ADIUVO

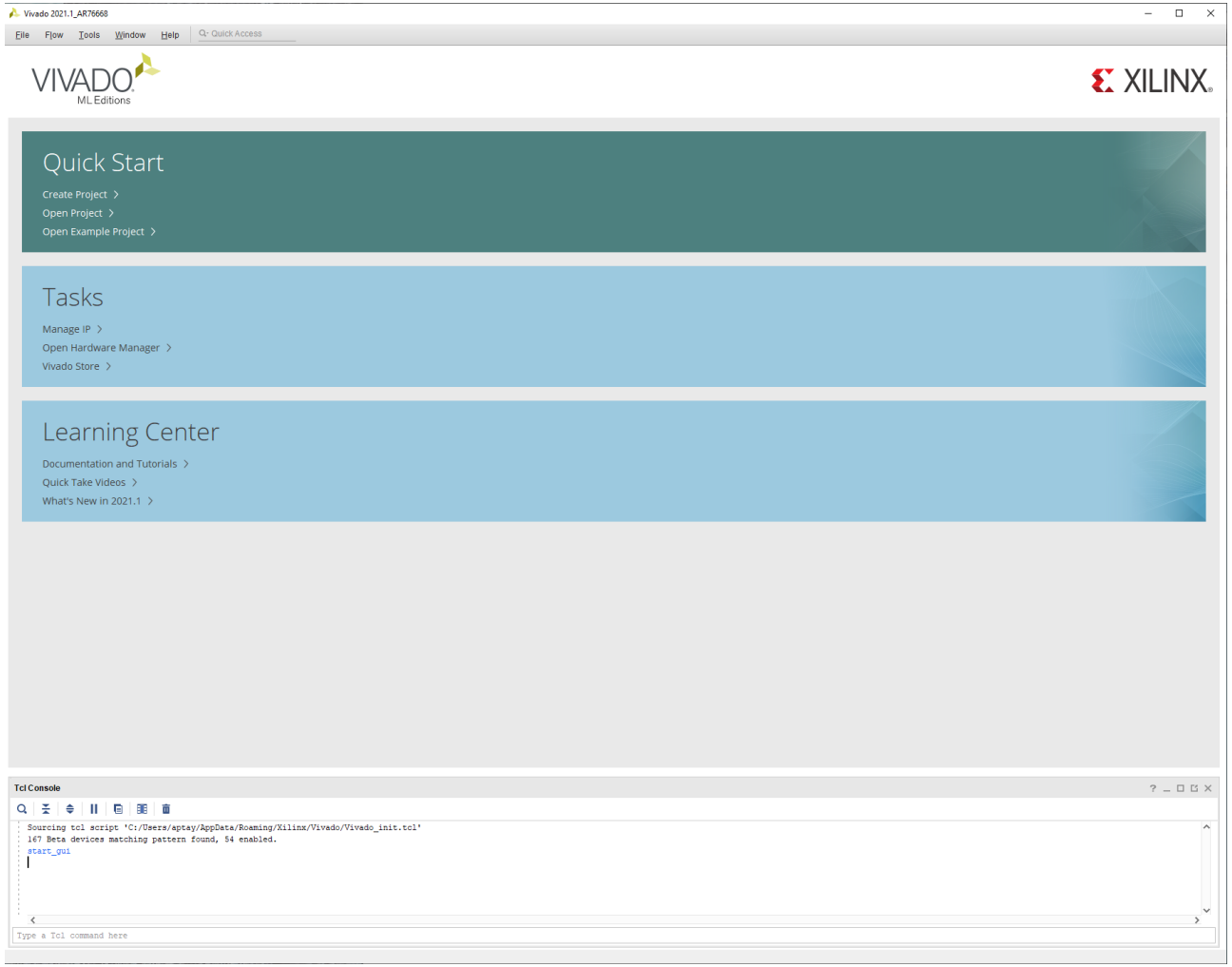
ENGINEERING AND TRAINING, LTD.

Mastering MicroBlaze

Lab Book

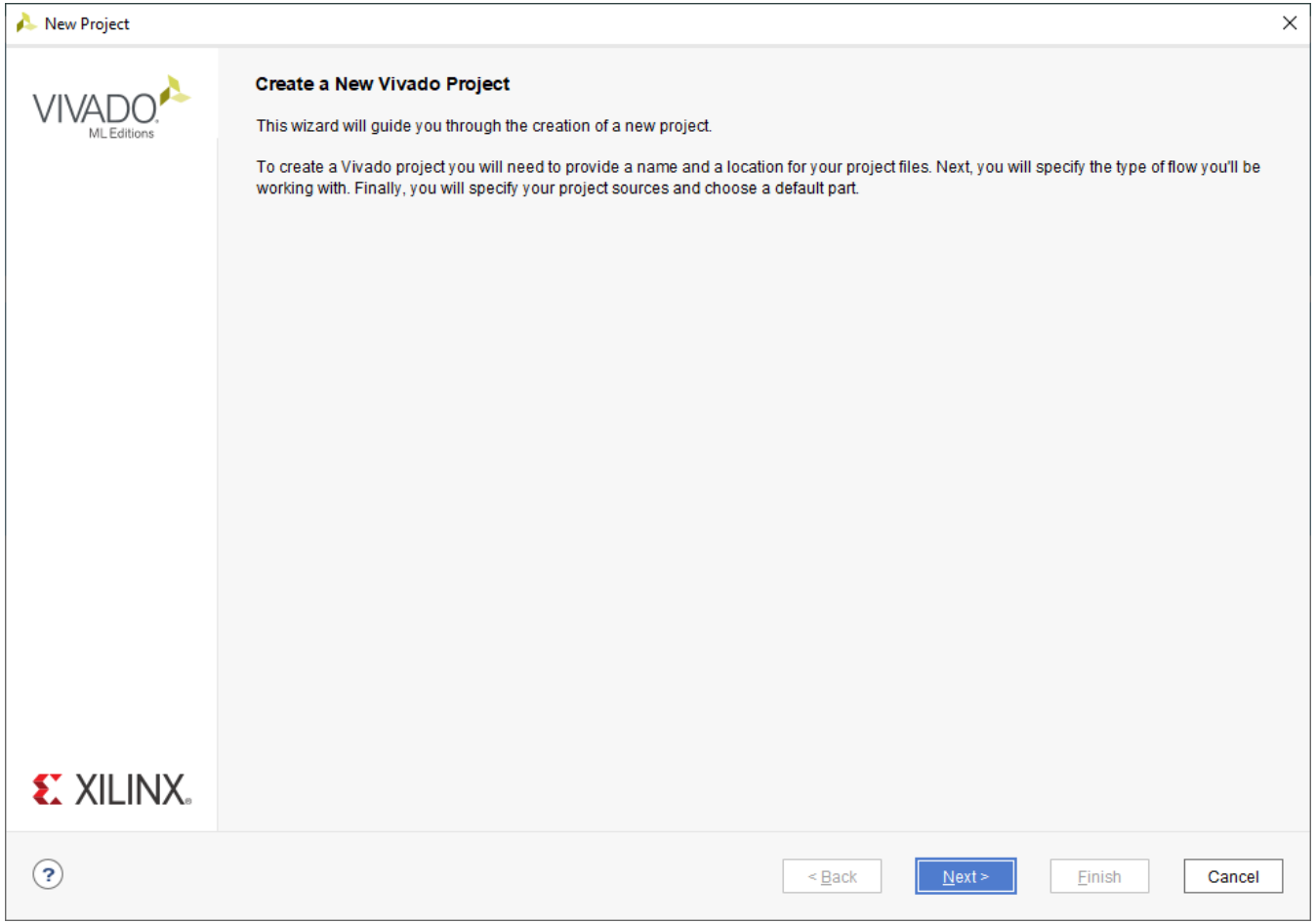
Lab: Mastering MicroBlaze

Open Vivado and Create a new project



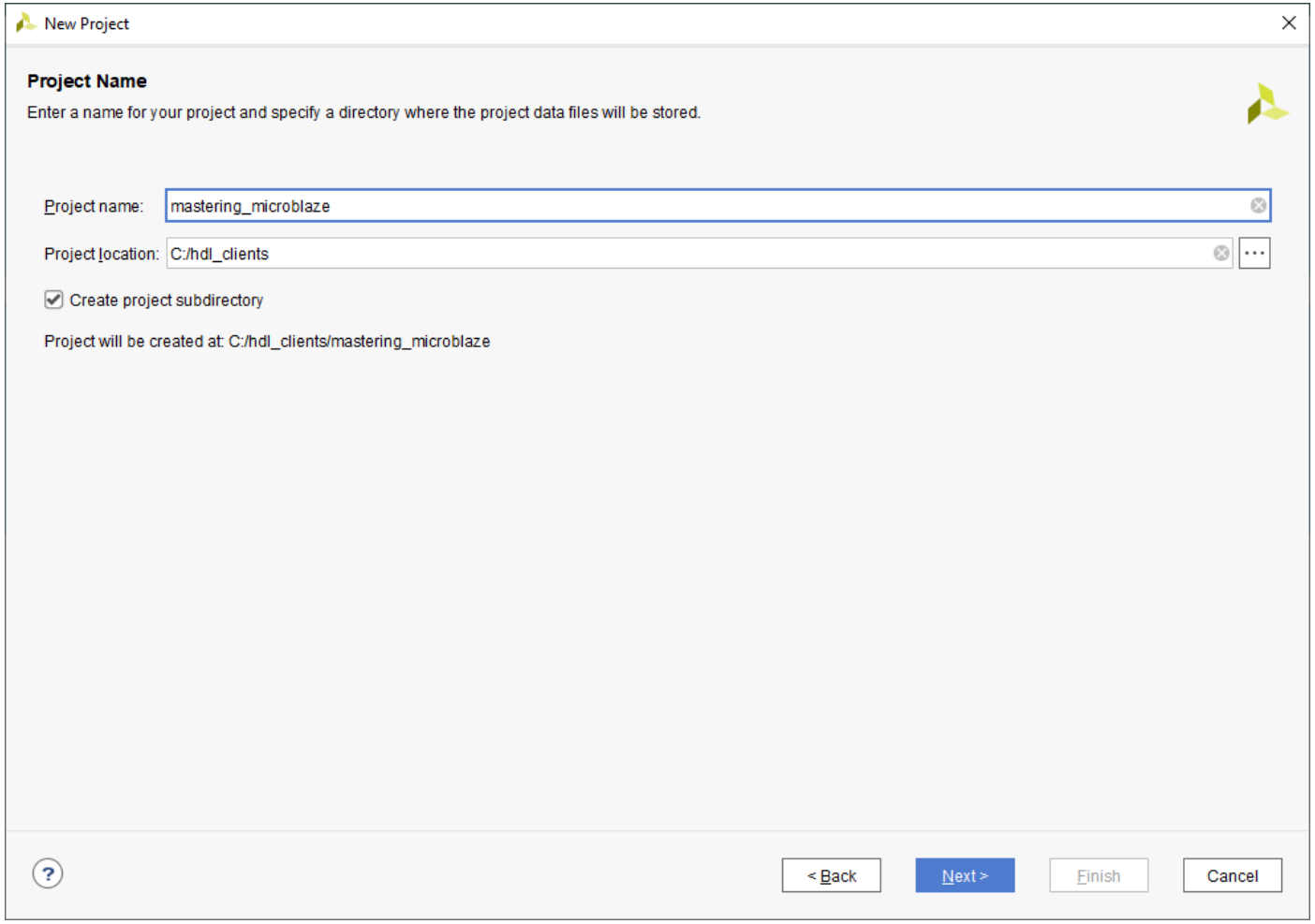
Lab: Mastering MicroBlaze

Click Next



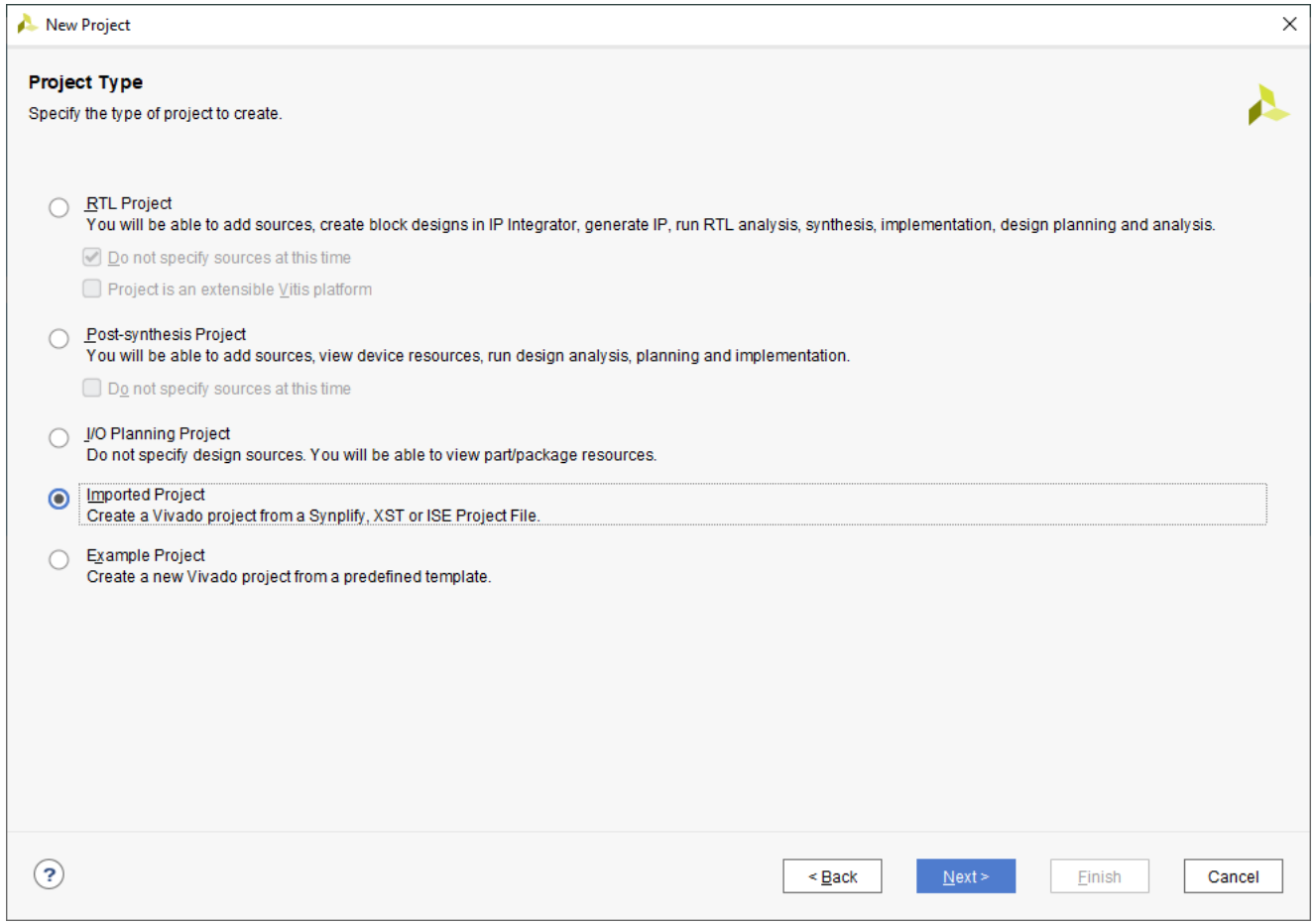
Lab: Mastering MicroBlaze

Enter a location and project name



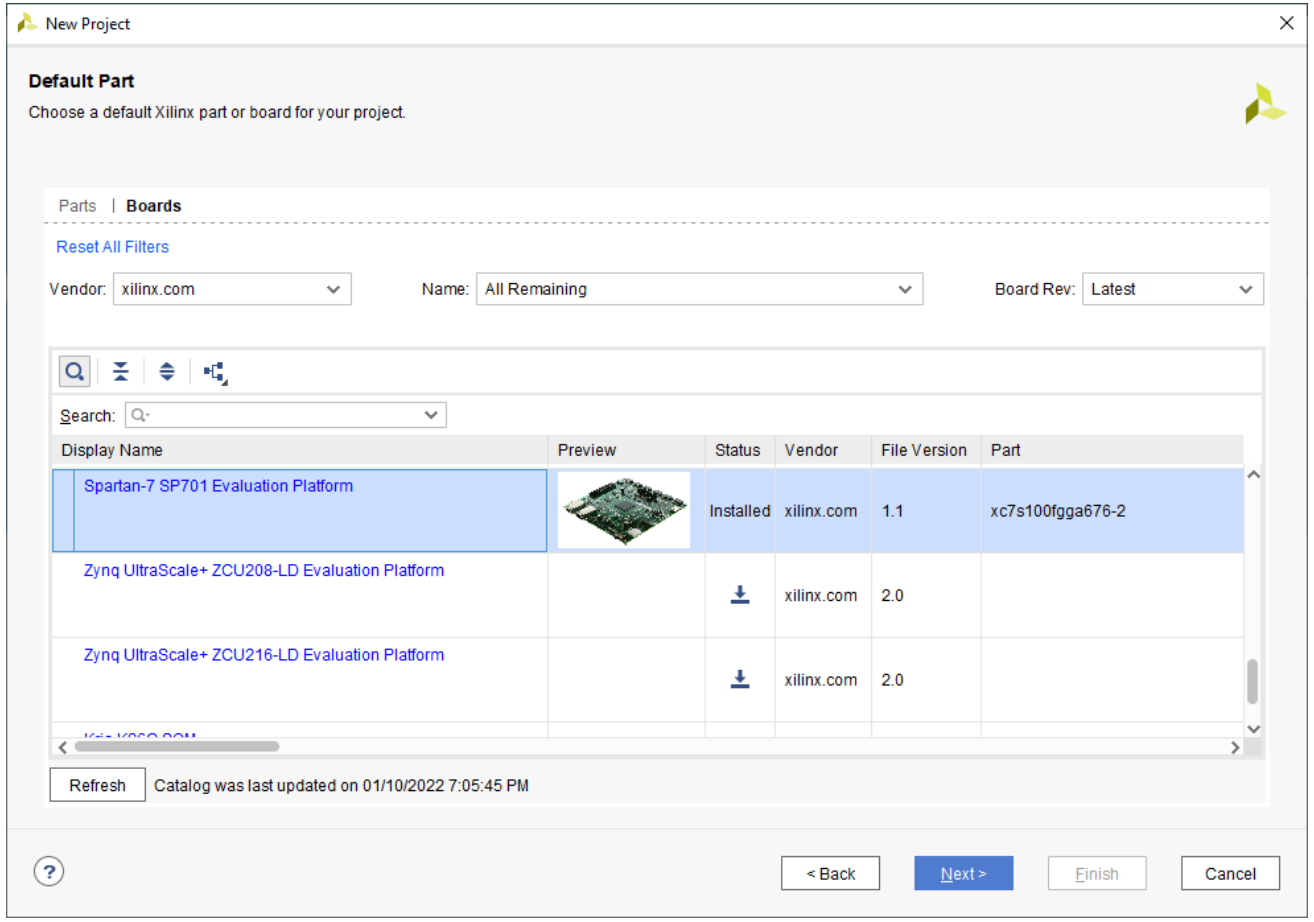
Lab: Mastering MicroBlaze

Select project type as RTL Project and check Do Not Specify Sources at this time.



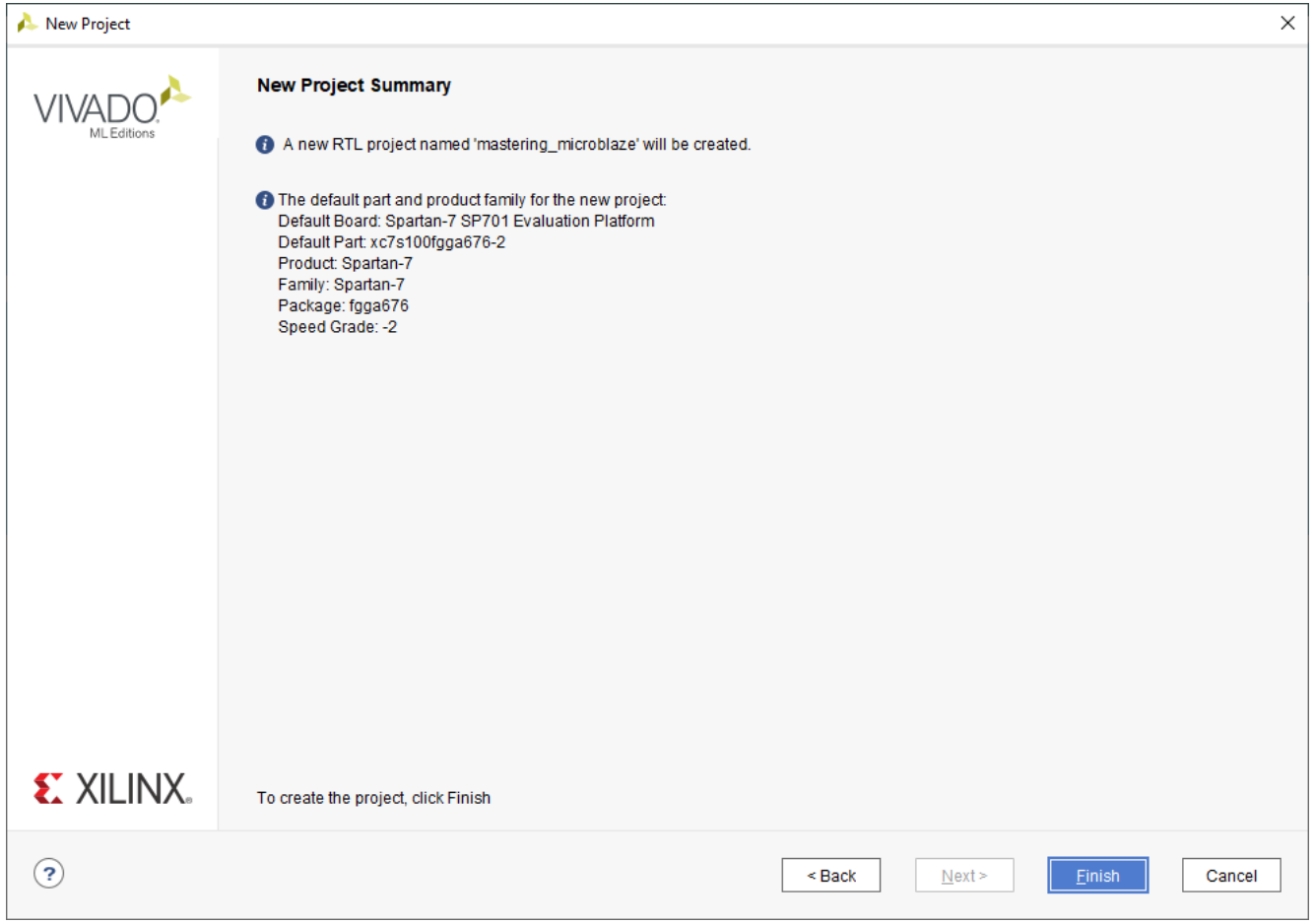
Lab: Mastering MicroBlaze

Select your development board – I will be using the SP701. But ANY 7 Series / UltraScale (+) board with a LED connected to the IO should work.



