# THỰC HÀNH SOC

## Bài 5 – GIAO TIẾP BỘ TIMER VÀ NGẮT

### 1. MỤC ĐÍCH

Thông qua bài thực hành này, sinh viên sẽ biết:

o Cấu trúc bộ Timer của Altera (các thanh ghi, chức năng…).

o Cách xây dựng hệ thống phần cứng bằng Qsys để giao tiếp với bộ Timer.

o Cách xây dựng chương trình C trên công cụ Nios II – EDS để giao tiếp với bộ Timer, và sử dụng ngắt của bộ Timer.

### 2. NỘI DUNG

#### 2.1. Lý thuyết về bộ Timer

##### 2.1.1. Giới thiệu

Module Timer của Altera có những tính năng cơ bản sau:

- Bộ đếm 32bit, hỗ trợ điều khiển Start, Stop.

- Hỗ trợ cấu hình chu kì đếm, hỗ trợ phát sinh ngắt khi kết thúc chu kì đếm.

##### 2.1.2. Thanh ghi của Timer

Module Timer bao gồm 6 thanh ghi cơ bản như bên dưới.

| Offset | Name | R/W | Description of Bits | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | 15 | ... | 4 | 3 | 2 | 1 | 0 |
| 0 | status | RW | (1) | | | | | RUN | TO |
| 1 | control | RW | (1) | | | STOP | START | CONT | ITO |
| 2 | periodl | RW | Timeout Period – 1 (bits [15:0]) | | | | | | |
| 3 | periodh | RW | Timeout Period – 1 (bits [31:16]) | | | | | | |
| 4 | snapl | RW | Counter Snapshot (bits [15:0]) | | | | | | |
| 5 | snaph | RW | Counter Snapshot (bits [31:16]) | | | | | | |

Dù mỗi thanh ghi của bộ Timer là 16 bit, địa chỉ của tất cả các thanh ghi này được sắp hàng theo 32bit. Nghĩa là nếu địa chỉ của thanh ghi "status" là BASE, thì địa chỉ của thanh ghi control là BASE + 4.

Thanh ghi "status".

| Bit | Name | R/W/C | Description |
|---|---|---|---|
| 0 | TO | R/WC | The TO (timeout) bit is set to 1 when the internal counter reaches zero. Once set by a timeout event, the TO bit stays set until explicitly cleared by a master peripheral. Write 0 or 1 to the status register to clear the TO bit. |
| 1 | RUN | R | The RUN bit reads as 1 when the internal counter is running; otherwise this bit reads as 0. The RUN bit is not changed by a write operation to the status register. |

Thanh ghi "control".

| Bit | Name | R/W/C | Description |
|---|---|---|---|
| 0 | ITO | RW | If the ITO bit is 1, the interval timer core generates an IRQ when the status register's TO bit is 1. When the ITO bit is 0, the timer does not generate IRQs. |
| 1 | CONT | RW | The CONT (continuous) bit determines how the internal counter behaves when it reaches zero. If the CONT bit is 1, the counter runs continuously until it is stopped by the STOP bit. If CONT is 0, the counter stops after it reaches zero. When the counter reaches zero, it reloads with the value stored in the period registers, regardless of the CONT bit. |
| 2 | START (1) | W | Writing a 1 to the START bit starts the internal counter running (counting down). The START bit is an event bit that enables the counter when a write operation is performed. If the timer is stopped, writing a 1 to the START bit causes the timer to restart counting from the number currently stored in its counter. If the timer is already running, writing a 1 to START has no effect. Writing 0 to the START bit has no effect. |
| 3 | STOP (1) | W | Writing a 1 to the STOP bit stops the internal counter. The STOP bit is an event bit that causes the counter to stop when a write operation is performed. If the timer is already stopped, writing a 1 to STOP has no effect. Writing a 0 to the stop bit has no effect.<br>If the timer hardware is configured with **Start/Stop control bits** off, writing the STOP bit has no effect. |

Thanh ghi "periodh" và "periodl".

Hai thanh ghi này quy định chu kì đếm của Timer. "periodh" chứa 16 bit cao của chu kì đếm và "periodl" chứa 16 bit thấp của chu kì đếm.

Thanh ghi “snaph” và “snapl”.

Hai thanh ghi này thể hiện giá trị hiện tại của bộ đếm của Timer. “snaph” chứa 16 bit cao của và “snapl” chứa 16 bit thấp.

