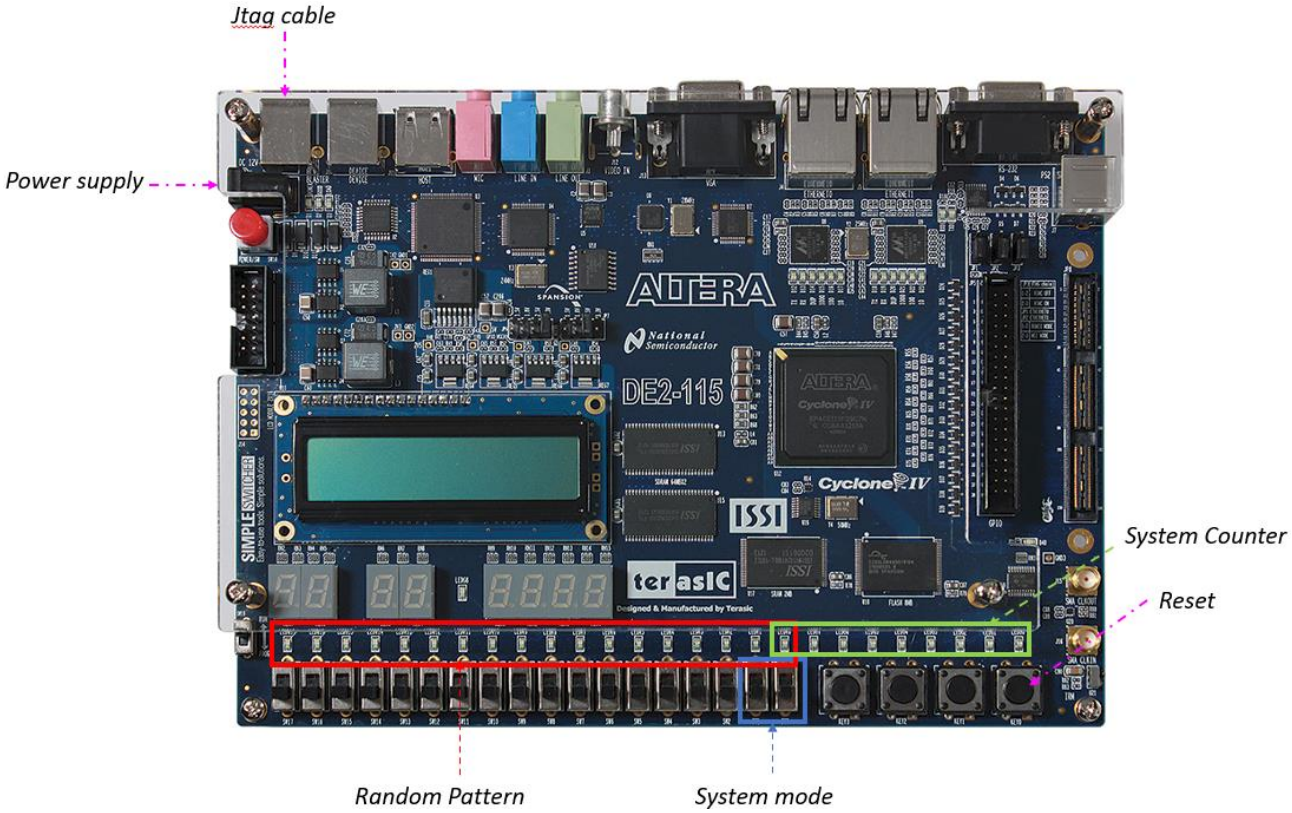
Project Objective

The main purpose of this project is to learn how to use the parallel I/O devices (peripherals) in embedded systems and how to use the PIO functions in your application software. The figure below shows the parallel I/O devices that you are required to add into your system.



Hardware Requirements

A. The design of Project1 will be used as a base and you will upgrade the design based on Project2 requirements. Before you start, make sure that both data and instruction cache of NIOS processor are set to **4KB** and Nios2 version is **Nios II/f.** Add the three PIO components which are shown in Table (1) and specify their features as in the table.

Table 1: PIO Required Features

PIO board device	Direction	Width	name	Conduit name
SW[1:0]	Input	2-bit	system_modes	modes
LEDG[7:0]	Output	8-bit	system_counter	cout
LEDR[17:0]	Output	18-bit	random_pattern	pattern

B. The first column shows that I/O pins that will be used in the Verilog code and connected with the exported signal of your Nios system.

C. The second and third columns show the PIO direction and width that you need to specify when you select the component from IP Catalog. Figure (1) shows how to find the PIO from the IP catalog and how to specify its features.

D. The fourth column shows the name that you need to rename the component with when you add it to your Nios system. The fifth column shows the signal name that will be exported in the conduit signal. Figure 2 shows how to rename the component and export a signal.