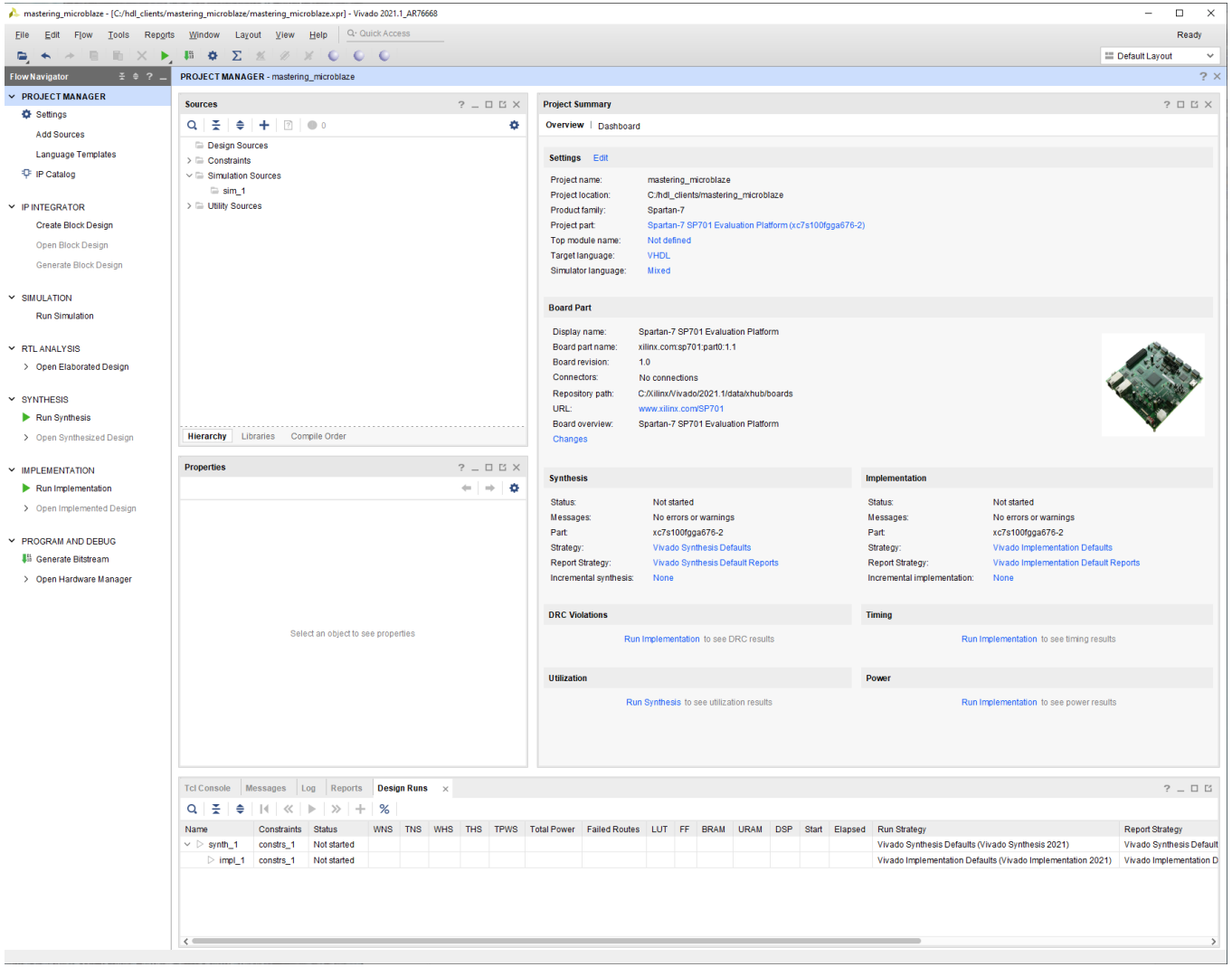
Lab: Mastering MicroBlaze

Click Finish



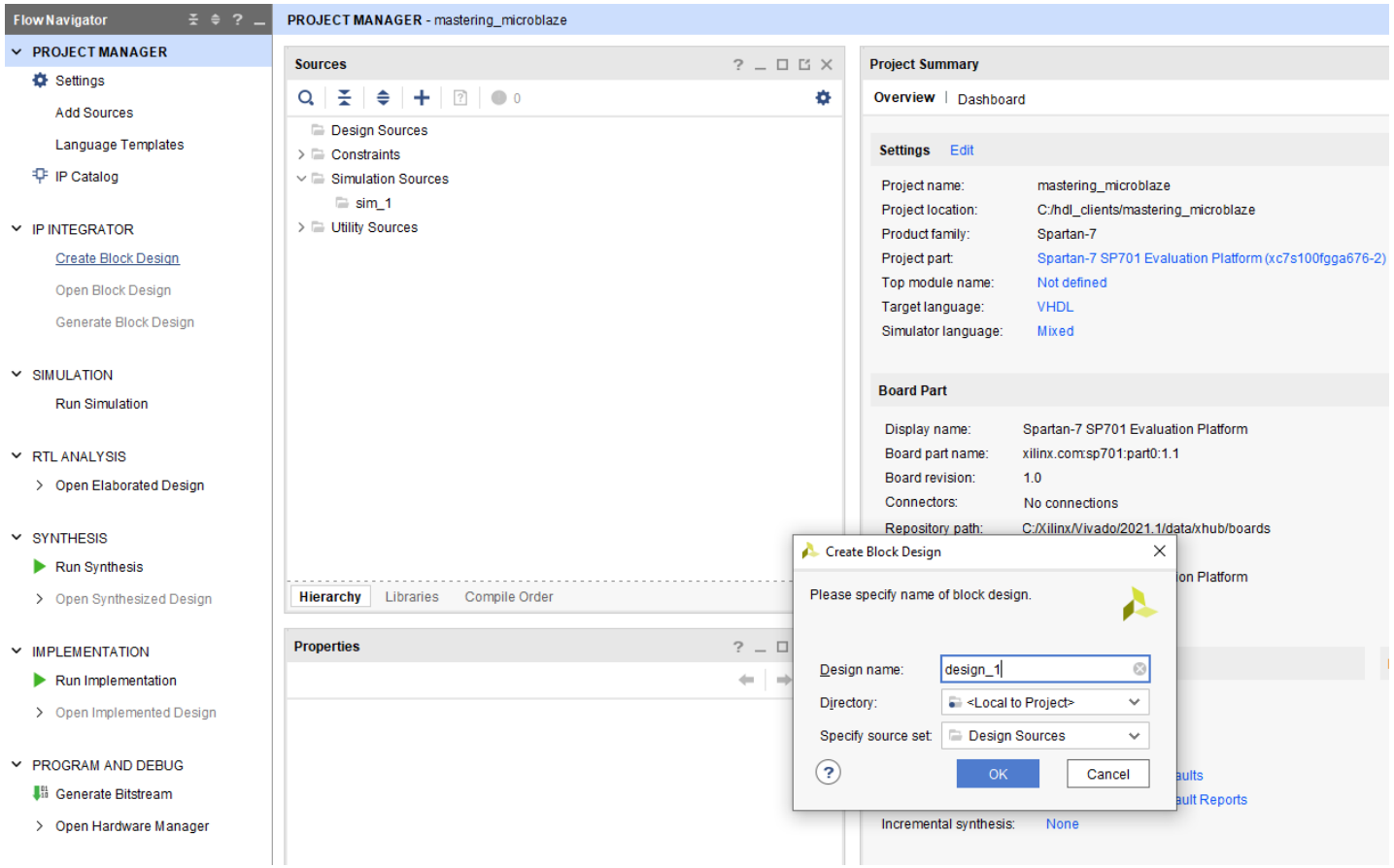
Lab: Mastering MicroBlaze

The open project should look as below



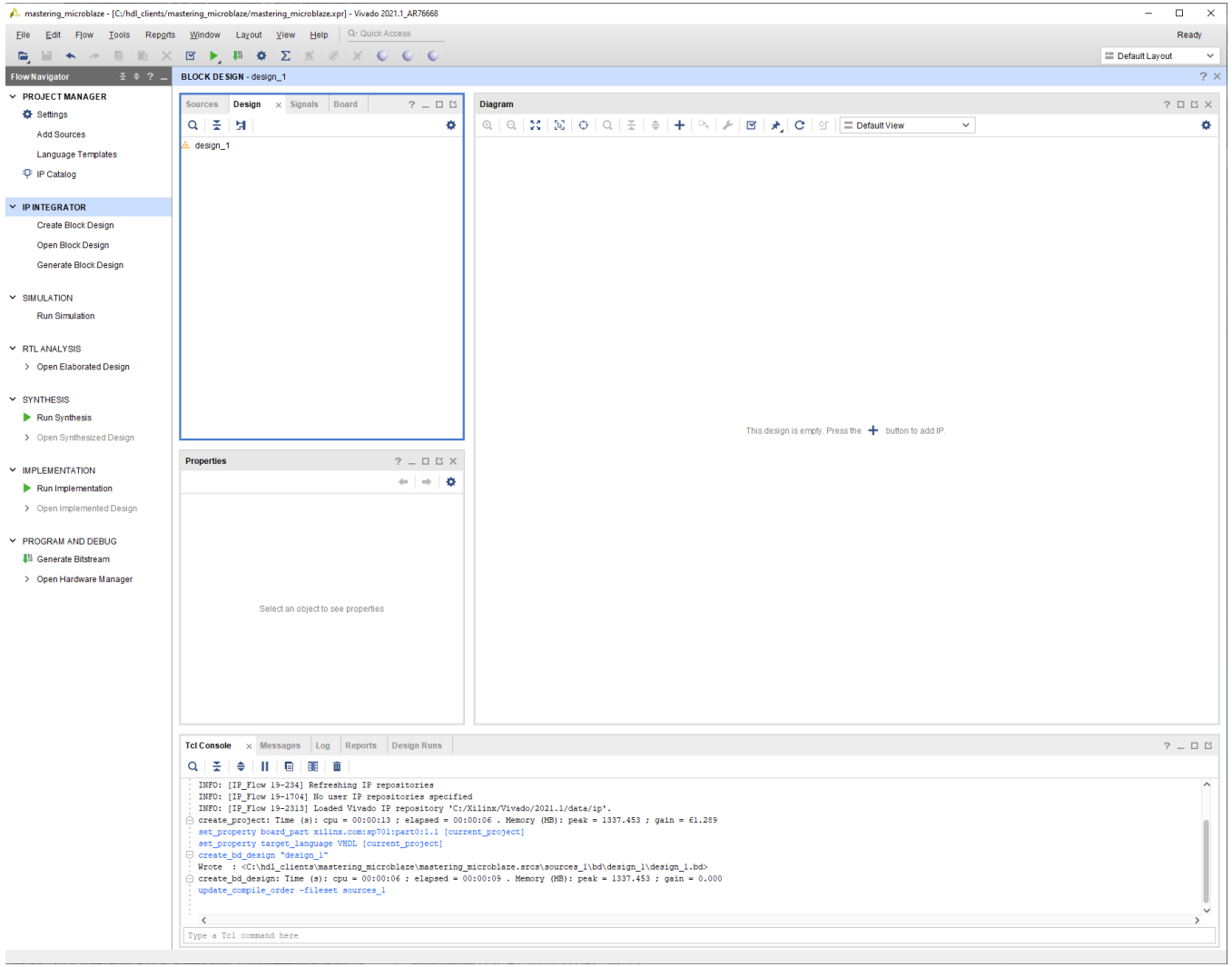
Lab: Mastering MicroBlaze

Click Create Block Design and leave the name unchanged



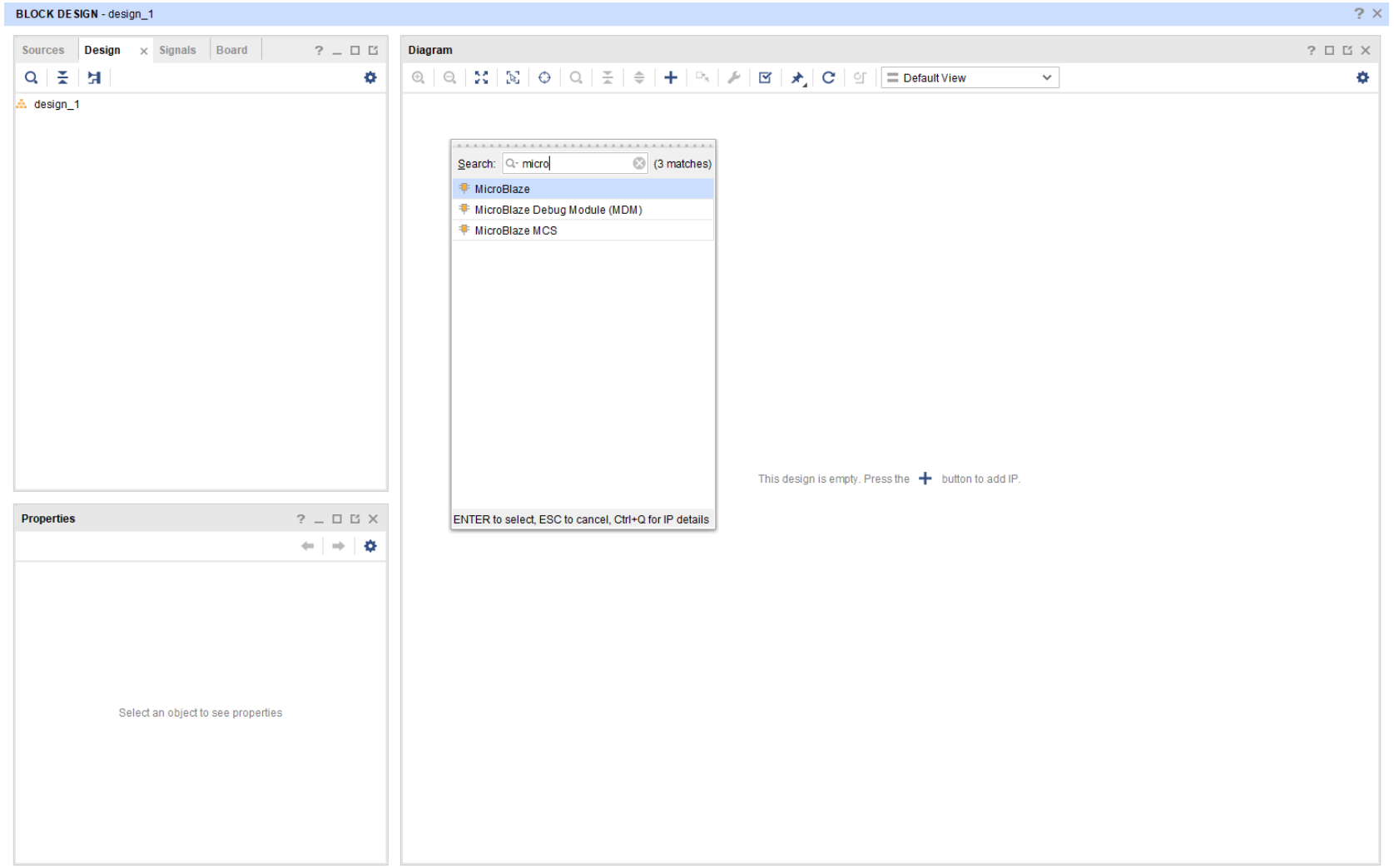
Lab: Mastering MicroBlaze

The Block Diagram should look as below



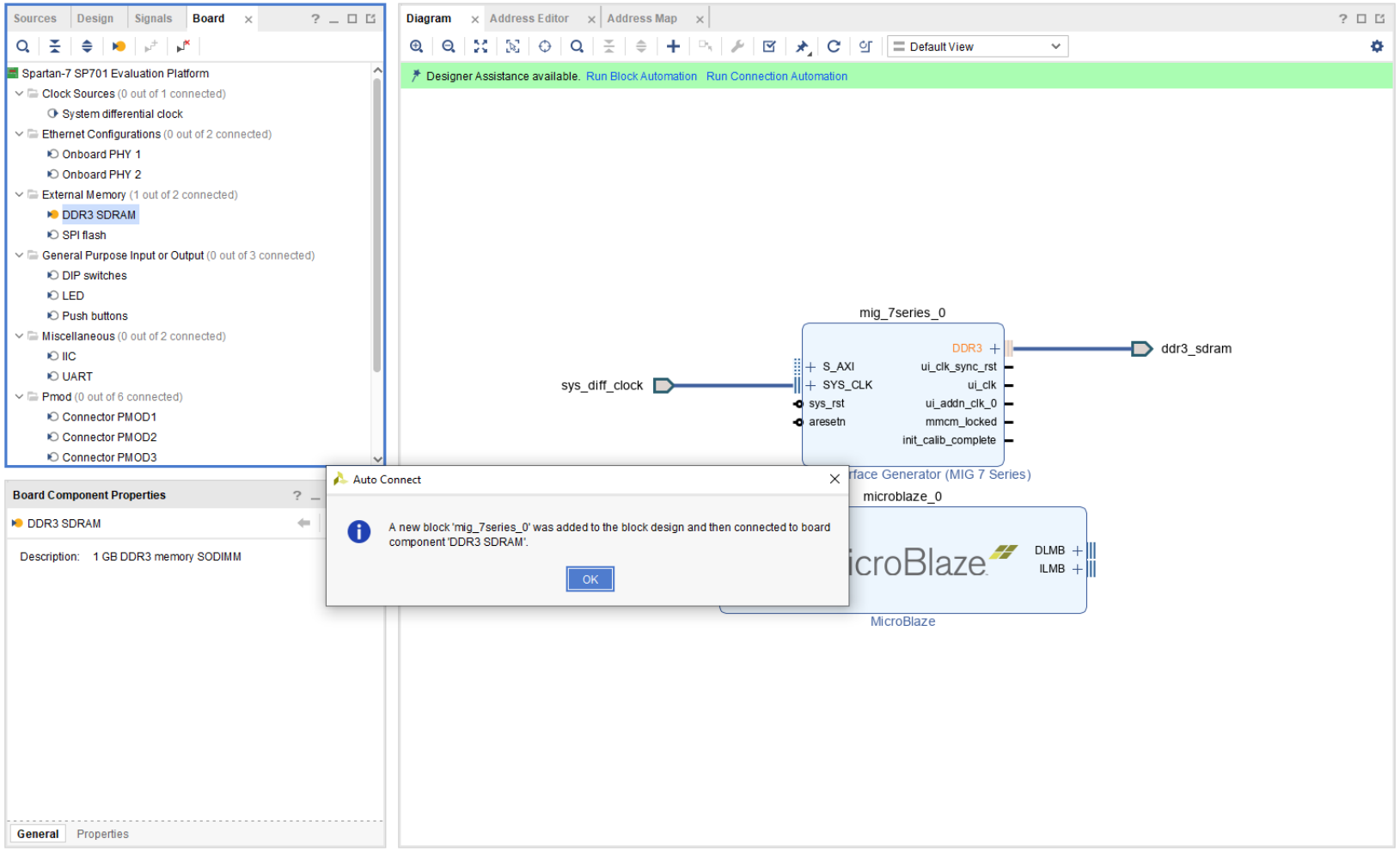
Lab: Mastering MicroBlaze

Click on the + in the block diagram and type in MicroBlaze



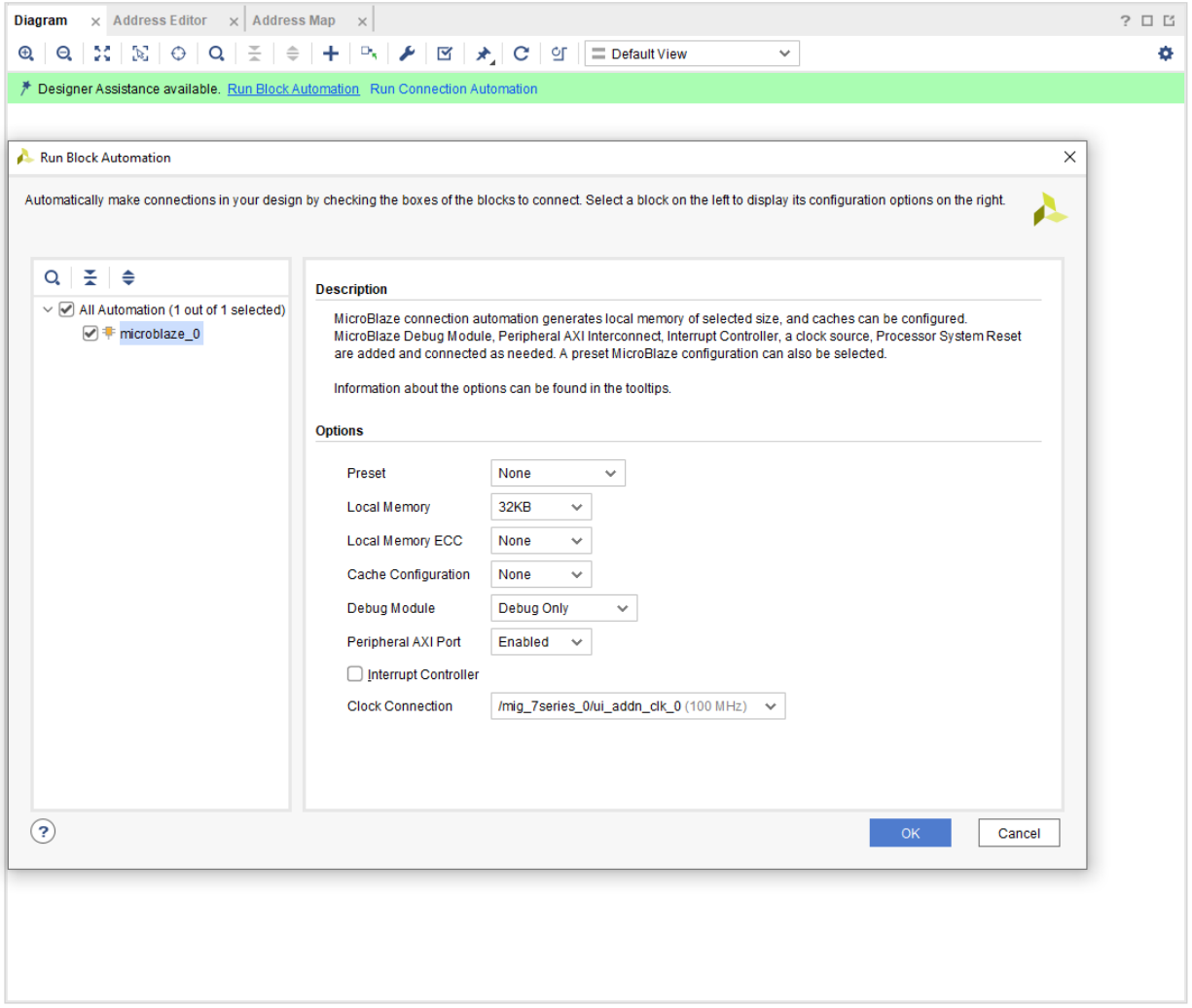
Lab: Mastering MicroBlaze