### 2.1.3. Hoạt động của Timer

Khi bắt đầu hoạt động, bộ đếm của Timer sẽ lấy giá trị khởi tạo từ thanh ghi “periodh” và “periodl”. Sau đó, bộ đếm sẽ bắt đầu đếm xuống tại mỗi chu kì xung clock. Khi giá trị của bộ đếm bằng 0 thì:

- Bộ đếm sẽ được khởi tạo lại giá trị ban đầu từ thanh ghi “periodh” và “periodl”.
- Bit “TO” của thanh ghi “status” sẽ bằng 1.
- Ngắt sẽ được sinh ra nếu bit “ITO” của thanh ghi “control” bằng 1.

Trong hàm xử lý ngắt Timer, xóa ngắt bằng cách xóa bit “TO” của thanh ghi “status”.

## 2.2. Hệ thống phần cứng

### 2.2.1. Tạo project Quartus

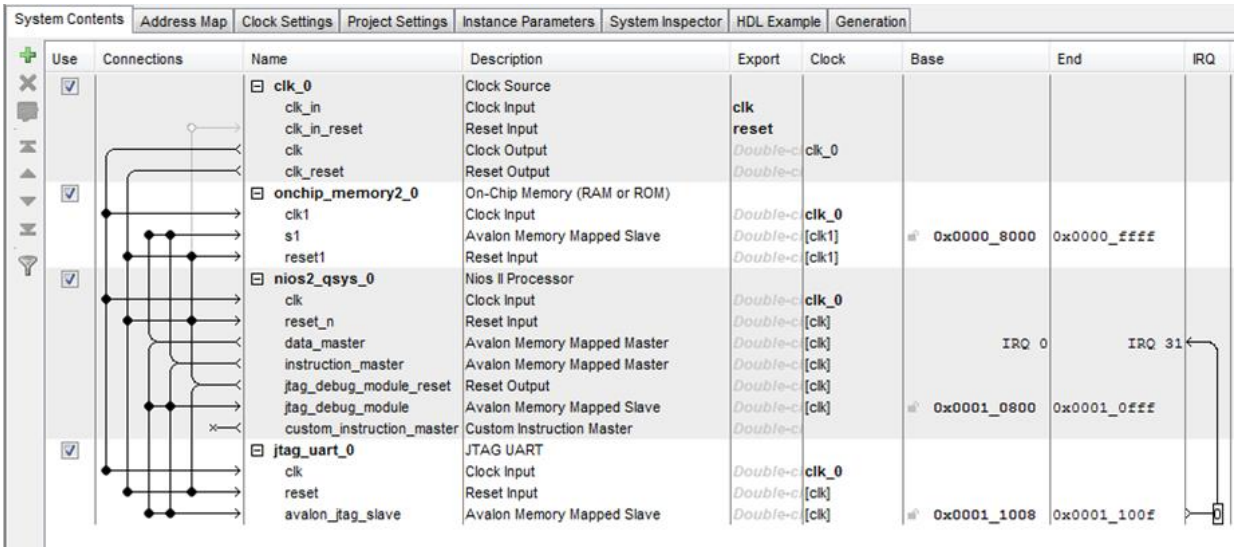
Tạo project Quartus tên là “lab5”. Lưu ý đường dẫn thư mục project không được có khoảng trắng.

Nếu sử dụng board DE2-115, chọn Family là Cyclone IV E, device là EP4CE115F29C8.

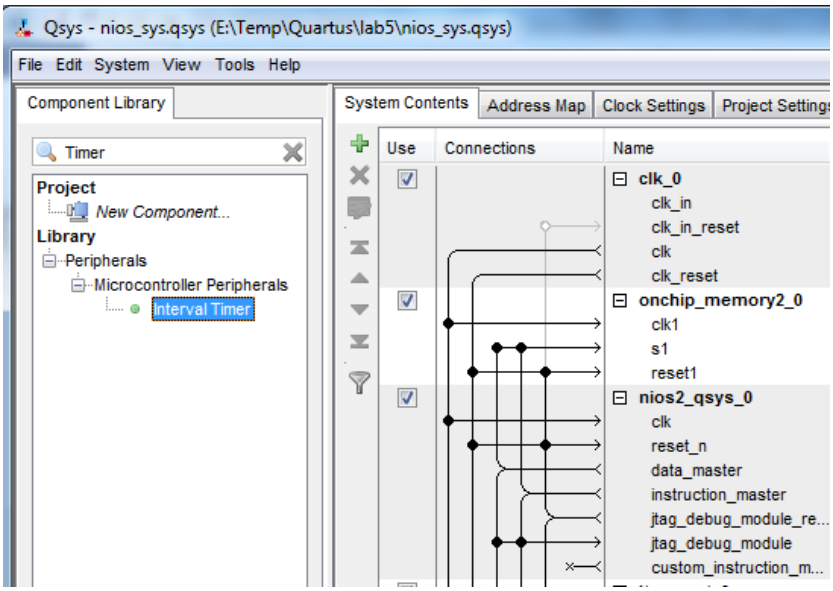
Nếu sử dụng board DE2, chọn Family là Cyclone II, device là EP2C35F672C6.

