

# Lab 4 七段顯示器實驗

銘傳大學電腦與通訊工程學系

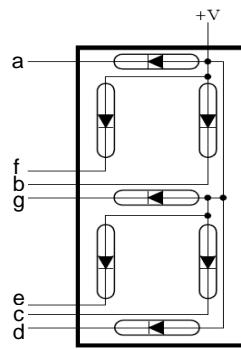
陳慶逸

## 一、背景知識

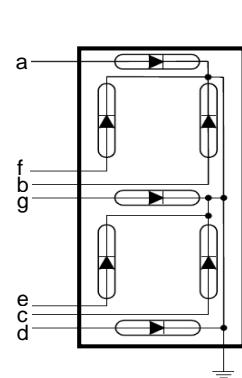
DE2-70 實驗板上設有八顆共陽極的七段顯示器，每一個七段顯示器的內部分別由 8 個 LED 燈組成，構成七個筆畫與一個小數點；當 a~g 接點接低電位(0V)時，便可點亮顯示器中對應的 LED。



(a)外觀



(b)共陽極架構



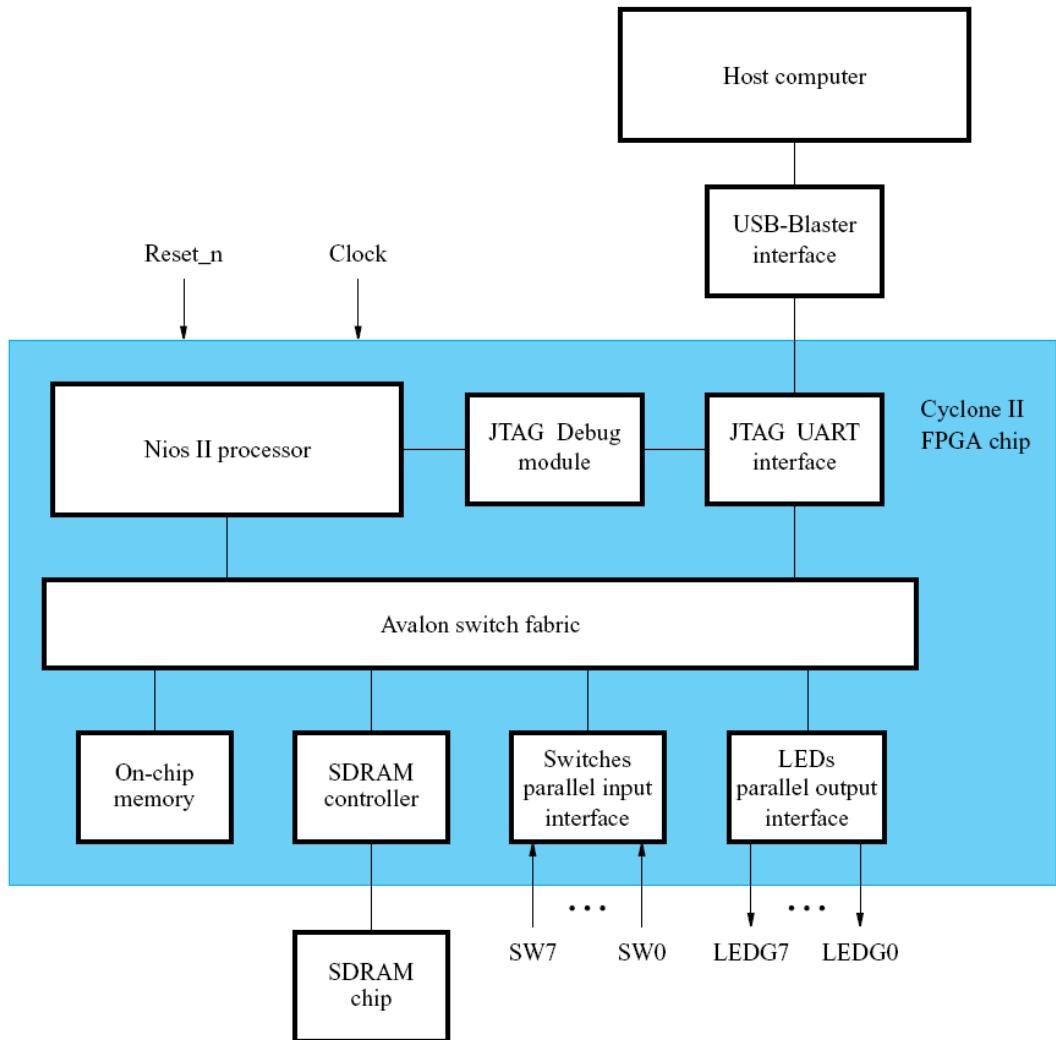
(c)共陰極架構

七段顯示器外觀及內部結構

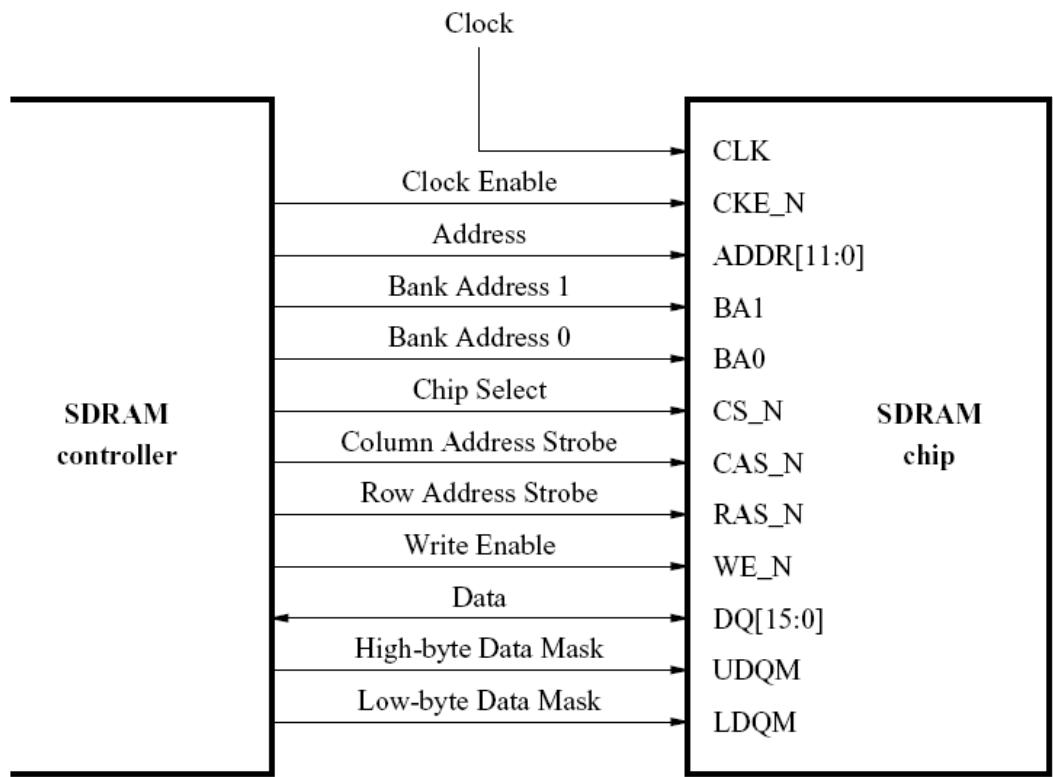
共陽極七段顯示器控制碼

dp	g	f	e	d	c	b	a	十六進位	顯示數字
1	1	0	0	0	0	0	0	C0	0
1	1	1	1	1	0	0	1	F9	1
1	0	1	0	0	1	0	0	A4	2
1	0	1	1	0	0	0	0	B0	3
1	0	0	1	1	0	0	1	99	4
1	0	0	1	0	0	1	0	92	5
1	0	0	0	0	0	1	0	82	6
1	1	1	1	1	0	0	0	F8	7
1	0	0	0	0	0	0	0	80	8
1	0	0	1	0	0	0	0	90	9
1	0	0	0	1	0	0	0	88	A
1	0	0	0	0	0	1	1	83	b
1	1	0	0	0	1	1	0	C6	C
1	0	1	0	0	0	0	1	A1	d
1	0	0	0	0	1	1	0	86	E
1	0	0	0	1	1	1	0	8E	F

