111-1　535313

進階可程式邏輯系統設計與應用

Advanced Programmable Logic System Design and Application

實驗編號 : LAB08

實驗名稱 : HPS FPGA and Custom QSYS Component

結報完成日期 : 2022.12.05

姓名 : 王語
系級 : 機械四
學號 : 0811127

一、 實驗目的

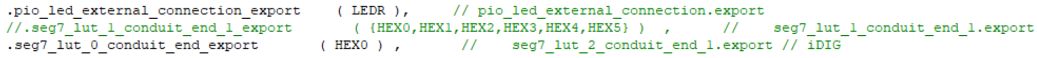
本實驗藉由Qsys連接DE1-SoC上的ARM以及FPGA，以熟悉基於SoC的 FPGA 嵌入式系統硬體與軟體開發流程。

說明[1]：HPS是基於ARM的處理器，具有豐富的周邊電路和存儲介面。HPS和 FPGA不僅能夠獨立工作，也能通過高性能 AXI 總線橋接實現高速寬頻行數據傳輸，此部分配置可以藉由Quartus II 中的Custom QSYS完成開發，這個總線是雙向的且支持同時讀寫的功能。

ARM部分以 C語言開發，而FPGA部分則支持Verilog。

二、 Verilog 程式碼

為了節省空間，只貼由我撰寫的兩行程式碼。

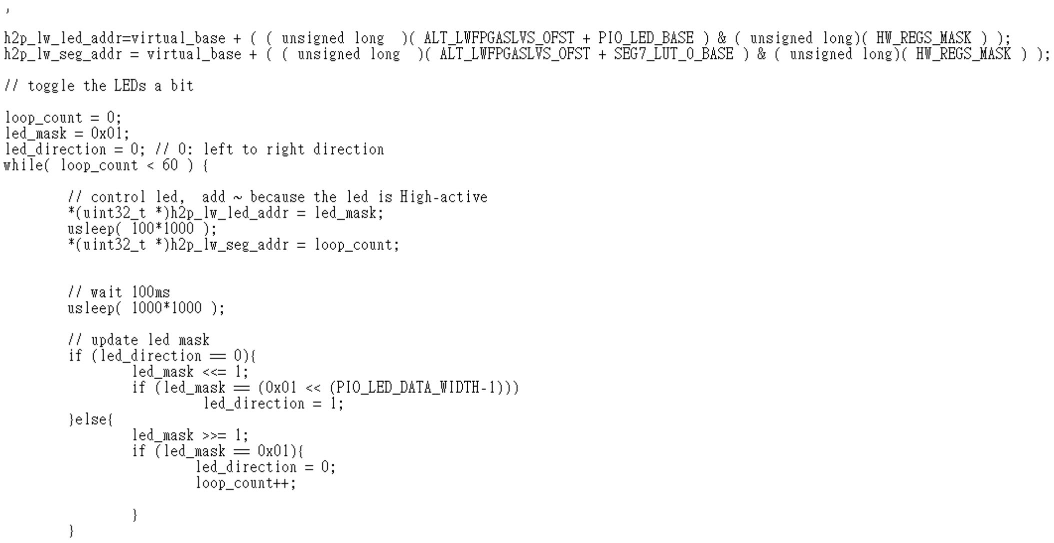


```
.pio_led_external_connection_export    ( LEDR ),     // pio_led_external_connection.export
//.seg7_lut_1_conduit_end_1_export       ( {HEX0,HEX1,HEX2,HEX3,HEX4,HEX5} )  ,      //    seg7_lut_1_conduit_end_1.export
.seg7_lut_0_conduit_end_export       ( HEX0 ) ,       //    seg7_lut_2_conduit_end_1.export // iDIG
```

另外，E3下載的v檔案中有部分wire沒有宣告，已將其更正。

三、 C 程式碼

由於執行過程七節管沒有正常工作，我將助教給的程式碼做了小更動，移除了兩行註解。



```
,

h2p_lw_led_addr=virtual_base + ( ( unsigned long  )( ALT_LWFPGASLVS_OFST + PIO_LED_BASE ) & ( unsigned long)( HW_REGS_MASK ) );
h2p_lw_seg_addr = virtual_base + ( ( unsigned long  )( ALT_LWFPGASLVS_OFST + SEG7_LUT_0_BASE ) & ( unsigned long)( HW_REGS_MASK ) );

// toggle the LEDs a bit

loop_count = 0;
led_mask = 0x01;
led_direction = 0; // 0: left to right direction
while( loop_count < 60 ) {

        // control led,  add ~ because the led is High-active
        *(uint32_t *)h2p_lw_led_addr = led_mask;
        usleep( 100*1000 );
        *(uint32_t *)h2p_lw_seg_addr = loop_count;


        // wait 100ms
        usleep( 1000*1000 );

        // update led mask
        if (led_direction == 0){
                led_mask <<= 1;
                if (led_mask == (0x01 << (PIO_LED_DATA_WIDTH-1)))
                         led_direction = 1;
        }else{
                led_mask >>= 1;
                if (led_mask == 0x01){
                        led_direction = 0;
                        loop_count++;

                }
        }
```