### 2.2.2. Xây dựng hệ thống Qsys

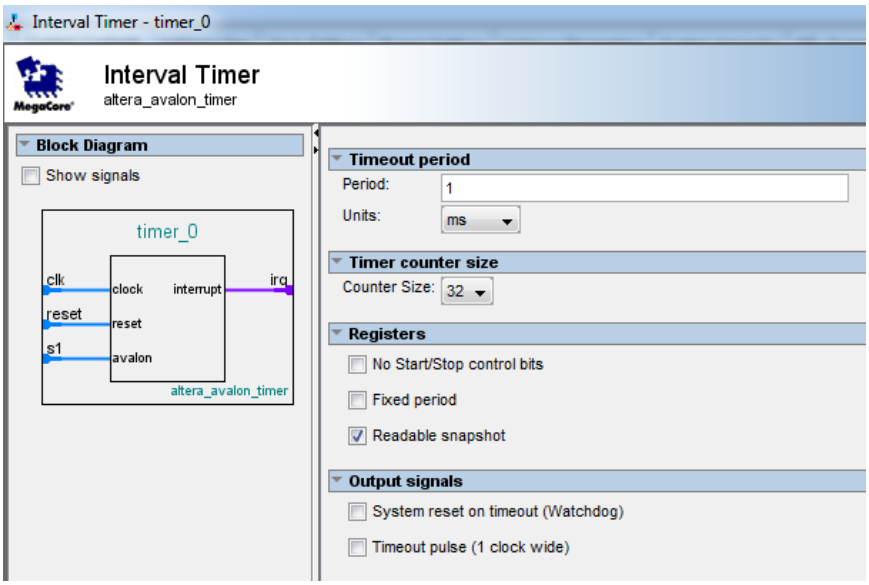
Xây dựng hệ thống phần cứng như hình bên dưới.



Trong cửa sổ thư viện, tìm bộ Timer và thêm vào hệ thống.

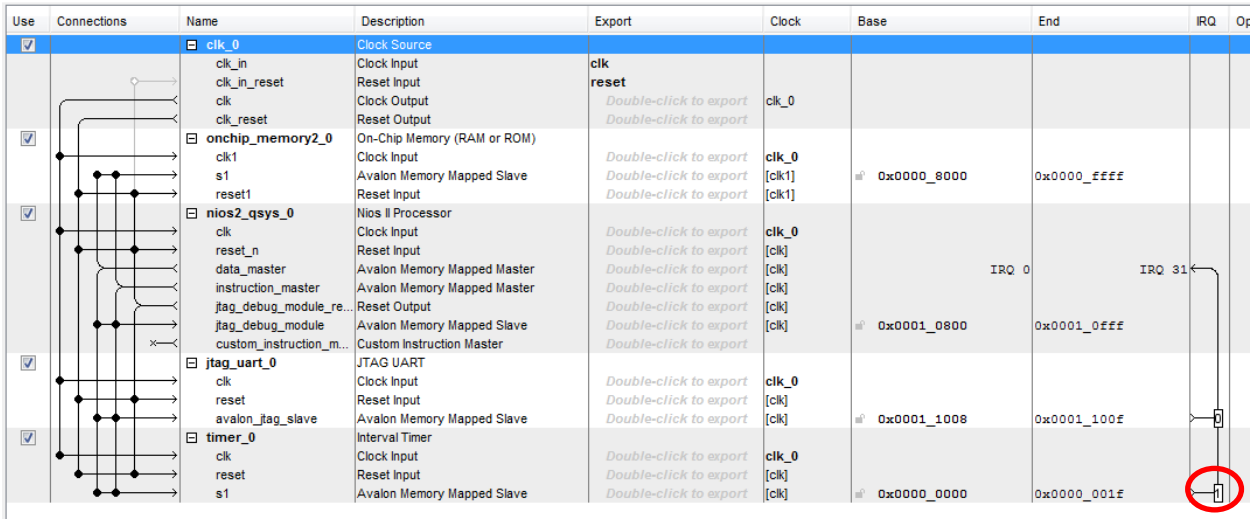



Cấu hình bộ Timer như bên dưới, click “Finish”.