- 如下例所示，本次實驗中我們將在SOPC Builder中加入SDRAM 介面到Nios II system之中。

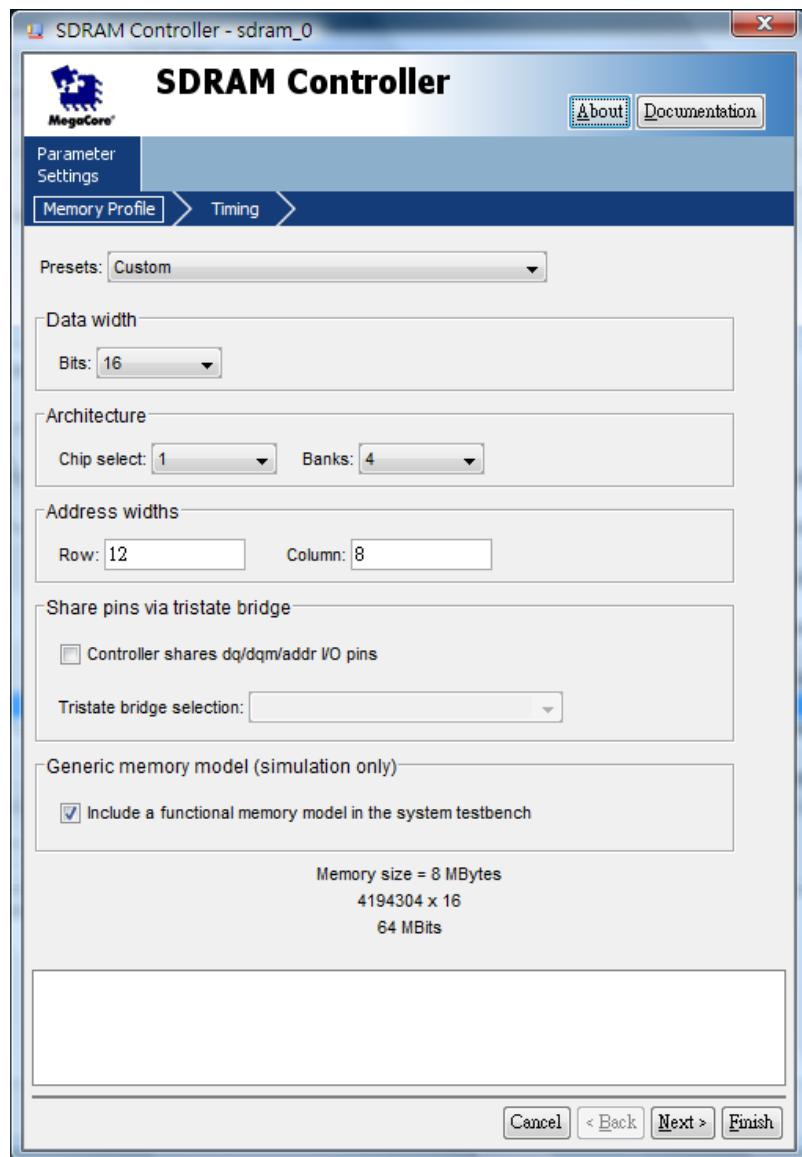


## 2. The SDRAM Interface

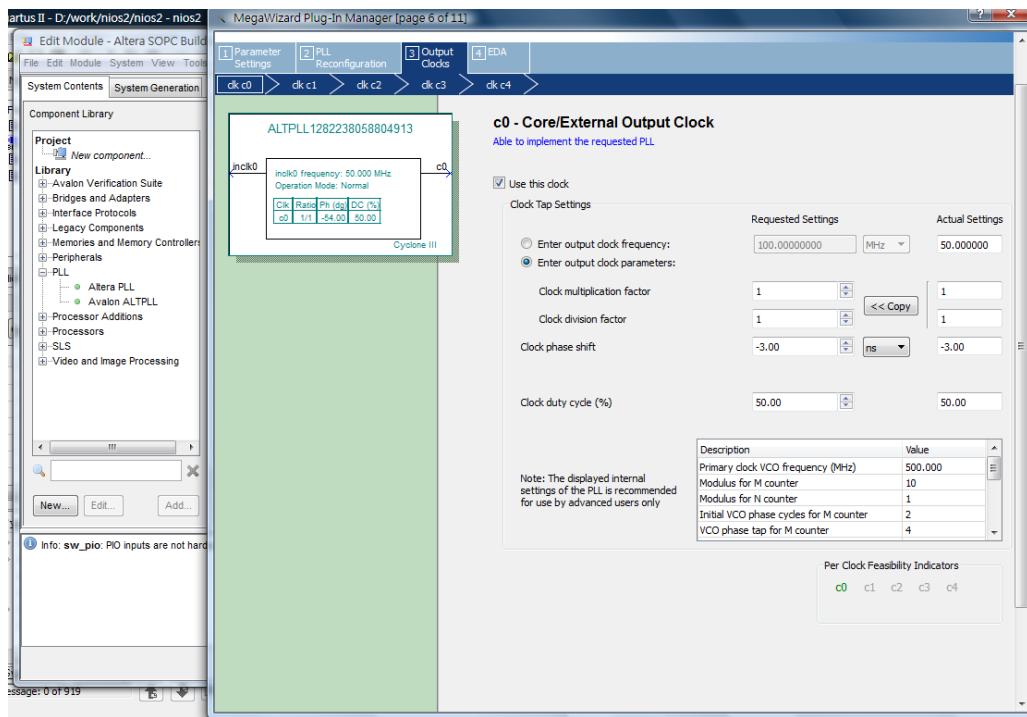


### 3. Using the SOPC Builder to Generate the Nios II System

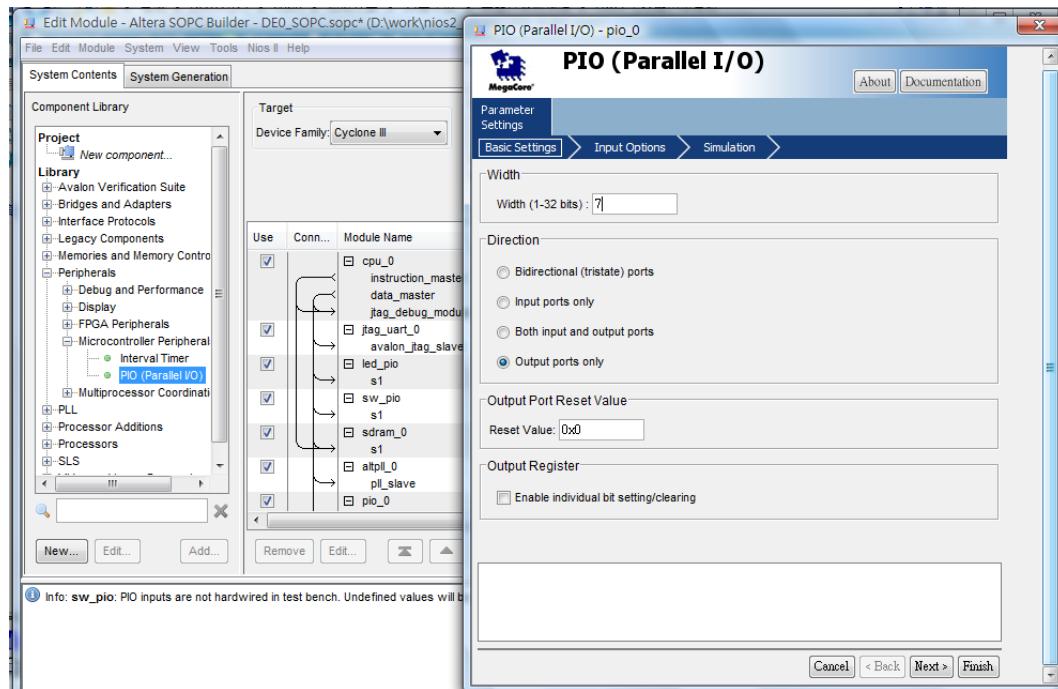
#### (1) SDRAM Controller

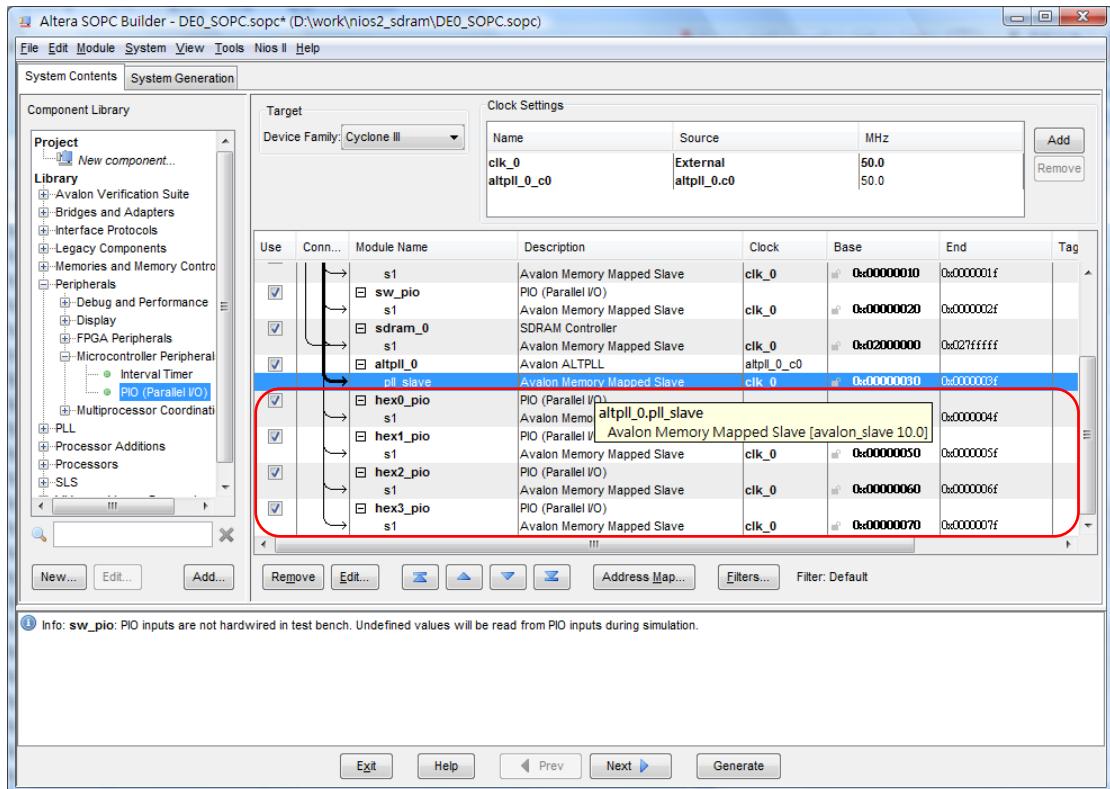


## (2) PLL

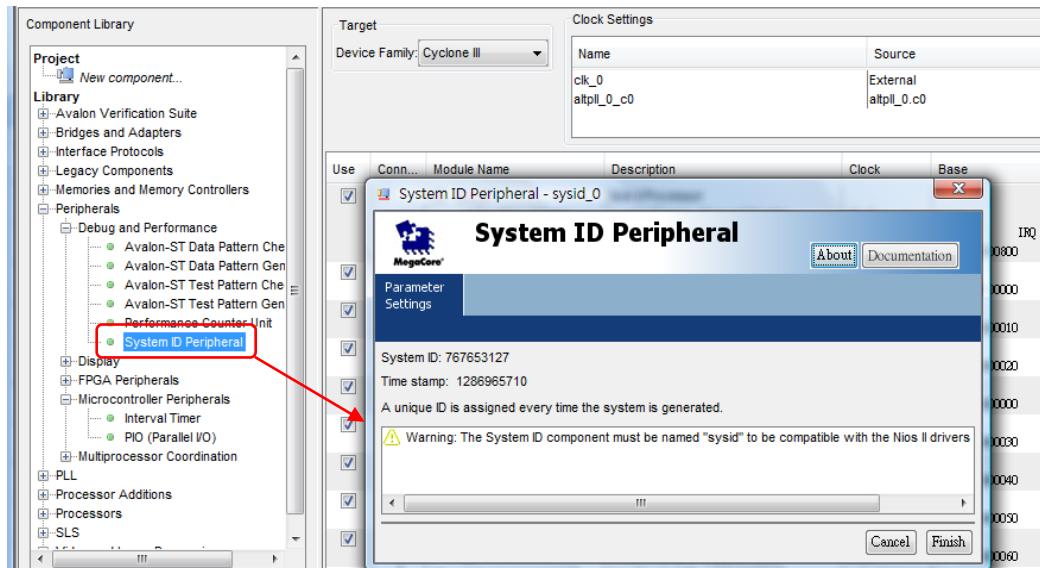


### (3) 增加四組七段顯示器 pio 輸出



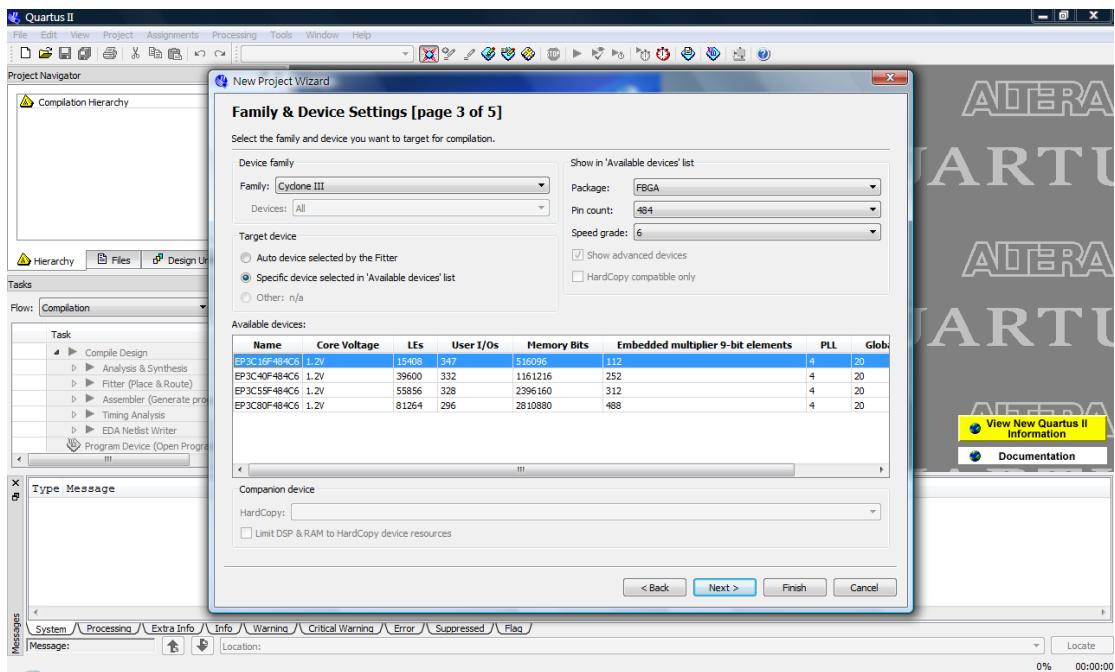
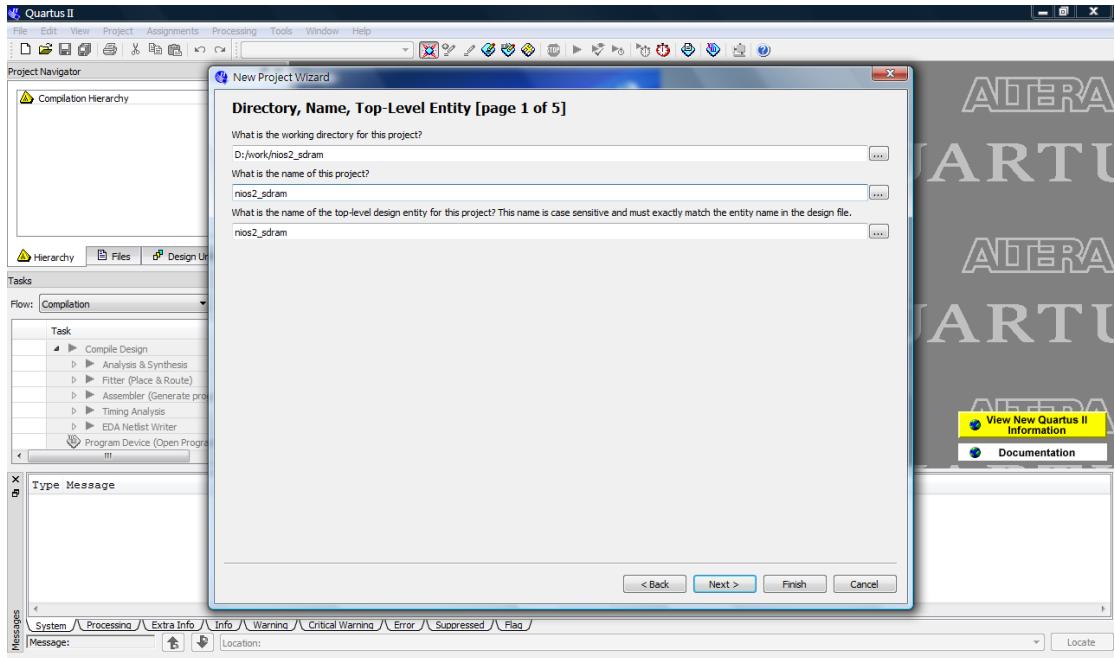


## (4) System ID Peripheral



## 二、硬體設計(Nios II Hardware Development)

### 1. Create New Project (nios2\_sdram)



### 2. Using the SOPC Builder to Generate the Nios II System..

2-1 Choose SOPC Builder (Tools menu), SOPC Builder displays the Create New System dialog box.

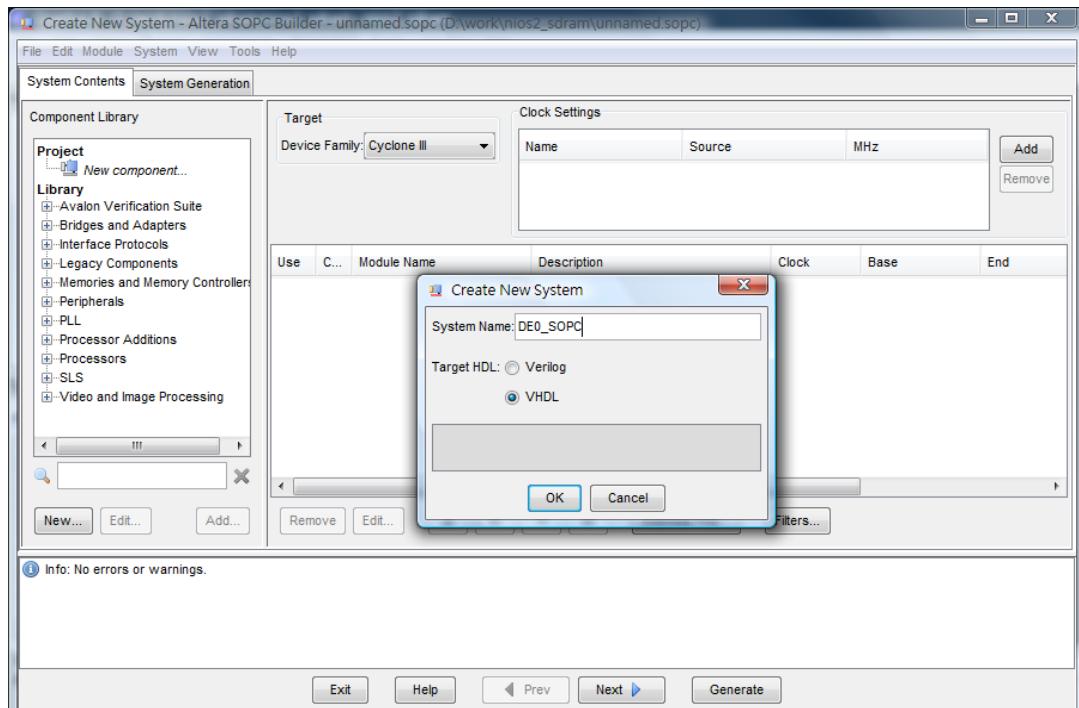


Fig.3 Type **DE0\_SOPC** in the System Name field.

