Text, logo

Description automatically generated **San Francisco Bay University**

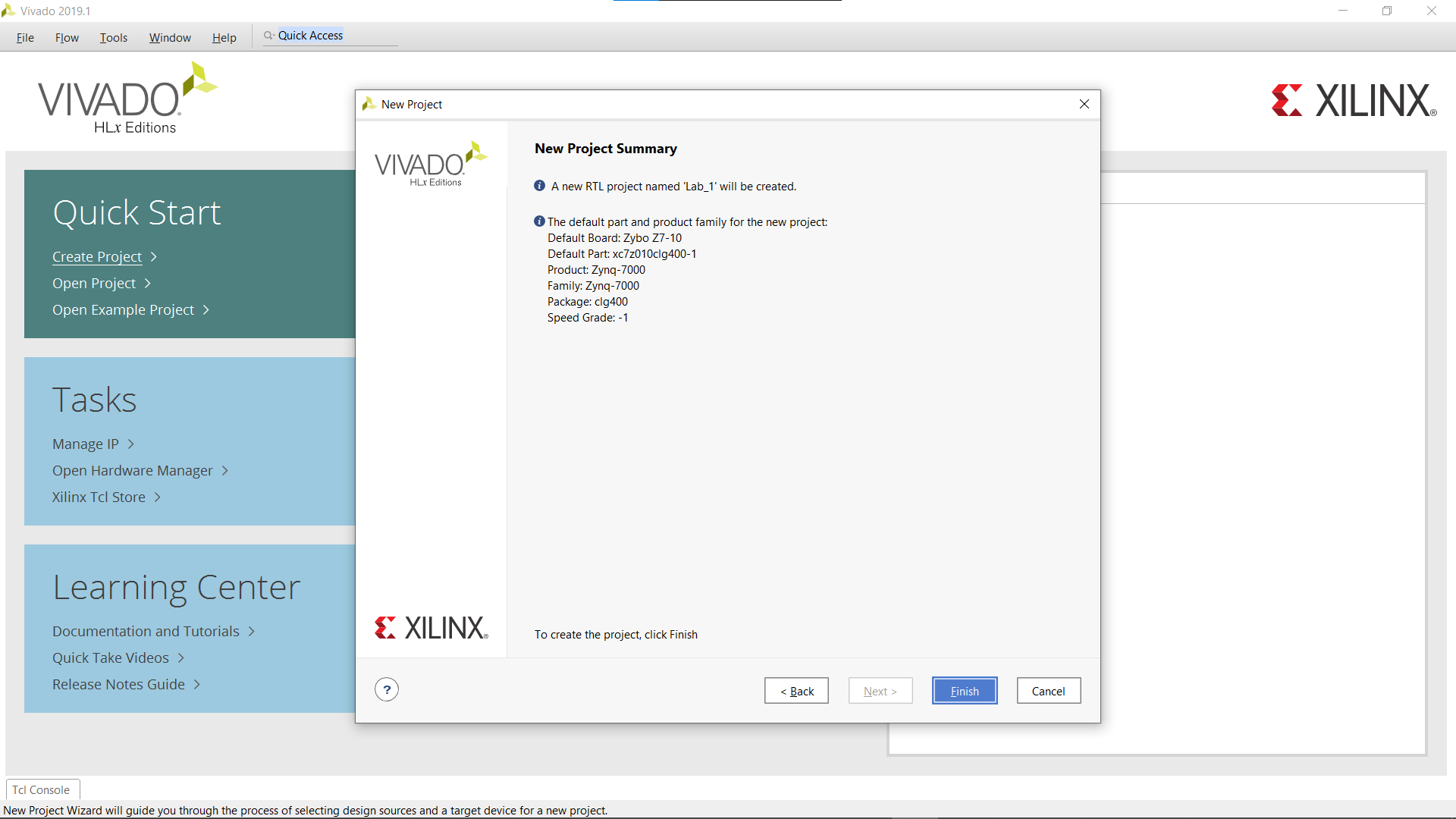
**EE553 - System on Chip (SoC) Design**

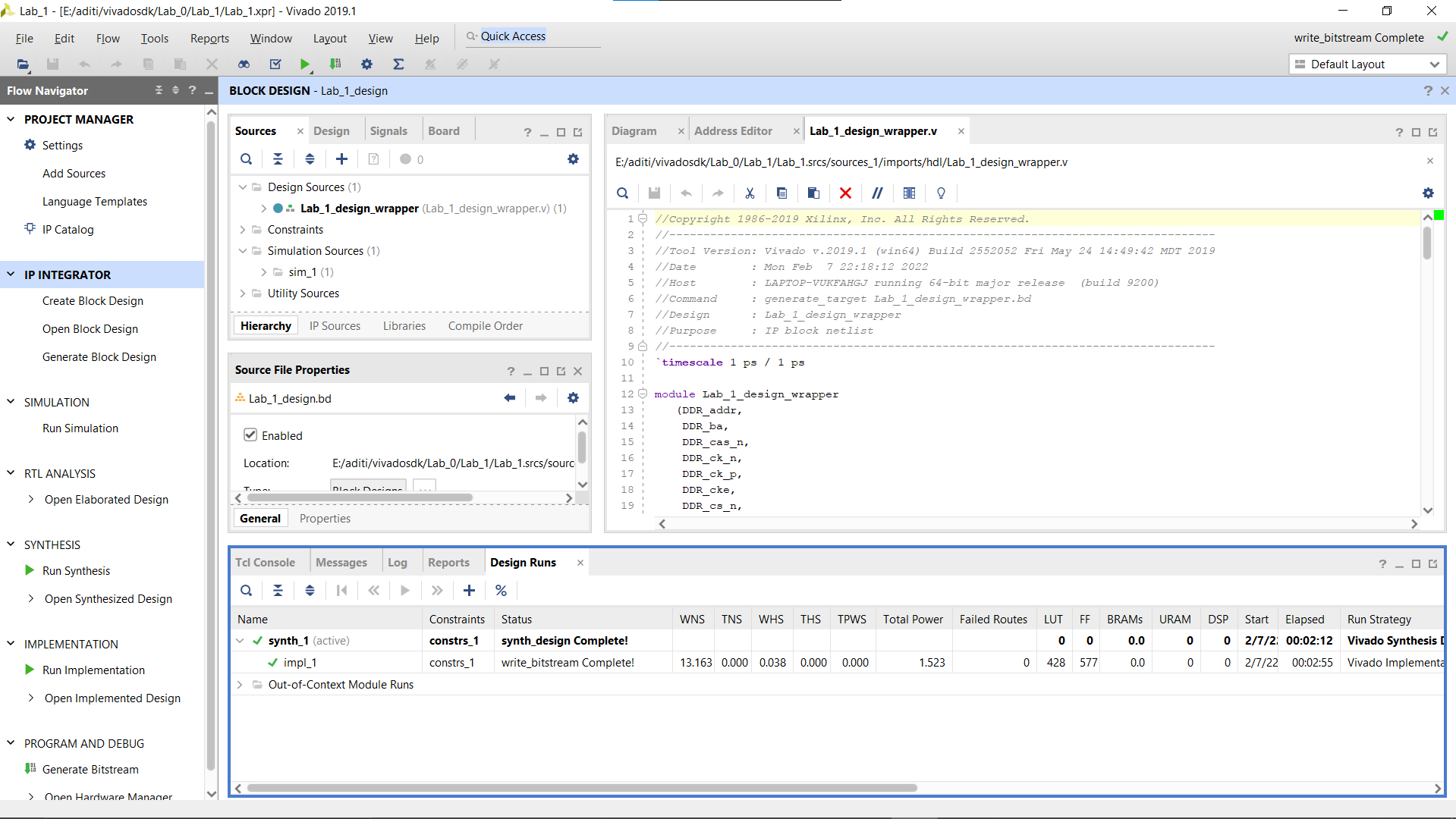