四、 hps_0

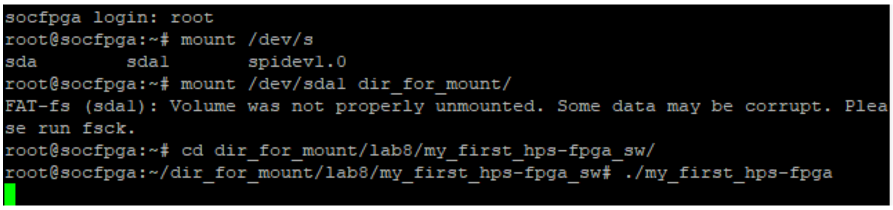
```
#ifndef _ALTERA_HPS_0_H_
#define _ALTERA_HPS_0_H_

/*
 * This file was automatically generated by the swinfo2header utility.
 *
 * Created from SOPC Builder system 'soc_system' in
 * file './soc_system.sopcinfo'.
 */

/*
 * This file contains macros for module 'hps_0' and devices
 * connected to the following masters:
 *     h2f_axi_master
 *     h2f_lw_axi_master
 *
 * Do not include this header file and another header file created for a
 * different module or master group at the same time.
 * Doing so may result in duplicate macro names.
 * Instead, use the system header file which has macros with unique names.
 */

/*
 * Macros for device 'onchip_memory2_0', class 'altera_avalon_onchip_memory2'
 * The macros are prefixed with 'ONCHIP_MEMORY2_0_'.
 * The prefix is the slave descriptor.
 */
#define ONCHIP_MEMORY2_0_COMPONENT_TYPE altera_avalon_onchip_memory2
#define ONCHIP_MEMORY2_0_COMPONENT_NAME onchip_memory2_0
#define ONCHIP_MEMORY2_0_BASE 0x0
#define ONCHIP_MEMORY2_0_SPAN 65536
#define ONCHIP_MEMORY2_0_END 0xffff
#define ONCHIP_MEMORY2_0_ALLOW_IN_SYSTEM_MEMORY_CONTENT_EDITOR 0
#define ONCHIP_MEMORY2_0_ALLOW_MRAM_SIM_CONTENTS_ONLY_FILE 0
#define ONCHIP_MEMORY2_0_CONTENTS_INFO ""
#define ONCHIP_MEMORY2_0_DUAL_PORT 0
#define ONCHIP_MEMORY2_0_GUI_RAM_BLOCK_TYPE AUTO
#define ONCHIP_MEMORY2_0_INIT_CONTENTS_FILE soc_system_onchip_memory2_0
#define ONCHIP_MEMORY2_0_INIT_MEM_CONTENT 1
#define ONCHIP_MEMORY2_0_INSTANCE_ID NONE
#define ONCHIP_MEMORY2_0_NON_DEFAULT_INIT_FILE_ENABLED 0
#define ONCHIP_MEMORY2_0_RAM_BLOCK_TYPE AUTO
#define ONCHIP_MEMORY2_0_READ_DURING_WRITE_MODE DONT_CARE
#define ONCHIP_MEMORY2_0_SINGLE_CLOCK_OP 0
#define ONCHIP_MEMORY2_0_SIZE_MULTIPLE 1
#define ONCHIP_MEMORY2_0_SIZE_VALUE 65536
#define ONCHIP_MEMORY2_0_WRITABLE 1
#define ONCHIP_MEMORY2_0_MEMORY_INFO_DAT_SYM_INSTALL_DIR SIM_DIR
#define ONCHIP_MEMORY2_0_MEMORY_INFO_GENERATE_DAT_SYM 1
#define ONCHIP_MEMORY2_0_MEMORY_INFO_GENERATE_HEX 1
#define ONCHIP_MEMORY2_0_MEMORY_INFO_HAS_BYTE_LANE 0
#define ONCHIP_MEMORY2_0_MEMORY_INFO_HEX_INSTALL_DIR QPF_DIR
#define ONCHIP_MEMORY2_0_MEMORY_INFO_MEM_INIT_DATA_WIDTH 64