## 2-2 Under Altera SOPC Builder Components category, select **Nios II Processor**.

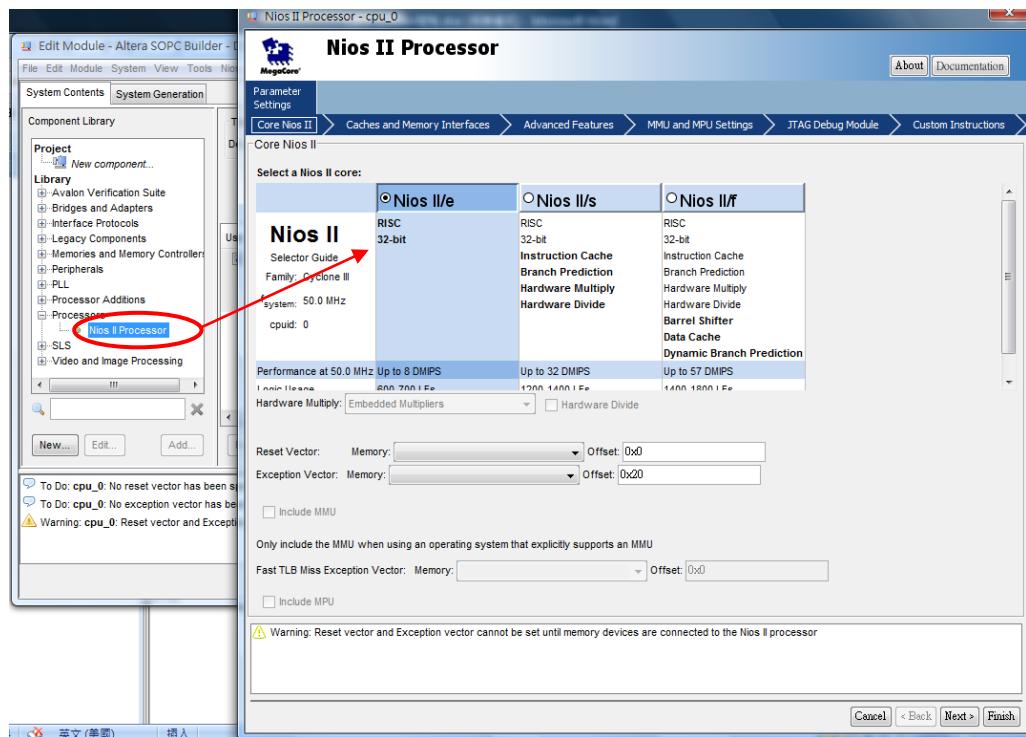


Fig. 4 Nios II Processor

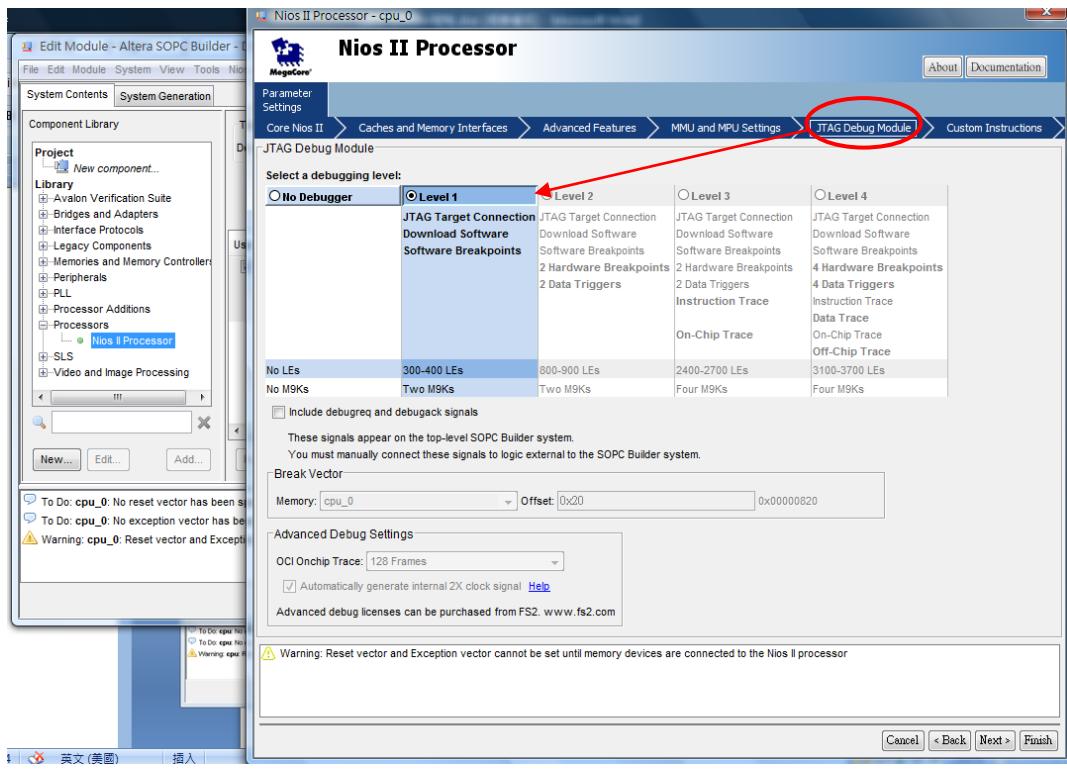
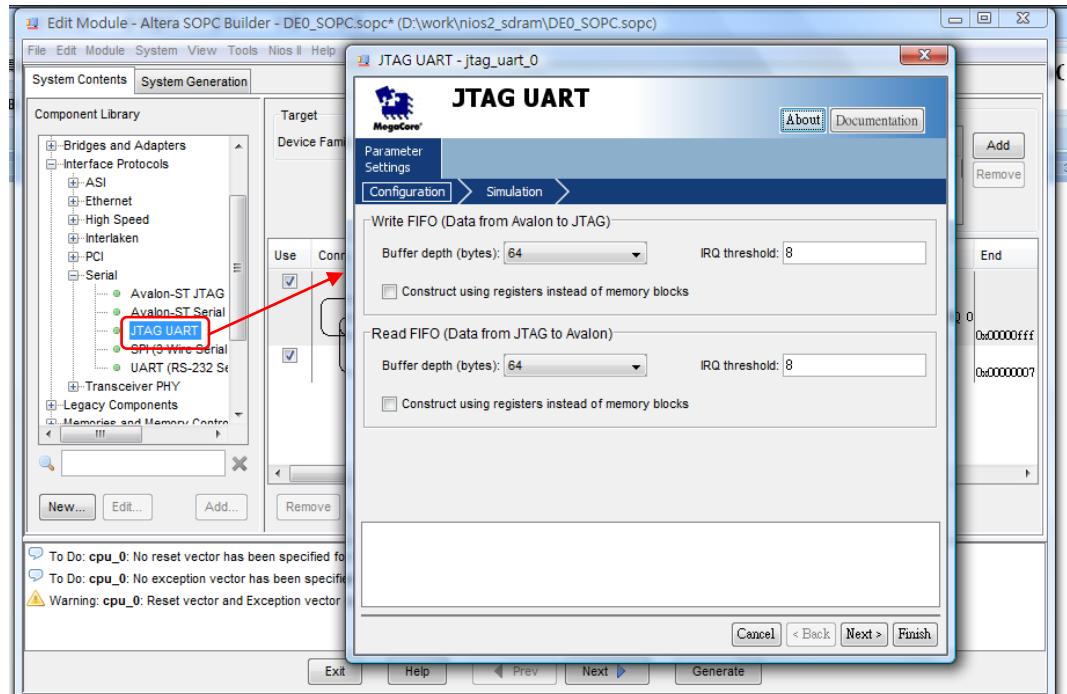


Fig. 5 Click the **JTAG Debug Module** tab and choose the “Level 1” debugging level.

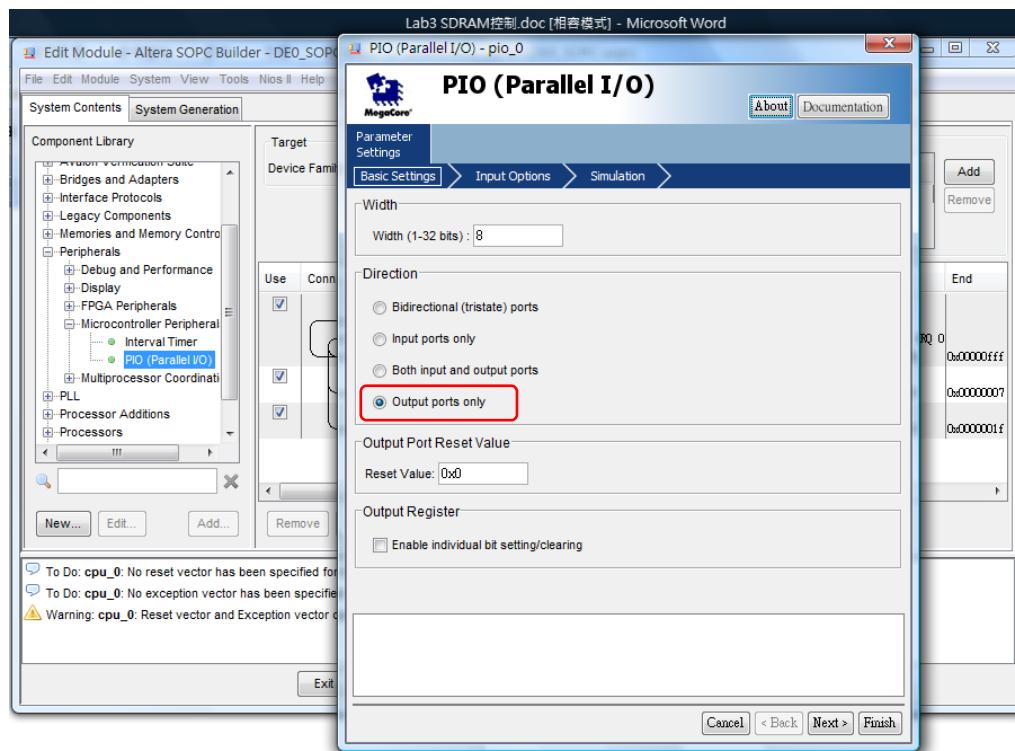
2-3 Select **JTAG UART** under the **Altera SOPC Builder > Interface Protocols > Serial** category. Add JTAG\_UART.



Add JTAG UART

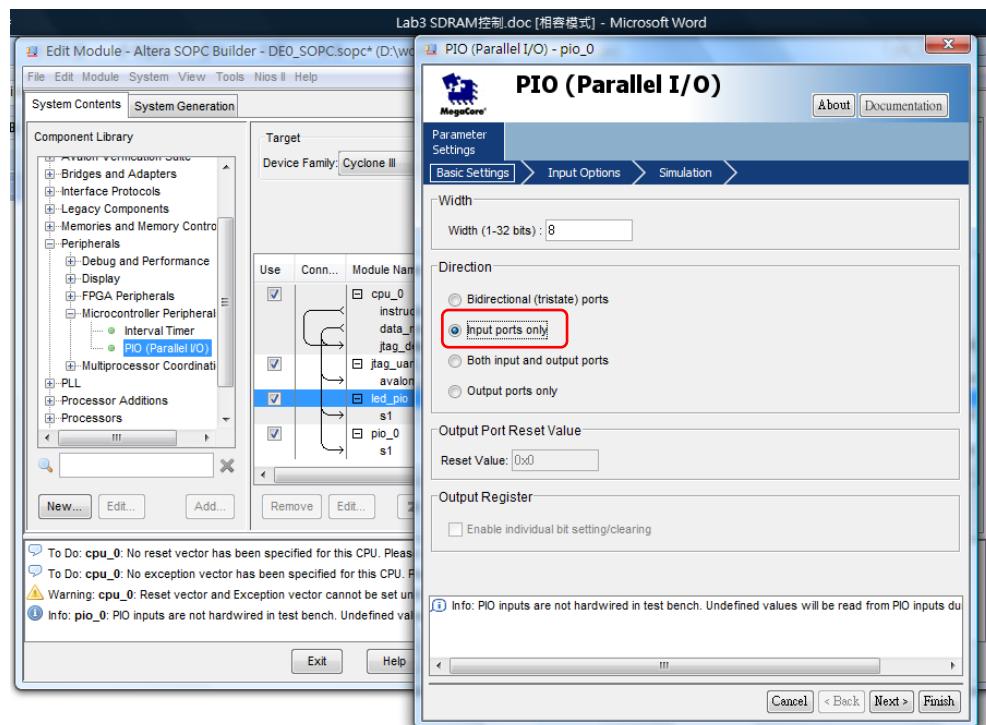
2-4 Select **PIO (Parallel I/O)** under the **Altera SOPC Builder> Peripherals >**

Microcontroller Peripherals category. Add Output PIO. Right-click **pio** and select Rename. Type **led\_pio** and press Enter.

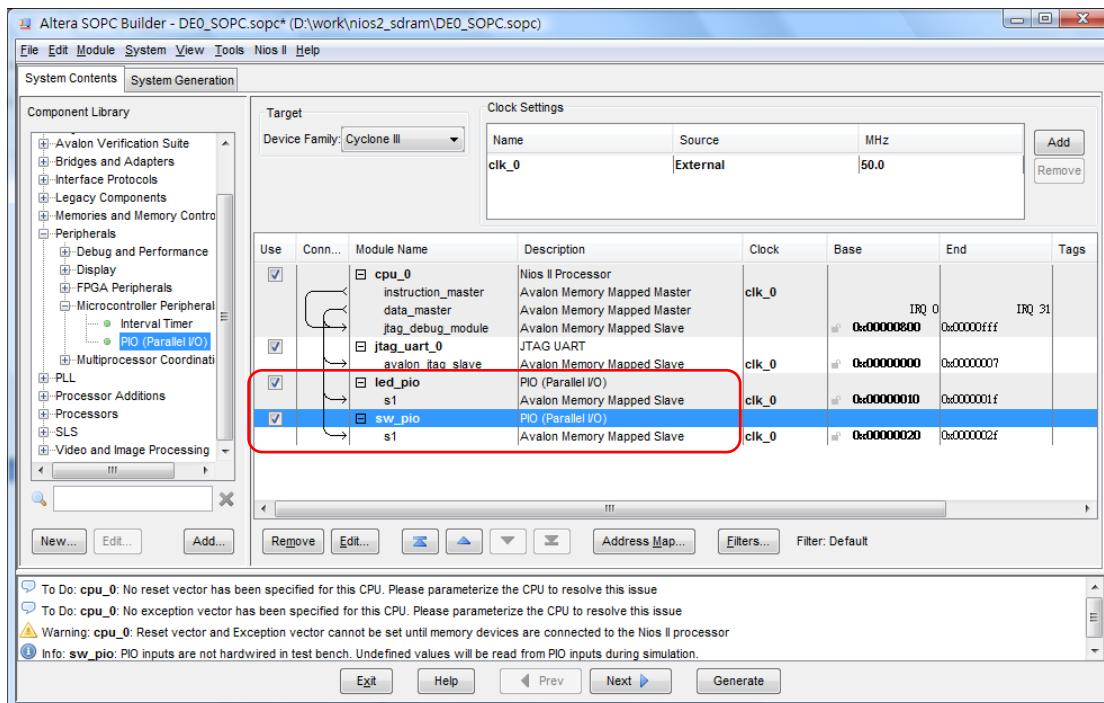


Add Output PIO (Rename “led\_pio”)

Add Input PIO. Right-click **pio** and select Rename. Type **sw\_pio** and press Enter.