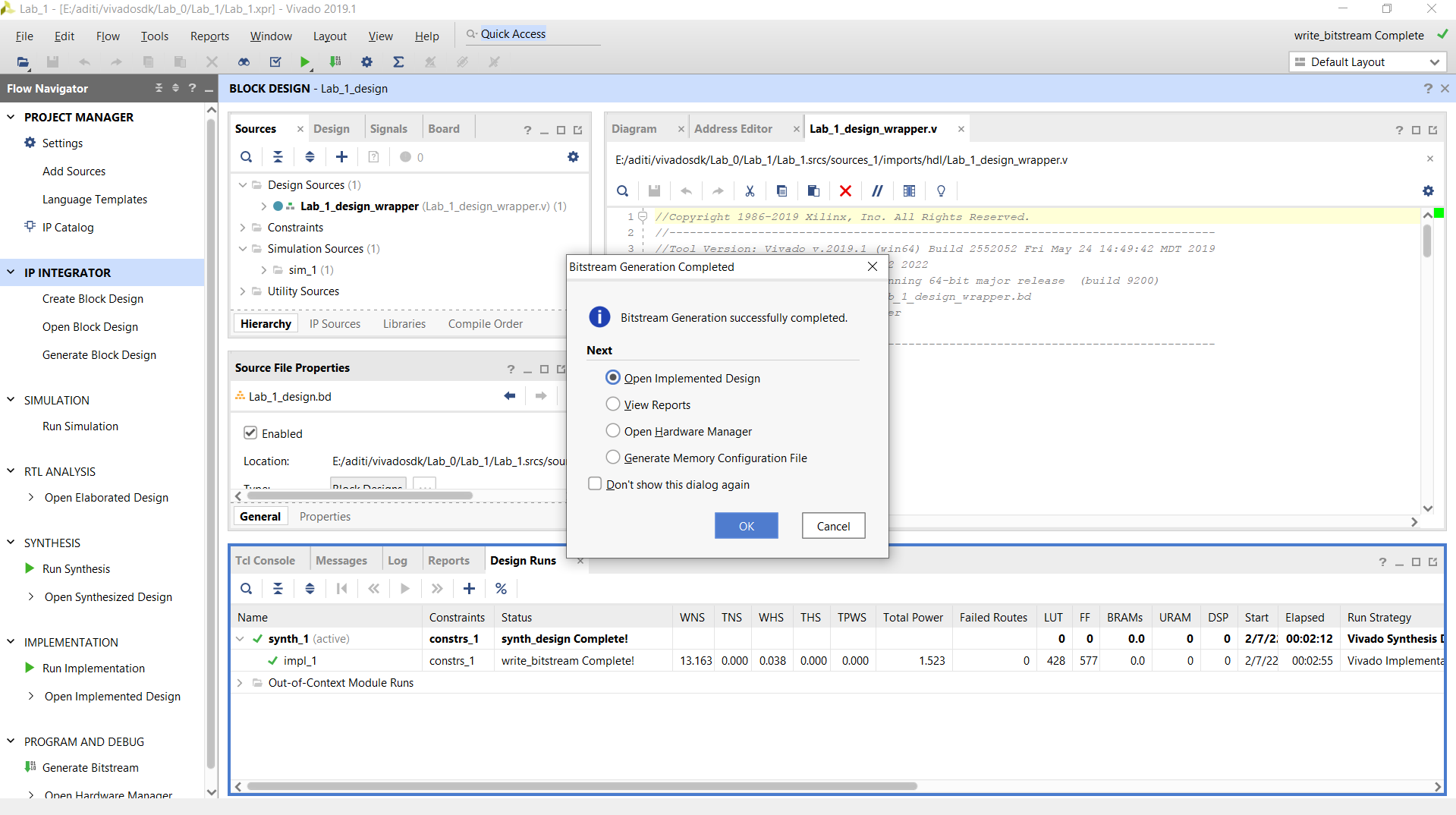
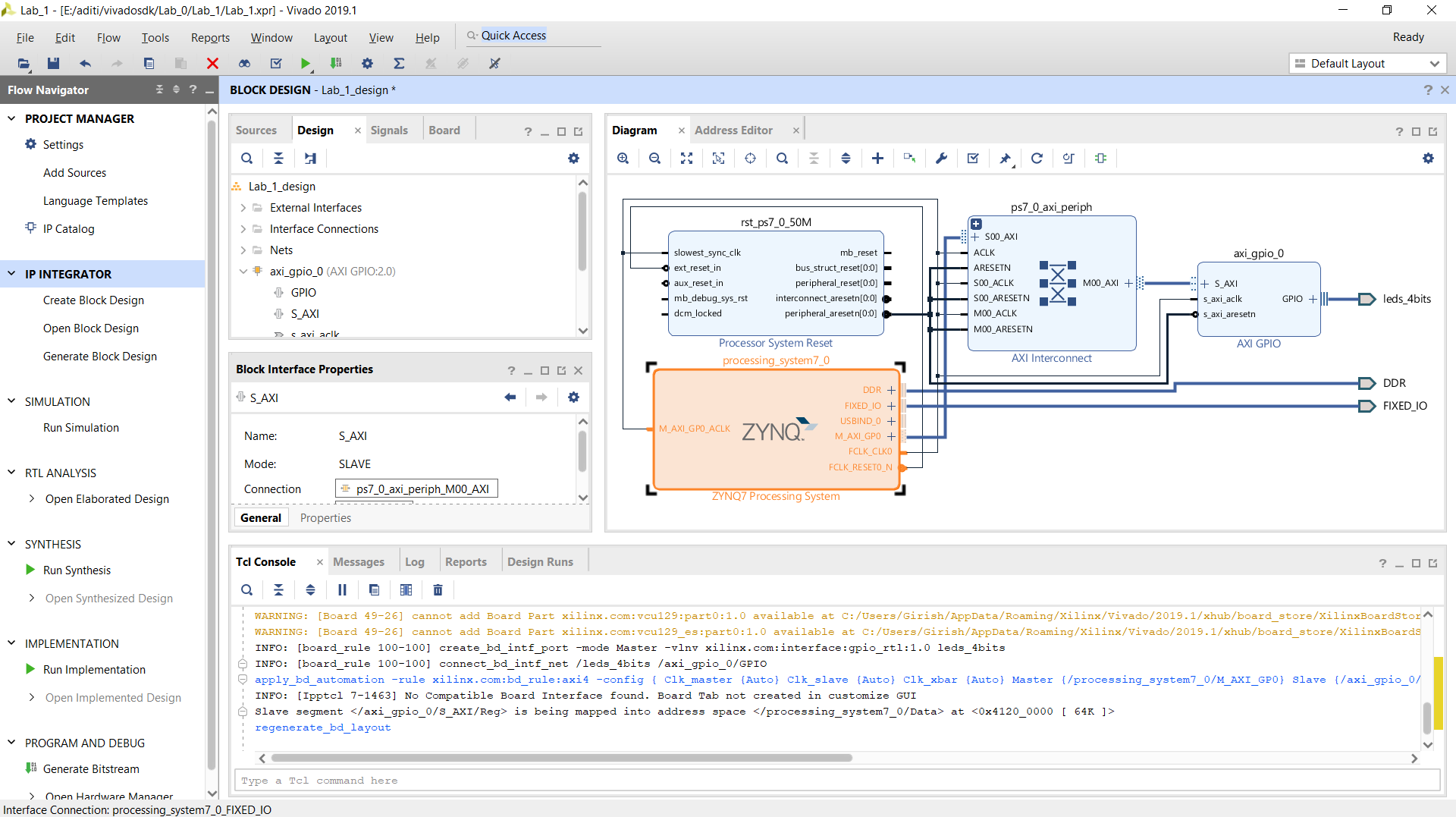