#define ONCHIP_MEMORY2_0_MEMORY_INFO_MEM_INIT_FILENAME soc_system_onchip_memory2_0
```

```
/*
 * Macros for device 'pio_led', class 'altera_avalon_pio'
 * The macros are prefixed with 'PIO_LED_'.
 * The prefix is the slave descriptor.
 */
#define PIO_LED_COMPONENT_TYPE altera_avalon_pio
#define PIO_LED_COMPONENT_NAME pio_led
#define PIO_LED_BASE 0x0
#define PIO_LED_SPAN 32
#define PIO_LED_END 0x1f
#define PIO_LED_BIT_CLEARING_EDGE_REGISTER 0
#define PIO_LED_BIT_MODIFYING_OUTPUT_REGISTER 0
#define PIO_LED_CAPTURE 0
#define PIO_LED_DATA_WIDTH 10
#define PIO_LED_DO_TEST_BENCH_WIRING 0
#define PIO_LED_DRIVEN_SIM_VALUE 0
#define PIO_LED_EDGE_TYPE NONE
#define PIO_LED_FREQ 50000000
#define PIO_LED_HAS_IN 0
#define PIO_LED_HAS_OUT 1
#define PIO_LED_HAS_TRI 0
#define PIO_LED_IRQ_TYPE NONE
#define PIO_LED_RESET_VALUE 1023

/*
 * Macros for device 'SEG7_LUT_0', class 'SEG7_LUT'
 * The macros are prefixed with 'SEG7_LUT_0_'.
 * The prefix is the slave descriptor.
 */
#define SEG7_LUT_0_COMPONENT_TYPE SEG7_LUT
#define SEG7_LUT_0_COMPONENT_NAME SEG7_LUT_0
#define SEG7_LUT_0_BASE 0x20
#define SEG7_LUT_0_SPAN 4
#define SEG7_LUT_0_END 0x23

/*
 * Macros for device 'sysid_qsys', class 'altera_avalon_sysid_qsys'
 * The macros are prefixed with 'SYSID_QSYS_'.
 * The prefix is the slave descriptor.
 */
#define SYSID_QSYS_COMPONENT_TYPE altera_avalon_sysid_qsys
#define SYSID_QSYS_COMPONENT_NAME sysid_qsys
#define SYSID_QSYS_BASE 0x10000
#define SYSID_QSYS_SPAN 8
#define SYSID_QSYS_END 0x10007
#define SYSID_QSYS_ID 2899645186
#define SYSID_QSYS_TIMESTAMP 1670153700

/*
 * Macros for device 'jtag_uart', class 'altera_avalon_jtag_uart'
 * The macros are prefixed with 'JTAG_UART_'.
 * The prefix is the slave descriptor.
 */
#define JTAG_UART_COMPONENT_TYPE altera_avalon_jtag_uart
#define JTAG_UART_COMPONENT_NAME jtag_uart
#define JTAG_UART_BASE 0x20000
#define JTAG_UART_SPAN 16
#define JTAG_UART_END 0x2000f
```

```
#define JTAG_UART_IRQ 2
#define JTAG_UART_READ_DEPTH 64
#define JTAG_UART_READ_THRESHOLD 8
#define JTAG_UART_WRITE_DEPTH 64
#define JTAG_UART_WRITE_THRESHOLD 8

#endif /* _ALTERA_HPS_0_H_ */
```

