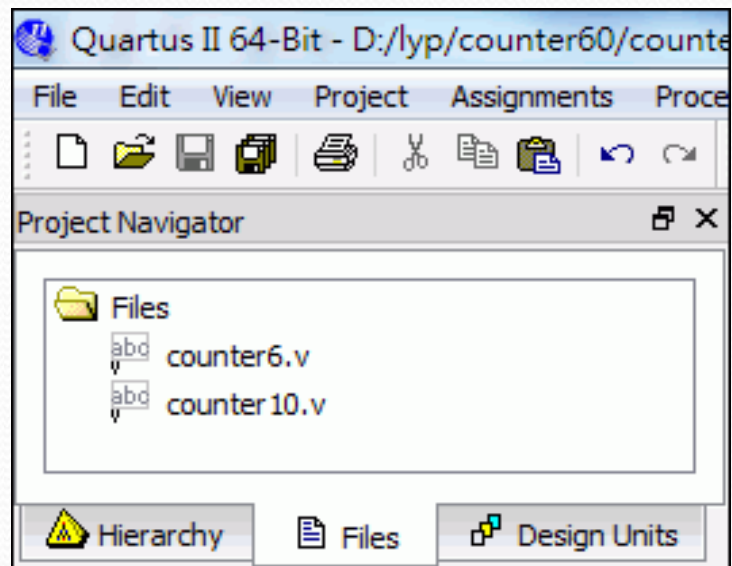
```
1 module counter6 ( load, D, en, clrn, clk, Q, Co);  
2   input load,en,clrn,clk;  
3   input [2:0] D;  
4   output [2:0] Q;  
5   output Co;  
6   reg [2:0] Q;  
7
```