## Add Input PIO (Rename “sw\_pio”)



2-5 Select **SDRAM** under the **Altera SOPC Builder > Memory and Memory Controllers category > SDRAM> SDRAM Controller.**

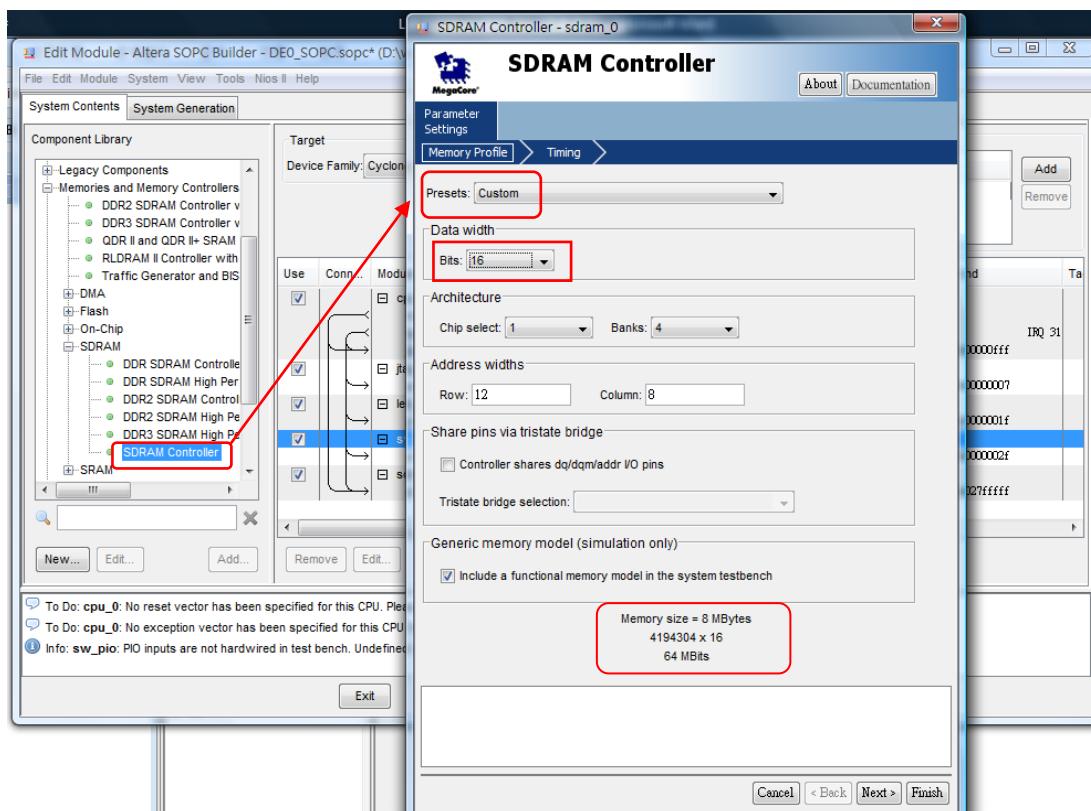


Fig.6 Add SDRAM Controller

## 2-6 Modify the Reset Vector & Exception Vector of the NIOS II Processor.

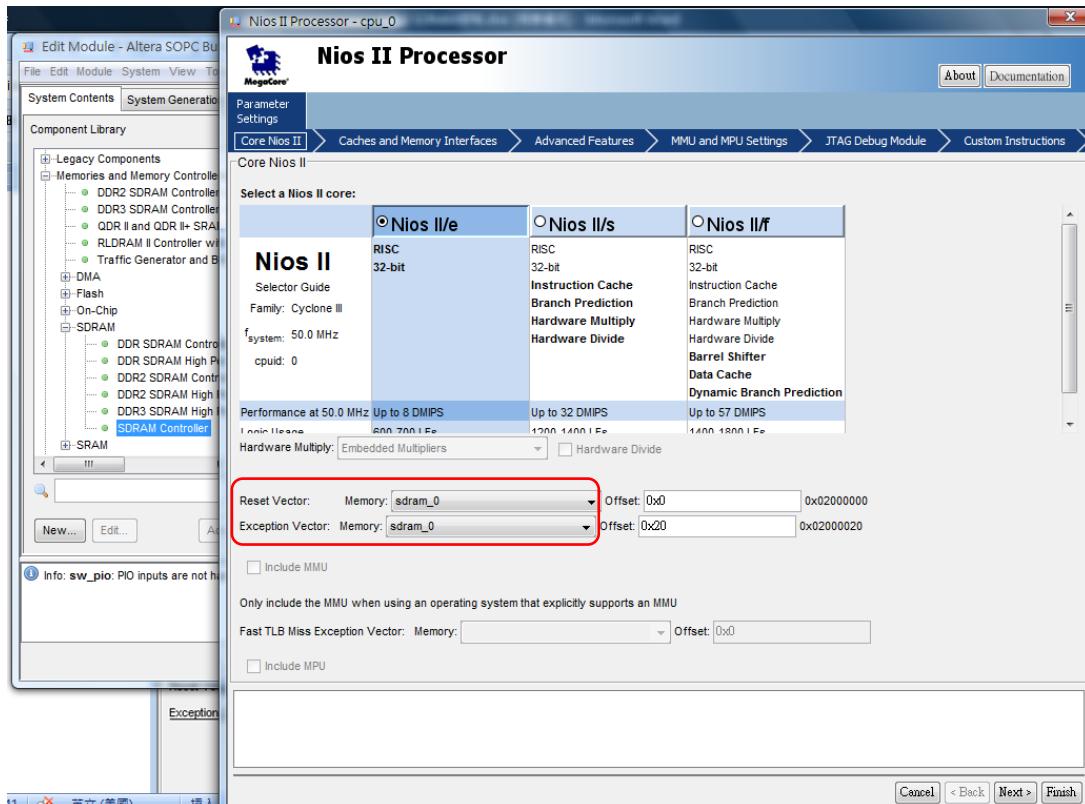
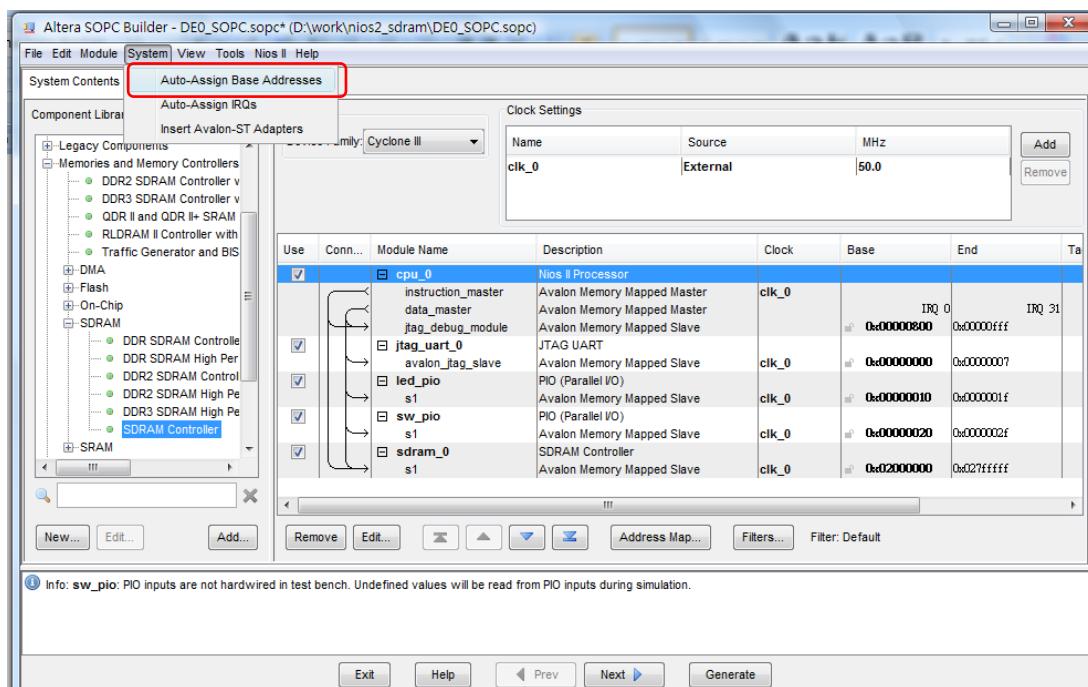


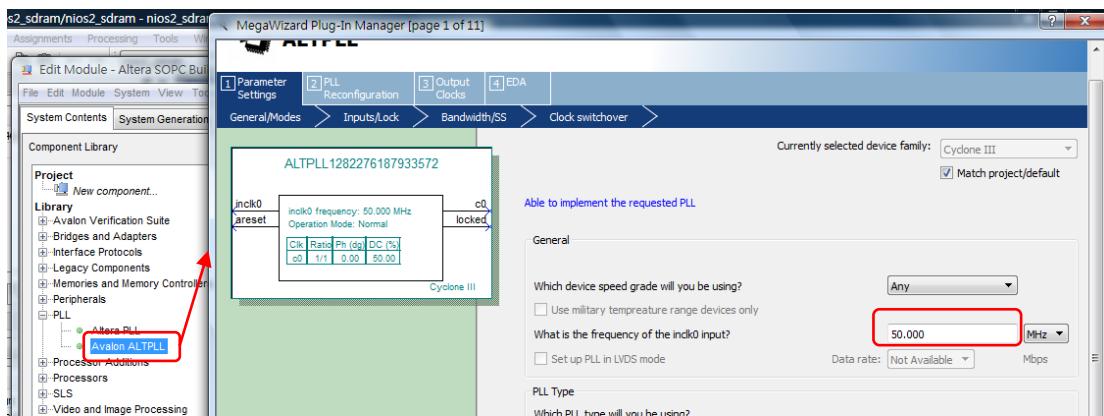
Fig.7 Modify the Reset Vector & Exception Vector

註：若發生位址衝突的錯誤訊息，可執行 System > Auto-Assign Base Address

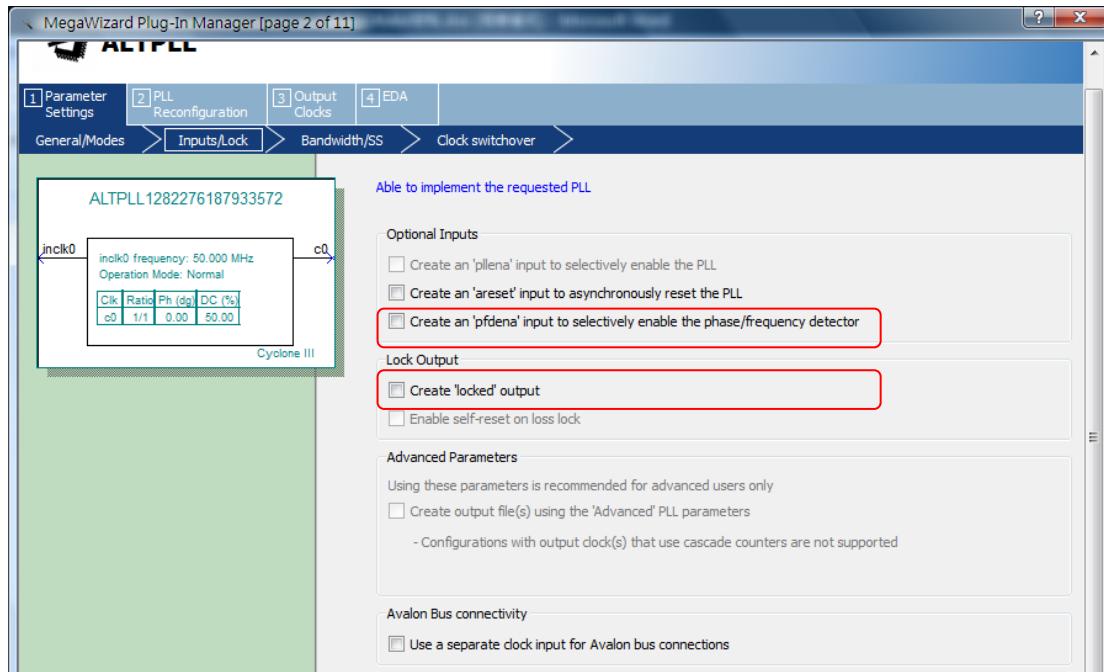


2-7 Select ALTPLL under the Altera SOPC Builder > PLL > Avalon ALTPPLL.