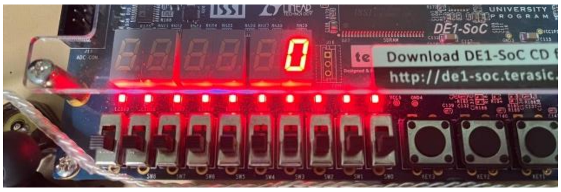
五、　實驗結果

Putty 執行

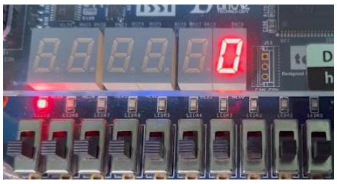


```
socfpga login: root
root@socfpga:~# mount /dev/s
sda         sdal        spidevl.0
root@socfpga:~# mount /dev/sdal dir_for_mount/
FAT-fs (sdal): Volume was not properly unmounted. Some data may be corrupt. Plea
se run fsck.
root@socfpga:~# cd dir_for_mount/lab8/my_first_hps-fpga_sw/
root@socfpga:~/dir_for_mount/lab8/my_first_hps-fpga_sw# ./my_first_hps-fpga
```

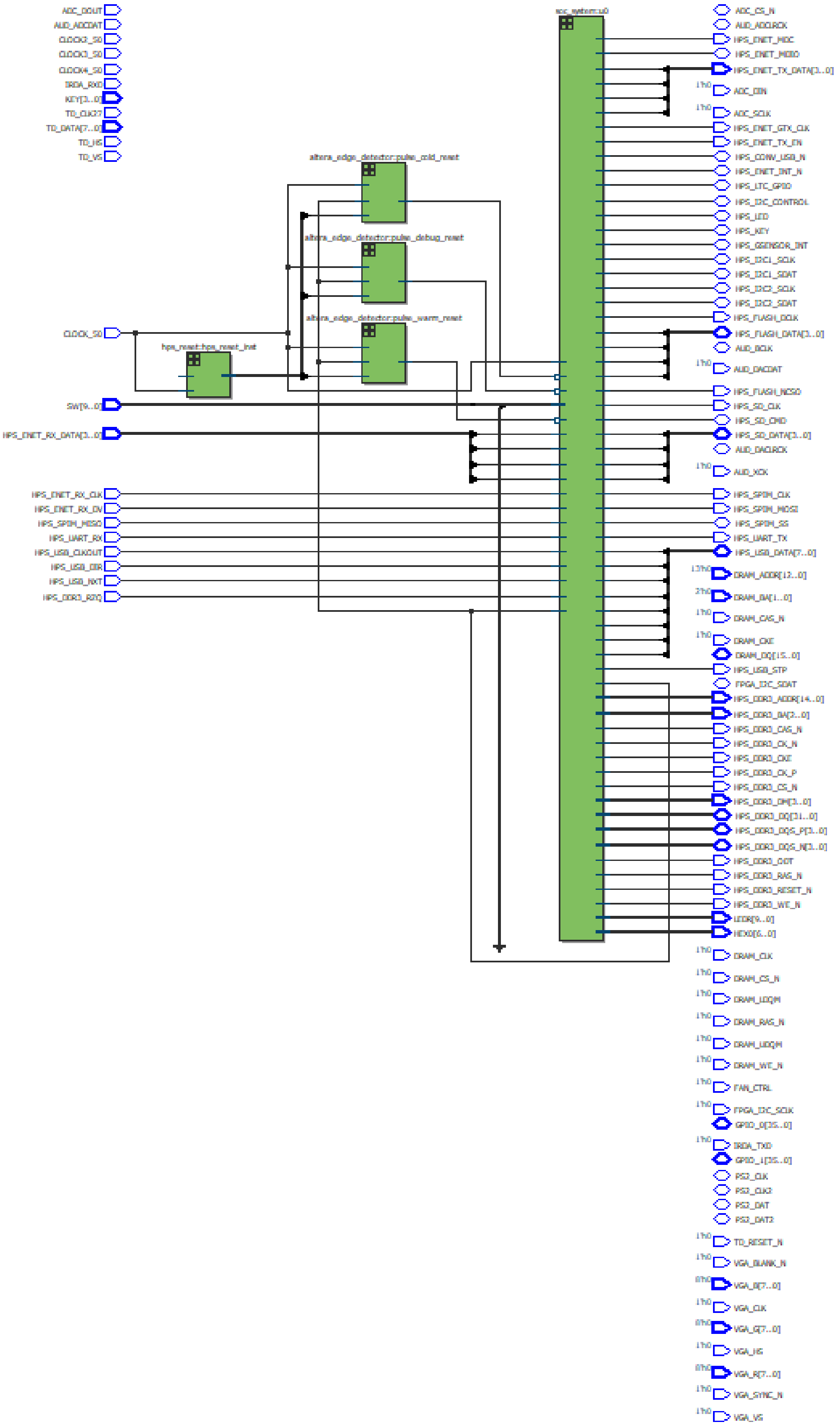
Sof 檔案燒錄完成



運行結果如同 DEMO 影片，執行跑馬燈，LED 燈從最右邊開始閃爍至最左邊，再由最左邊閃爍回到最右邊，當最右邊的 LED 燈再次亮起時，七節管的數字+1。

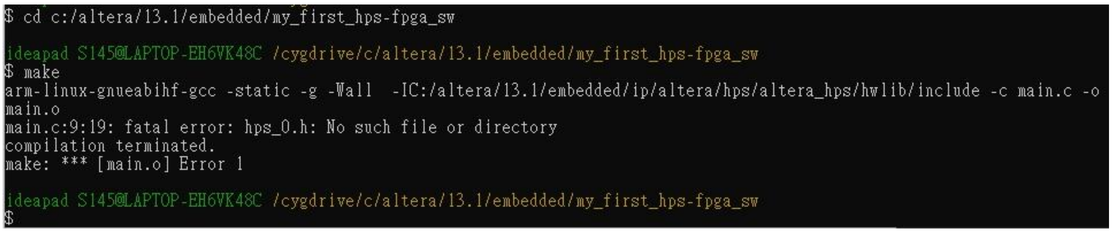


六、 RTL


ADC_DOUT
AUD_ADCDAT
CLOCK2_50
CLOCK3_50
CLOCK4_50
IRDA_RXD
KEY[3..0]
TD_CLK27
TD_DATA[7..0]
TD_HS
TD_VS
altera_edge_detector:pulse_cold_reset
altera_edge_detector:pulse_debug_reset
altera_edge_detector:pulse_warm_reset
CLOCK_50
hps_reset:hps_reset_inst
SW[9..0]
HPS_ENET_RX_DATA[3..0]
HPS_ENET_RX_CLK
HPS_ENET_RX_DV
HPS_SPIM_MISO
HPS_UART_RX
HPS_USB_CLKOUT
HPS_USB_DIR
HPS_USB_NXT
HPS_DDR3_RZQ
soc_system:u0
ADC_CS_N
AUD_ADCLRCK
HPS_ENET_MDC
HPS_ENET_MDIO
HPS_ENET_TX_DATA[3..0]