From the board tab select and drag the DDR3 onto the block diagram



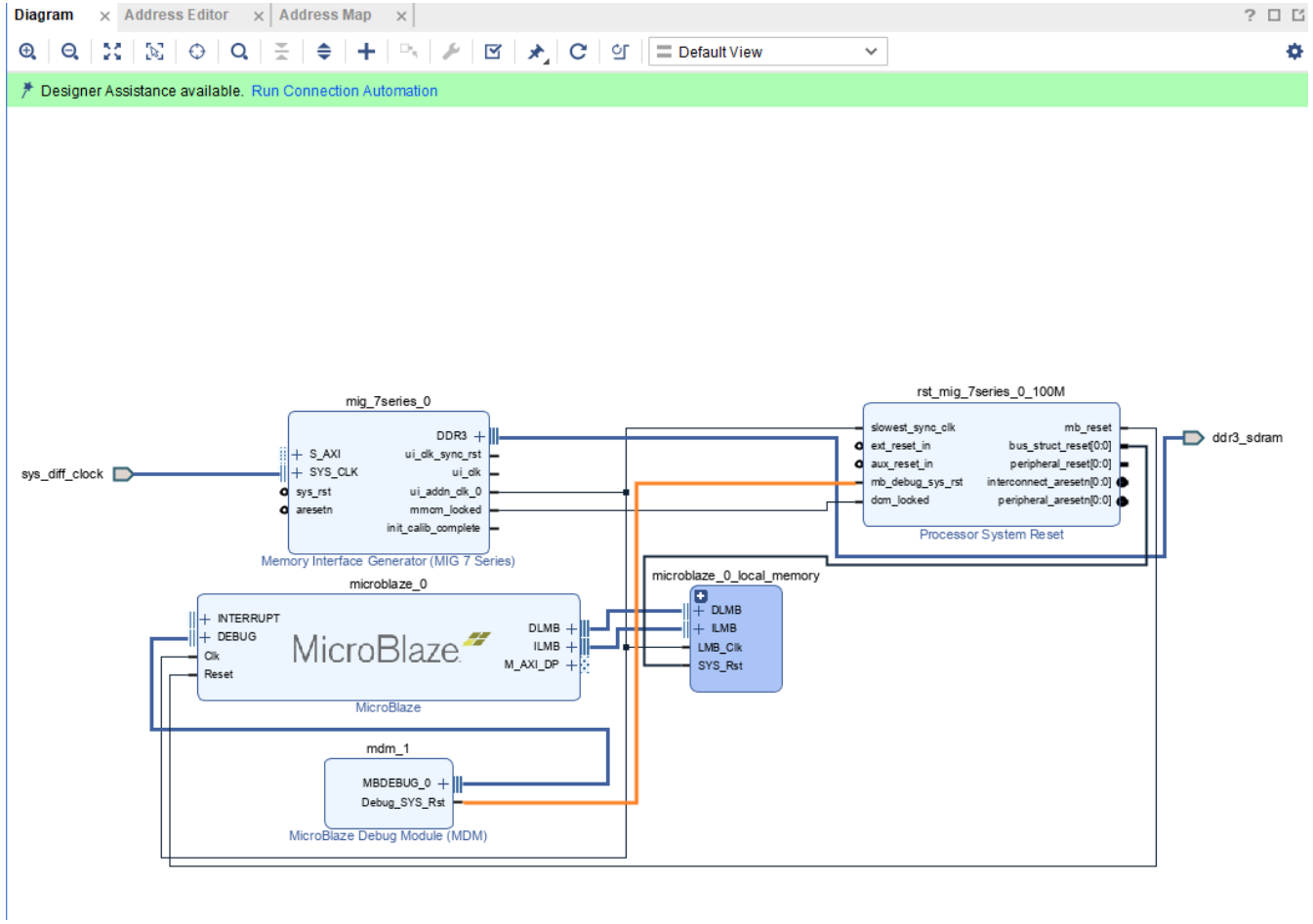
Lab: Mastering MicroBlaze

Run the Block Automation and configure 32KB of local memory and no cache



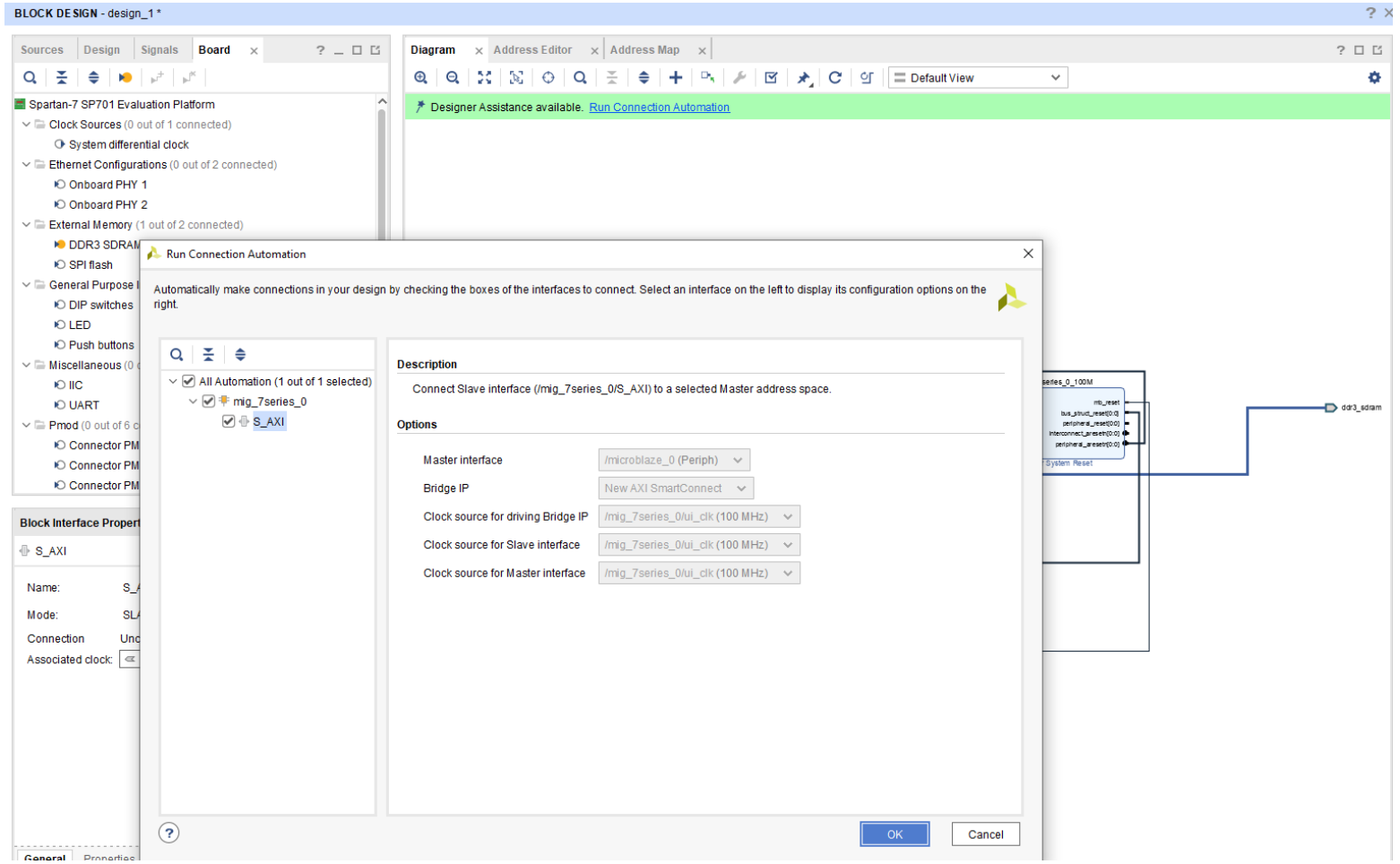
Lab: Mastering MicroBlaze

The result of the block automation should be as below



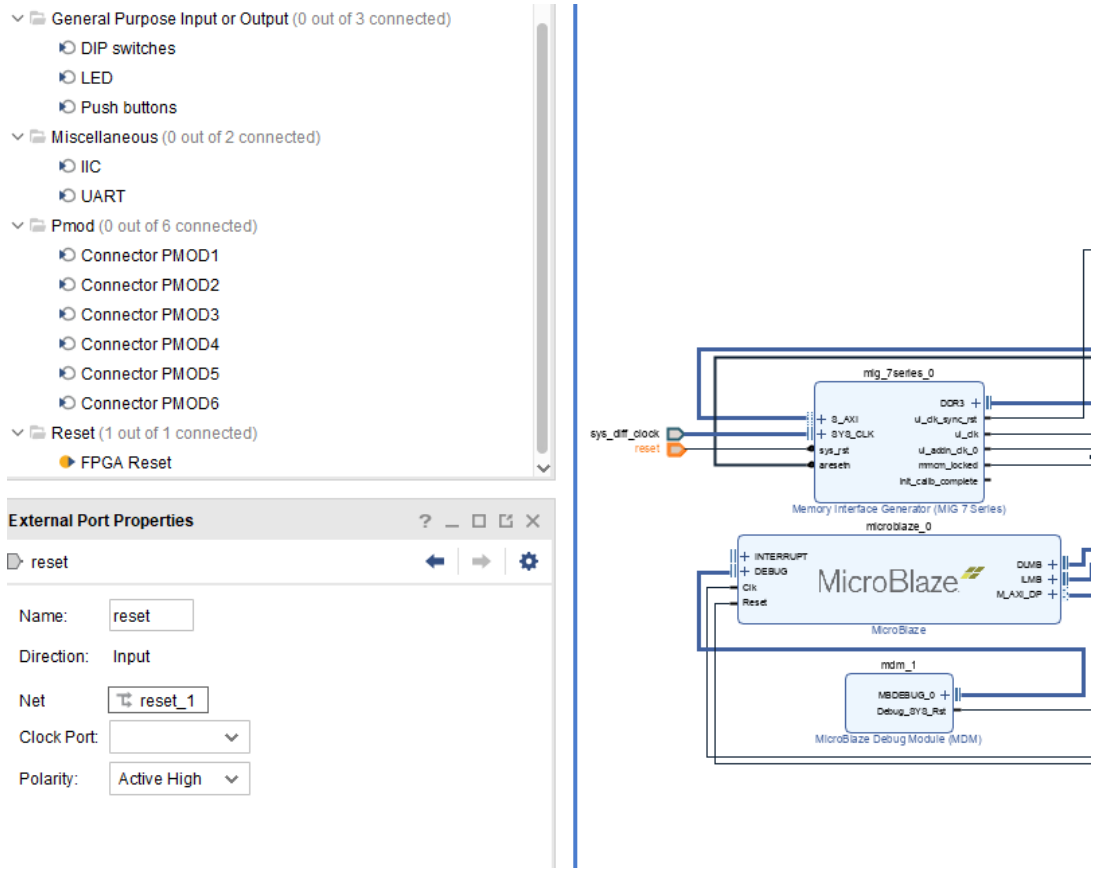
Lab: Mastering MicroBlaze

Run the Connection Automation to connect in the DDR3



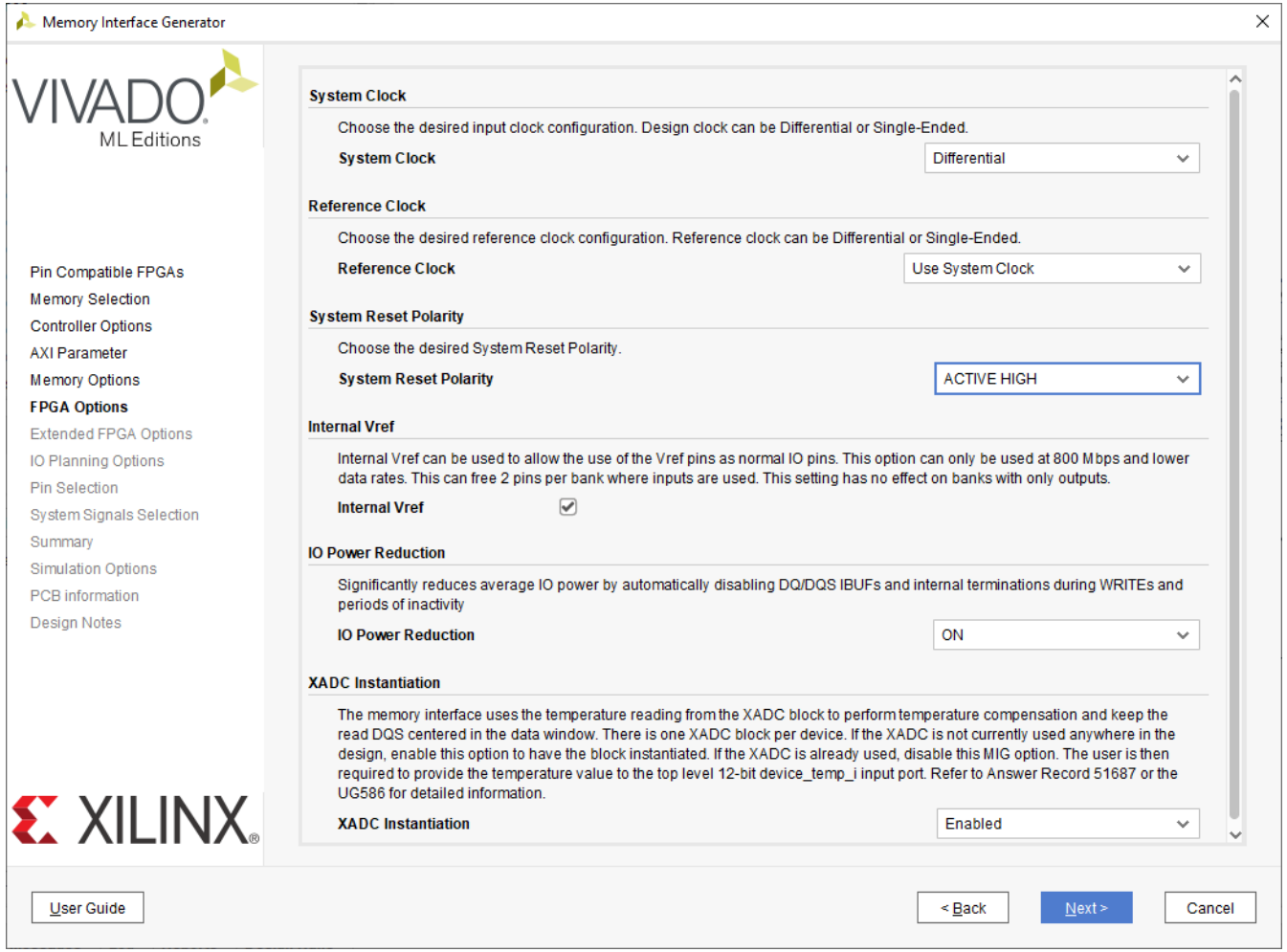
Lab: Mastering MicroBlaze

Drag and drop the reset port onto the diagram, select the port properties to determine its polarity