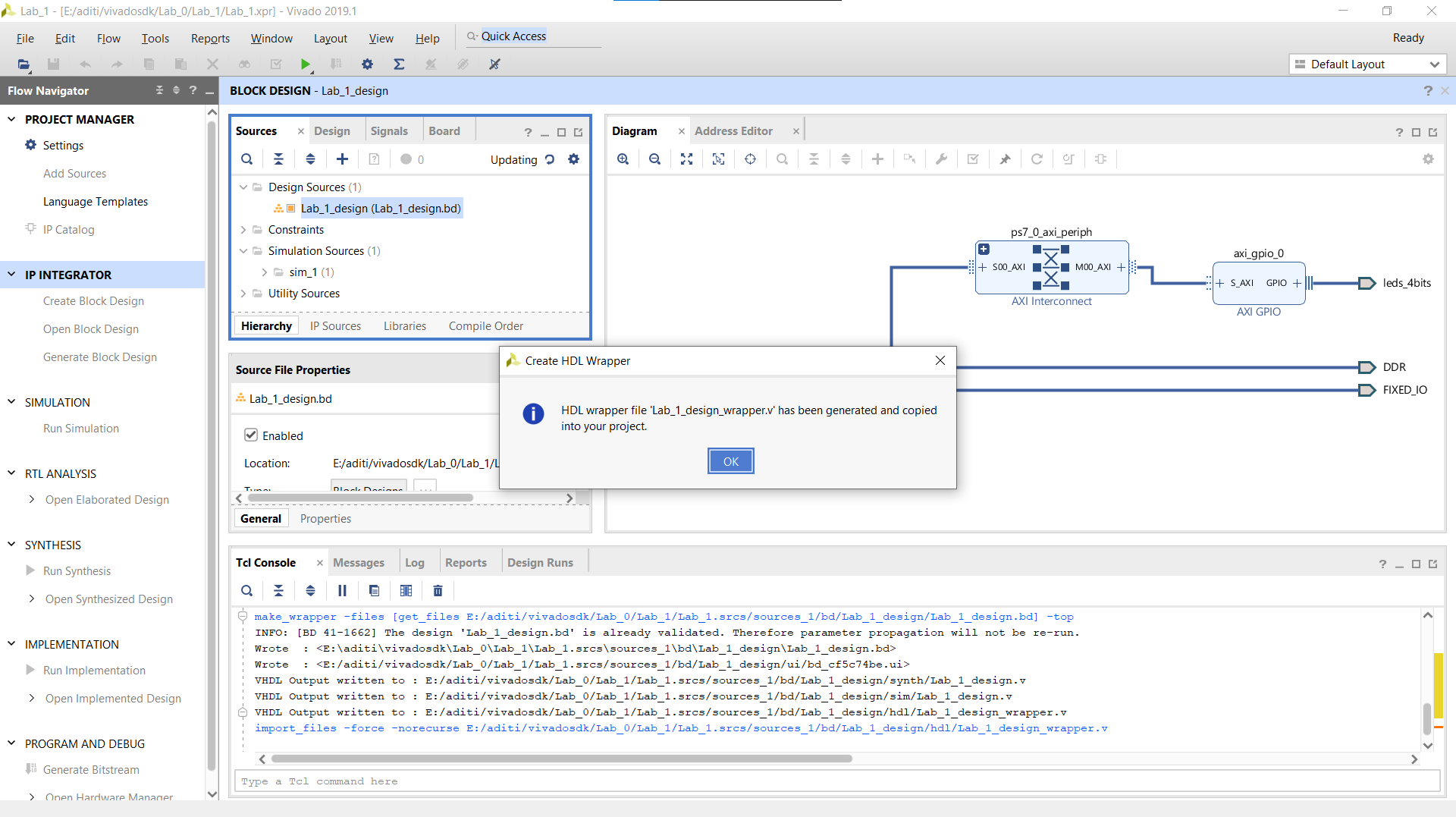
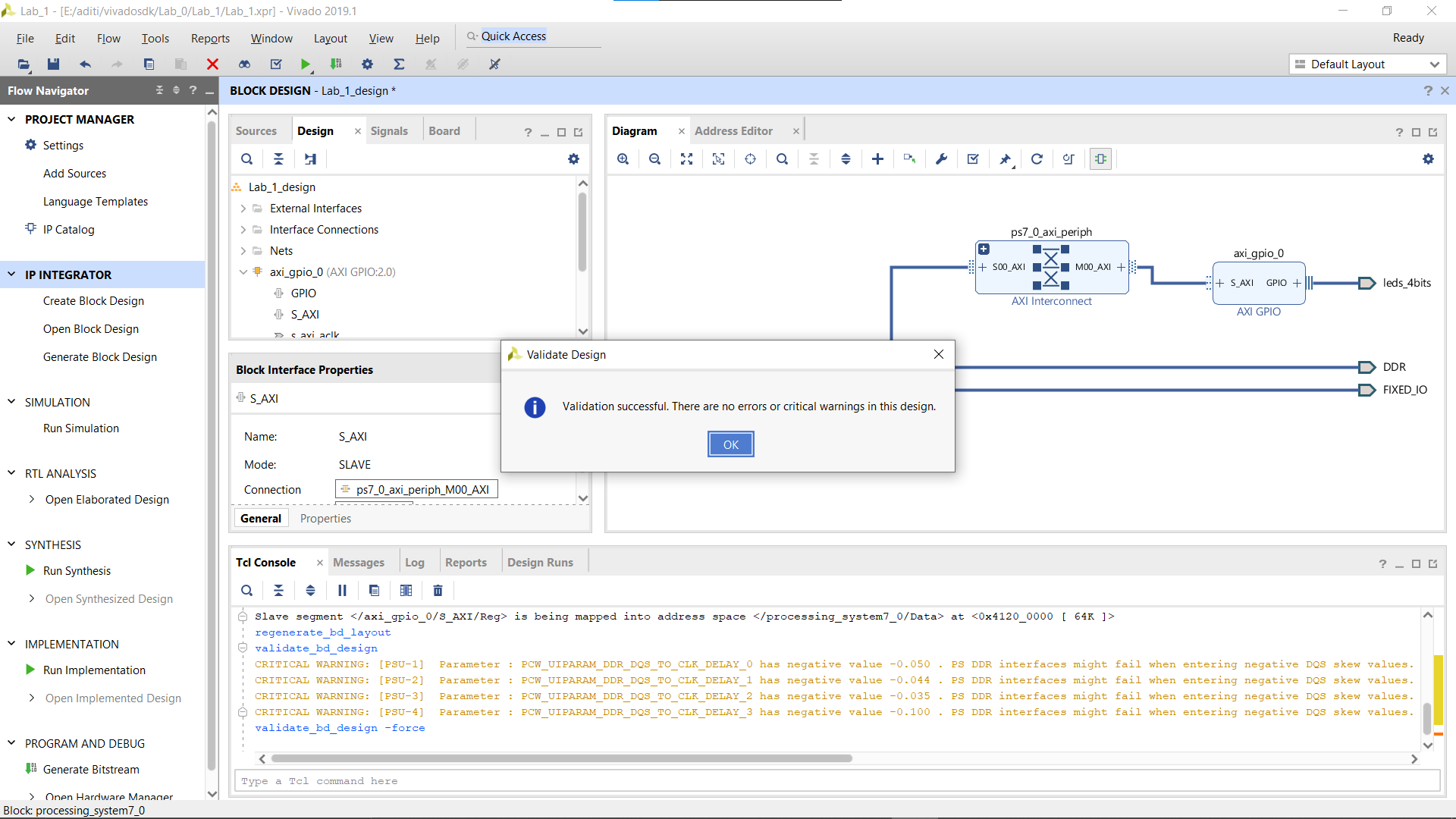
**Homework Assignment #3**