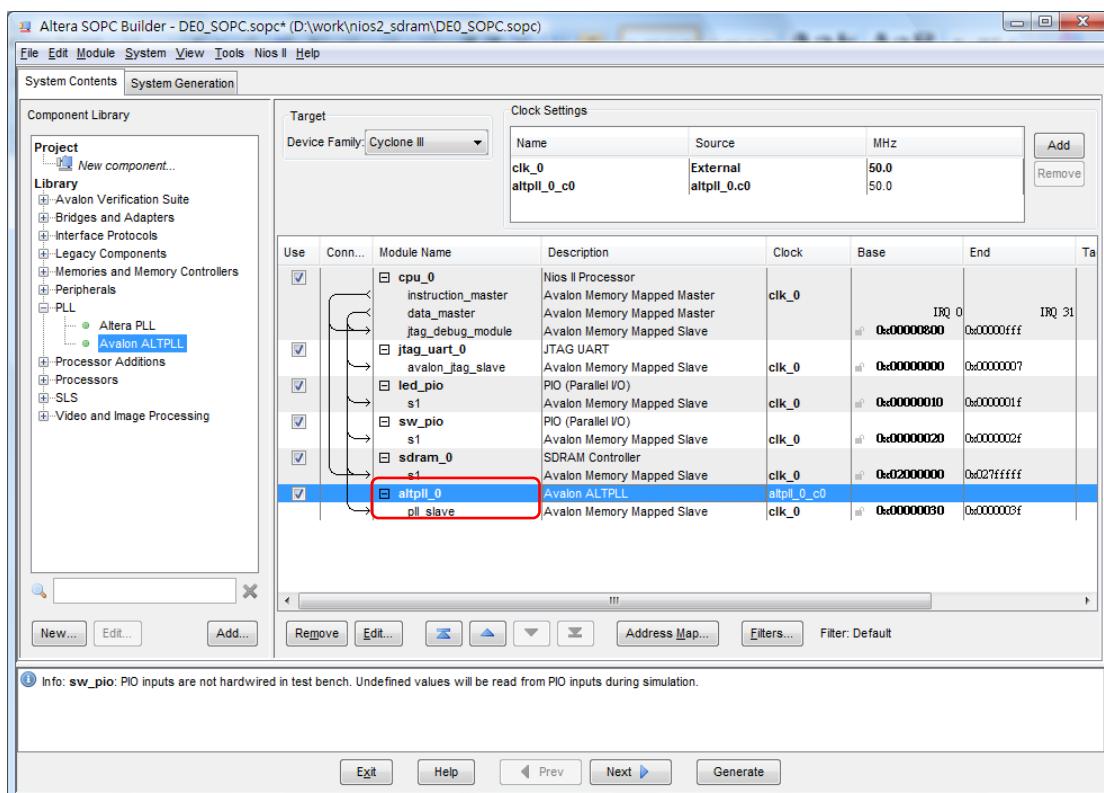
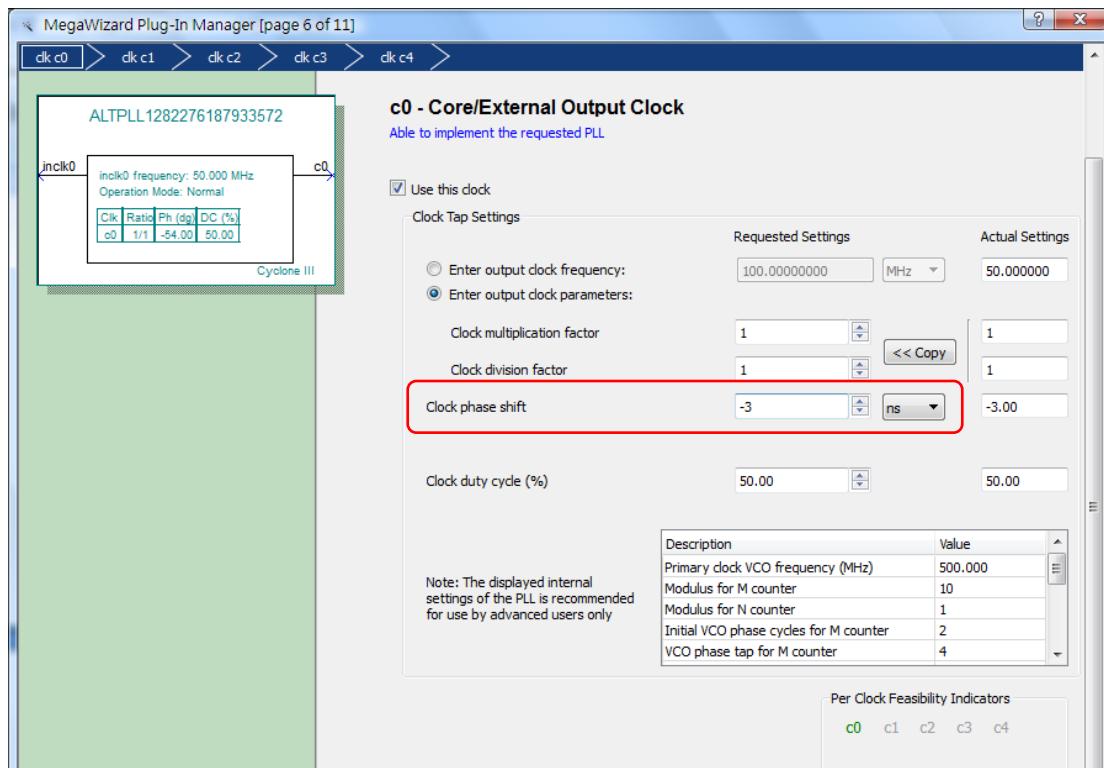
## 2-7-1 Select the frequency of the inclock0 input.



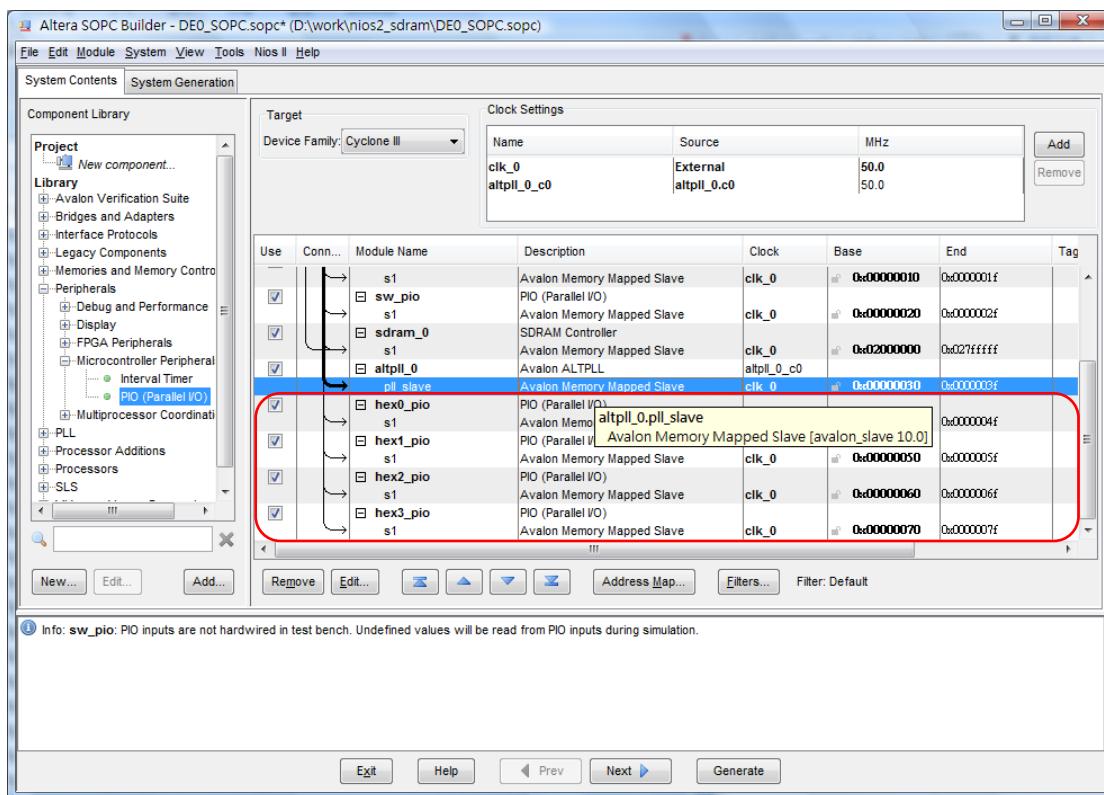
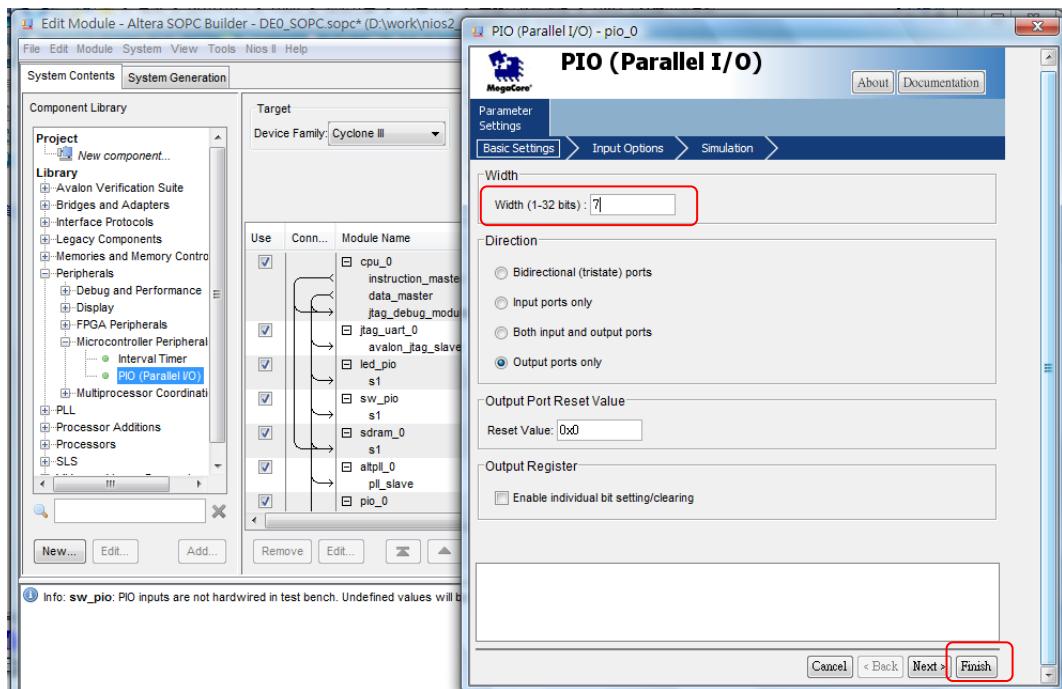
## 2-7-2 Optional inputs



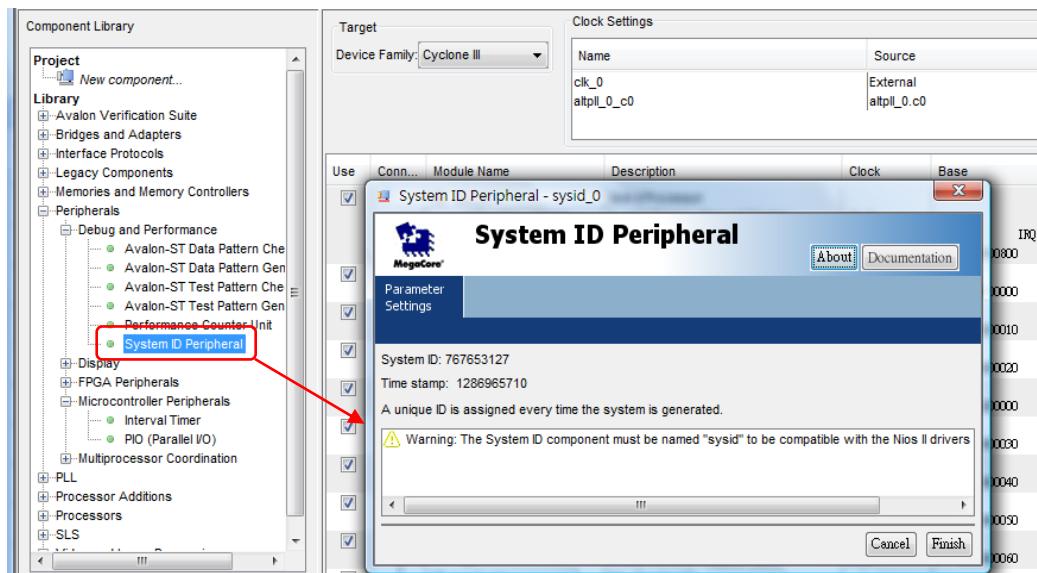
## 2-7-3 Clock Tap Settings (Clock phase shift: -3ns)



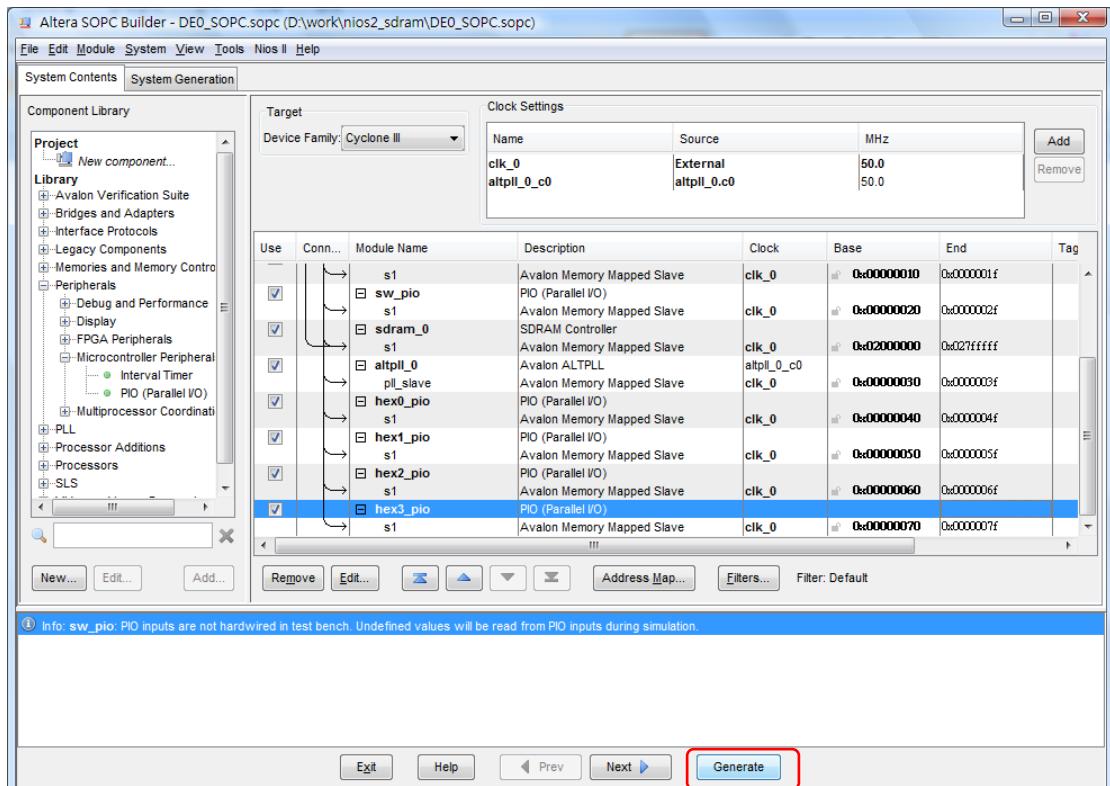
2-8 Add 4 output PIOs. Right-click **pio** and select Rename, respectively. Type hex0\_pio, hex1\_pio, hex2\_pio and hex3\_pio, and press Enter.