1'h0
ADC_DIN
1'h0
ADC_SCLK
HPS_ENET_GTX_CLK
HPS_ENET_TX_EN
HPS_CONV_USB_N
HPS_ENET_INT_N
HPS_LTC_GPIO
HPS_I2C_CONTROL
HPS_LED
HPS_KEY
HPS_GSENSOR_INT
HPS_I2C1_SCLK
HPS_I2C1_SDAT
HPS_I2C2_SCLK
HPS_I2C2_SDAT
HPS_FLASH_DCLK
HPS_FLASH_DATA[3..0]
AUD_BCLK
1'h0
AUD_DACDAT
HPS_FLASH_NCSO
HPS_SD_CLK
HPS_SD_CMD
HPS_SD_DATA[3..0]
AUD_DACLRCK
1'h0
AUD_XCK
HPS_SPIM_CLK
HPS_SPIM_MOSI
HPS_SPIM_SS
HPS_UART_TX
HPS_USB_DATA[7..0]
13'h0
DRAM_ADDR[12..0]
2'h0
DRAM_BA[1..0]
1'h0
DRAM_CAS_N
1'h0
DRAM_CKE
DRAM_DQ[15..0]
HPS_USB_STP
FPGA_I2C_SDAT
HPS_DDR3_ADDR[14..0]
HPS_DDR3_BA[2..0]
HPS_DDR3_CAS_N
HPS_DDR3_CK_N
HPS_DDR3_CKE
HPS_DDR3_CK_P
HPS_DDR3_CS_N
HPS_DDR3_DM[3..0]
HPS_DDR3_DQ[31..0]
HPS_DDR3_DQS_P[3..0]
HPS_DDR3_DQS_N[3..0]
HPS_DDR3_ODT
HPS_DDR3_RAS_N
HPS_DDR3_RESET_N
HPS_DDR3_WE_N
LEDR[9..0]
HEX0[6..0]
1'h0
DRAM_CLK
1'h0
DRAM_CS_N
1'h0
DRAM_LDQM
1'h0
DRAM_RAS_N
1'h0
DRAM_UDQM
1'h0
DRAM_WE_N
1'h0
FAN_CTRL
1'h0
FPGA_I2C_SCLK
GPIO_0[35..0]
1'h0
IRDA_TXD
GPIO_1[35..0]
PS2_CLK
PS2_CLK2
PS2_DAT
PS2_DAT2
1'h0
TD_RESET_N
1'h0
VGA_BLANK_N
8'h0
VGA_B[7..0]
1'h0
VGA_CLK
8'h0
VGA_G[7..0]
1'h0
VGA_HS
8'h0
VGA_R[7..0]
1'h0
VGA_SYNC_N
1'h0
VGA_VS