» 圖89 Verilog HDL編輯六模計數器 腳位設定

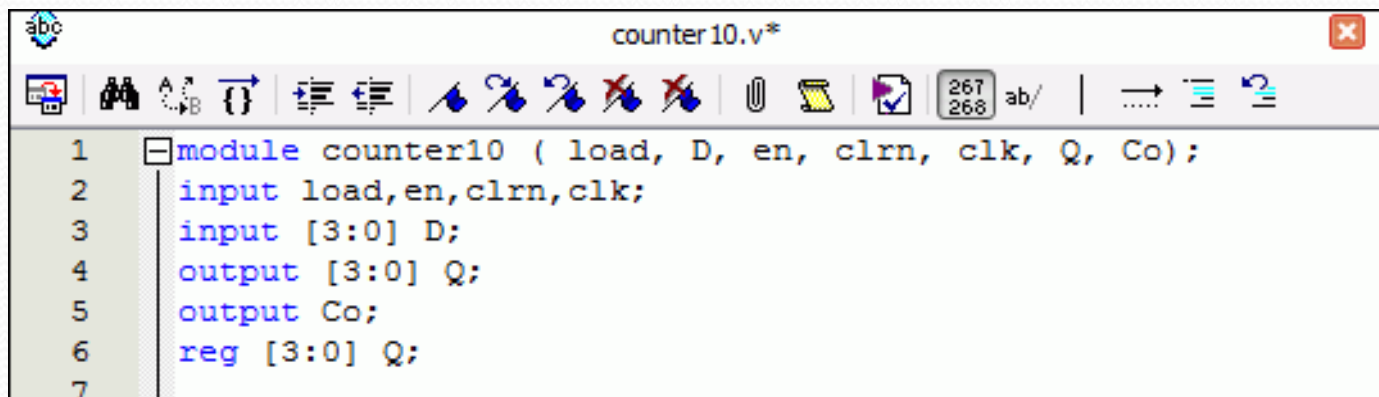




» 圖91 RTL Viewer



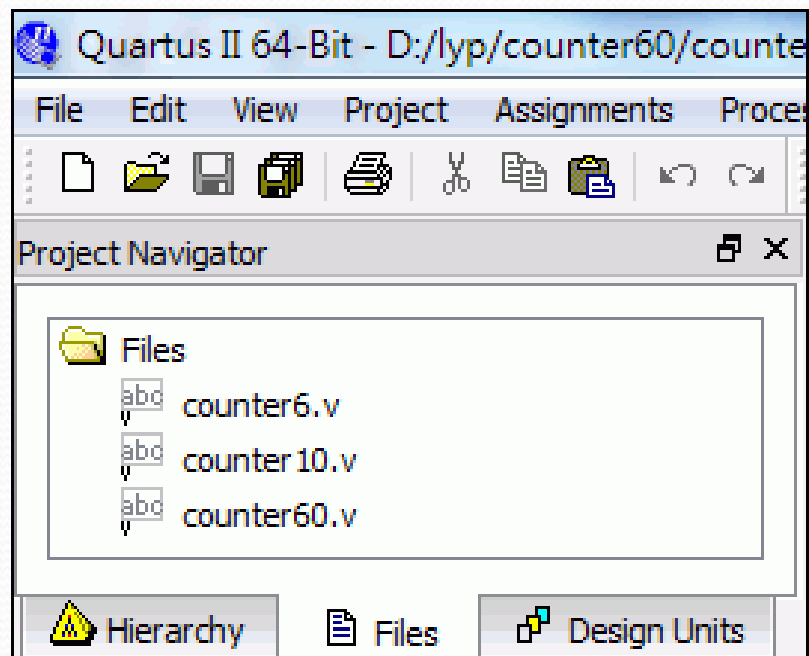
» 圖92 專案檔案導覽



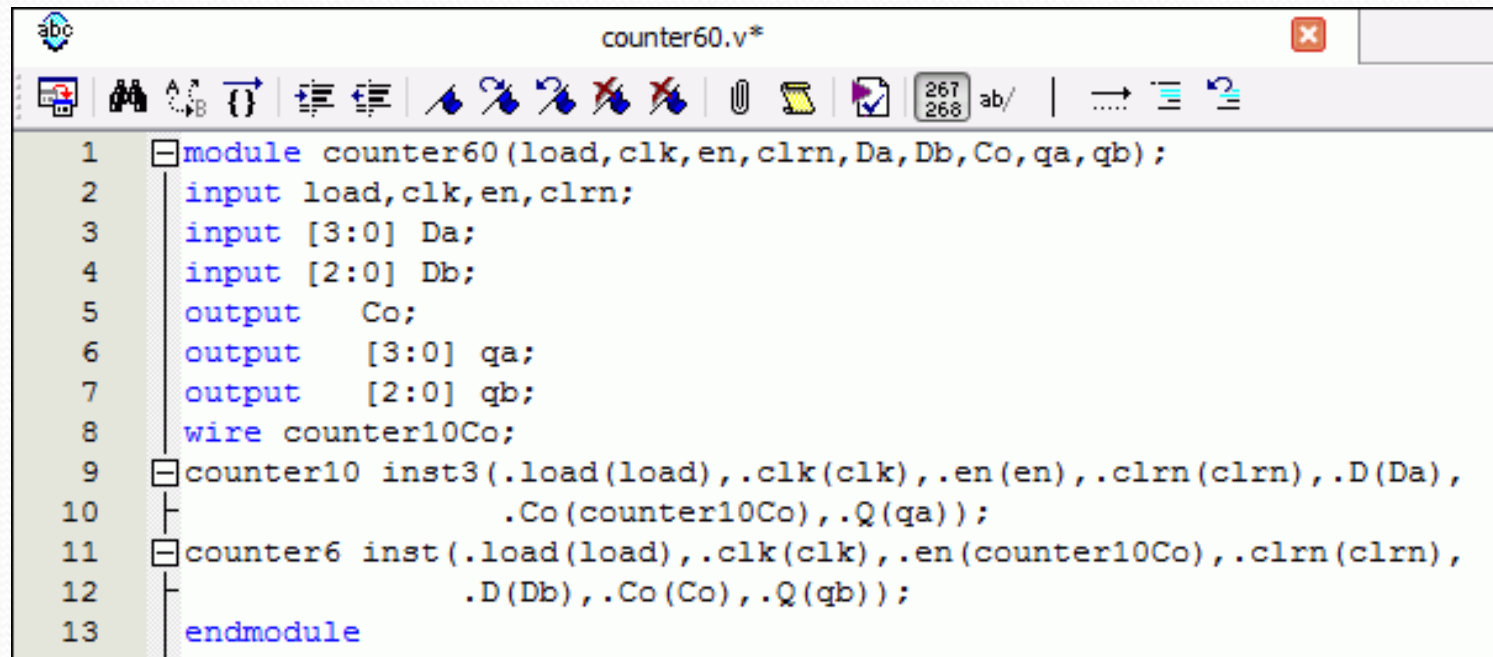
```
1 module counter10 ( load, D, en, clrn, clk, Q, Co);  
2   input load,en,clrn,clk;  
3   input [3:0] D;  
4   output [3:0] Q;  
5   output Co;  
6   reg [3:0] Q;  
7
```