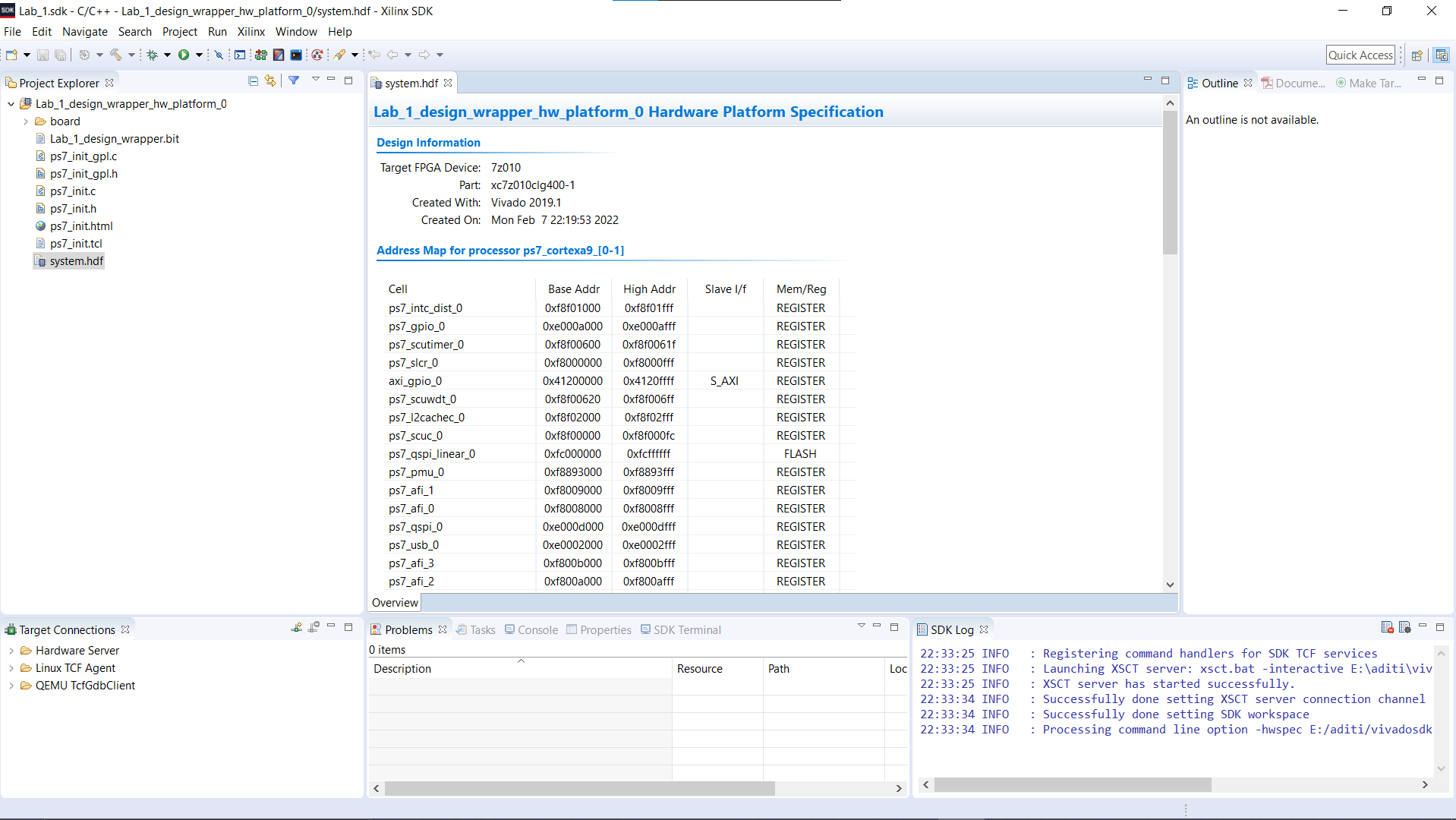
**Due day: 3/9/2022**