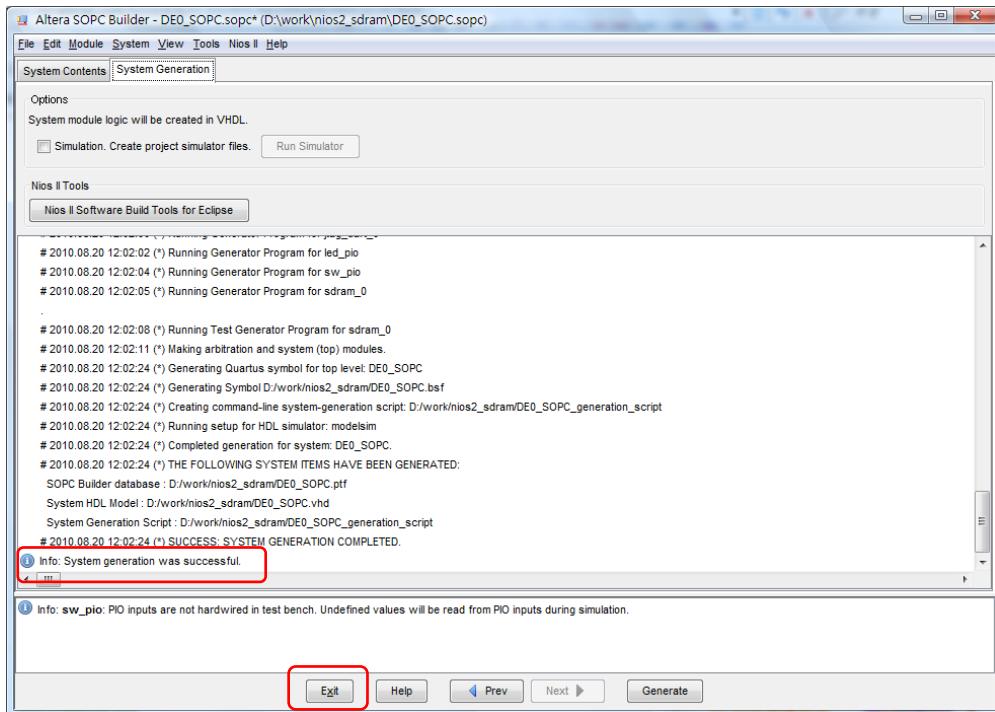
## 2-9 Add System ID Peripheral



**2-10 Click the **Generate** tab. The system generation process begins. The generation process can take several minutes. When it completes, the System Generation tab displays a message "**SUCCESS: SYSTEM GENERATION COMPLETED**".**



**Fig. 13 Click **Generate****



### 3. Integration of the Nios II System into the Quartus II Project

#### 3-1 Adding the Quartus II Symbol to the BDF:

##### 3-1-1 File>New>Block Diagram/Schematic File

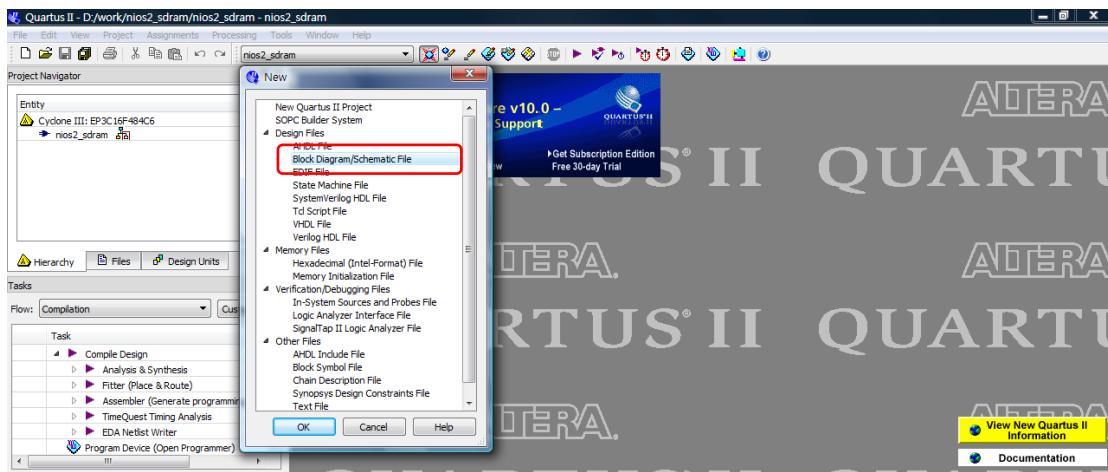


Fig.14 File>New>Block Diagram/Schematic File

##### 3-1-2 Select Project >DE0\_SOPC. The Symbol dialog box displays the DE0\_SOPC symbol. Click OK.

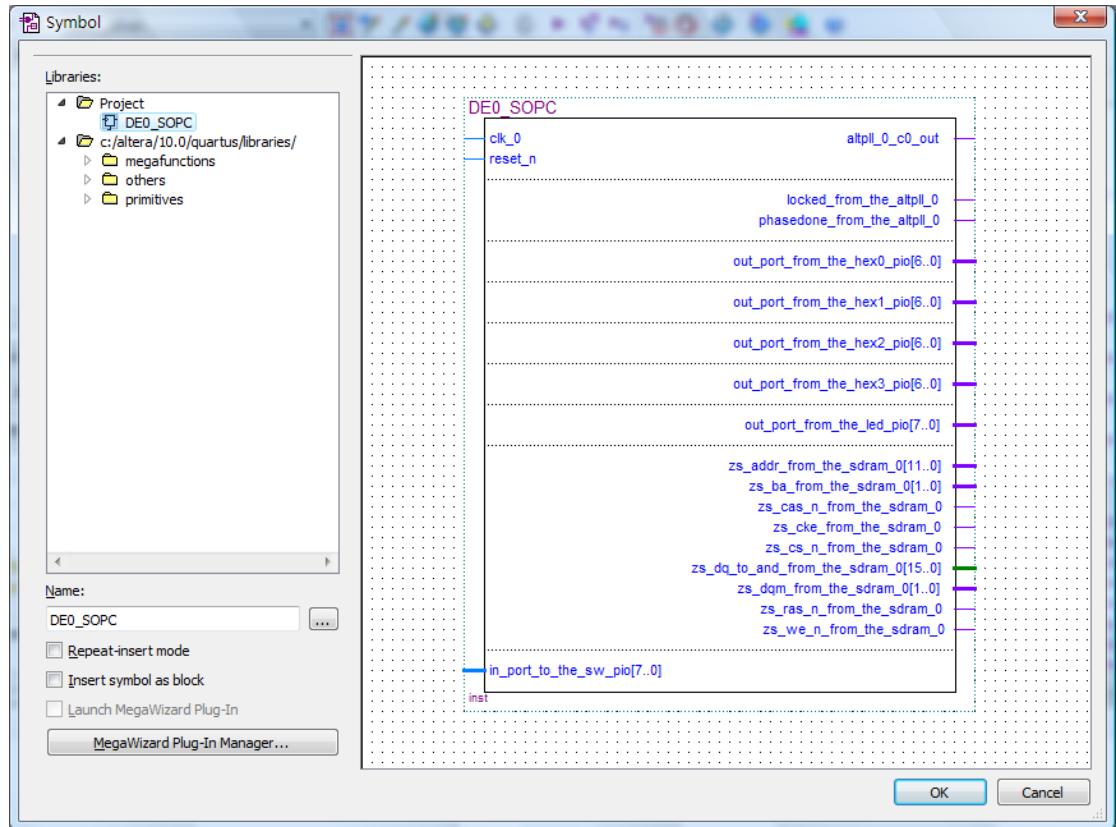
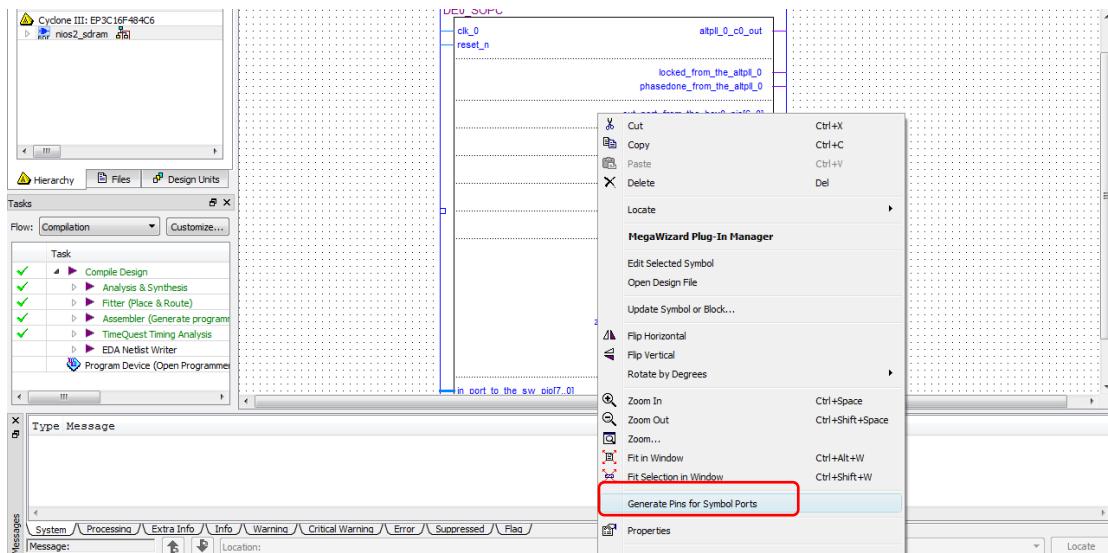


Fig.15 Add DE0\_SOPC symbol

3-1-3 Right-click **DE0\_SOPC** symbol and select “Generate pins for Symbol ports”:



3-1-4 Rename Symbol ports:

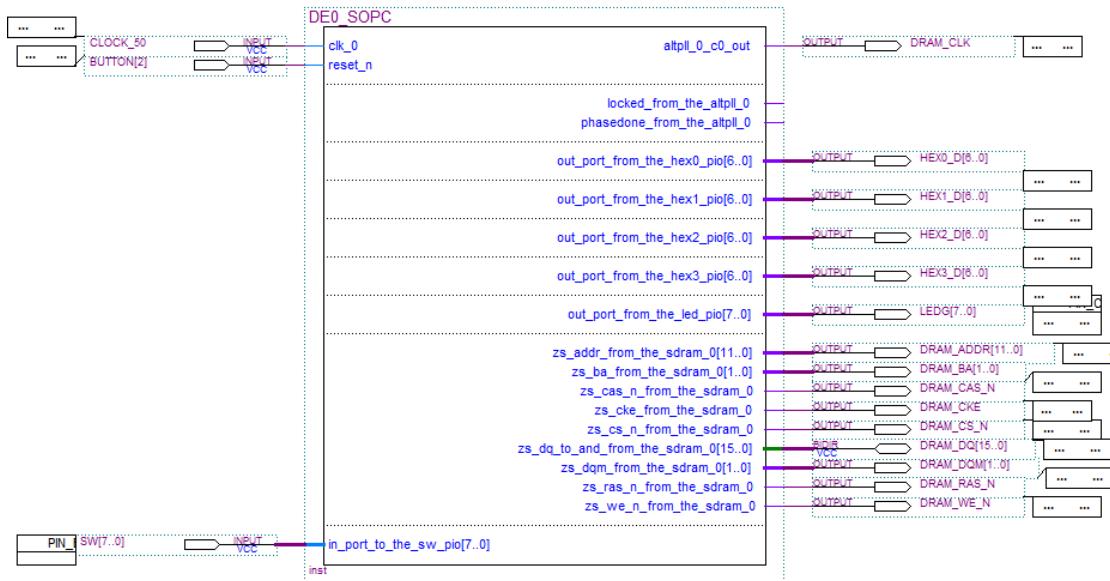


Fig.17 Rename Symbol ports

### 3-2 Compilier the Design.

## 4. Pin Assignments & Download

### 4-1 Choose Assignments > Import Assignments

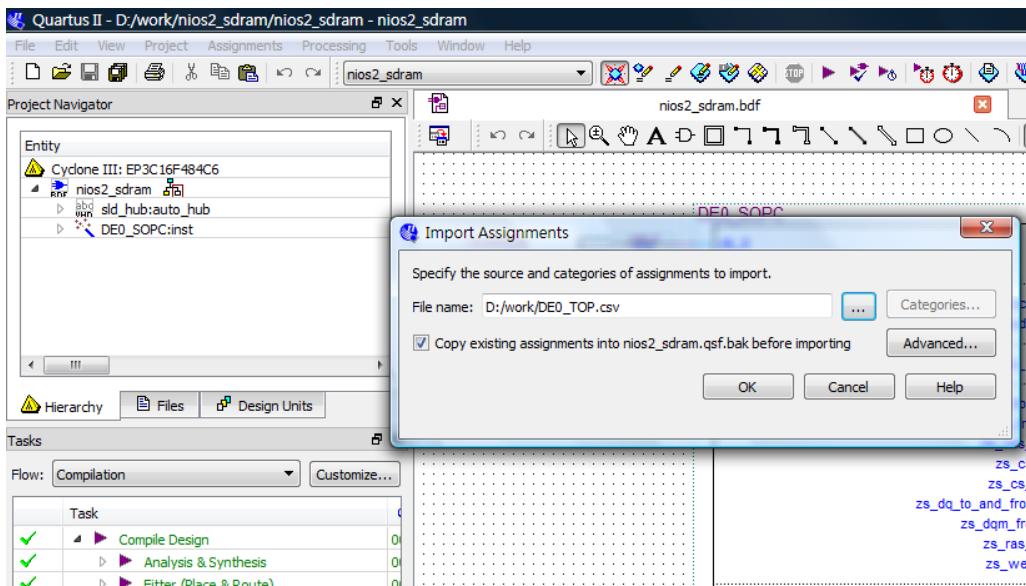


Fig.20 Specify assignment source

### 4-2 Modify the pin assignments of DRAM\_BA[1..0] & DRAM\_DQM[1..0]

**Delete**

HEX3_D[3]	Unknown	PIN_B5	8	B7_N0	3.3-
HEX3_D[2]	Unknown	PIN_A4	8	B8_N1	3.3-
HEX3_D[1]	Unknown	PIN_E7	8	B8_N1	3.3-
HEX3_D[0]	Unknown	PIN_B8	8	B8_N0	3.3-
HEX3_DP	Unknown			B7_N0	3.3-
DRAM_BA_0	Output	PIN_B5	8	B8_N1	3.3-
DRAM_BA_1	Output	PIN_A4	8	B8_N1	3.3-
DRAM_LDQM	Unknown	PIN_E7	8	B8_N1	3.3-
DRAM_UDQM	Unknown	PIN_B8	8	B8_N0	3.3-
CLOCK_50_2	Unknown	PIN_B12	7	B7_N1	3.3-
DRAM_ADDR[12]	Unknown	PIN_C8	8	B8_N0	3.3-
<<new node>>					