» 圖93 Verilog HDL編輯十模計數器 腳位設定



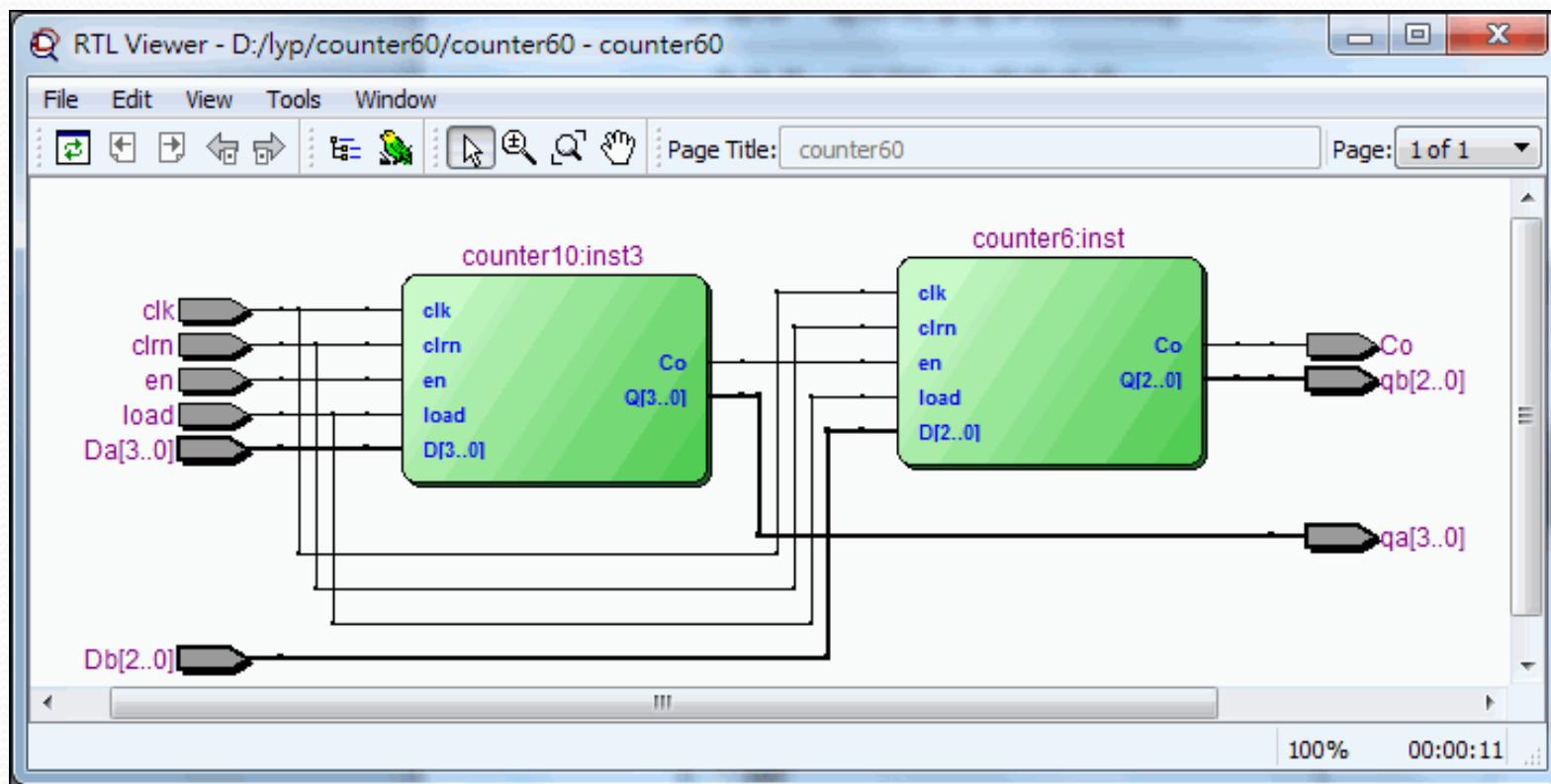


» 圖94 專案檔案導覽



```
1 module counter60 (load, clk, en, clrn, Da, Db, Co, qa, qb);
2     input load, clk, en, clrn;
3     input [3:0] Da;
4     input [2:0] Db;
5     output Co;
6     output [3:0] qa;
7     output [2:0] qb;
8     wire counter10Co;
9     counter10 inst3 (.load(load), .clk(clk), .en(en), .clrn(clrn), .D(Da),
10                    .Co(counter10Co), .Q(qa));
11     counter6 inst (.load(load), .clk(clk), .en(counter10Co), .clrn(clrn),
12                  .D(Db), .Co(Co), .Q(qb));
13 endmodule
```