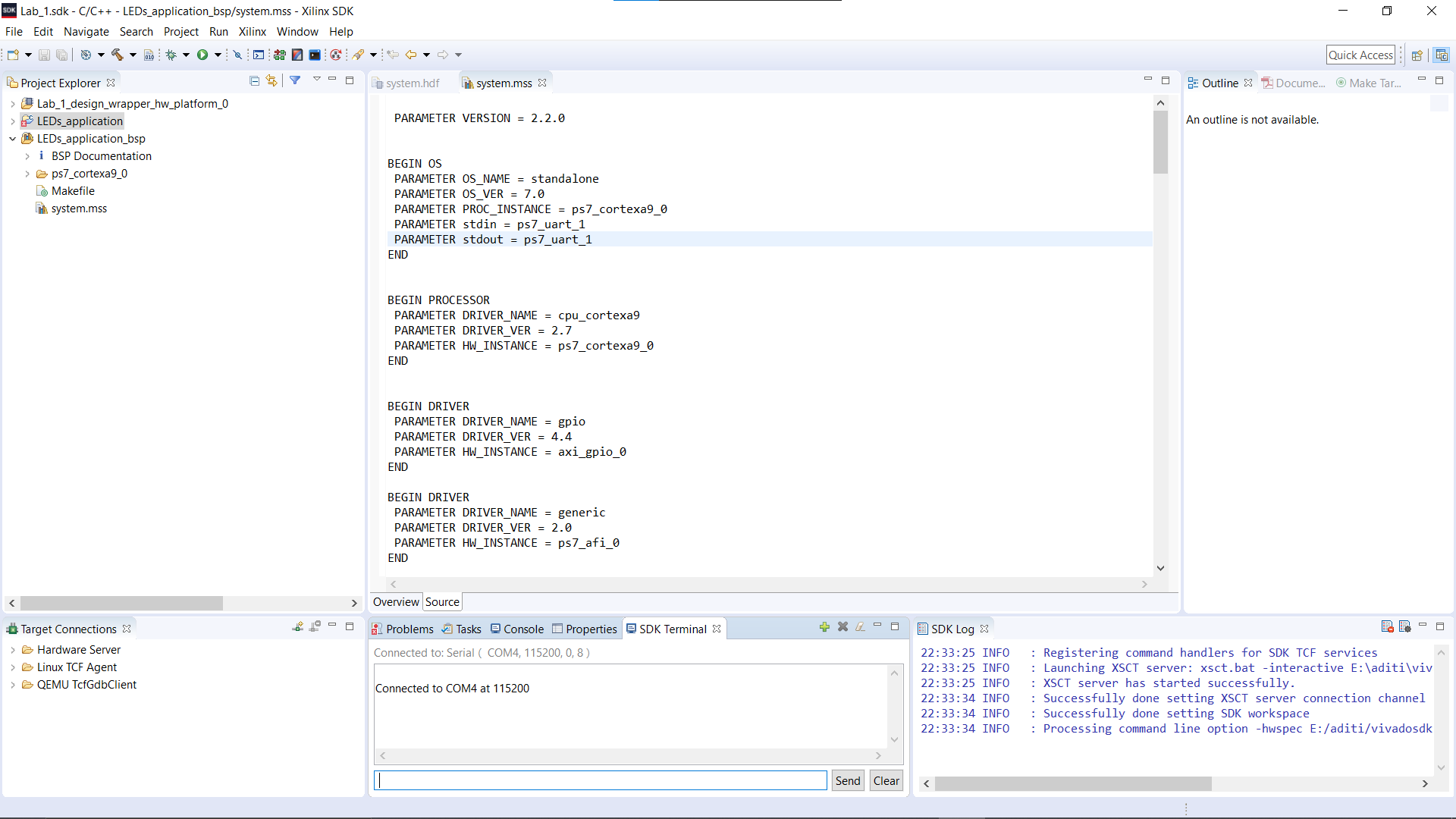
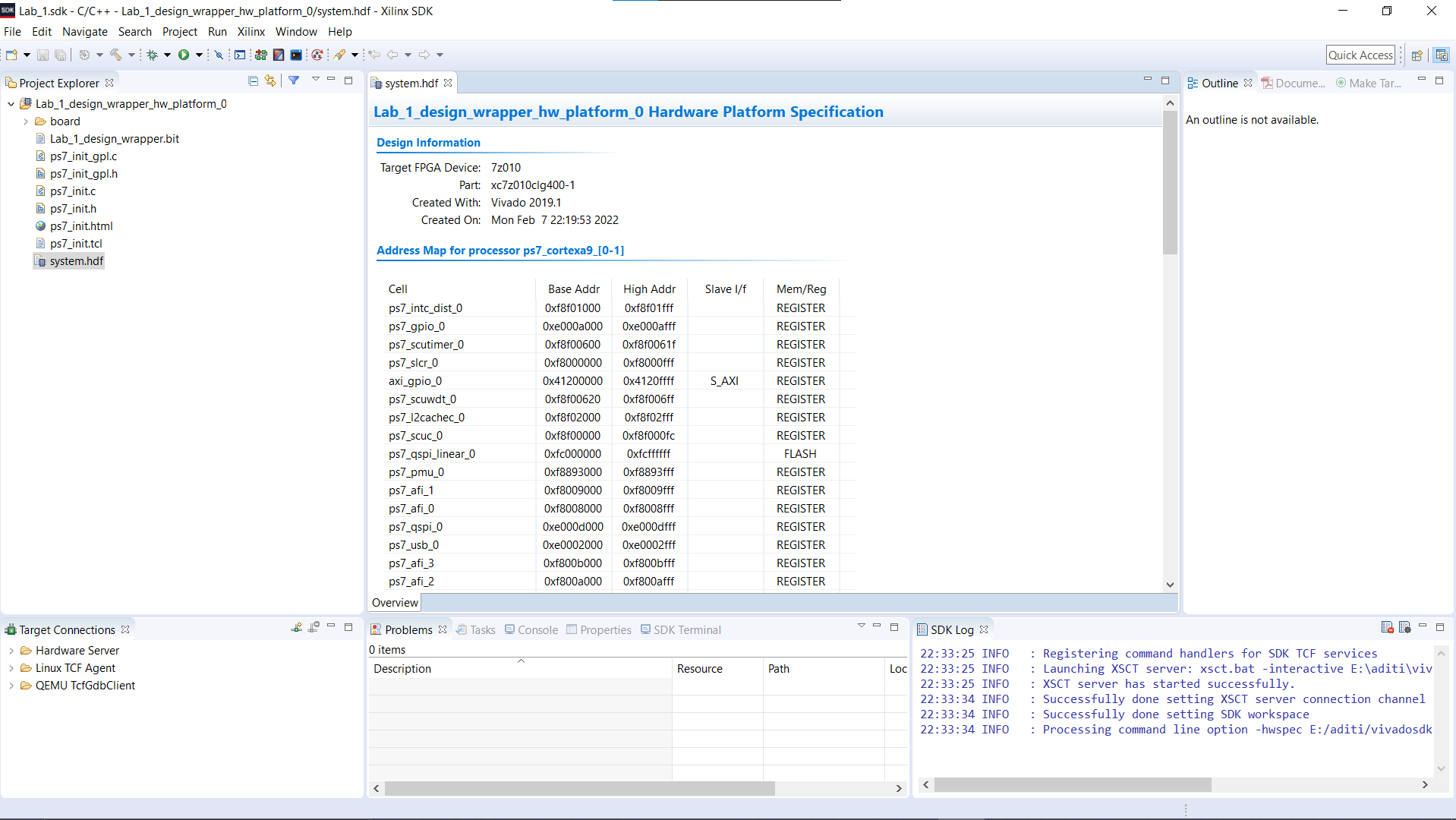