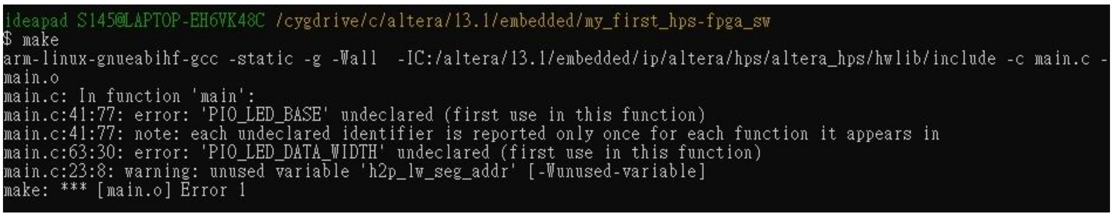
七、 問題與討論

1. 照著實驗講義之步驟操作，卻出現缺少 hps_0 的錯誤



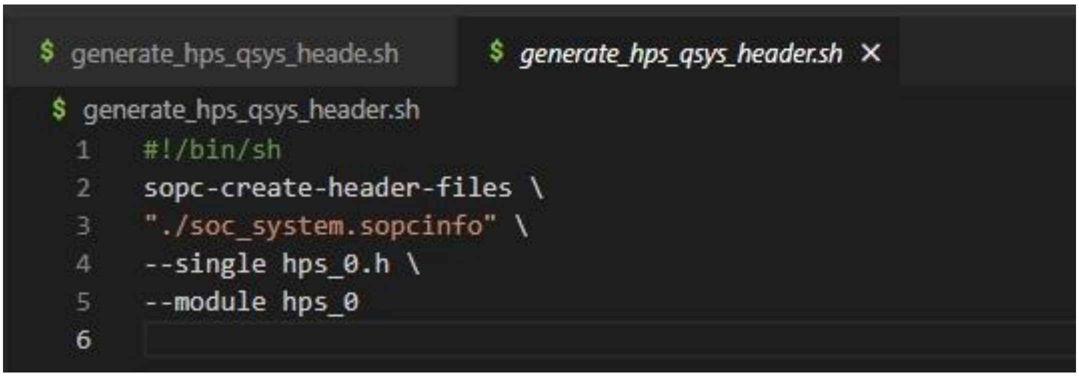
```
$ cd c:/altera/13.1/embedded/my_first_hps-fpga_sw

ideapad S145@LAPTOP-EH6VK48C /cygdrive/c/altera/13.1/embedded/my_first_hps-fpga_sw
$ make
arm-linux-gnueabihf-gcc -static -g -Wall  -IC:/altera/13.1/embedded/ip/altera/hps/altera_hps/hwlib/include -c main.c -o
main.o
main.c:9:19: fatal error: hps_0.h: No such file or directory
compilation terminated.
make: *** [main.o] Error 1

ideapad S145@LAPTOP-EH6VK48C /cygdrive/c/altera/13.1/embedded/my_first_hps-fpga_sw
$
```

而且生成出來的 hps_0 似乎也存在錯誤



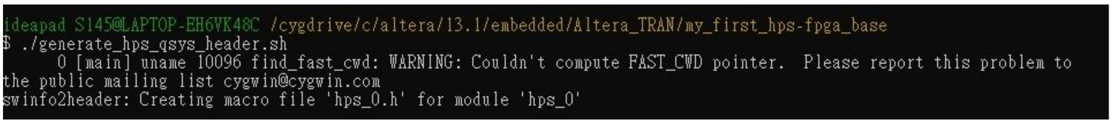
```
ideapad S145@LAPTOP-EH6VK48C /cygdrive/c/altera/13.1/embedded/my_first_hps-fpga_sw
$ make
arm-linux-gnueabihf-gcc -static -g -Wall  -IC:/altera/13.1/embedded/ip/altera/hps/altera_hps/hwlib/include -c main.c -
main.o
main.c: In function 'main':
main.c:41:77: error: 'PIO_LED_BASE' undeclared (first use in this function)
main.c:41:77: note: each undeclared identifier is reported only once for each function it appears in
main.c:63:30: error: 'PIO_LED_DATA_WIDTH' undeclared (first use in this function)
main.c:23:8: warning: unused variable 'h2p_lw_seg_addr' [-Wunused-variable]
make: *** [main.o] Error 1
```

為了解決此問題，自行創建一個新的 generate_hps_qsys_header.sh 檔，方法如同參考資料[2]


$ generate_hps_qsys_heade.sh
$ generate_hps_qsys_header.sh ×
$ generate_hps_qsys_header.sh

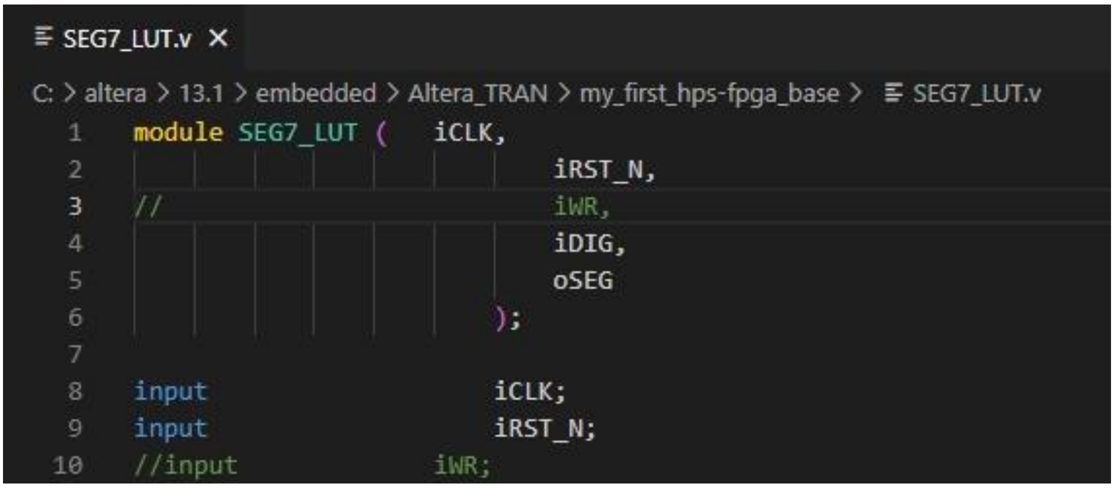
```
#!/bin/sh
sopc-create-header-files \
"./soc_system.sopcinfo" \
--single hps_0.h \
--module hps_0
```