» 圖95 Verilog HDL編輯六十模計數器

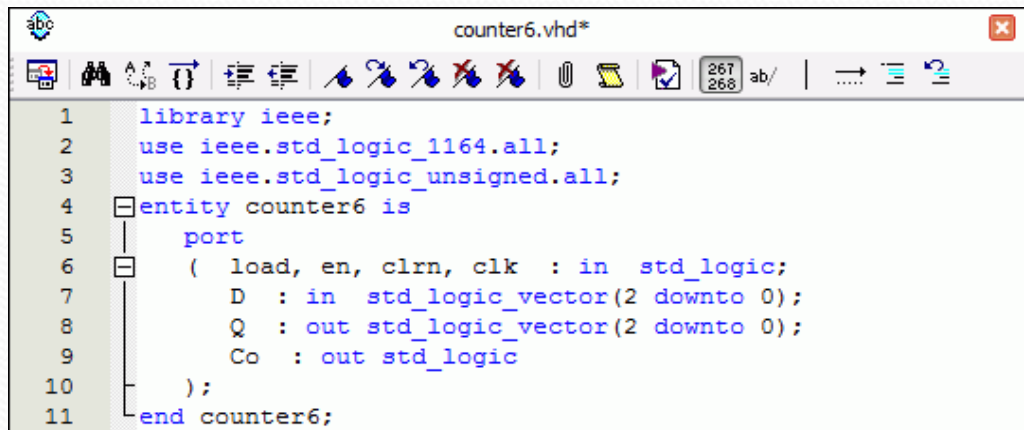


» 圖96 RTL Viewer



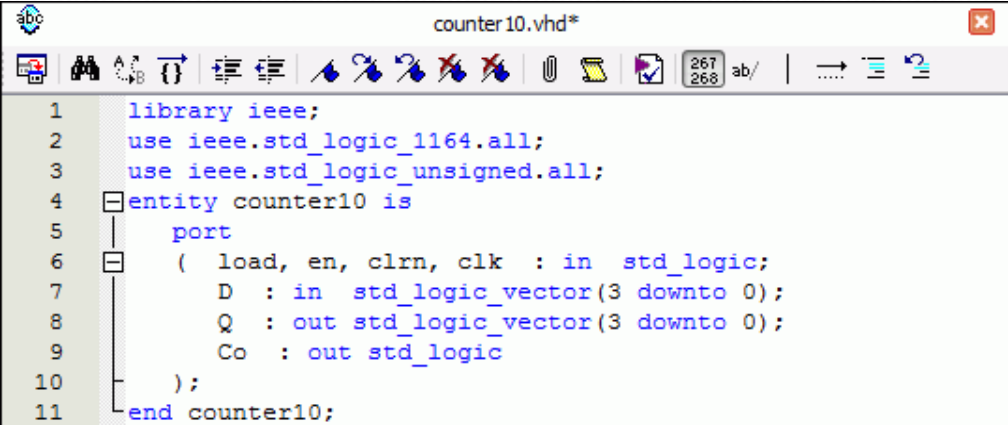
## ● 6-3 VHDL編輯

開啟新增專案精靈：建立counter60專案 → 建立6模計數器檔案 → 另存新檔：儲存為counter6.vhd → 建立10模計數器 → 另存新檔：儲存為counter10.vhd → 建立60模計數器 → 另存新檔：儲存為counter60.vhd → 存檔並組譯 → 模擬驗證



```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_unsigned.all;
4  entity counter6 is
5      port
6      (   load, en, clrn, clk : in  std_logic;
7          D   : in  std_logic_vector(2 downto 0);
8          Q   : out std_logic_vector(2 downto 0);
9          Co  : out std_logic
10     );
11 end counter6;
```