**Assignment pins:**

DRAM\_BA[1] -> PIN\_A4

DRAM\_BA[0] -> PIN\_B5

**Assignment pins:**

DRAM\_DQM[1] -> PIN\_B8

DRAM\_DQM[0] -> PIN\_E7

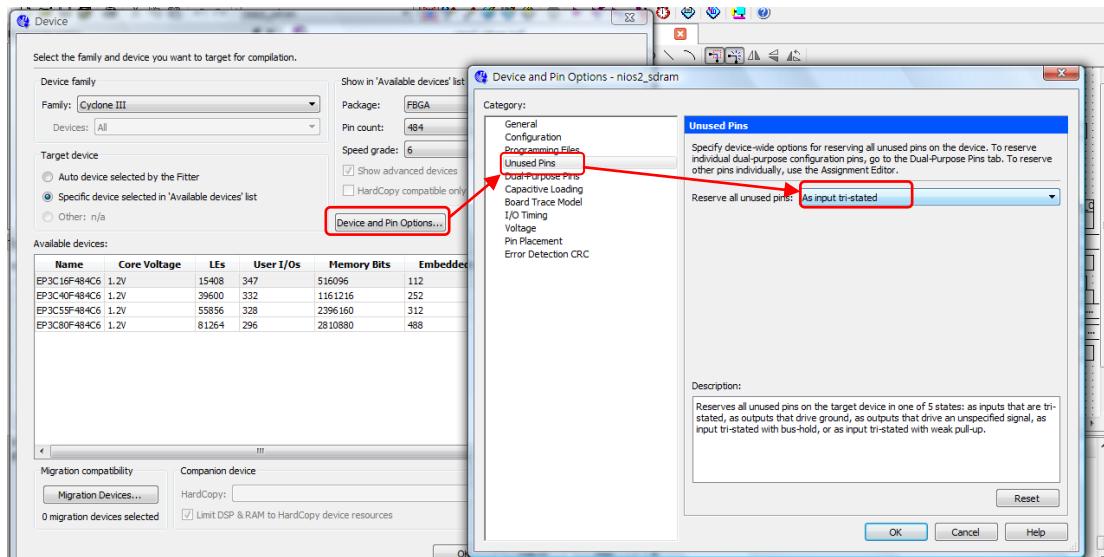
Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved
DRAM_ADDR[1]	Output	PIN_A3	8	B8_N1	3.3-V LVTTL	
DRAM_ADDR[0]	Output	PIN_C4	8	B8_N1	3.3-V LVTTL	
DRAM_BA[1]	Output				2.5 V (default)	
DRAM_BA[0]	Output				2.5 V (default)	
DRAM_CS_N	Output	PIN_G8	8	B8_N1	3.3-V LVTTL	
DRAM_CKE	Output	PIN_E6	8			
DRAM_CLK	Output	PIN_E5	8			
DRAM_CS_N	Output	PIN_G7	8			
DRAM_DQ[15]	Bidir	PIN_F10	8			
DRAM_DQ[14]	Bidir	PIN_E10	8			
DRAM_DQ[13]	Bidir	PIN_A10	8			
DRAM_DQ[12]	Bidir	PIN_B10	8			
DRAM_DQ[11]	Bidir	PIN_C10	8			
DRAM_DQ[10]	Bidir	PIN_A9	8			
DRAM_DQ[9]	Bidir	PIN_B9	8			
DRAM_DQ[8]	Bidir	PIN_A8	8			
DRAM_DQ[7]	Bidir	PIN_F8	8			
DRAM_DQ[6]	Bidir	PIN_H9	8			
DRAM_DQ[5]	Bidir	PIN_G9	8			
DRAM_DQ[4]	Bidir	PIN_F9	8			
DRAM_DQ[3]	Bidir	PIN_E9	8			
DRAM_DQ[2]	Bidir	PIN_H10	8			
DRAM_DQ[1]	Bidir	PIN_G10	8			
DRAM_DQ[0]	Bidir	PIN_D10	8			
DRAM_DQM[1]	Output				3.3-V LVTTL	
DRAM_DQM[0]	Output				3.3-V LVTTL	
DRAM_RAS_N	Output	PIN_F7	8	B8_N1	3.3-V LVTTL	
DRAM_WE_N	Output	PIN_D6	8	B8_N1	3.3-V LVTTL	

DRAM_ADDR[3]	Output	PIN_C3	8	B8_N1	3.3-V LVTTL
DRAM_ADDR[2]	Output	PIN_B3	8	B8_N1	3.3-V LVTTL
DRAM_ADDR[1]	Output	PIN_A3	8	B8_N1	3.3-V LVTTL
DRAM_ADDR[0]	Output	PIN_C4	8	B8_N1	3.3-V LVTTL
DRAM_BA[1]	Output	PIN_B5	8	B8_N1	3.3-V LVTTL
DRAM_BA[0]	Output	PIN_A4	8	B8_N1	3.3-V LVTTL
DRAM_CS_N	Output	PIN_G8	8	B8_N1	3.3-V LVTTL
DRAM_CKE	Output	PIN_E6	8	B8_N1	3.3-V LVTTL
DRAM_CLK	Output	PIN_E5	8	B8_N1	3.3-V LVTTL
DRAM_CS_N	Output	PIN_G7	8	B8_N1	3.3-V LVTTL
DRAM_DQ[15]	Bidir	PIN_F10	8	B8_N1	3.3-V LVTTL
DRAM_DQ[14]	Bidir	PIN_E10	8	B8_N0	3.3-V LVTTL
DRAM_DQ[13]	Bidir	PIN_A10	8	B8_N0	3.3-V LVTTL
DRAM_DQ[12]	Bidir	PIN_B10	8	B8_N0	3.3-V LVTTL
DRAM_DQ[11]	Bidir	PIN_C10	8	B8_N0	3.3-V LVTTL
DRAM_DQ[10]	Bidir	PIN_A9	8	B8_N0	3.3-V LVTTL
DRAM_DQ[9]	Bidir	PIN_B9	8	B8_N0	3.3-V LVTTL
DRAM_DQ[8]	Bidir	PIN_A8	8	B8_N0	3.3-V LVTTL
DRAM_DQ[7]	Bidir	PIN_F8	8	B8_N1	3.3-V LVTTL
DRAM_DQ[6]	Bidir	PIN_H9	8	B8_N0	3.3-V LVTTL
DRAM_DQ[5]	Bidir	PIN_G9	8	B8_N0	3.3-V LVTTL
DRAM_DQ[4]	Bidir	PIN_F9	8	B8_N1	3.3-V LVTTL
DRAM_DQ[3]	Bidir	PIN_E9	8	B8_N0	3.3-V LVTTL
DRAM_DQ[2]	Bidir	PIN_H10	8	B8_N0	3.3-V LVTTL
DRAM_DQ[1]	Bidir	PIN_G10	8	B8_N0	3.3-V LVTTL
DRAM_DQ[0]	Bidir	PIN_D10	8	B8_N0	3.3-V LVTTL
DRAM_DQM[1]	Output	PIN_B8	8	B8_N1	3.3-V LVTTL
DRAM_DQM[0]	Output	PIN_E7	8	B8_N1	3.3-V LVTTL
DRAM_RAS_N	Output	PIN_F7	8	B8_N1	3.3-V LVTTL
DRAM_WE_N	Output	PIN_D6	8	B8_N1	3.3-V LVTTL

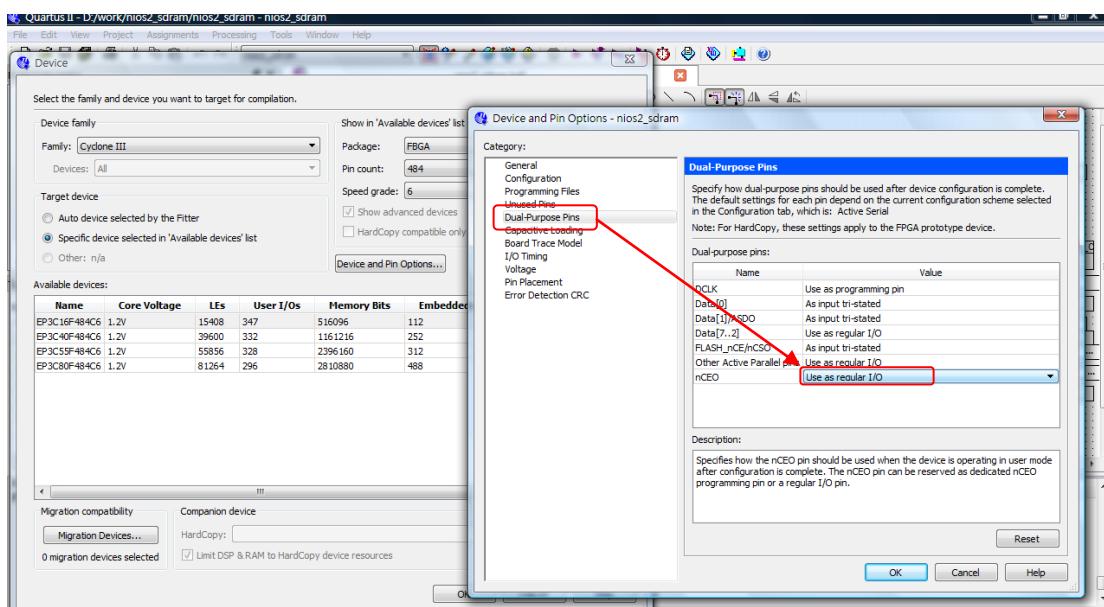
## 4-3 Compiler the design.

### 4-3-1 Select Assignments > Device > Device and Pin Options

(在 Quartus II 編譯 project 前，先選擇【Assignments】【Device】開啟 Settings 視窗以進行 Unused Pins 的設定，並將未使用的腳位都設定為高阻抗輸入。)



Setting Unused Pins



Setting Dual-Purpose Pins

### 4-3-2 Start Compilation and download the hardware

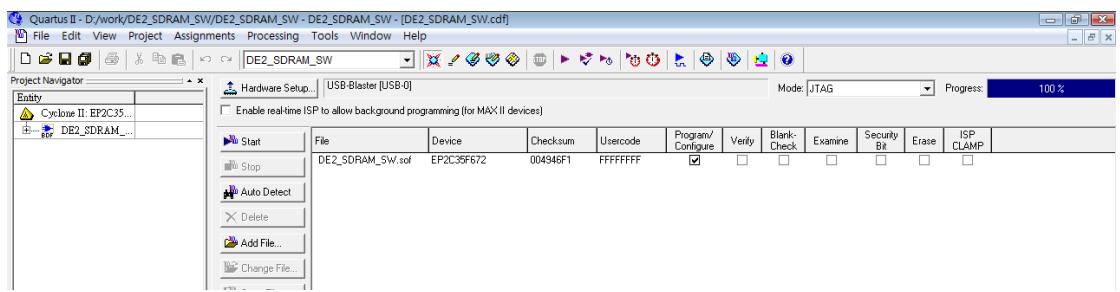


Fig.22 download the hardware

### 三、軟體設計

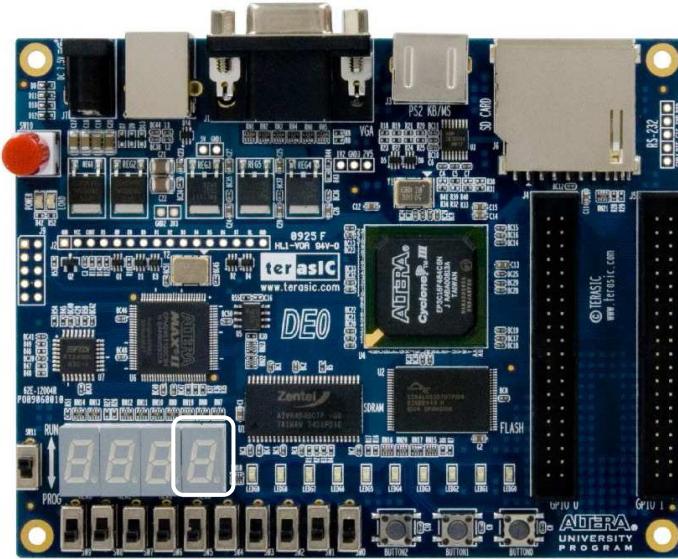
#### NIOS II C/C++ Applications

```
#include <stdio.h>
#include <io.h>
#include "system.h"
#include "unistd.h"

unsigned char i;
unsigned char
seven_seg[10]={0XC0,0XF9,0XA4,0XB0,0X99,0X92,0X82,0XF8,0X80,0X90};

int main()
{
    while (1)          /* Loop forever */
    {
        for (i=0;i<=9;i++)
        {
            IOWR(HEX0_PIO_BASE,0,seven_seg[i]);
            usleep(100000);
        }
    }

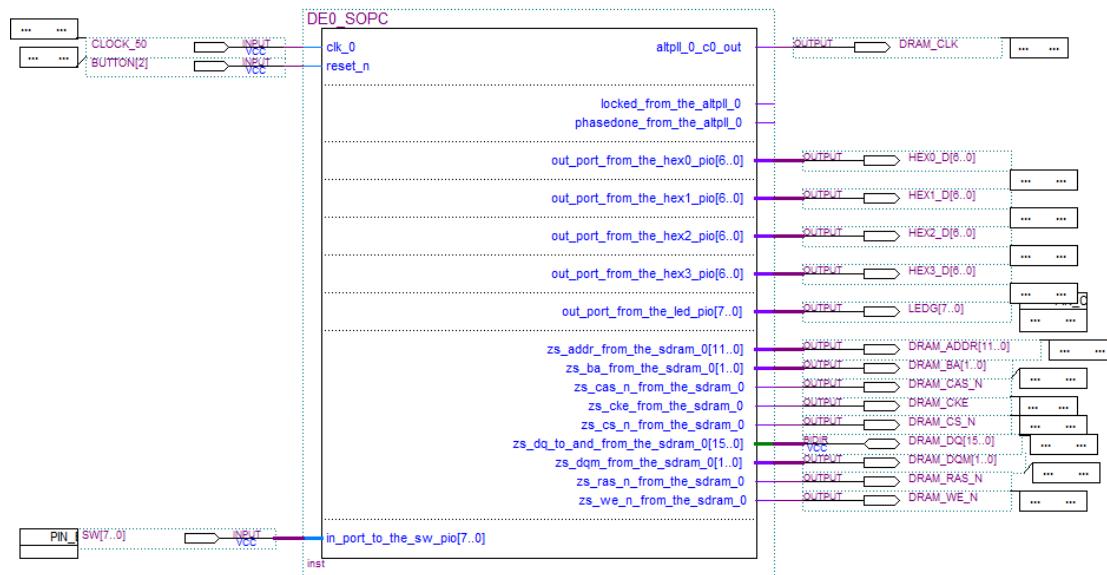
    return 0; // End program
}
```



隨堂練習一

- 請設計一個可由 switch 指撥開關輸入,對應產生七段顯示器輸出(0~F)的程式。  
(當指撥開關為 0x00 時，七段顯示器為數字“0”  
當指撥開關為 0x01 時，七段顯示器為數字“1”

...  
)