Kết nối các tín hiệu của bộ Timer vào hệ thống.

Lưu ý kết nối tín hiệu ngắt của Timer vào CPU NIOS II.




Gán địa chỉ cho các module (System > Assign Base Addresses).

Gán độ ưu tiên ngắt cho các module (System > Assign Interrupt Number).

Hệ thống phần cứng đã hoàn thành, không còn thông báo lỗi. Save lại hệ thống với tên "nios_sys".

Chuyển sang tab "Generation", click "Generate".

### 2.2.3. Tích hợp hệ thống Qsys vào project Quatus

Thực hiện tương tự như bài thực hành trước.

Tạo file top module, đặt tên là "lab5.v" với nội dung như sau.

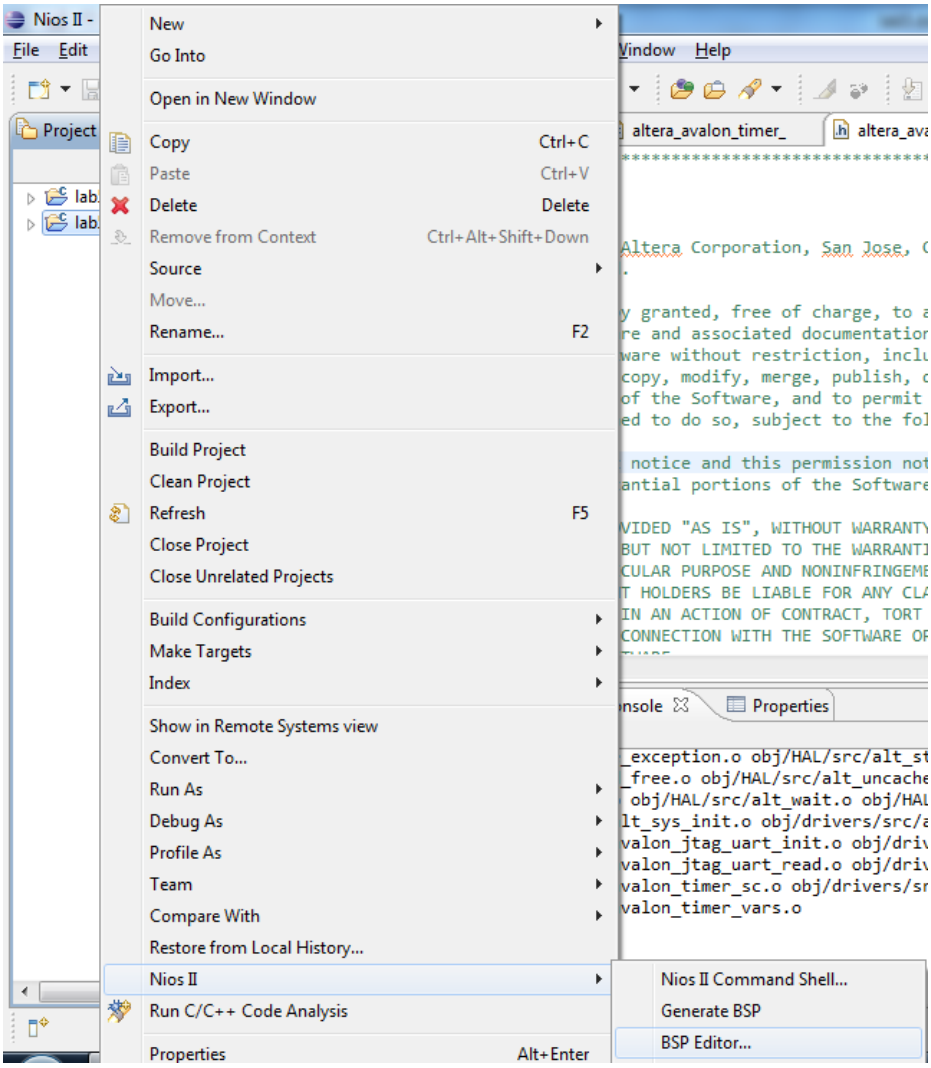
```
1   module lab5(CLOCK_50,KEY);
2
3   input CLOCK_50;
4   input [0:0] KEY;
5
6   nios_sys u0 (
7       .clk_clk       (CLOCK_50),
8       .reset_reset_n (KEY[0])
9    );
10
11  endmodule
```

Build project Quartus và download hệ thống phần cứng xuống board.