» 圖97 VHDL編輯六模計數器 腳位設定



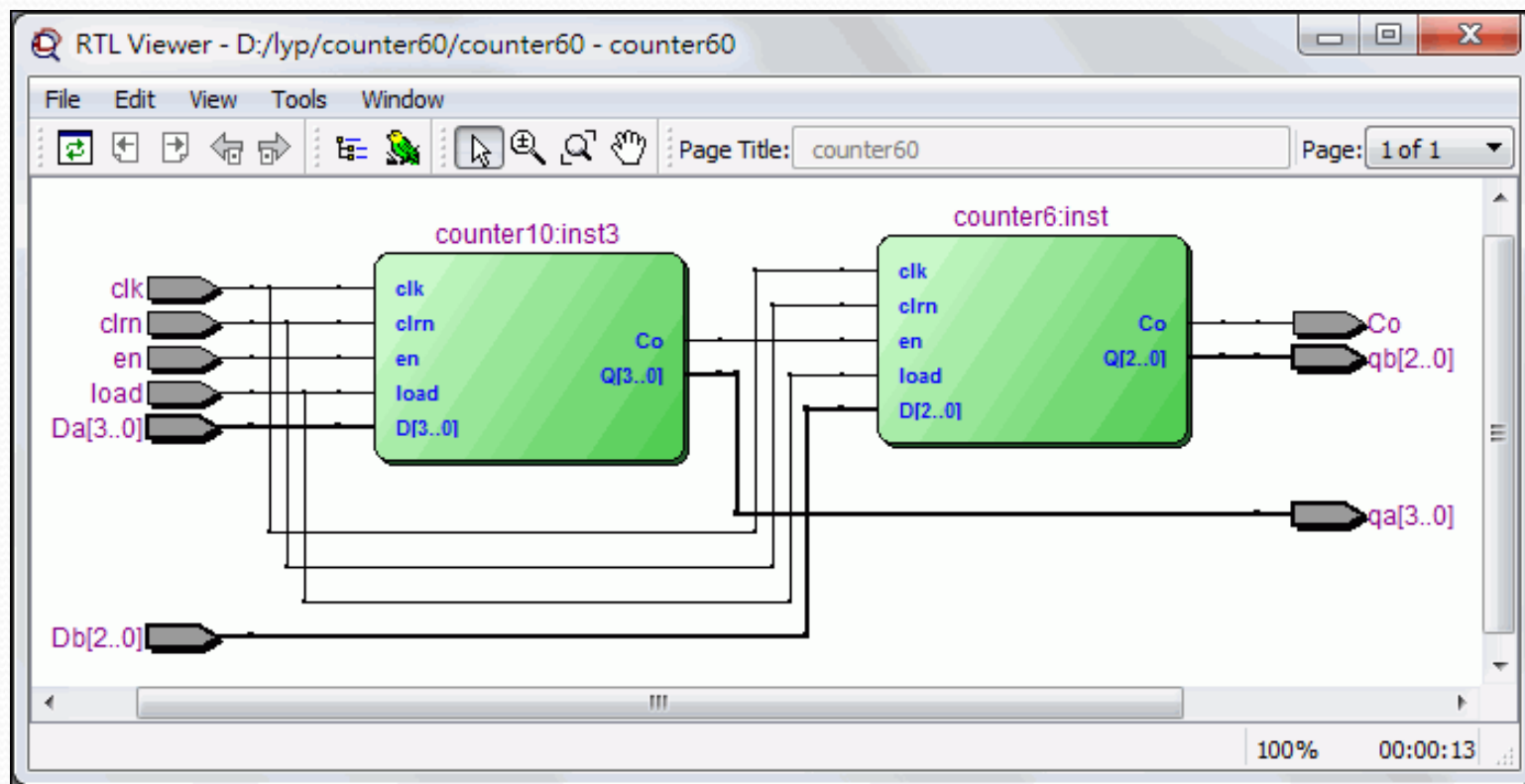
```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_unsigned.all;
4  entity counter10 is
5      port
6      (  load, en, clrn, clk : in  std_logic;
7          D : in  std_logic_vector(3 downto 0);
8          Q : out std_logic_vector(3 downto 0);
9          Co : out std_logic
10     );
11  end counter10;
```