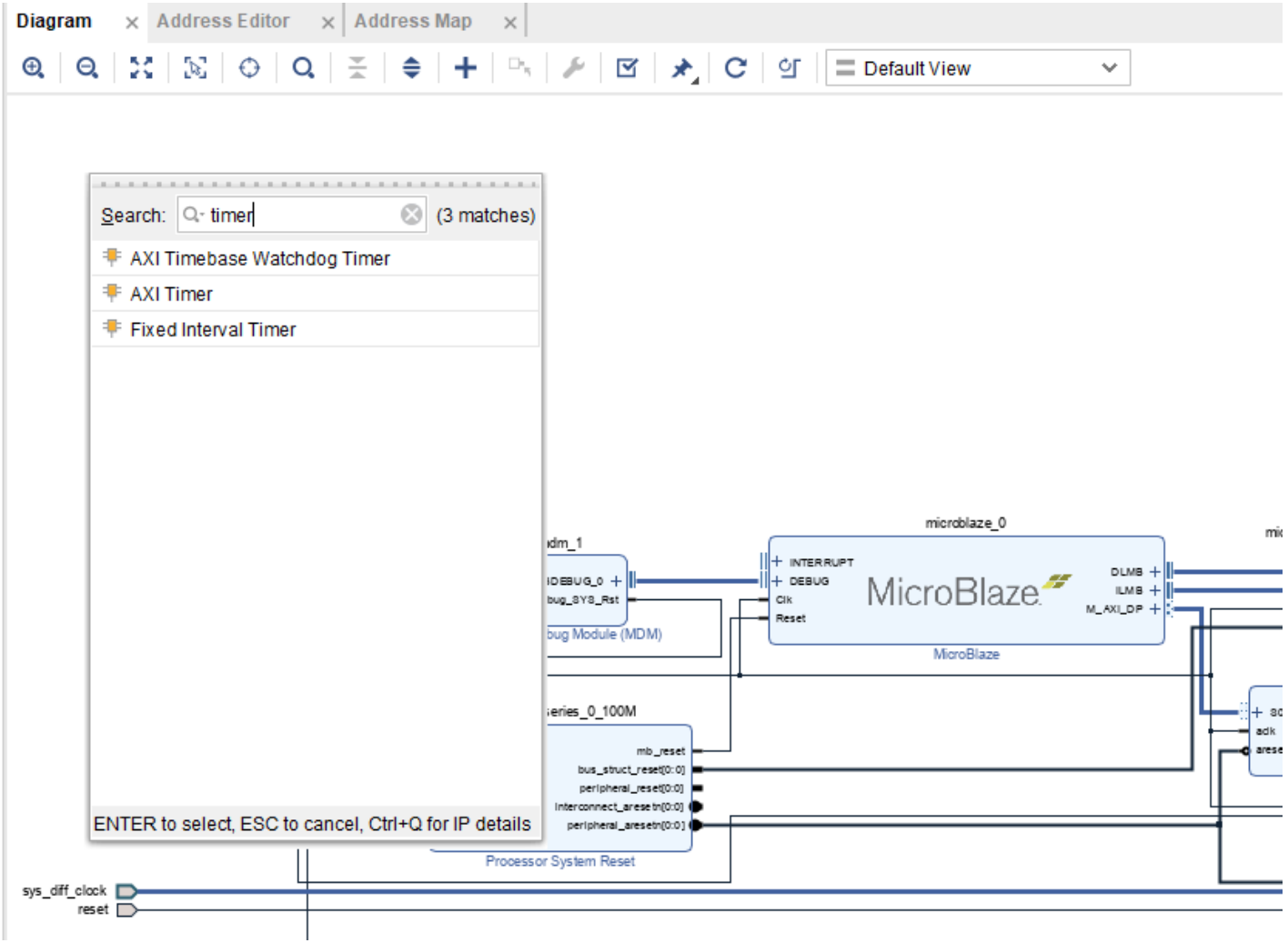
Lab: Mastering MicroBlaze

Re Customize the MIG and set the System Reset Polarity for Active High



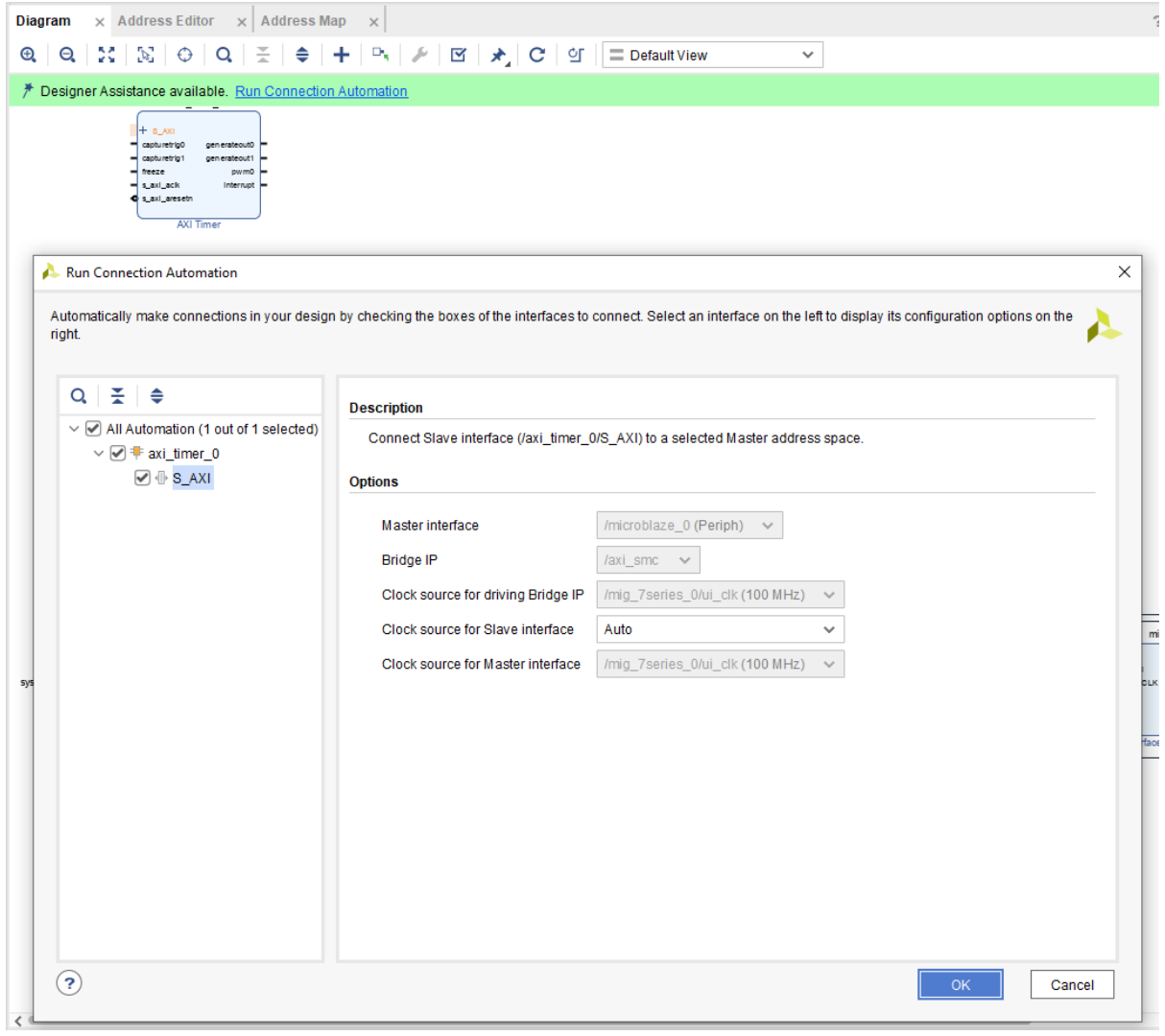
Lab: Mastering MicroBlaze

Add in a AXI Timer



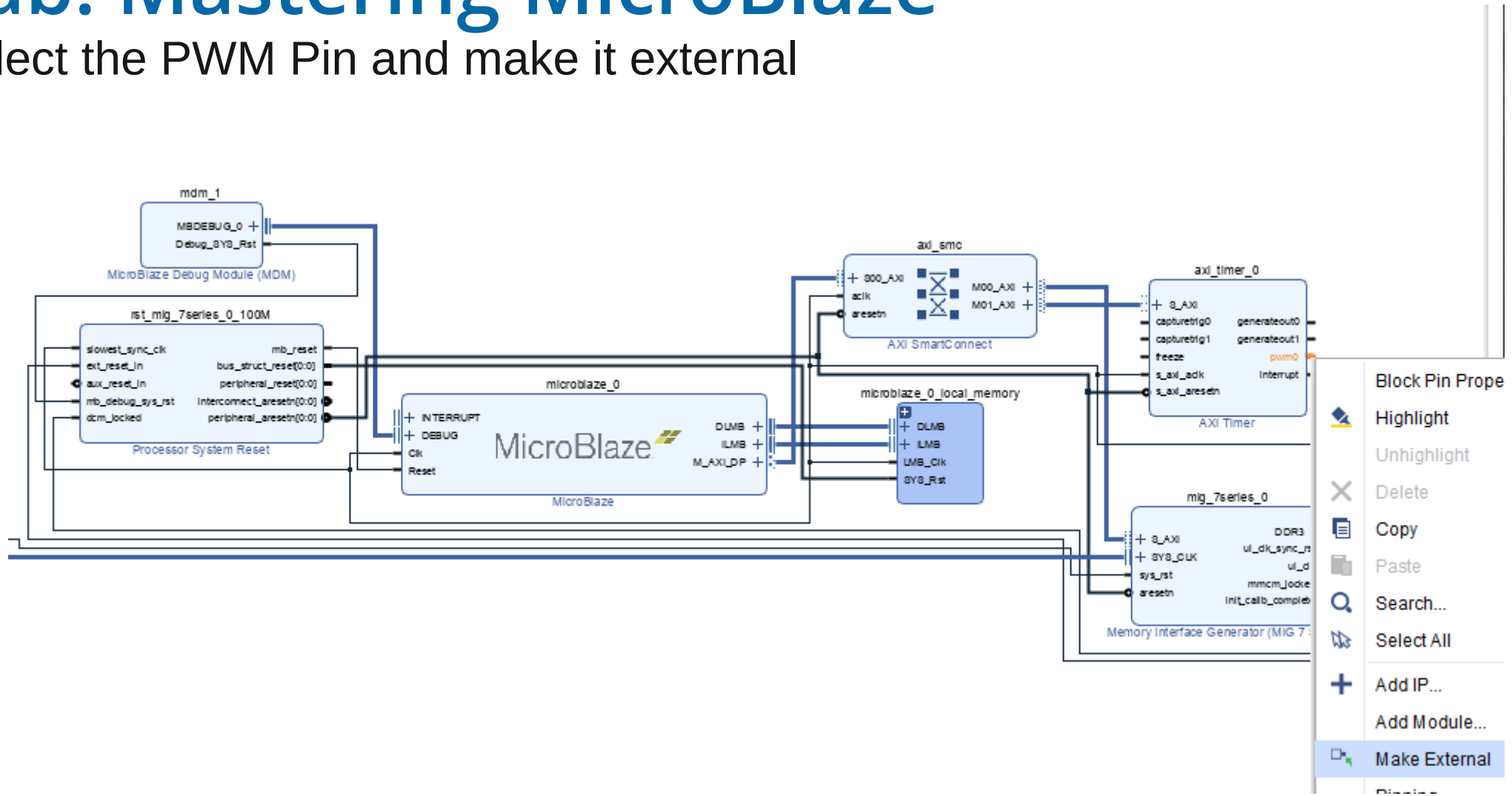
Lab: Mastering MicroBlaze

Run the Connection Automation



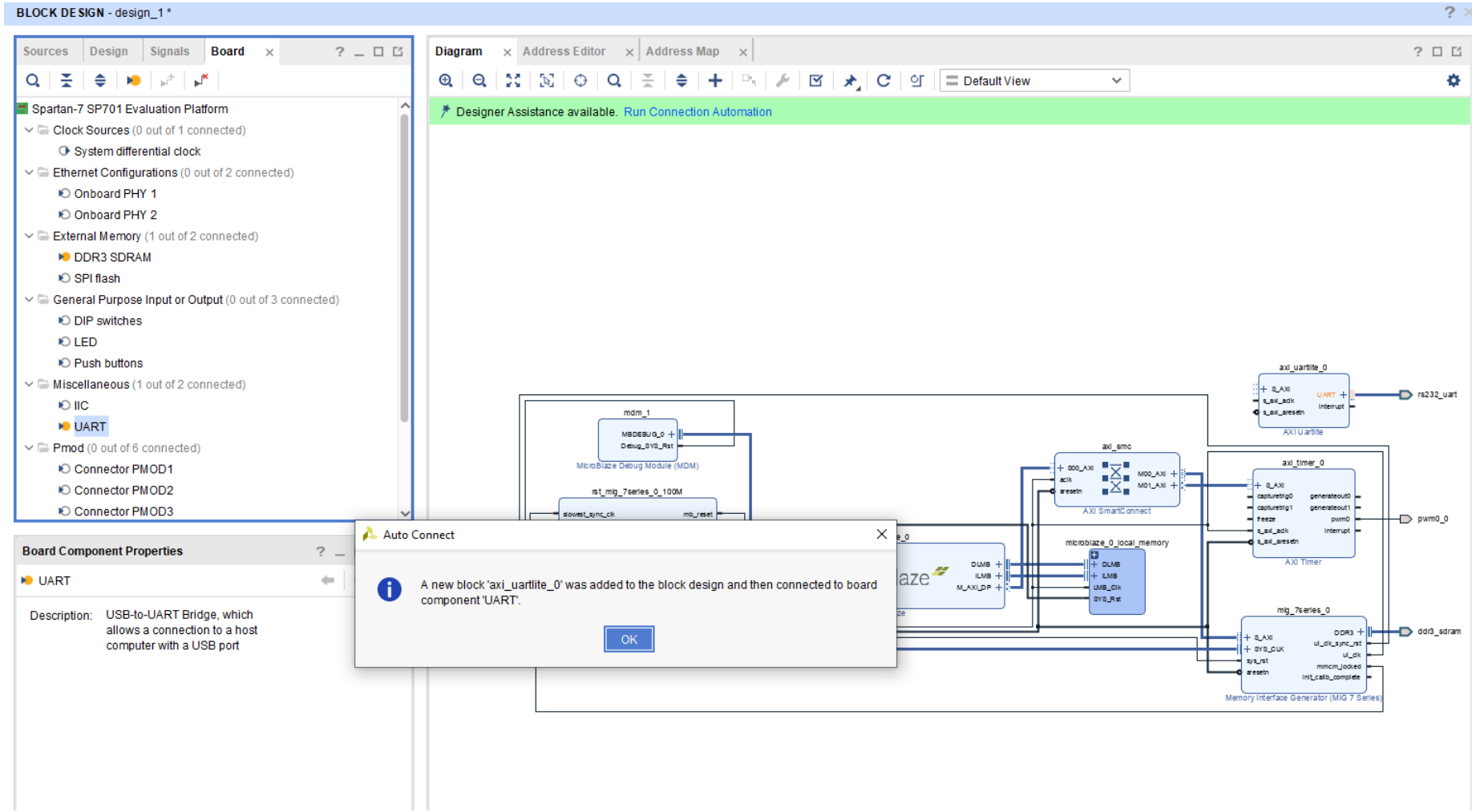
Lab: Mastering MicroBlaze

Select the PWM Pin and make it external



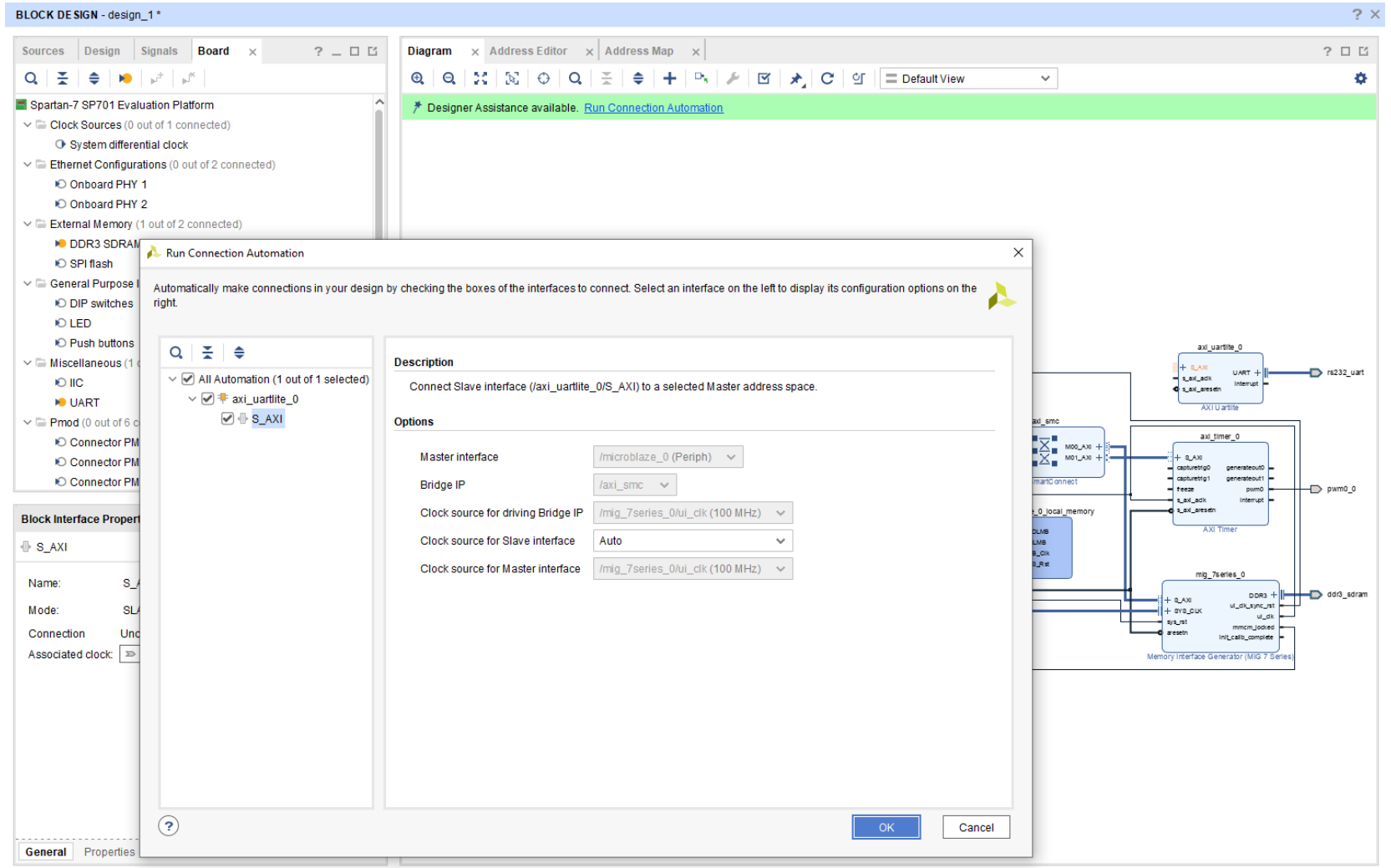
Lab: Mastering MicroBlaze

From the board menu drag and drop the UART onto the diagram



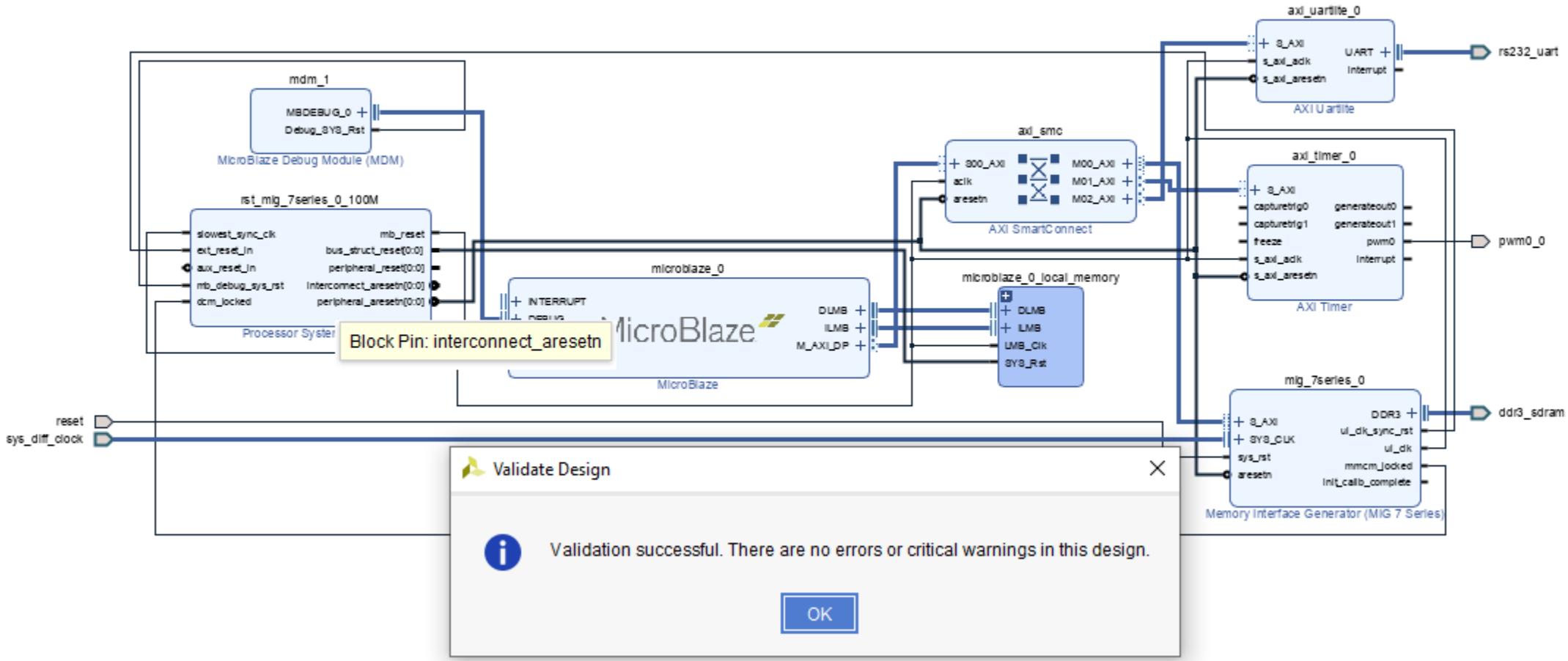
Lab: Mastering MicroBlaze

Run the connection automation



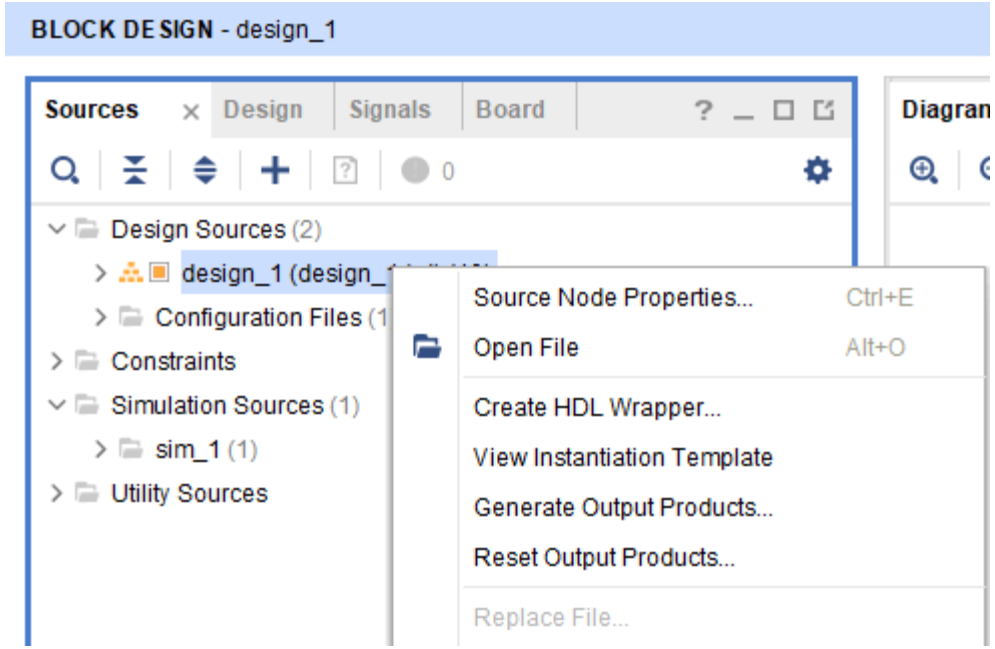
Lab: Mastering MicroBlaze

Validate the design = No errors or critical warnings should appear



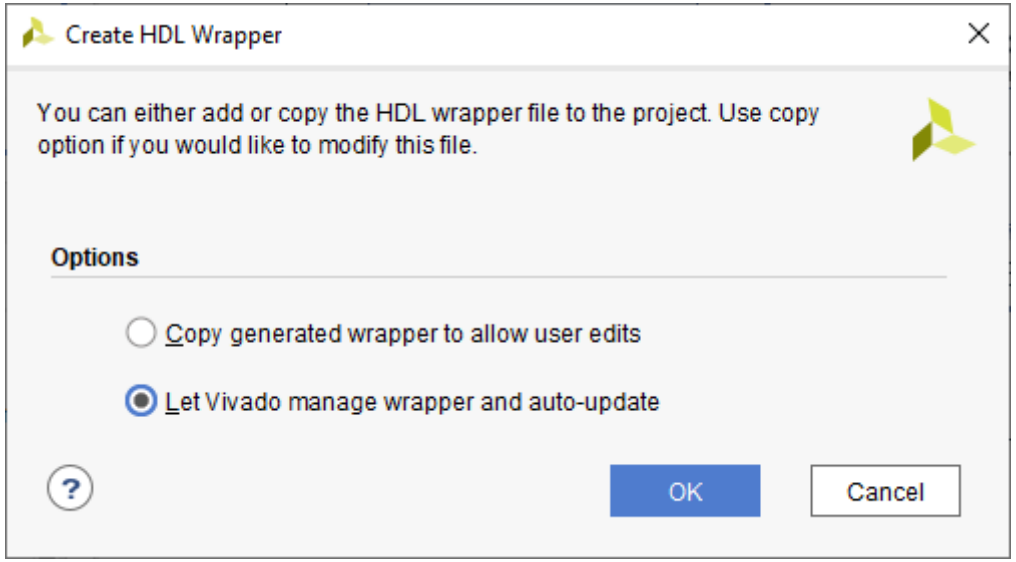
Lab: Mastering MicroBlaze