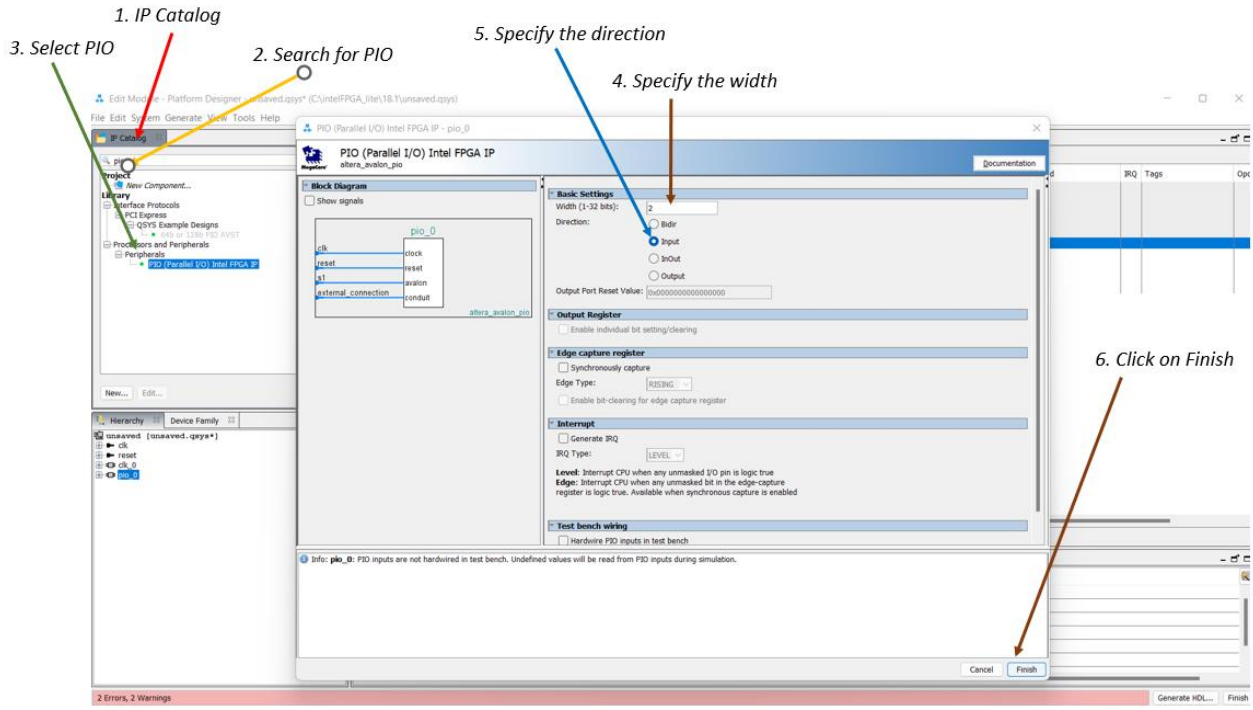


Figure 1: Find PIO and Specify its Features.

A. After you complete your platform design (*nios_system.qsys*), generate the system and click on Finish. Then, go back to quartus and Modify your test.v to include the following code (Type the given code into your test.v, as it might be encrypted if you copy and paste).Then, compile the design and program the device.

```
module test (
        // inputs of the top-level-design
        input CLOCK_50,                 // clock
        input [1:0] KEY,                // reset
        input [1:0] SW,
        output [17:0] LEDR,
        output [7:0] LEDG);
```

```
nios_system u0 (
        .clk_clk       (CLOCK_50),      //    clk.clk
        .reset_reset_n  (KEY[0]),       //  reset.reset_n
        .cout_export (LEDG),            // counter.export
        .modes_export (SW[1:0]),        // modes.export
        .pattern_export (LEDR)          // pattern.export
);

endmodule
```