» 圖100 VHDL編輯十模計數器 腳位設定



```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  entity counter60 is
4      port
5      (
6          load,clk,en,clrn : in  std_logic;
7          Da : in  std_logic_vector(3 downto 0);
8          Db : in  std_logic_vector(2 downto 0);
9          Co : out std_logic;
10         qa : out std_logic_vector(3 downto 0);
11         qb : out std_logic_vector(2 downto 0)
12     );
13 end counter60;
14 architecture a of counter60 is
15     component counter6
16     port(load,clk,en,clrn : in  std_logic;
17         D : in  std_logic_vector(2 downto 0);
18         Co : out std_logic;
19         q : out std_logic_vector(2 downto 0)
20     );
21 end component;
22 component counter10
23 port(load,clk,en,clrn : in  std_logic;
24     D : in  std_logic_vector(3 downto 0);
25     Co : out std_logic;
26     q : out std_logic_vector(3 downto 0)
27 );
28 end component;
29 signal counter10Co : std_logic;
30 begin
31     inst3 : counter10
32     port map(load => load, clk => clk,en => en,clrn => clrn,
33         D => Da, Co => counter10Co, q => qa);
34     inst : counter6
35     port map(load => load,clk => clk,en => counter10Co,clrn => clrn,
36         D => Db,Co => Co, q => qb);
37 end;
```

» 圖102 VHDL編輯六十模計數器



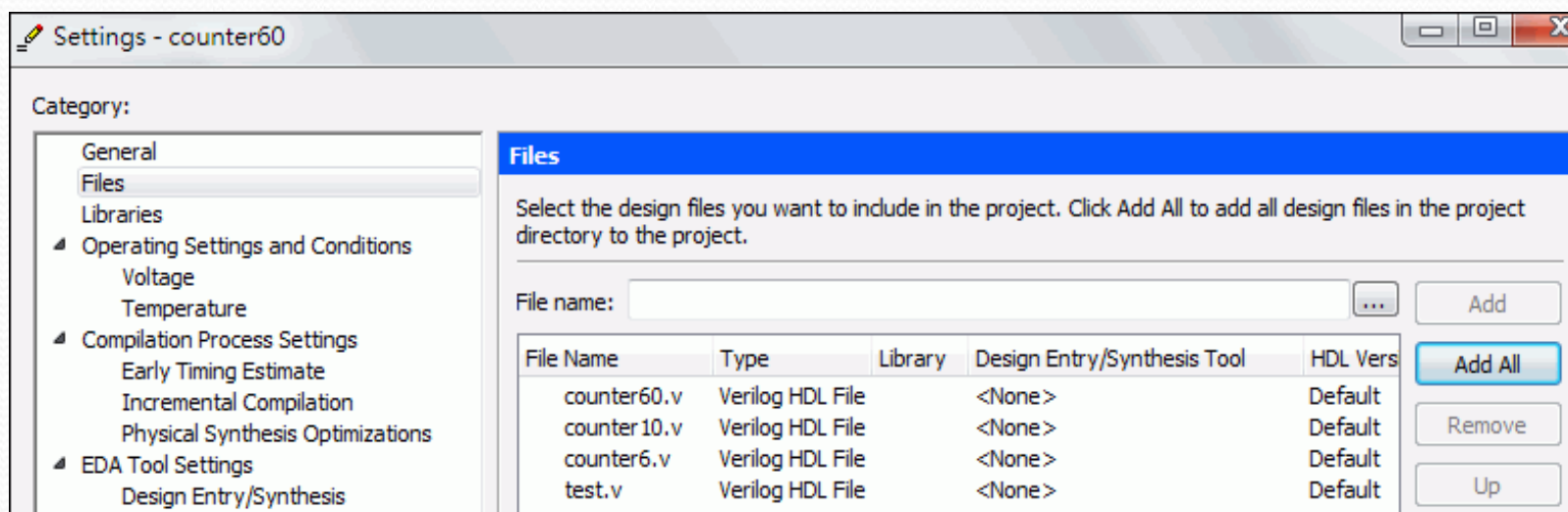
» 圖103 RTL Viewer

## ● 6-4 模擬驗證

### ● 編輯流程如下：

開啟專案 → 指定元件 → 建立測試平台(test bench)  
→ 另存新檔 → 建立測試模組名稱 → 加入reg接線  
→ 加入wire接線 → 引入counter60的模組 → 設定初  
始值 → 定義時間單位 → 變化波形 → 存檔 → 設定  
模擬工具路徑 → 設定模擬工具 → 組譯並模擬 → 調  
整視窗範圍 → 檢驗模擬結果 → 關閉ModelSim-  
Altera