Create an HDL Wrapper



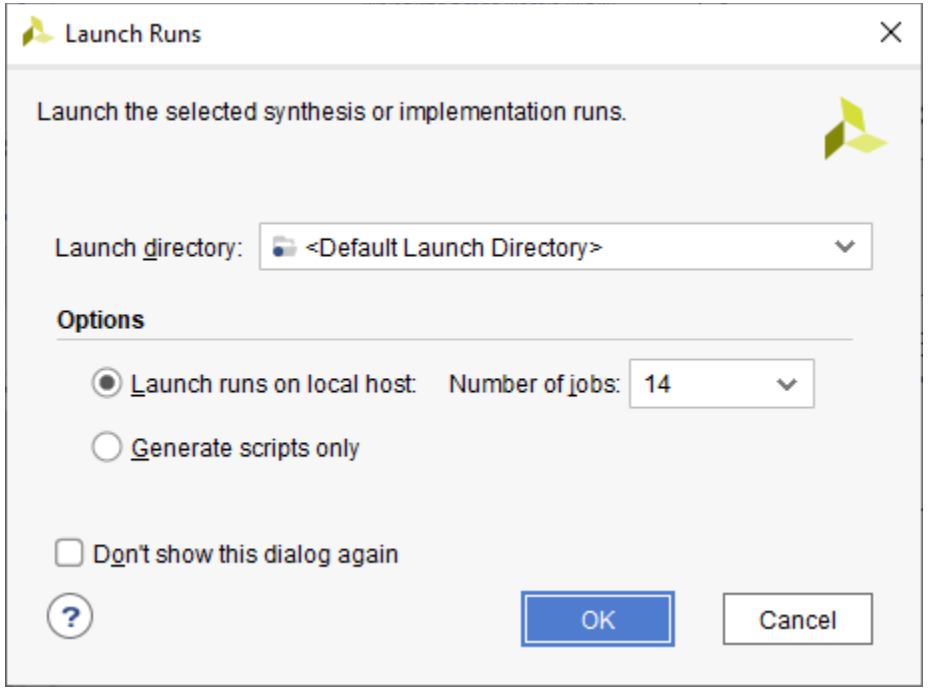
Lab: Mastering MicroBlaze

Let Vivado manage the wrapper



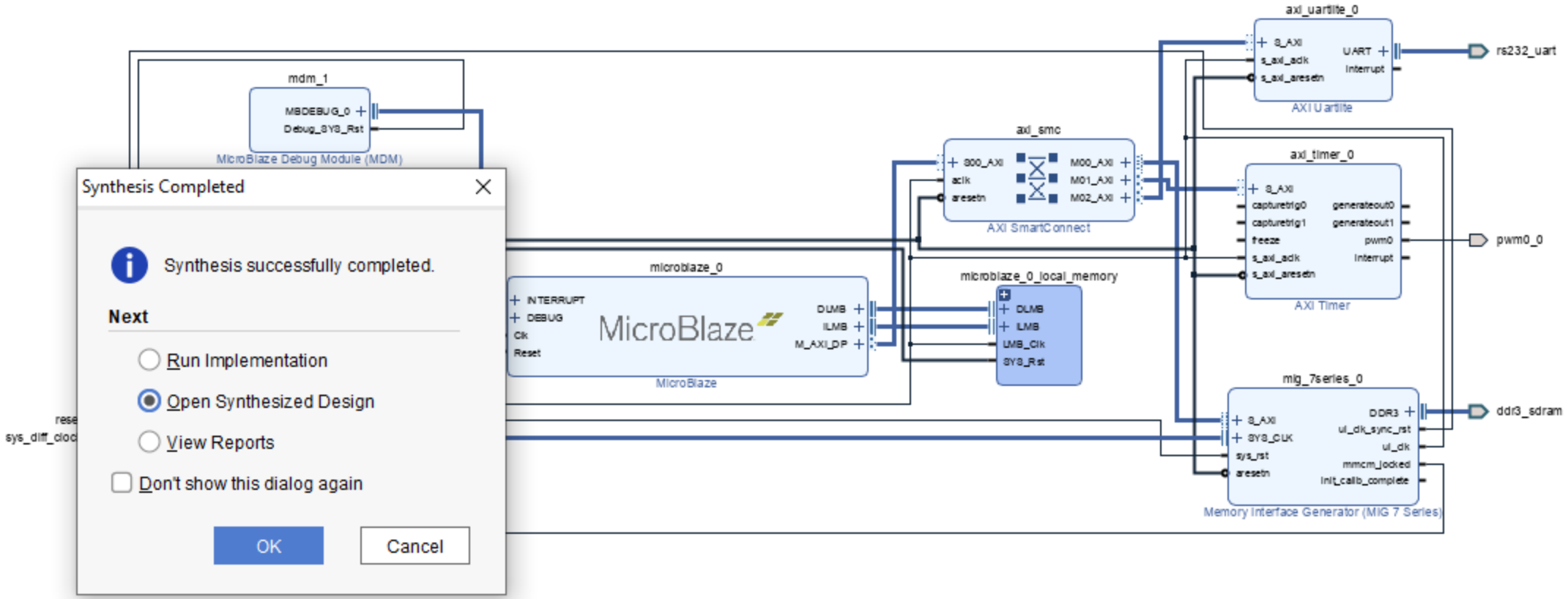
Lab: Mastering MicroBlaze

Synthesize the design



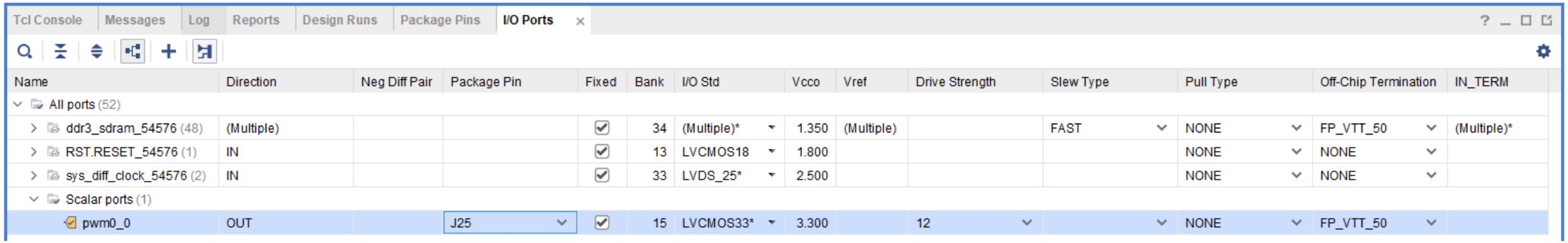
Lab: Mastering MicroBlaze

Once synthesis completes open the synthesized view



Lab: Mastering MicroBlaze

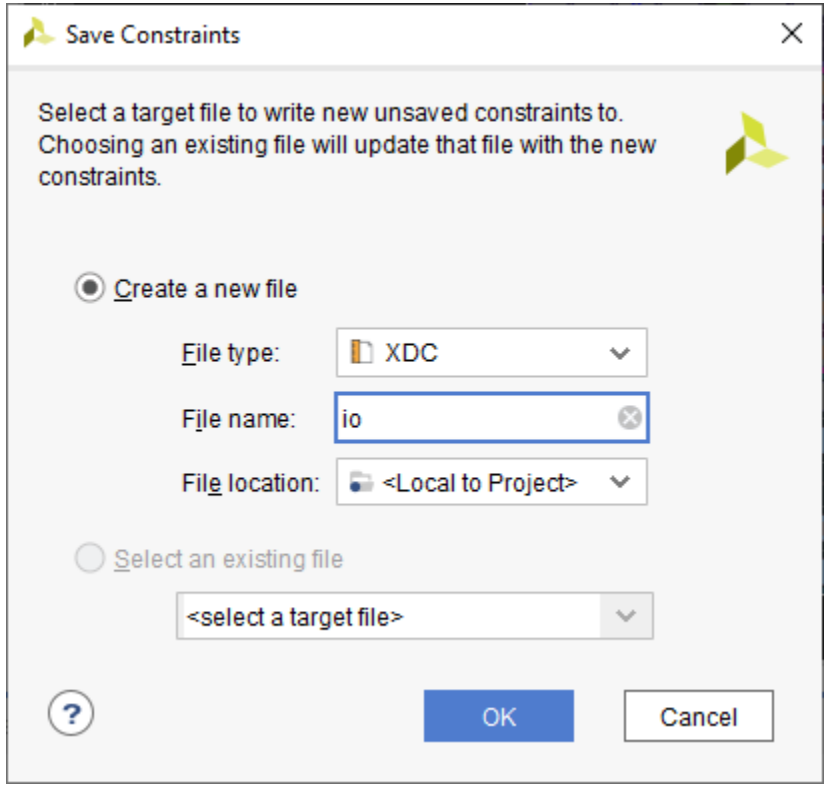
In the IO Ports view assign the PWM output to a pin connected to the LED



Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination	IN_TERM
All ports (52)													
ddr3_sdram_54576 (48)	(Multiple)			☑	34	(Multiple)*	1.350	(Multiple)		FAST	NONE	FP_VTT_50	(Multiple)*
RST.RESET_54576 (1)	IN			☑	13	LVCMOS18	1.800				NONE	NONE	
sys_diff_clock_54576 (2)	IN			☑	33	LVDS_25*	2.500				NONE	NONE	
Scalar ports (1)													
pwm0_0	OUT		J25	☑	15	LVCMOS33*	3.300		12		NONE	FP_VTT_50	