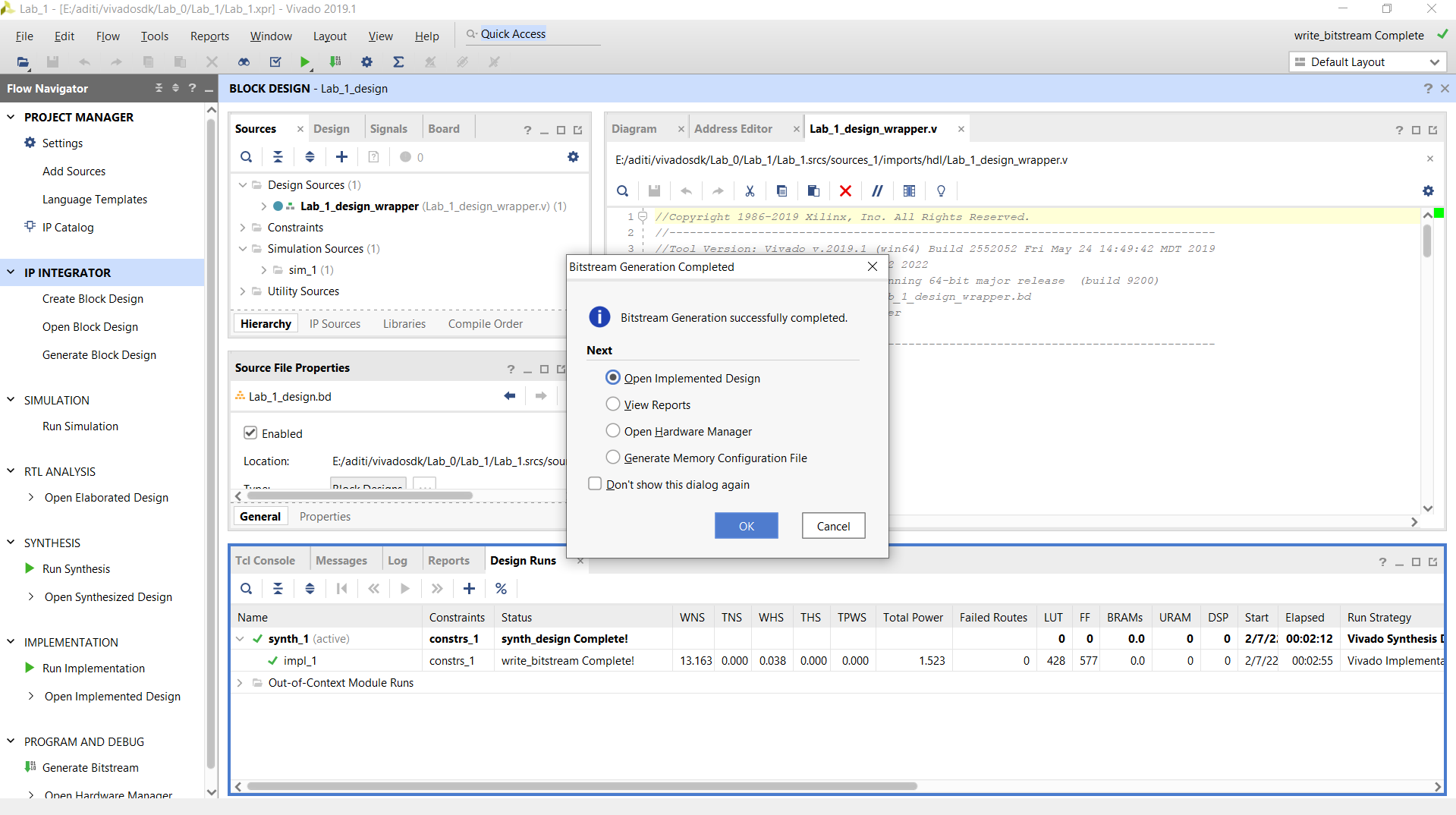
**Instruction:**