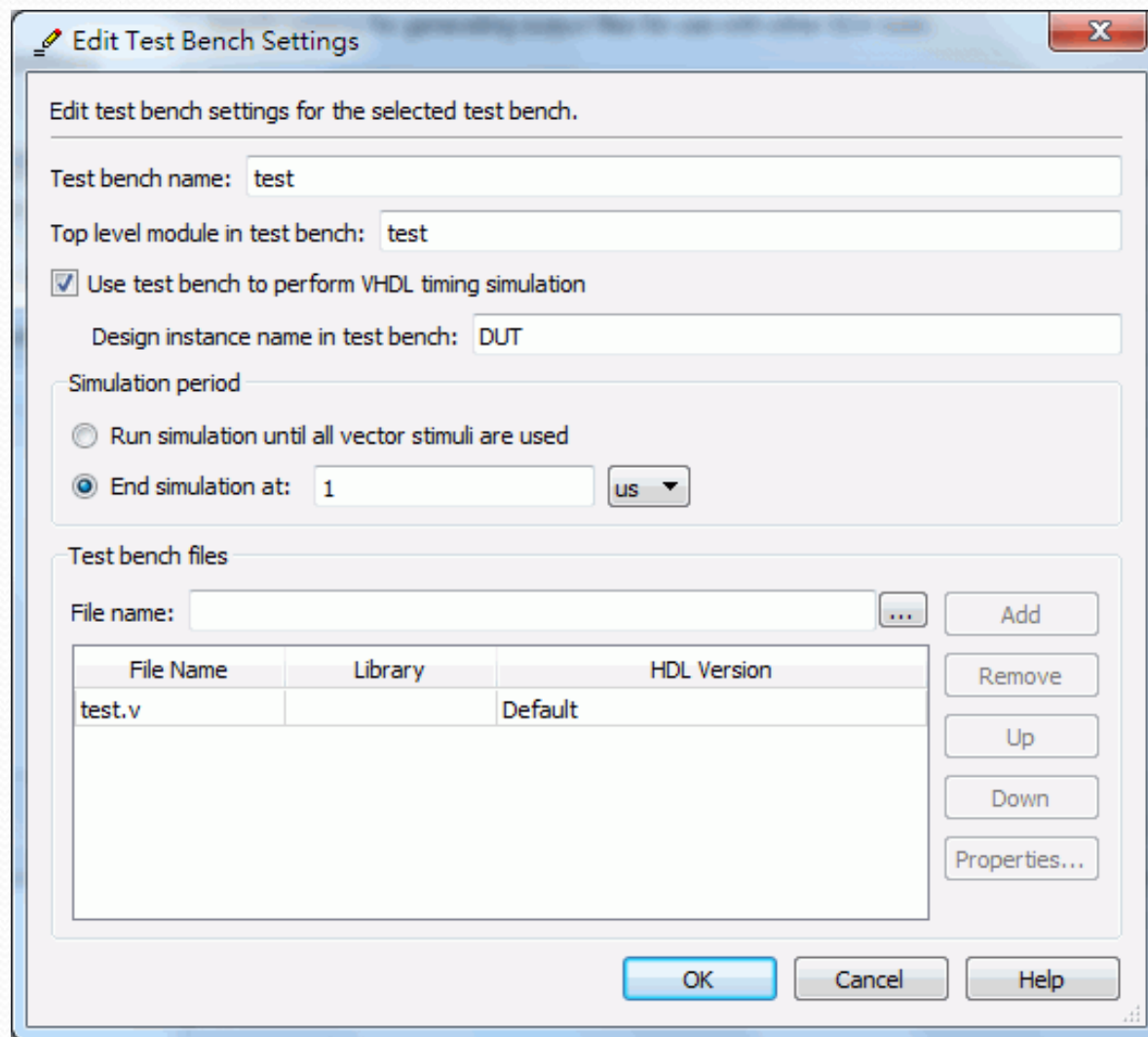




» 圖104 加入檔案

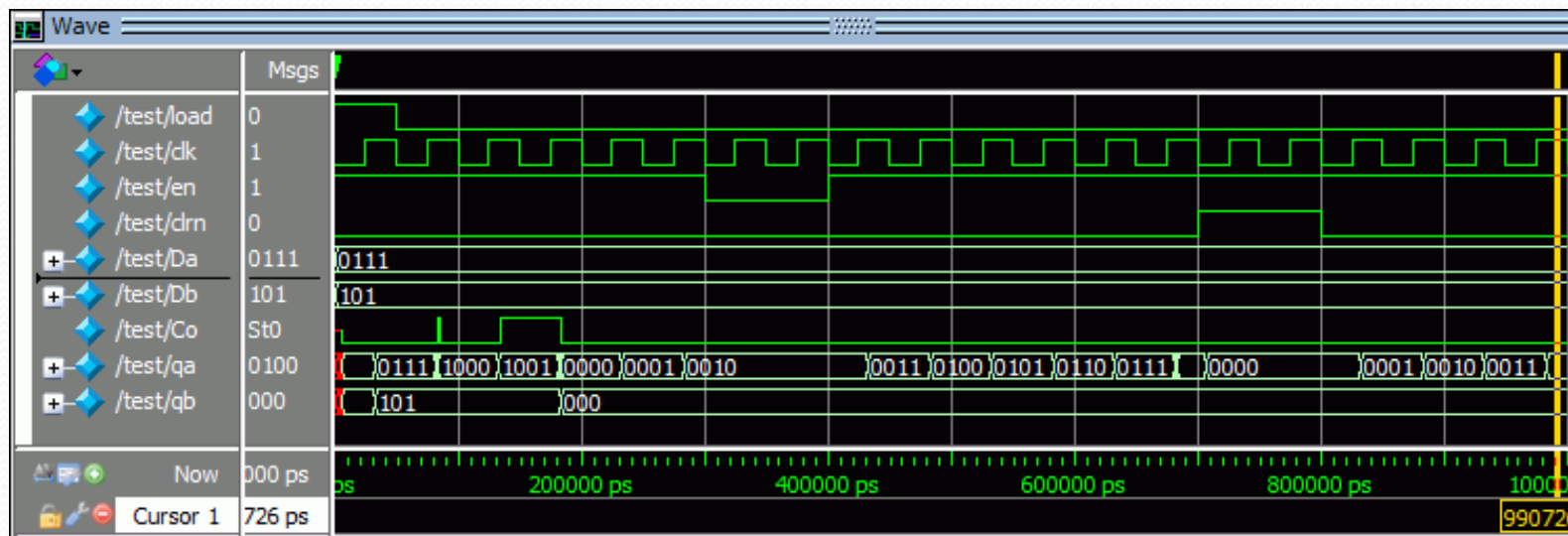
```
`timescale 1ns/10ps
module test;
reg      load,clk,en,clrn;
reg      [3:0] Da;
reg      [2:0] Db;
wire      Co;
wire      [3:0] qa;
wire      [2:0] qb;
counter60 DUT (.load(load), .clk(clk), .en(en), .clrn(clrn),
               .Da(Da), .Db(Db), .Co(Co), .qa(qa), .qb(qb));
```

» 表15 Test Bench 腳位設定



» 圖105 「New Test Benches Settings」視窗





» 圖106 模擬波型

- 檢驗模擬結果：觀察輸出結果，可按Processing→Simulation Report，再點選 'Simulation Waveforms' 觀察模擬波形，檢查結果如下：
  - 25ns處clk為正緣，clrn=0，load=1，載入Da=4'b0111，Db=3'101，輸出qa=Da=4'b0111，qb=Db=3'101。
  - 75ns處clk為正緣，clrn=0，load=0，en=1，輸出增加1，輸出qa變為4'b1000，qb保持3'101。
  - 125ns處clk為正緣，clrn=0，load=0，en=1，輸出增加1，輸出qa變為4'b1001，qb保持3'101。輸出Co=1。
  - 175ns處clk為正緣，clrn=0，load=0，en=1，輸出增加1，輸出qa變為4'b0000，qb變為3'b000。
  - 325ns處clk為正緣，clrn=0，load=0，en=0，故輸出不變。
  - 700ns處，clrn=1，輸出清除為0，qa=4'b0000，qb=3'b000。
- 與表14真值表做比較，對照無誤。



# 作業

- 請完成「非同步清除同步載入60模計數器」
- 主程式可使用Verilog或VHDL
- Test Bench使用Verilog
- 整個專案及Word檔(存放模擬波形截圖)並上傳
- 加分項目：課堂間完成Demo