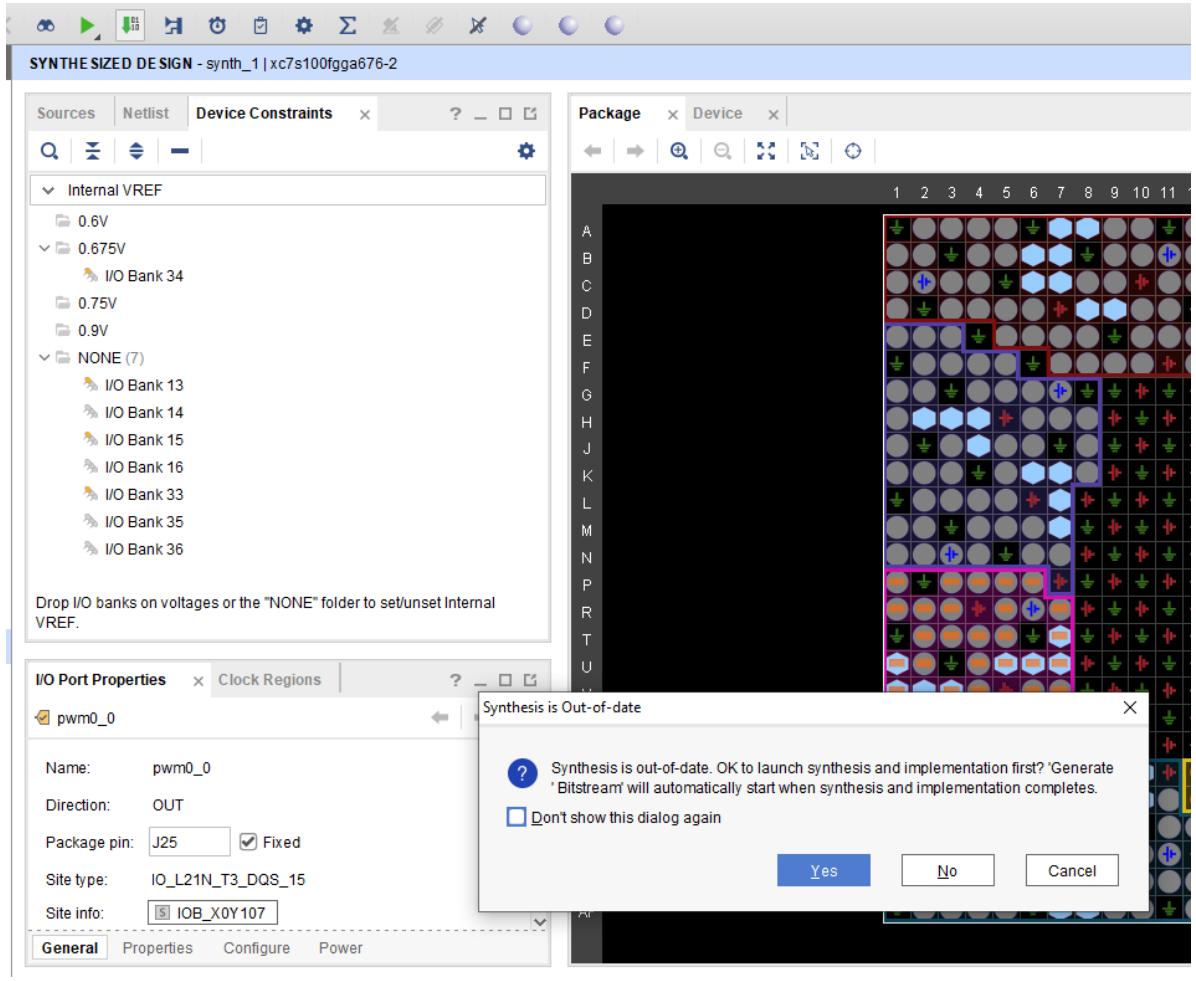
Lab: Mastering MicroBlaze

Save the constraints



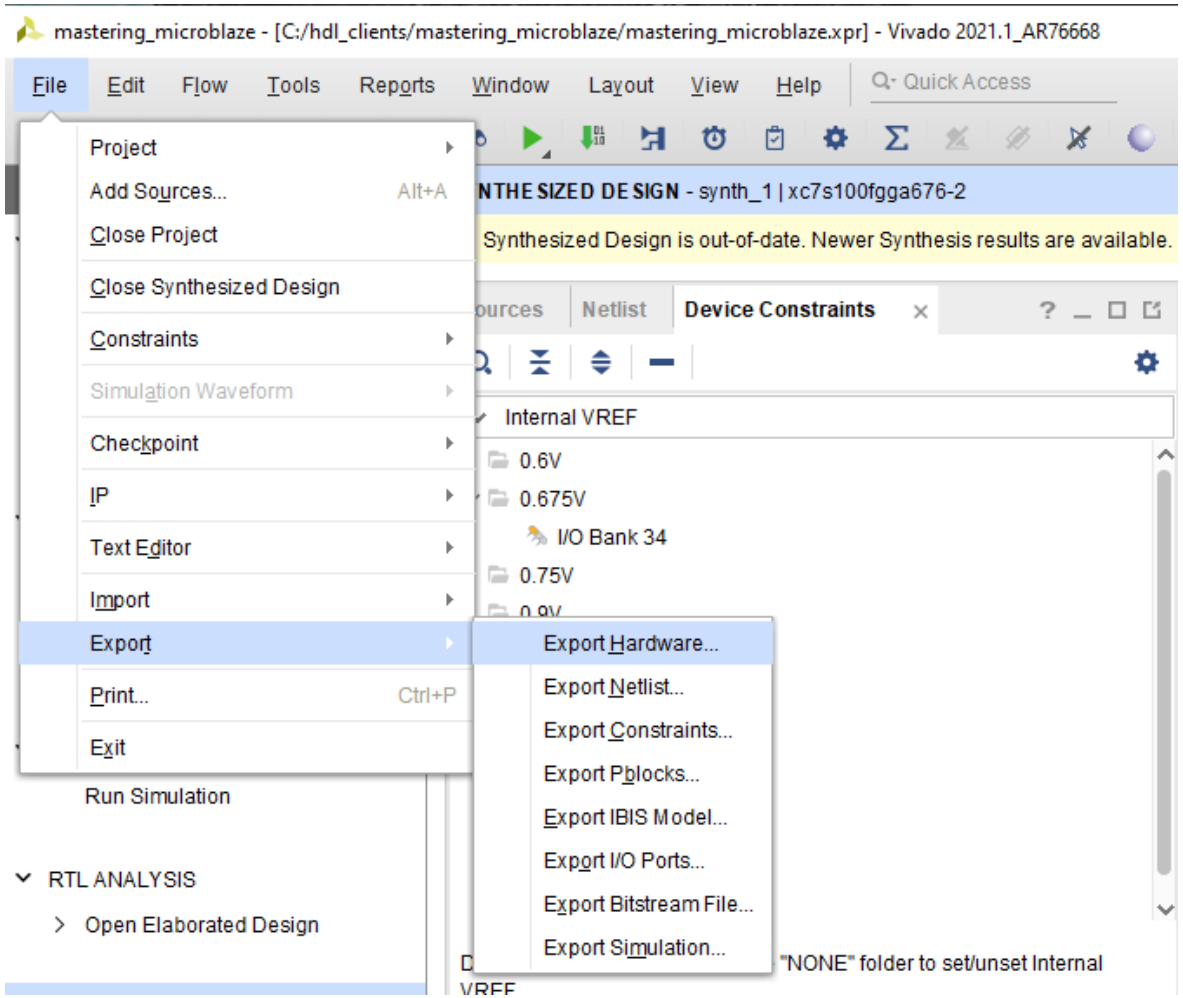
Lab: Mastering MicroBlaze

Generate the bitstream – if you see an out-of-date warning click yes



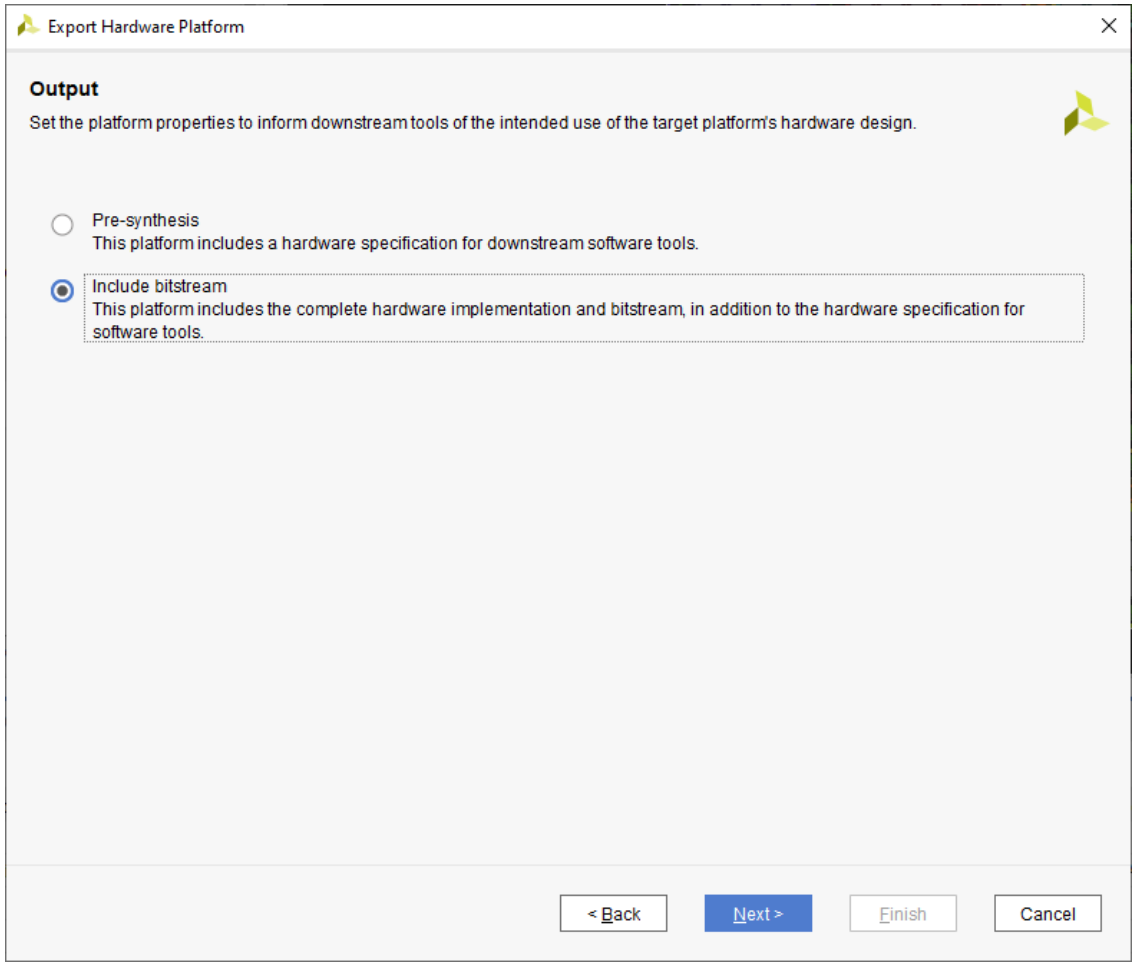
Lab: Mastering MicroBlaze

Once the bit stream is completed, export the hardware definition to Vitis



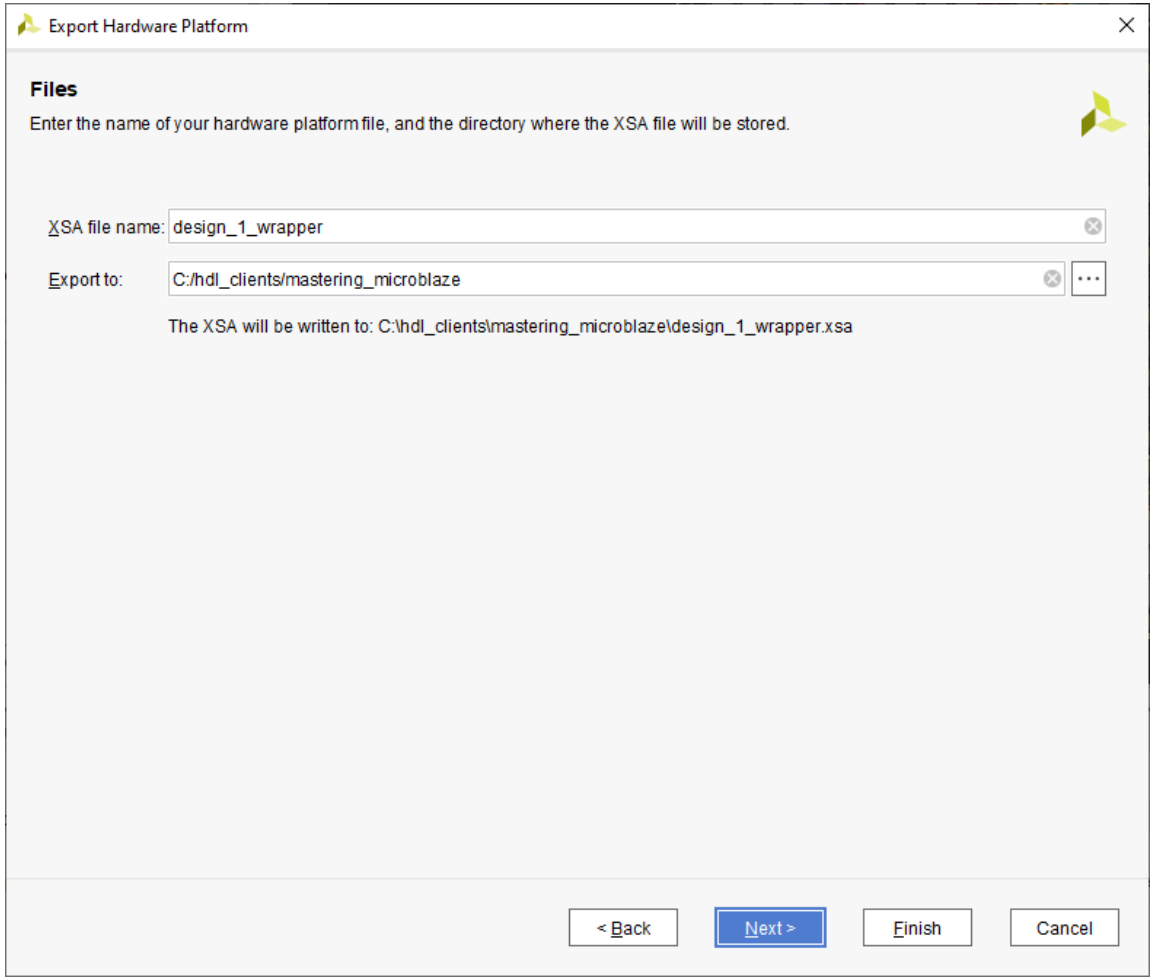
Lab: Mastering MicroBlaze

Export the Hardware with the bitstream



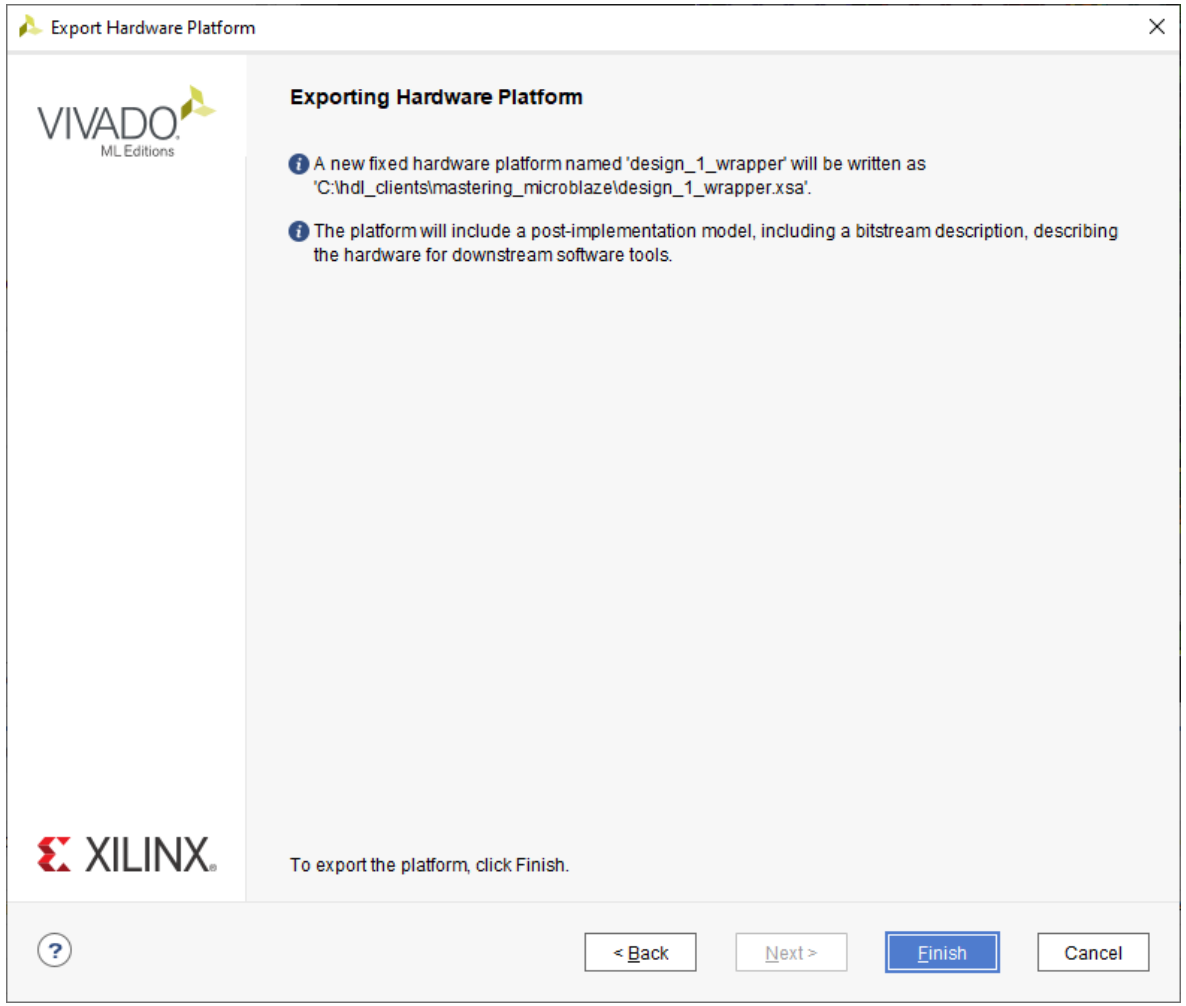
Lab: Mastering MicroBlaze

Leave the file name and location unchanged



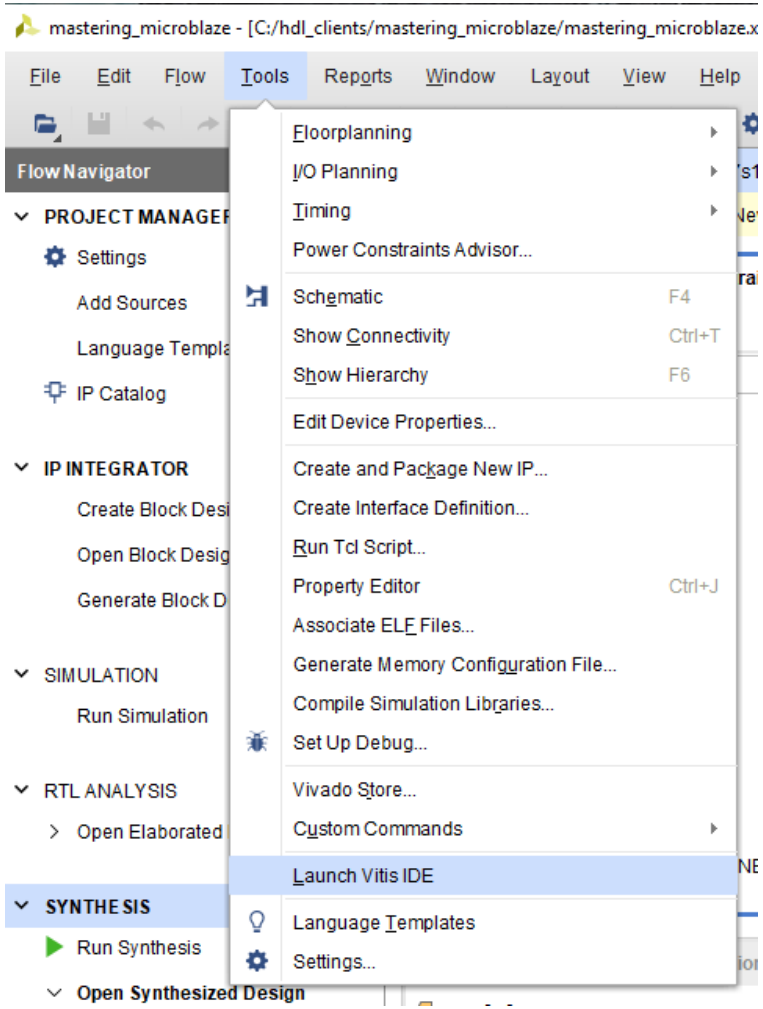
Lab: Mastering MicroBlaze

Click Finish



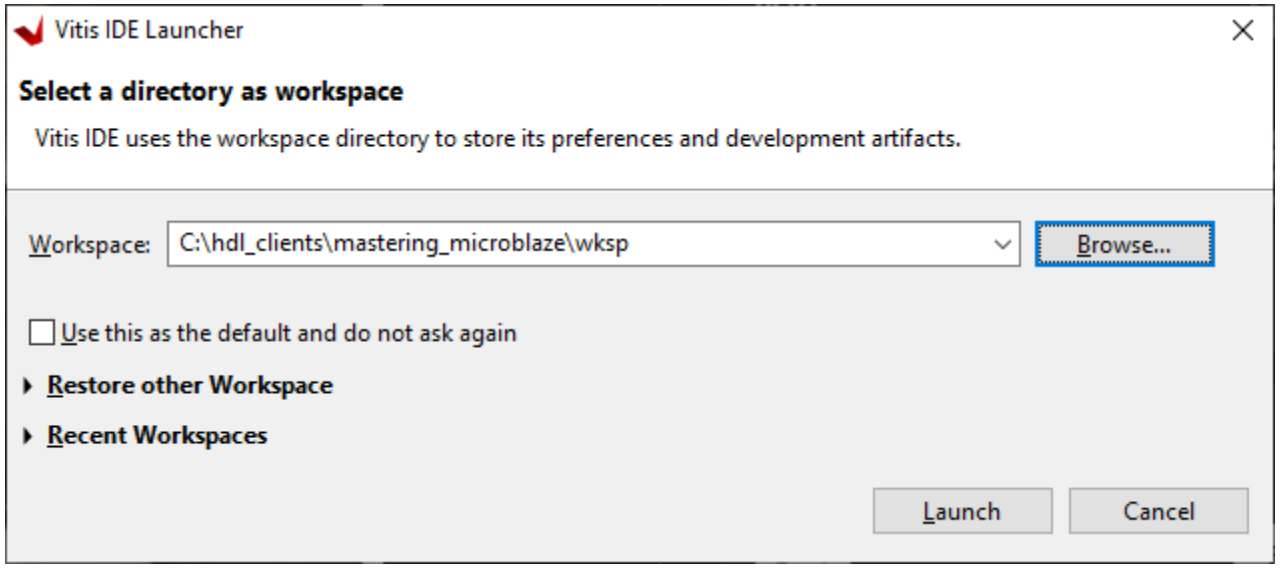
Lab: Mastering MicroBlaze

From the tools menu select Launch Vitis IDE



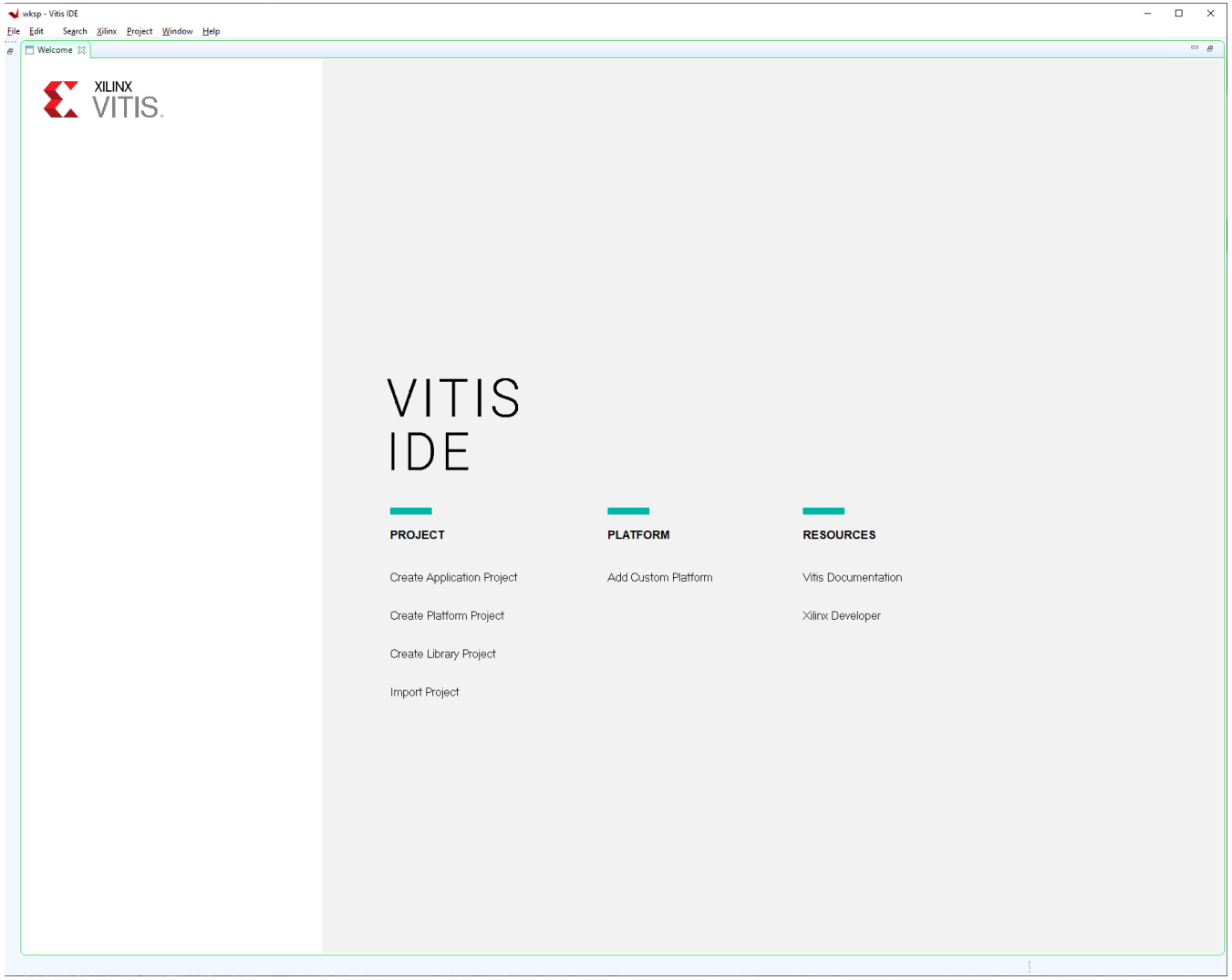
Lab: Mastering MicroBlaze

Select the workspace – this is where all the SW files and projects will be contained