## 2.3. Lập trình phần mềm

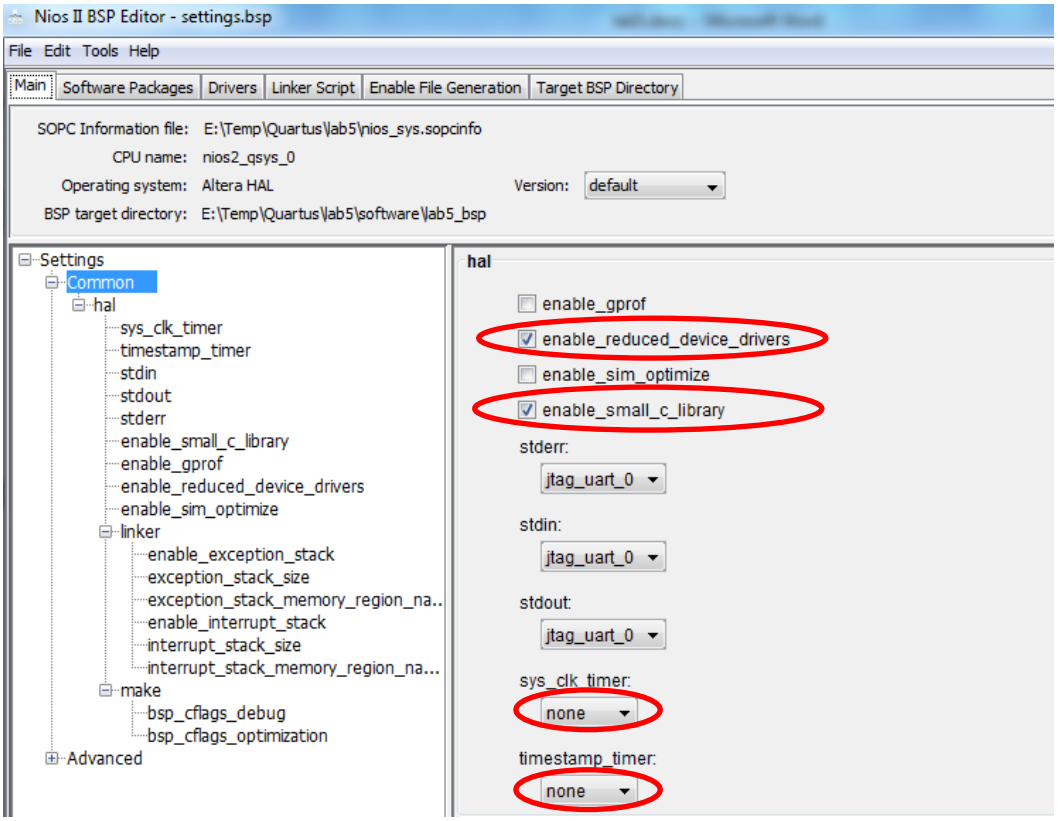
Tạo và đặt tên project là “lab5”.

Trong cửa sổ quản lý project, click chuột phải vào “lab5_bsp”,  chọn NIOS II > BSP Editor.



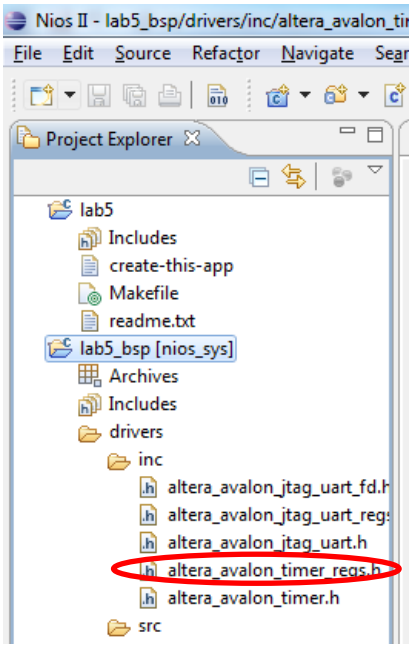

Tại tab “Main”, chọn “enable_reduced_device_drivers” và “enable_small_c_library” để giảm dung lượng thư viện chuẩn và driver.