並以新生成的檔案取代舊版的即可解決問題。



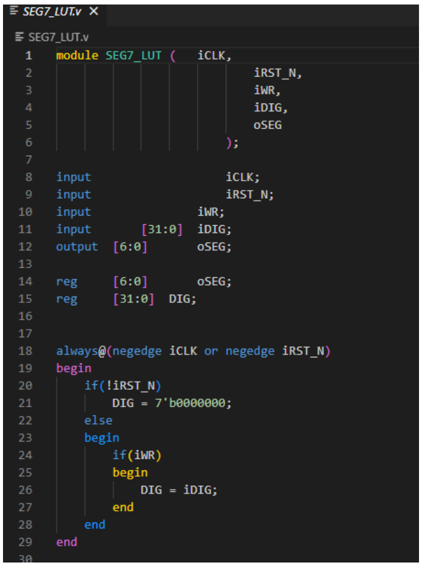
```
ideapad S145@LAPTOP-EH6VK48C /cygdrive/c/altera/13.1/embedded/Altera_TRAN/my_first_hps-fpga_base
$ ./generate_hps_qsys_header.sh
      0 [main] uname 10096 find_fast_cwd: WARNING: Couldn't compute FAST_CWD pointer.  Please report this problem to
the public mailing list cygwin@cygwin.com
swinfo2header: Creating macro file 'hps_0.h' for module 'hps_0'
```

2. 在 Qsys 中創建 SEG7_LUT 時，因為缺少了 write_en，因此無法順利生成，將 SEG7_LUT.v 中的註解取消即可順利運行。



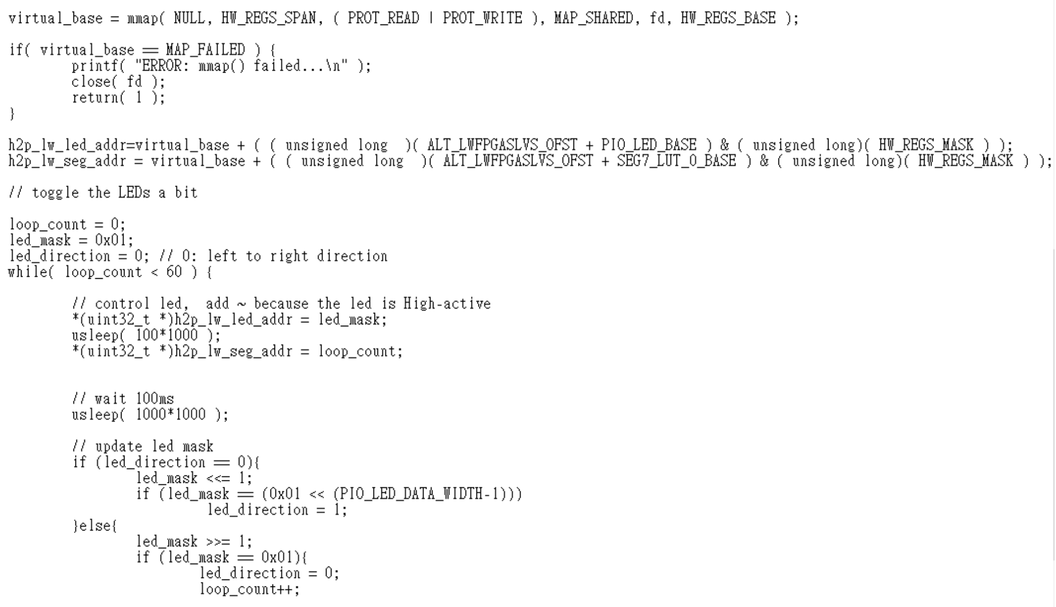
SEG7_LUT.v
C: > altera > 13.1 > embedded > Altera_TRAN > my_first_hps-fpga_base > SEG7_LUT.v

```
module SEG7_LUT (   iCLK,
                            iRST_N,
//                          iWR,
                            iDIG,
                            oSEG
                        );

input                   iCLK;
input                   iRST_N;
//input             iWR;
```