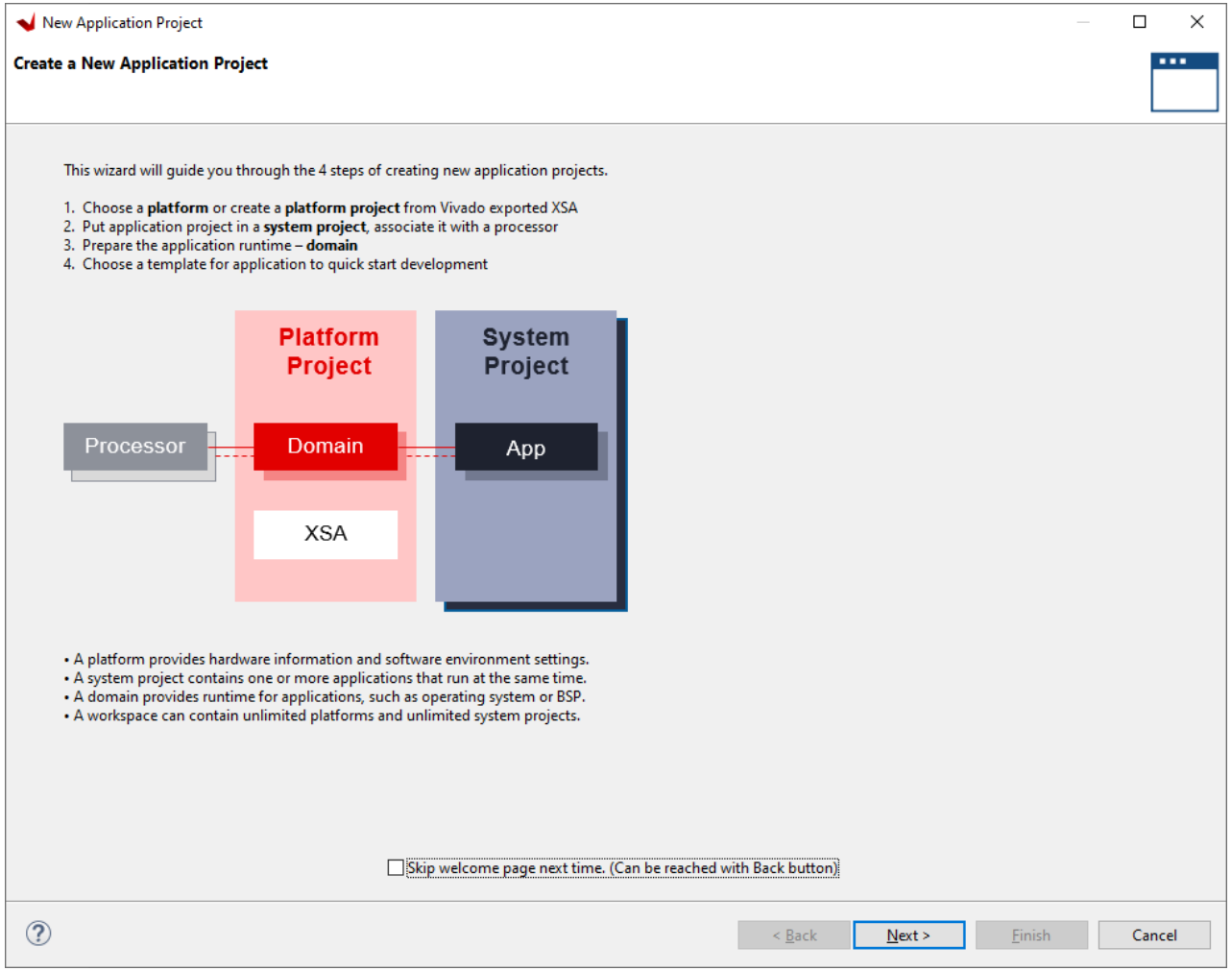
Lab: Mastering MicroBlaze

Once Vitis is Open, from the menu screen select Create-Application Project



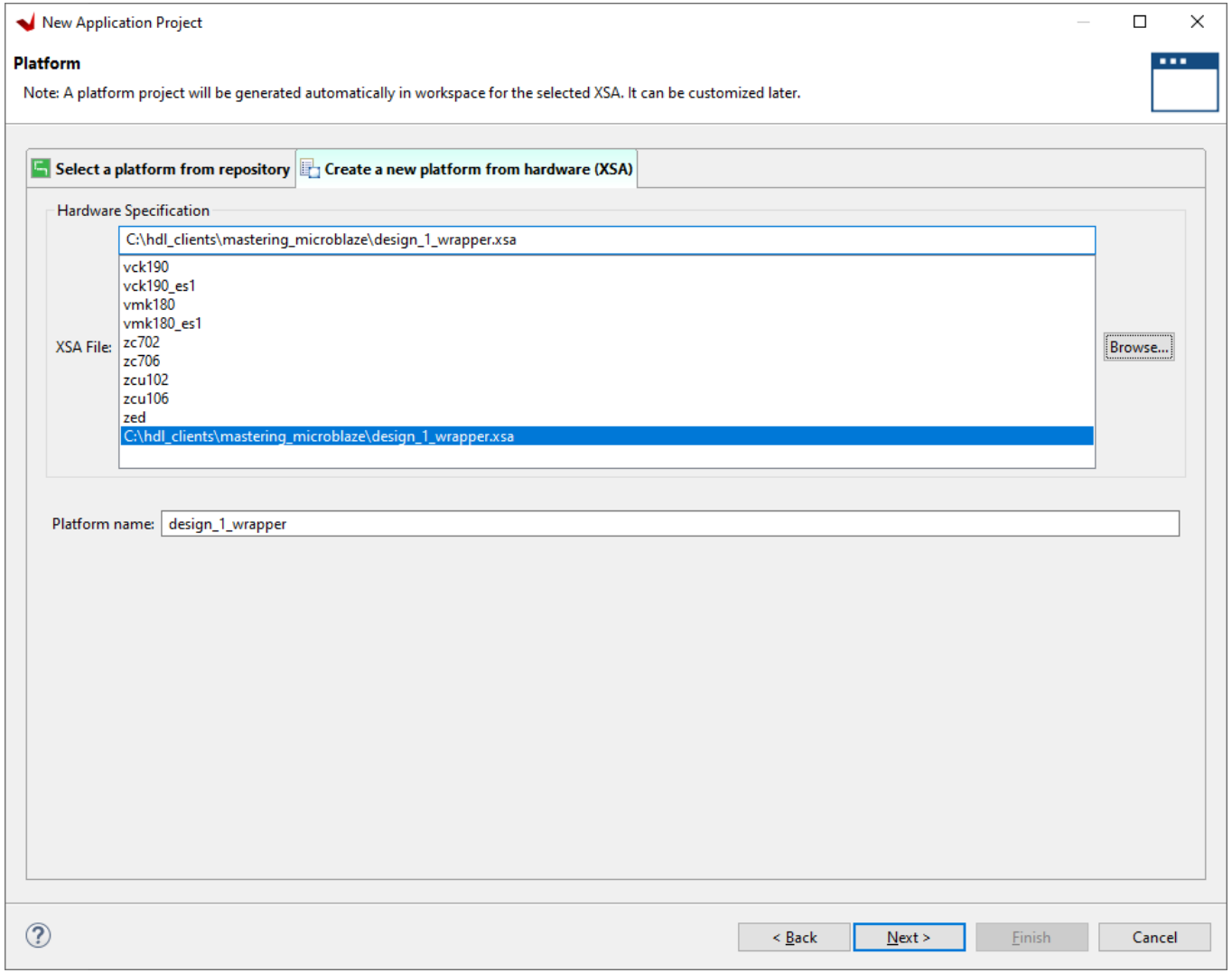
Lab: Mastering MicroBlaze

Click Next



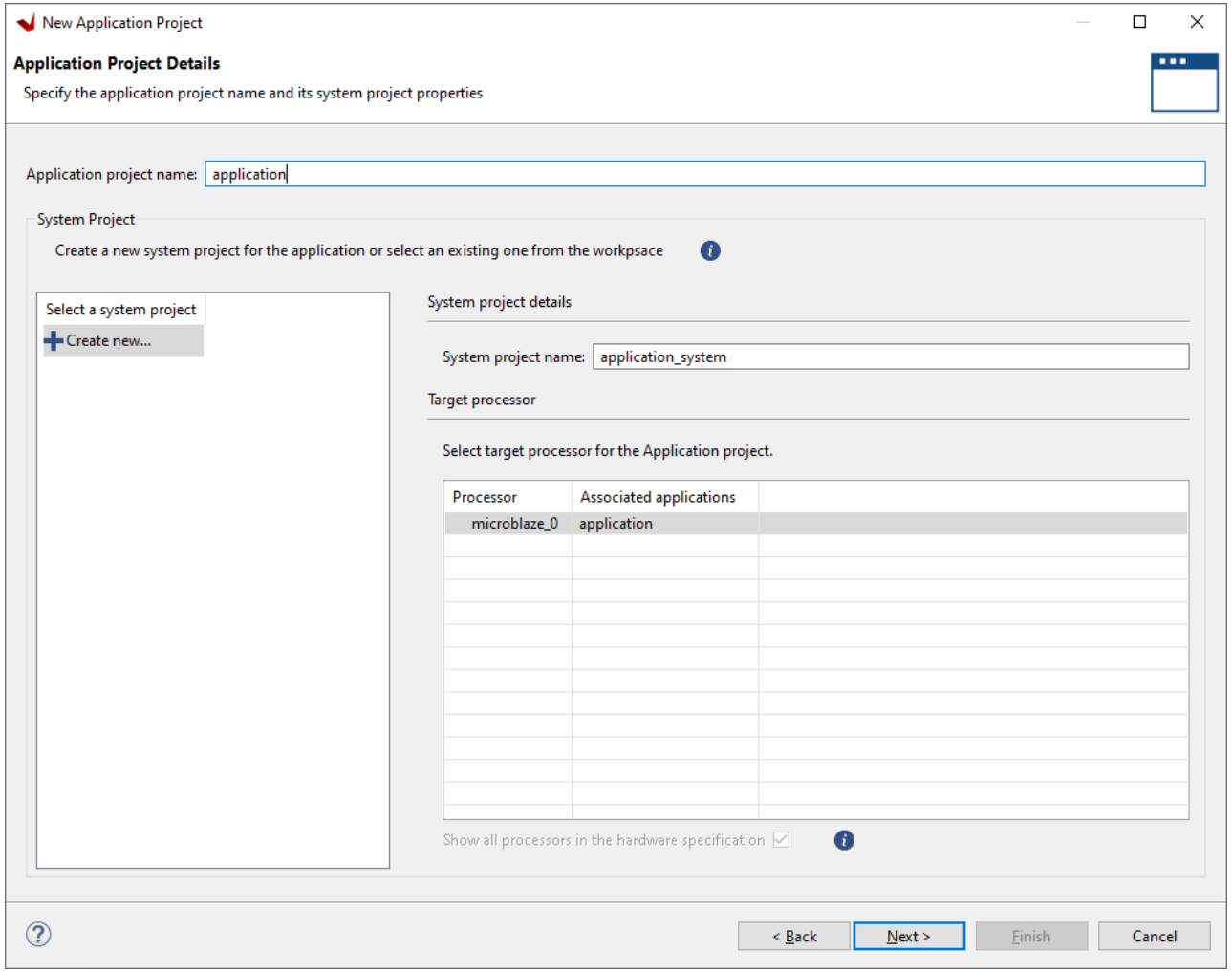
Lab: Mastering MicroBlaze

Select the XSA just exported from Vivado



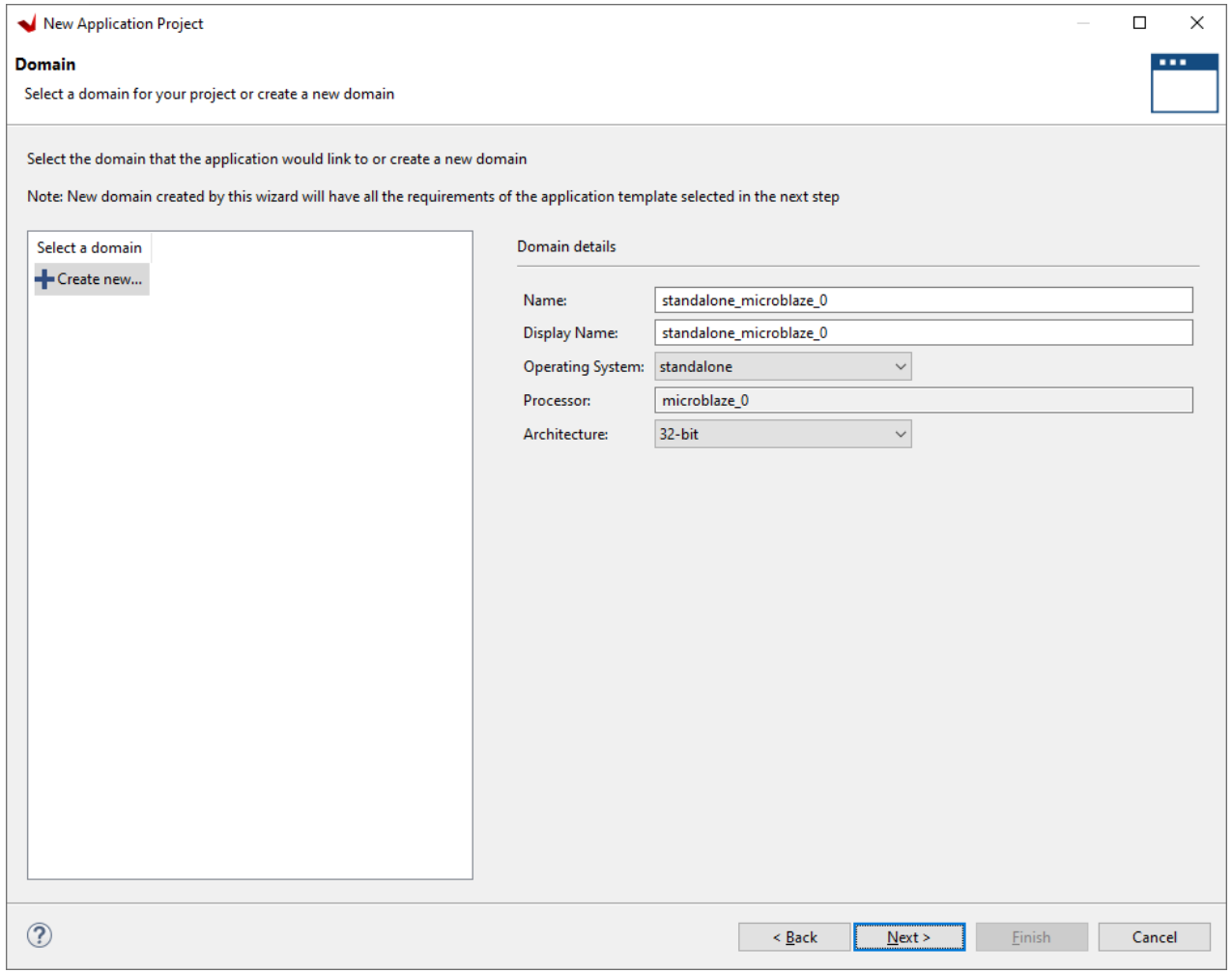
Lab: Mastering MicroBlaze

Enter a project name



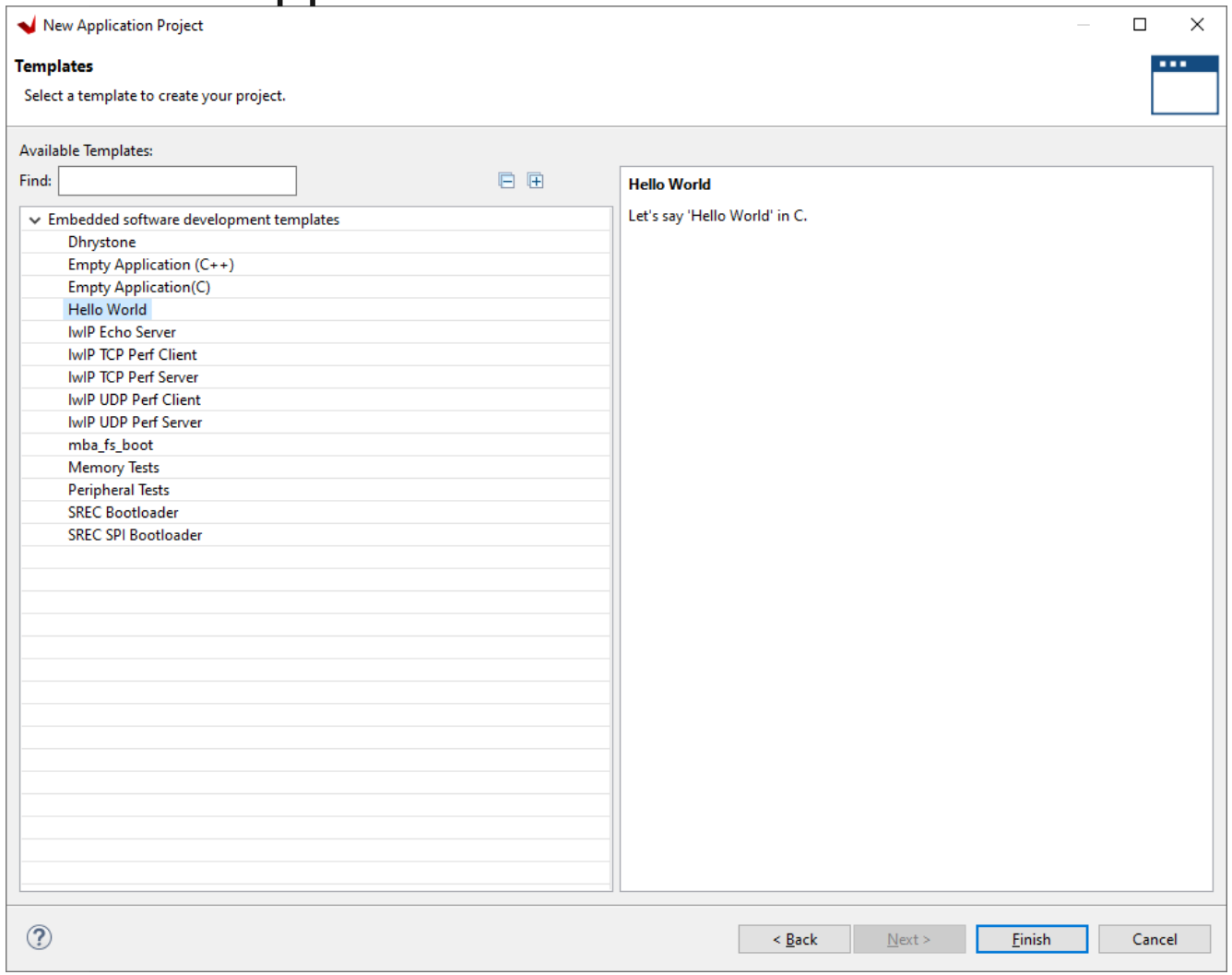
Lab: Mastering MicroBlaze

Leave the domain unchanged



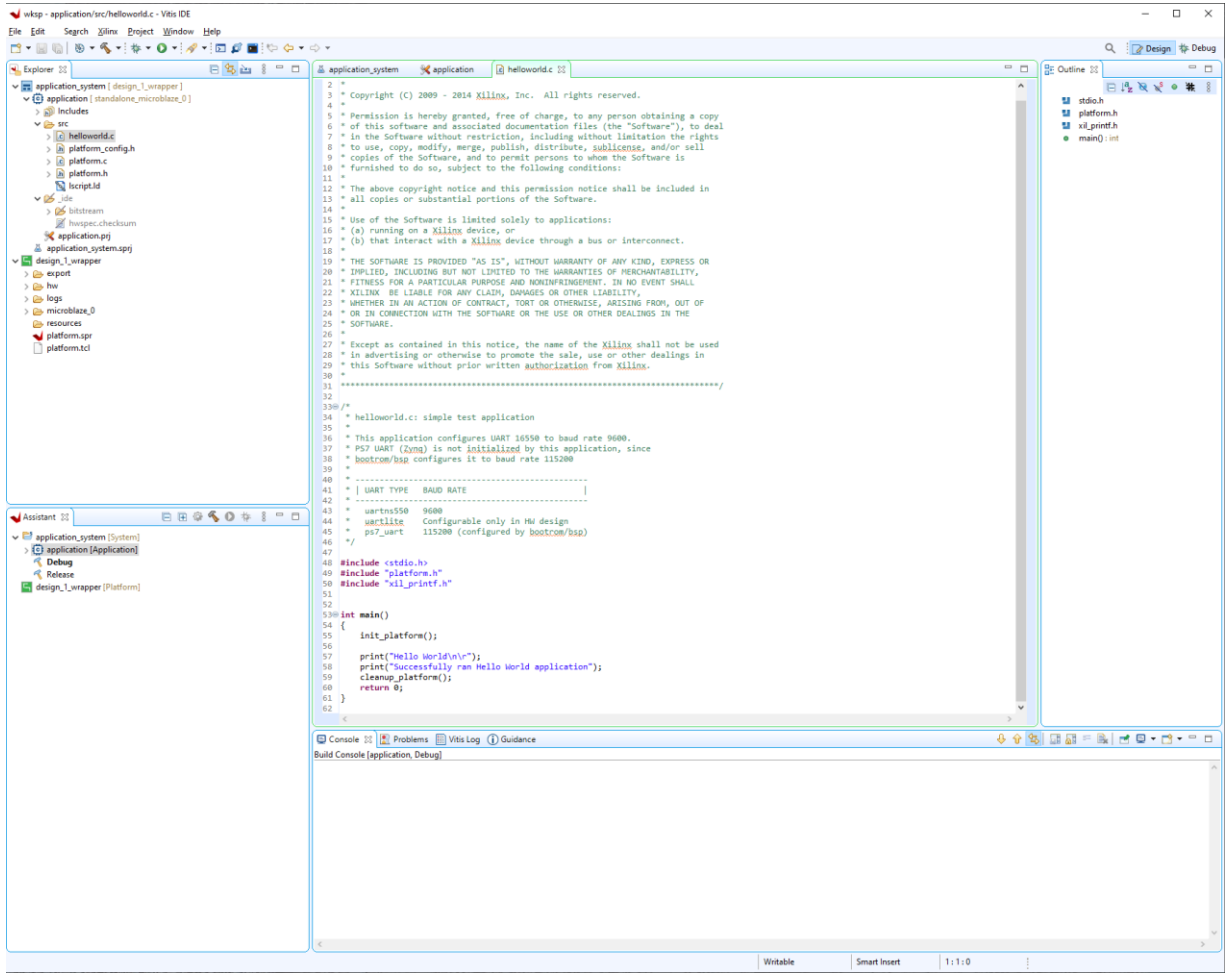
Lab: Mastering MicroBlaze

Select the Hello World Application



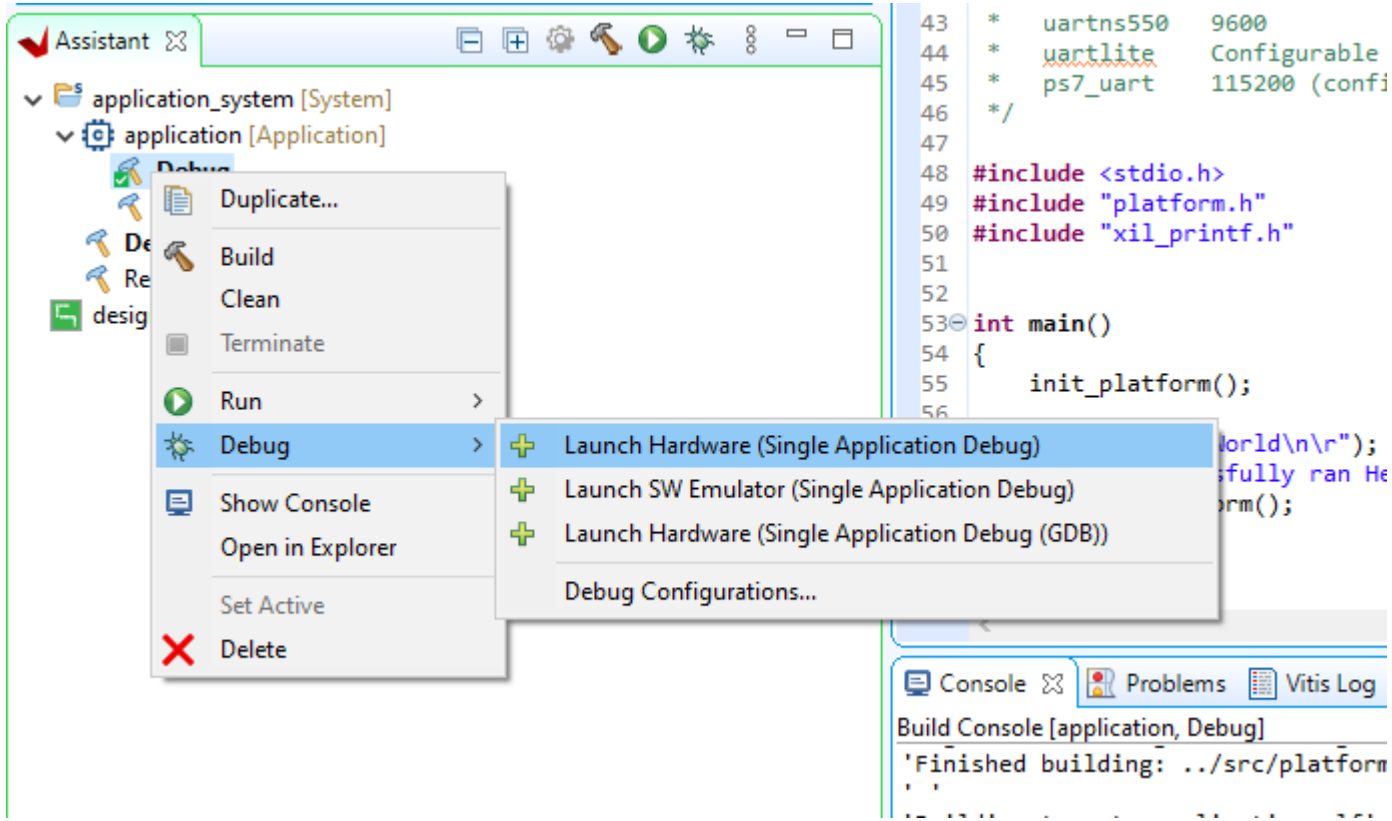
Lab: Mastering MicroBlaze

This will create a new application which prints hello world, we will use this to double check the correctness of the Vivado design. Click the Hammer to build the App