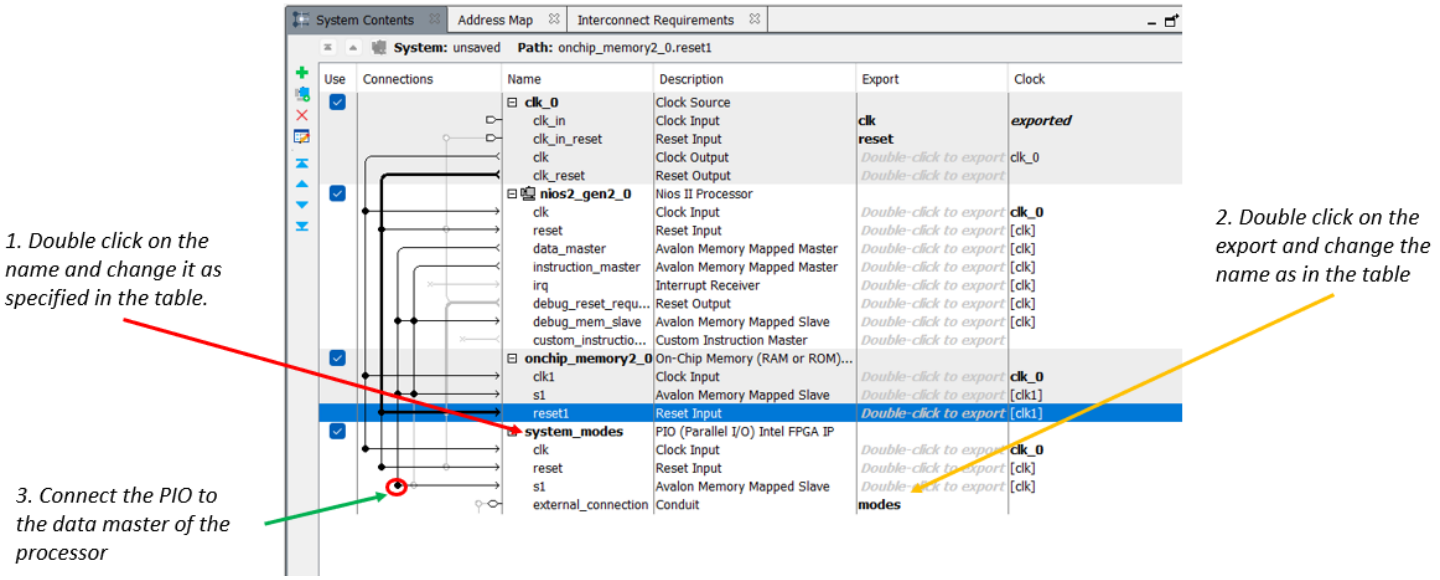


Figure 2: Rename the device and connect it to the processor.

Software Requirements

You are required to write an application software that controls the operation of the PIOs (switches and LEDs) that you added into your hardware system.

A. As shown in Table 2, there are three functions that the application will execute based on the three modes which are controlled using the switches. The output of the function will be displayed on the LEDs.

Table 2: Application Functions

SYSTEM MODE	SW [1:0]	FUNCTION
1	01	Both red and green LEDs light up
2	10	A counter starts incrementing from 0 to 255, and the value is displayed on Green LED [7:0]. If the mode changes, the counter stops on latest number. If the mode changes back to 1, the counter restarts.
3	11	A random pattern starts is displayed on Red LED [17:0]. If the mode changes, the pattern stops on latest value

A. To read from the input PIO (Switches), use *IORD_ALTERA_AVALON_PIO_DATA(baseAddress);*
B. To write on the output PIO (LEDs), use *IOWR_ALTERA_AVALON_PIO_DATA(baseAddress, value);*
C. To use both above functions, call this library *#include "altera_avalon_pio_regs.h"*
D. **Note:** A template of the application code is attached with assignment in blackboard. You may use it as a reference when you write the software.

Project Report (70%)

The project report will be graded out of 100, and the points will be distributed as following:

a. Professional preparation as described in Project1 (10 points).

b. Report Content (90 points):

1.0 *(15 points)* Use the system that you designed in this project as an example to briefly explain the meaning of System on Chip.

2.0 *(15 points)* Briefly explain the meaning of Avalon Memory Mapped interface. You may use the connections that you did in the system as examples.

3.0 *(15 points)* Briefly explain the meaning of Bare Metal Applications. You may use the application code that you developed in your application as an example.

4.0 *(10 points)* The following code is written in standard c code. Re-write the code using the hardware C code, where a is an input coming from switches and b is an output going out to the Green LEDs. In your code, include any necessary libraries, variables, and definitions.

```
#include <stdio.h>
#include <time.h>
#include <unistd.h>
#include <stdlib.h>


int main()
{
  printf("\n");

      int a, b;
      while(1) {

              if(a==0) printf("no output\n);
              else if (a==1) b = 10;
              usleep(1000000);
 }
return 0;
}
```

5.0 (*20 points total, each value is 5 points*) After you compiled and synthesized your system, read the summary report from Quartus, and fill out the below table with the numbers from the report.

Logical Elements	Registers	Total Pins	Memory Bits

6.0 (*15 points, each picture is 5 points*) Run the application software on the three different mode and include a picture that shows the output on the console for each case. You are required to submit only one picture per case.

Project Demo (30%)

- The main purpose of the demo is to test your project functionality and execution.
- Demos will be checked and graded by the TA.
- Demos will be graded out of 100, but worth 30% of total project grade.
- Demos will be conducted during the lab time on the following dates:
  - **Section 001**: Wed. Feb 1 or Wed Feb. 8
  - **Section 002**: Fri Feb 3 or Fri Feb. 10
  - Demo dates will be decided by the groups.
- Below are how the demo points will be distributed.
- Both partners must show up in that day. If a member didn't show up, he/she receives 0 unless an excused absence was provided.

Tasks	Point
LEDs lights (mode = 1)	/25
Random pattern (mode = 2)	/25
Counter (mode = 3)	/25
Answer questions	/25

Project Submission

1. Save the project report as **r2_username1_username2.pdf**, username of both students in the group.
2. **Only one attempt** is allowed.
3. **Only one group member** submits the project.
4. **Remember:** Any grade dispute must be raised within one week of the grade posting.