Theo mặc định, chương trình sẽ dử dụng Timer duy nhất trong hệ thống làm “sys_clk” (thường dùng trong các hệ điều hành RTOS). Vì không sử dụng RTOS nên chúng ta hủy bỏ cấu hình “sys_clk” như hình bên dưới.




Click “Generate”, click “Exit”.

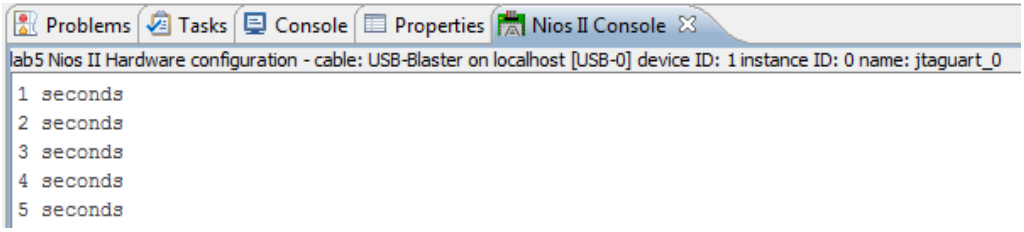
Trong thư mục “drivers/inc” của project “lab5_bsp” có file “altera_avalon_timer_regs.h”. Đây là file định nghĩa các thanh ghi Timer và các bit của các thanh ghi này. Chúng ta sẽ sử dụng file này để cấu hình Timer.



Thêm file “lab5.c” vào project “lab5”. File “lab5.c” có nội dung như bên dưới.

```c
/*
 * lab5.c
 */

#include <stdio.h>
#include "system.h"
#include "altera_avalon_timer_regs.h"
#include "sys/alt_irq.h"

// global variable, increase 1 after 1 second
unsigned int counter = 0;

// Interrupt handler of Timer
void Timer_IRQ_Handler(void* isr_context){

    // Increase counter and print it to console
    counter++;
    printf ("%d seconds\n",counter);

    // Clear Timer interrupt bit
    IOWR_ALTERA_AVALON_TIMER_STATUS(TIMER_0_BASE,ALTERA_AVALON_TIMER_STATUS_TO_MSK);
}

// Initialize function of Timer
void Timer_Init(void){

    unsigned int period = 0;

    // Stop Timer
    IOWR_ALTERA_AVALON_TIMER_CONTROL (TIMER_0_BASE,ALTERA_AVALON_TIMER_CONTROL_STOP_MSK);

    // The Timer's clock is 50 Mhz
    // In order to set the period of the Timer to 1 second, the period register must be (50000000-1)
    period = 50000000 - 1;
    IOWR_ALTERA_AVALON_TIMER_PERIODL (TIMER_0_BASE, period);
    IOWR_ALTERA_AVALON_TIMER_PERIODH (TIMER_0_BASE, (period>>16));

    // Configure and Start Timer
    IOWR_ALTERA_AVALON_TIMER_CONTROL (TIMER_0_BASE,ALTERA_AVALON_TIMER_CONTROL_CONT_MSK |   // Continue counting mode
                                                   ALTERA_AVALON_TIMER_CONTROL_ITO_MSK  |   // Interrupt enable
                                                   ALTERA_AVALON_TIMER_CONTROL_START_MSK);  // Start Timer
}

// entry point main()
int main(void){

    // Configure the Timer
    Timer_Init();

    // Register Timer's interrupt handler
    alt_ic_isr_register(0, TIMER_0_IRQ, Timer_IRQ_Handler, (void*)0, (void*)0);

    while(1);
}
```

Build và download phần mềm xuống board. Kiểm tra giá trị in ra màn hình console.



# BÁO CÁO THỰC HÀNH

## Bài 5 – GIAO TIẾP BỘ TIMER VÀ NGẮT

Sinh viên:

……...……

Lớp: …… Nhóm: ……..……..……..

Bài 1:

Sử dụng bộ Timer của Altera, xây dựng ứng dụng đồng hồ:

- Hai LED 7 đoạn hiển thị giờ, 2 LED 7 đoạn hiển thị phút, 2 LED 7 đoạn hiển thị giây
- 18 Switch được sử dụng để chỉnh giờ cho đồng hồ (cách chỉnh do sinh viên tự quy định)