1. **Push the answer sheet to Github**
2. **Overdue homework submission could not be accepted.**
3. **Takes academic honesty and integrity seriously (Zero Tolerance of Cheating & Plagiarism)**
4. Complete SoC design to blink one LED and shift back and forth among 4 LEDs on Zybo Z7-10 board









CODE-

#include "xparameters.h"

#include "xgpio.h"

#include "xstatus.h"

#include "xil\_printf.h"

/\* Definitions \*/

#define GPIO\_DEVICE\_ID XPAR\_AXI\_GPIO\_0\_DEVICE\_ID

#define LED 0x9

#define LED\_DELAY 10000000

#define LED\_CHANNEL 1

#define printf xil\_printf

XGpio Gpio;

int LEDOutputExample(void)

{

volatile int Delay;

int Status;

int led = LED; /\* Hold current LED value. Initialise to LED definition \*/

/\* GPIO driver initialisation \*/

Status = XGpio\_Initialize(&Gpio, GPIO\_DEVICE\_ID);

if (Status != XST\_SUCCESS) {

return XST\_FAILURE;

}

/\*Set the direction for the LEDs to output. \*/

XGpio\_SetDataDirection(&Gpio, LED\_CHANNEL, 0x0);

/\* Loop forever blinking the LED. \*/

while (1) {

/\* Write output to the LEDs. \*/

XGpio\_DiscreteWrite(&Gpio, LED\_CHANNEL, led);

/\* Flip LEDs. \*/

led = ~led;

/\* Wait a small amount of time so that the LED blinking is visible. \*/

for (Delay = 0; Delay < LED\_DELAY; Delay++);

}

return XST\_SUCCESS;

}

/\* Main function. \*/

int main(void){

int Status;

/\* Execute the LED output. \*/

Status = LEDOutputExample();

if (Status != XST\_SUCCESS) {

xil\_printf("GPIO output to the LEDs failed!\r\n");

}

return 0;

}

/\* Include Files \*/

#include "xparameters.h"

#include "xgpio.h"

#include "xstatus.h"

#include "xil\_printf.h"

/\* Definitions \*/

#define GPIO\_DEVICE\_ID XPAR\_AXI\_GPIO\_0\_DEVICE\_ID /\* GPIO device that LEDs are connected to \*/

#define LED 0x1 /\* Initial LED value - X00X \*/

#define LED1 0x8

#define LED\_DELAY 100000000 /\* Software delay length \*/

#define LED\_CHANNEL 1 /\* GPIO port for LEDs \*/

#define printf xil\_printf /\* smaller, optimised printf \*/

XGpio Gpio; /\* GPIO Device driver instance \*/

int LEDOutputExample(void)

{

volatile int Delay;

int Status;

int led = LED; /\* Hold current LED value. Initialise to LED definition \*/

int led1=LED1;

/\* GPIO driver initialisation \*/

Status = XGpio\_Initialize(&Gpio, GPIO\_DEVICE\_ID);

if (Status != XST\_SUCCESS) {

return XST\_FAILURE;

}

/\*Set the direction for the LEDs to output. \*/

XGpio\_SetDataDirection(&Gpio, LED\_CHANNEL, 0x0);

/\* Loop forever blinking the LED. \*/

while (1) {

/\* Write output to the LEDs. \*/

XGpio\_DiscreteWrite(&Gpio, LED\_CHANNEL, led);

XGpio\_DiscreteWrite(&Gpio, LED\_CHANNEL, led1);

led = led<<0x1;

led1 = led1>>0x1;

for (Delay = 0; Delay < LED\_DELAY; Delay++);

}

return XST\_SUCCESS; /\* Should be unreachable \*/

}

/\* Main function. \*/

int main(void){

int Status;

/\* Execute the LED output. \*/

Status = LEDOutputExample();

if (Status != XST\_SUCCESS) {

xil\_printf("GPIO output to the LEDs failed!\r\n");

}

return 0;

}

1. Design hardware and software to make 2 LEDs at the two ends of 4 LEDs on the board blinking and move in different directions, and then back and forth