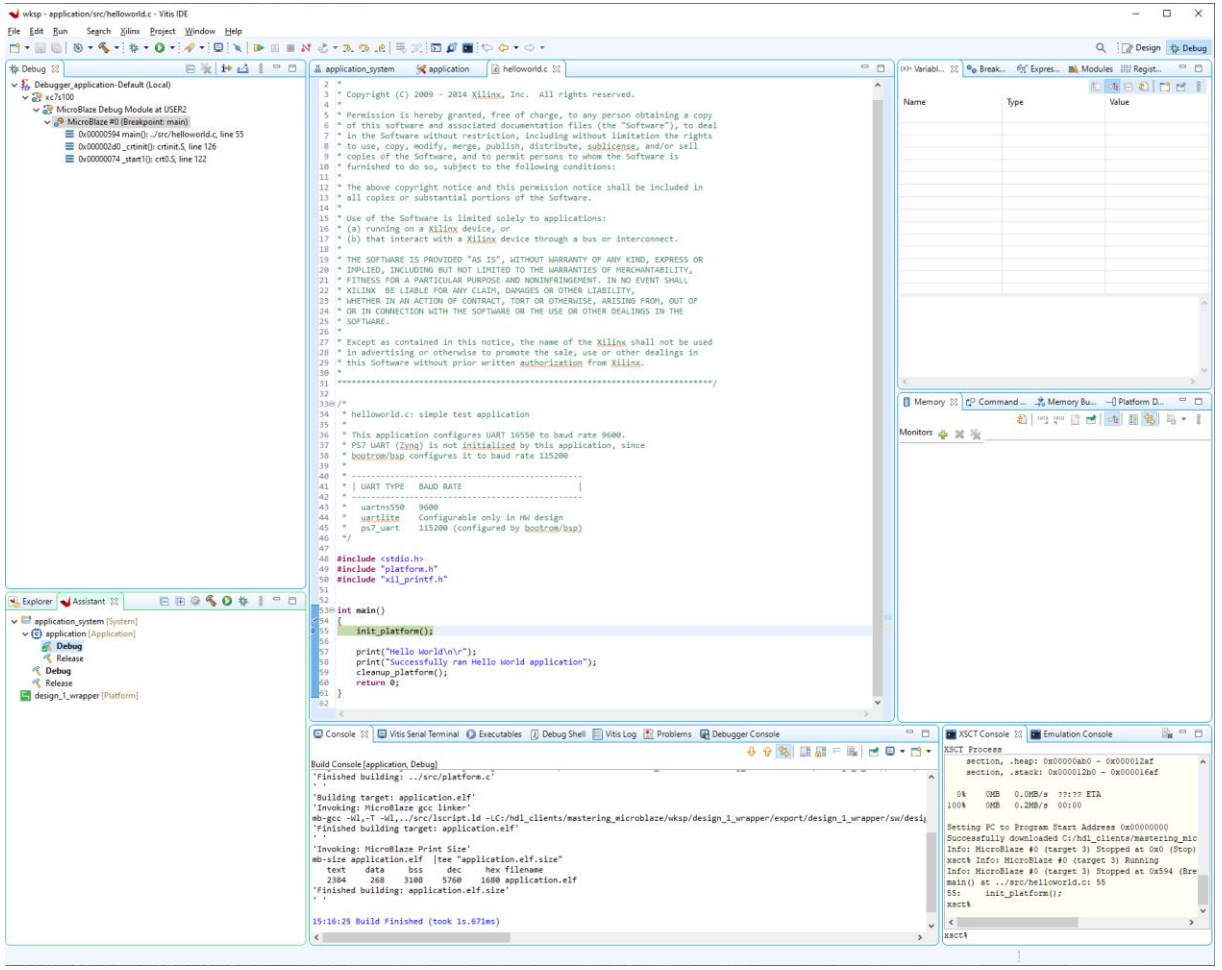
Lab: Mastering MicroBlaze

In the Assistant window select Debug, Launch on Hardware



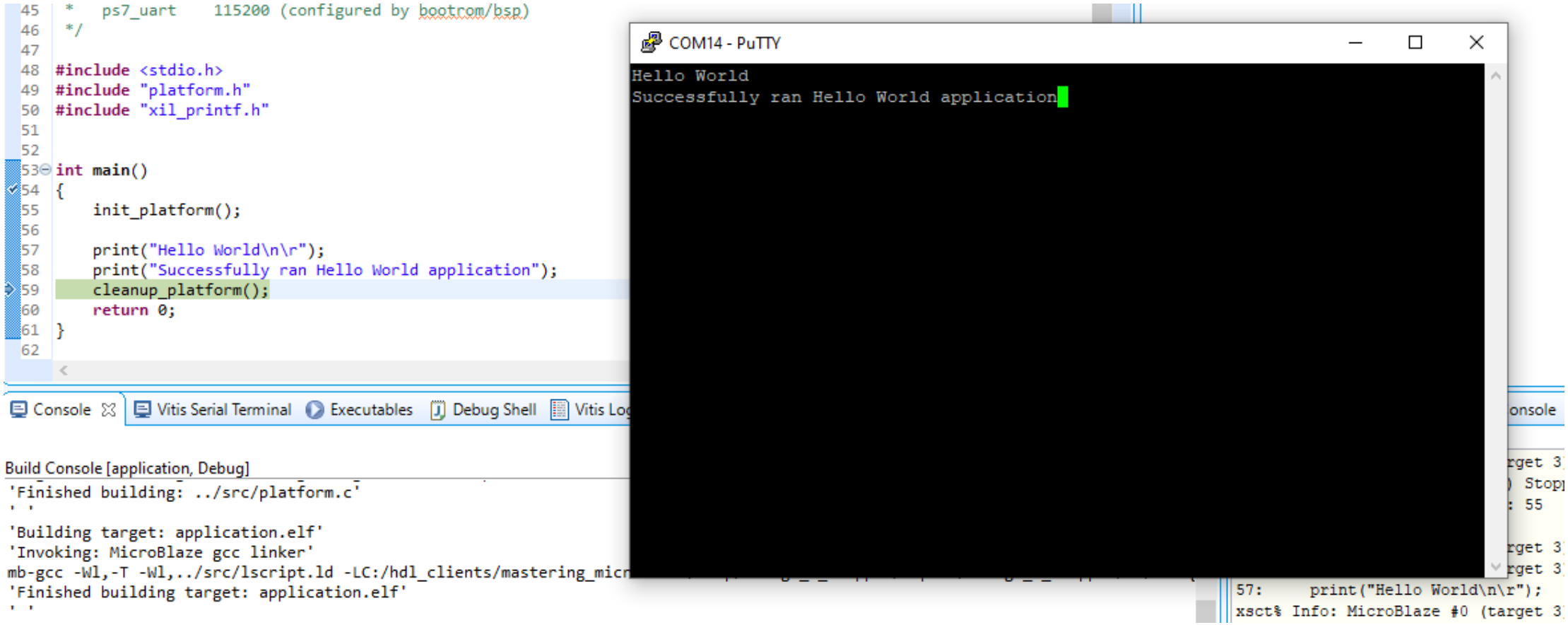
Lab: Mastering MicroBlaze

This will program the FPGA and download the application, halting it for execution at the first line.



Lab: Mastering MicroBlaze

Single step through the code and observe the output in a terminal window



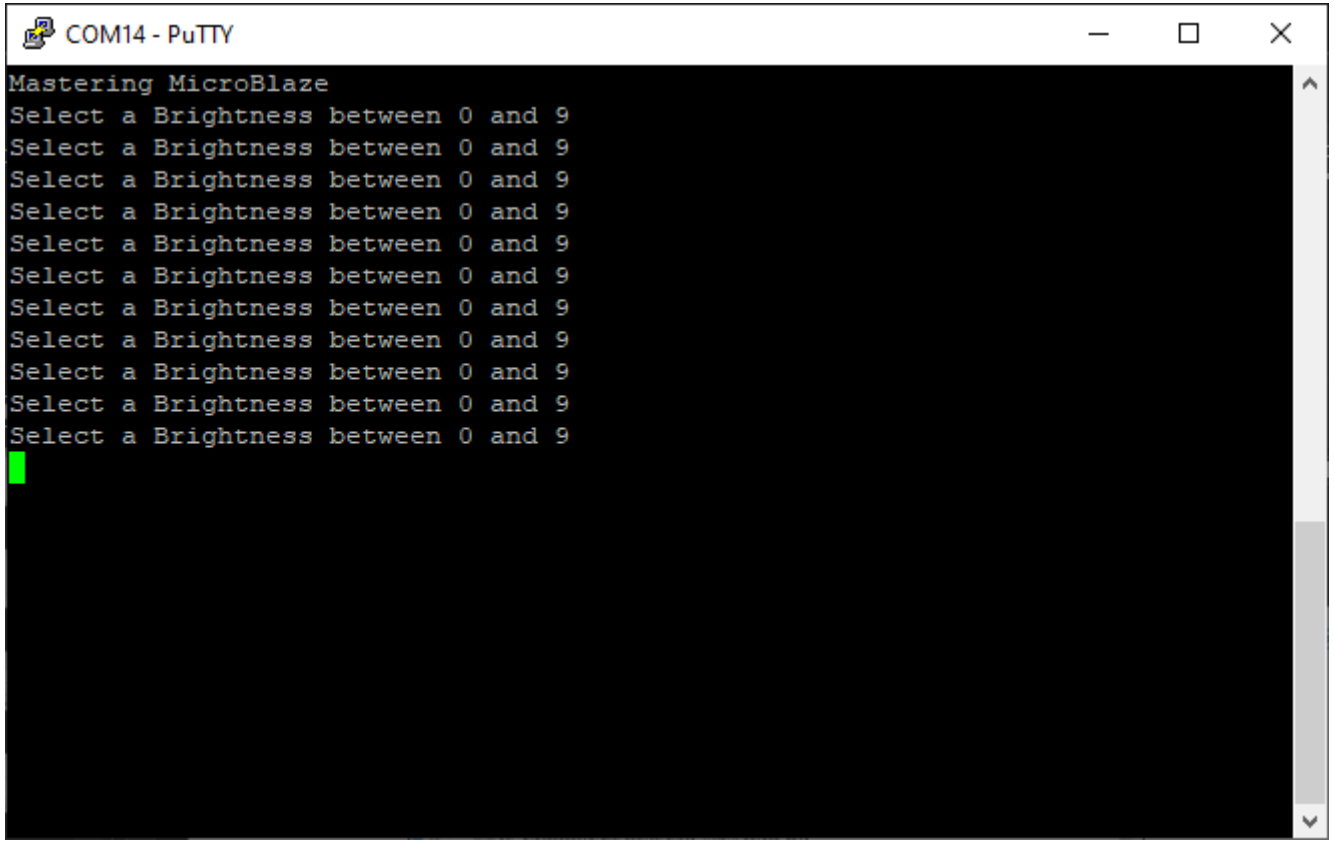
Lab: Mastering MicroBlaze

Replace the code in the hello world application with that in the Git Repo, rebuild the code and download it to the development board.

```c
int main()
{
    u8  DutyCycle;
    u8  NoOfCycles;
    u8  Div;
    u8  Value;
    u32 Period;
    u32 HighTime;
    u64 WaitCount;
    int Status;
    float DivF;

    init_platform();

    print("Mastering MicroBlaze\n\r");


    XTmrCtr_Initialize(&TimerCounterInst, TMRCTR_DEVICE_ID);

    XTmrCtr_PwmDisable(&TimerCounterInst);

    Div = DUTYCYCLE_DIVISOR;
    Period = PWM_PERIOD;
    HighTime = PWM_PERIOD * (Div/10) ;

    DutyCycle = XTmrCtr_PwmConfigure(&TimerCounterInst, Period, HighTime);

    xil_printf("PWM Configured for Duty Cycle = %d\r\n", DutyCycle);

    /* Enable PWM */
    XTmrCtr_PwmEnable(&TimerCounterInst);

    while(1){
         print("Select a Brightness between 0 and 9\n\r");

         /* Read an input value from the console. */
         Value = inbyte();
         Div = Value - 0x30;
         DivF = (float) Div /10;
         HighTime = PWM_PERIOD * DivF;

         XTmrCtr_PwmDisable(&TimerCounterInst);
         DutyCycle = XTmrCtr_PwmConfigure(&TimerCounterInst, Period, HighTime);
         XTmrCtr_PwmEnable(&TimerCounterInst);
    }

    cleanup_platform();
    return 0;
}
```

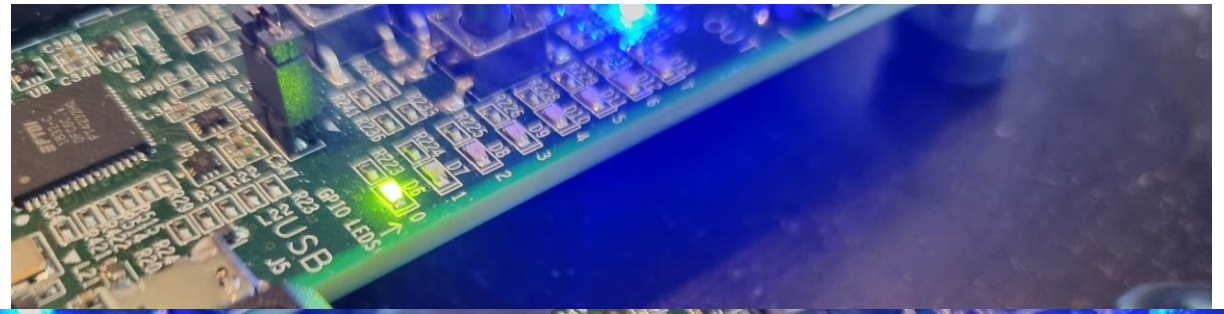
Lab: Mastering MicroBlaze

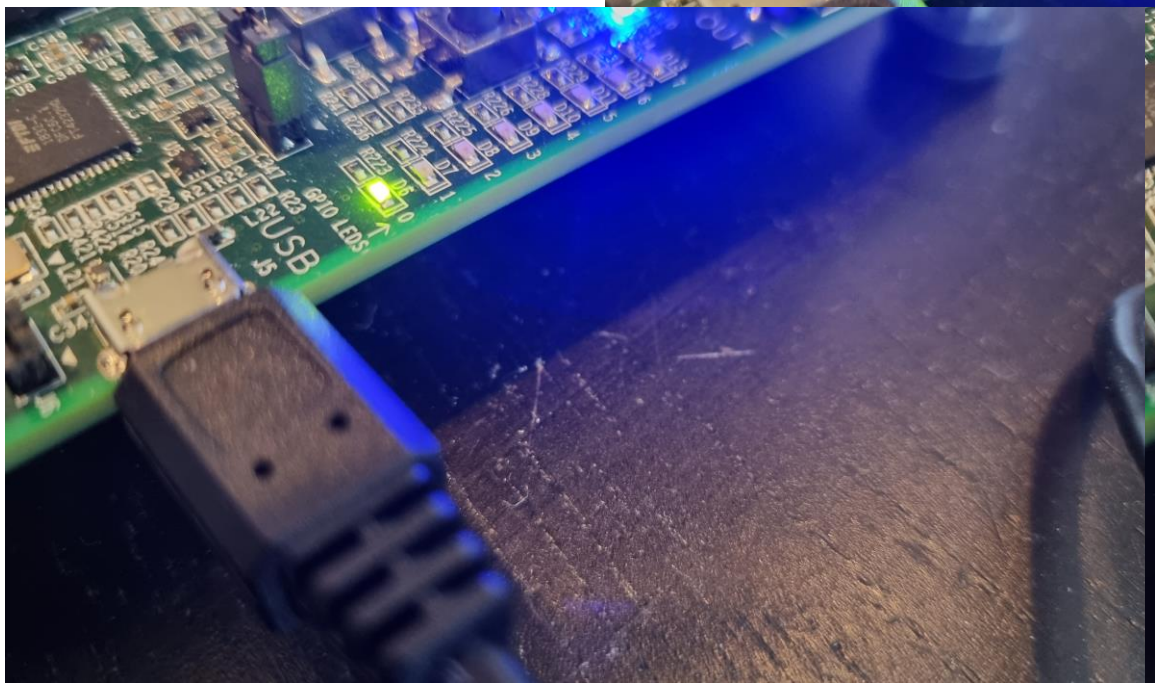
In a terminal window change the PWM to the LED and see the intensity change

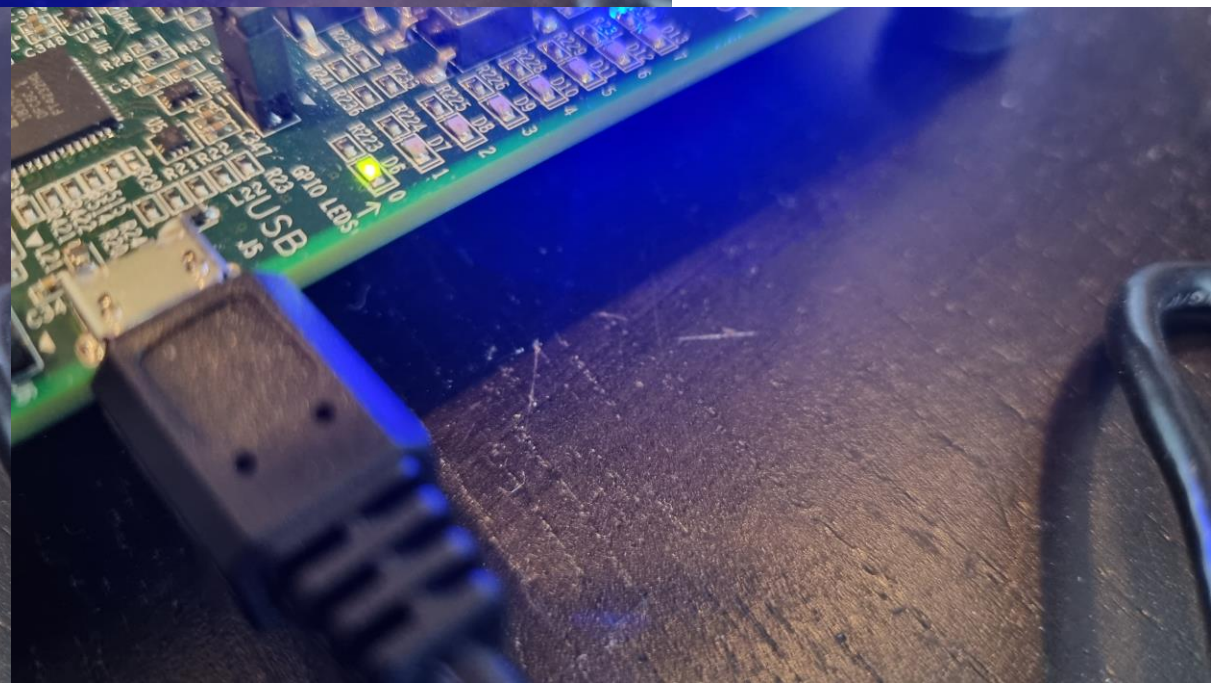


ADIUVO
ENGINEERING AND TRAINING, LTD.

Lab: Mastering MicroBlaze







ADIUVO

ENGINEERING AND TRAINING, LTD.

www.adiuvoengineering.com

adam@adiuvoengineering.com