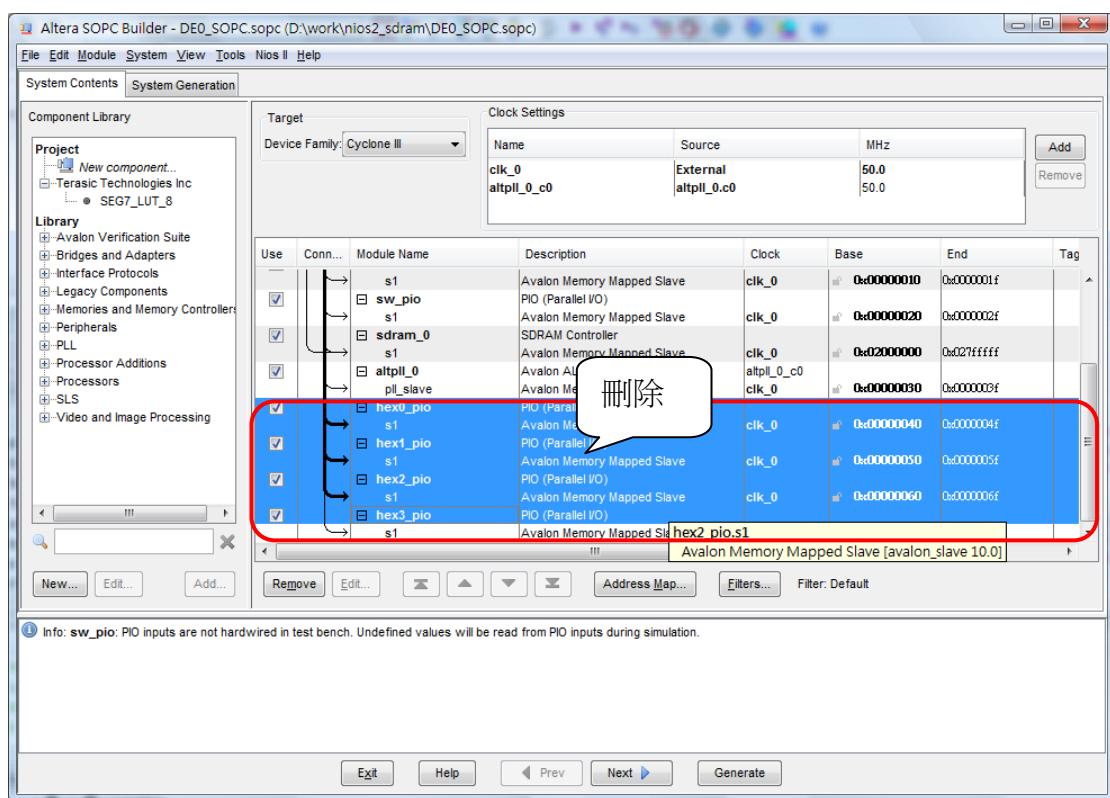
}

# 七段顯示器控制二

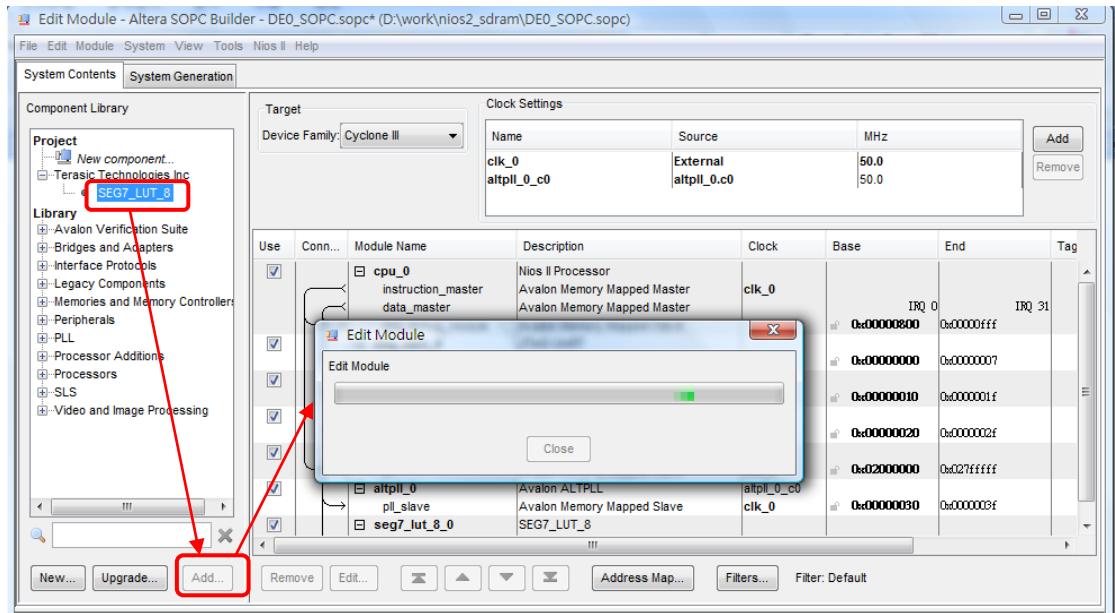
## -在 SOPC Builder 加入硬體 component

利用 SOPC Builder 建立 Nios II system 與 Quartus II 專案整合設計

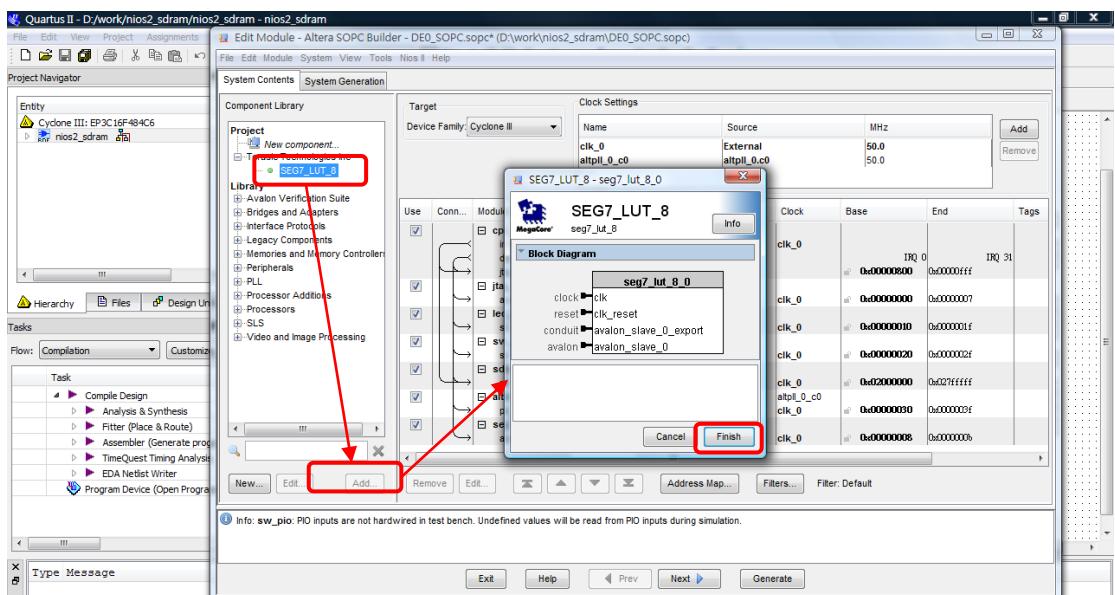
1. 從 moodle 下載 nios2\_sdram 專案壓縮檔。
2. 從 moodle 下載 SEG7\_LUT\_8\_10.0.rar，並將之解壓縮至工作專案資料夾中(產生 SEG7\_LUT\_8 資料夾)。
3. 刪除之前所建立的 hex0\_pio~hex3\_pio。



4. 在 SOPC Builder 中新增七段顯示器控制元件(SEG7\_LUT\_8)



要等候一段時間，出現 SEG7\_LUT\_8\_0 Settings 後按“Finish”結束。



執行“Generate”

P.S. 關於 SEG7\_LUT\_8 之 HDL 程式內容如下

```
module SEG7_LUT  (      oSEG,iDIG );
input [3:0] iDIG;
output      [6:0] oSEG;
```

```

reg      [6:0] oSEG;

always @(iDIG)
begin
    case(iDIG)
        4'h1: oSEG = 7'b1111001; // ---t---
        4'h2: oSEG = 7'b0100100; // | |
        4'h3: oSEG = 7'b0110000; // lt rt
        4'h4: oSEG = 7'b0011001; // | |
        4'h5: oSEG = 7'b0010010; // ---m---
        4'h6: oSEG = 7'b0000010; // | |
        4'h7: oSEG = 7'b1111000; // lb rb
        4'h8: oSEG = 7'b0000000; // | |
        4'h9: oSEG = 7'b0011000; // ---b---
        4'ha: oSEG = 7'b0001000;
        4'hb: oSEG = 7'b0000011;
        4'hc: oSEG = 7'b1000110;
        4'hd: oSEG = 7'b0100001;
        4'he: oSEG = 7'b0000110;
        4'hf: oSEG = 7'b0001110;
        4'h0: oSEG = 7'b1000000;
    endcase
end
endmodule

```

```

module SEG7_LUT_8 (      oSEG0,oSEG1,oSEG2,oSEG3,oSEG4,oSEG5,oSEG6,oSEG7,
                        iDIG,iWR,iCLK,iRST_N );
input [31:0]      iDIG;
input            iWR,iCLK,iRST_N;
output [6:0]      oSEG0,oSEG1,oSEG2,oSEG3,oSEG4,oSEG5,oSEG6,oSEG7;
reg   [31:0]      rDIG;

always@(posedge iCLK or negedge iRST_N)
begin
    if(!iRST_N)
        rDIG <= 0;
    else

```

```

begin
    if(iWR)
        rDIG <= iDIG;
    end
end

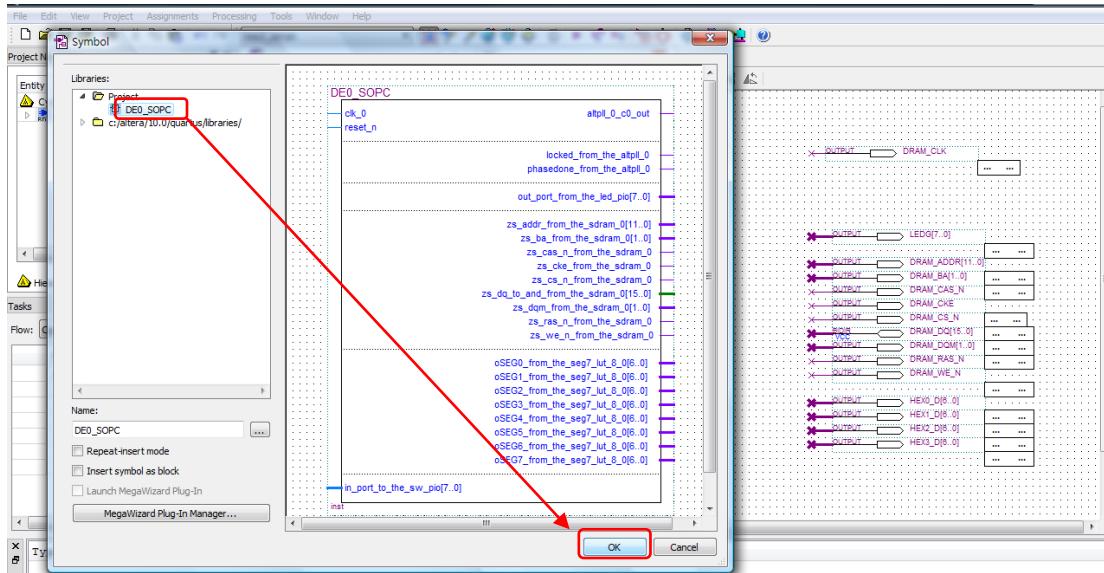
SEG7_LUT    u0  (    oSEG0,rDIG[3:0]      );
SEG7_LUT    u1  (    oSEG1,rDIG[7:4]      );
SEG7_LUT    u2  (    oSEG2,rDIG[11:8]     );
SEG7_LUT    u3  (    oSEG3,rDIG[15:12]    );
SEG7_LUT    u4  (    oSEG4,rDIG[19:16]    );
SEG7_LUT    u5  (    oSEG5,rDIG[23:20]    );
SEG7_LUT    u6  (    oSEG6,rDIG[27:24]    );
SEG7_LUT    u7  (    oSEG7,rDIG[31:28]    );

endmodule

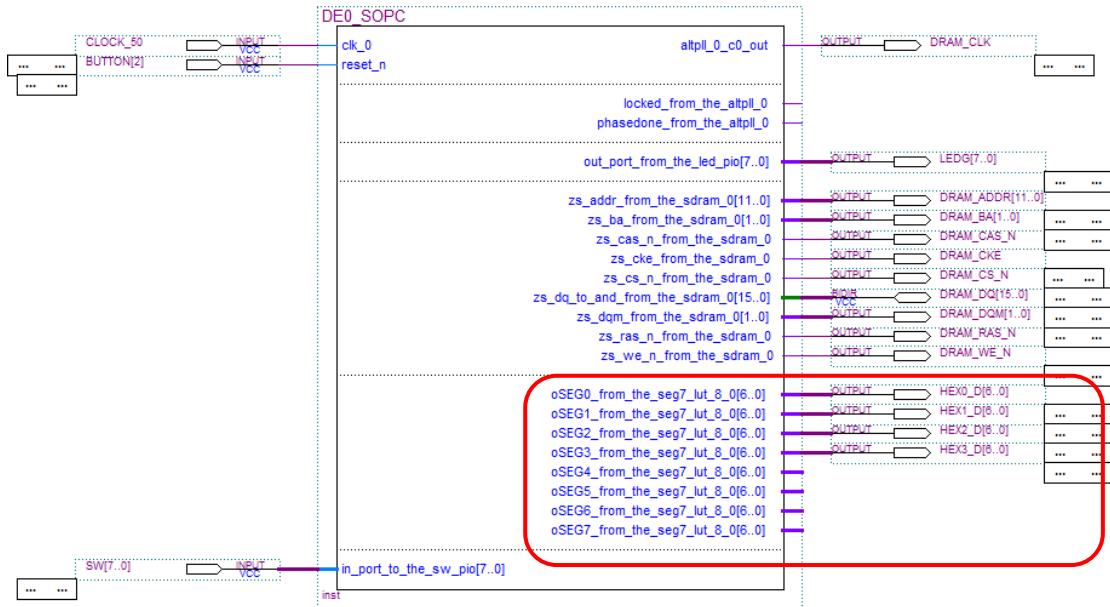
```

## 5. 電路整合設計 & Pin Assignments & Download

### (1) 更新 DE0\_SOPC symbol



### (2) 重新完成連線。



- (3) Compiler the design.  
(4) download the hardware

### 三、軟體設計

```
#include <iio.h>
#include "system.h"
#include "unistd.h"

int main() {
    int i;

    while(1)
    {
        for(i = 0; i != 0xFFFF; i++)
        {
            usleep(100000);
            IOWR(SEG7_LUT_8_0_BASE, 0, i);
        }
    }
    return 0;
}
```

## 隨堂練習二

請設計一個可透過指撥開關 SW0 控制七段顯示器上、下數(0x0000~0xFFFF)的程式。