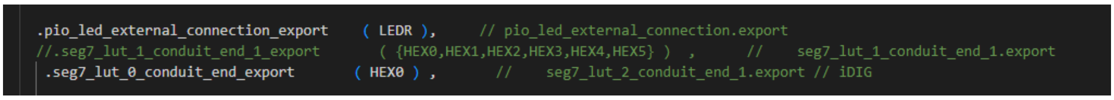
```
module SEG7_LUT (   iCLK,
                        iRST_N,
                        iWR,
                        iDIG,
                        oSEG
                    );

input                   iCLK;
input                   iRST_N;
input               iWR;
input       [31:0]  iDIG;
output  [6:0]       oSEG;

reg     [6:0]       oSEG;
reg     [31:0]  DIG;


always@(negedge iCLK or negedge iRST_N)
begin
    if(!iRST_N)
        DIG = 7'b0000000;
    else
    begin
        if(iWR)
        begin
            DIG = iDIG;
        end
    end
end
```

3. 燒錄 sof 檔並在 putty 上運行，不過 LED 跑馬燈跑完一輪後，七節管上的數字並沒有增加，花費數個小時，反覆檢查 Qsys 走線，以及 SEG7_LUT.v 是否存在錯誤，結果發現是 C 檔案中七節管數字的程式碼備註解掉了，不知道是不是助教故意設下的陷阱，但有一點考以很確定，這個程式的設計者是個狠人，因為使用記事本，程式碼沒有顏色，真的很難發現。



```
virtual_base = mmap( NULL, HW_REGS_SPAN, ( PROT_READ | PROT_WRITE ), MAP_SHARED, fd, HW_REGS_BASE );

if( virtual_base == MAP_FAILED ) {
        printf( "ERROR: mmap() failed...\n" );
        close( fd );
        return( 1 );
}

h2p_lw_led_addr=virtual_base + ( ( unsigned long  )( ALT_LWFPGASLVS_OFST + PIO_LED_BASE ) & ( unsigned long)( HW_REGS_MASK ) );
h2p_lw_seg_addr = virtual_base + ( ( unsigned long  )( ALT_LWFPGASLVS_OFST + SEG7_LUT_0_BASE ) & ( unsigned long)( HW_REGS_MASK ) );

// toggle the LEDs a bit

loop_count = 0;
led_mask = 0x01;
led_direction = 0; // 0: left to right direction
while( loop_count < 60 ) {

        // control led,  add ~ because the led is High-active
        *(uint32_t *)h2p_lw_led_addr = led_mask;
        usleep( 100*1000 );
        *(uint32_t *)h2p_lw_seg_addr = loop_count;


        // wait 100ms
        usleep( 1000*1000 );

        // update led mask
        if (led_direction == 0){
                led_mask <<= 1;
                if (led_mask == (0x01 << (PIO_LED_DATA_WIDTH-1)))
                         led_direction = 1;
        }else{
                led_mask >>= 1;
                if (led_mask == 0x01){
                        led_direction = 0;
                        loop_count++;
```

4. 在講義中並沒有說明.seg7_lut_0_conduit_end_export 要接到哪裡，經過一番研究後發現要接在 HEX0 上。



```
 .pio_led_external_connection_export    ( LEDR ),     // pio_led_external_connection.export
 //.seg7_lut_1_conduit_end_1_export       ( {HEX0,HEX1,HEX2,HEX3,HEX4,HEX5} )  ,     //    seg7_lut_1_conduit_end_1.export
  .seg7_lut_0_conduit_end_export       ( HEX0 ) ,      //    seg7_lut_2_conduit_end_1.export // iDIG
```

5. 若要使結果和 DEMO 影片一致，則不可以用 4’b0000 作為 HEX1~HEX5 的輸入，即便沒有給予訊號也不可使他默認接地，否則七節管會輸出 8。為了解決此問題，我直接將 Output 的 HEX1~HEX5 移除。

八、 心得

這次實驗好困難，做的我心好累，不是不會打，是不知道錯在哪，對於嵌入式我一向很害怕，如果只是要打程式，我很會抓 BUG，不過這種系統類、架構相關的問題，在加上對於開發軟體的不熟悉，讓我挫敗連連。

心好累，希望期末專題可以順順利利。

九、 參考資料

[1] HPS 基本概念及其設計 - noticeable - 博客園 (cnblogs.com)
https://www.cnblogs.com/noticeable/p/9378394.html

[2] 【友晶科技 Terasic】用 sopc-create-header-files 工具生成 FPGA..
https://www.cnblogs.com/DoreenLiu/p/15308574.html