CEC Continuing Education Center

DesignNews

Field-Programmable Gate Array (FPGA) Primer

Day 5:

Hardware and Software Design with Vivado and Vitis

Sponsored by

Digi-Key Electronics

informa markets

Webinar Logistics

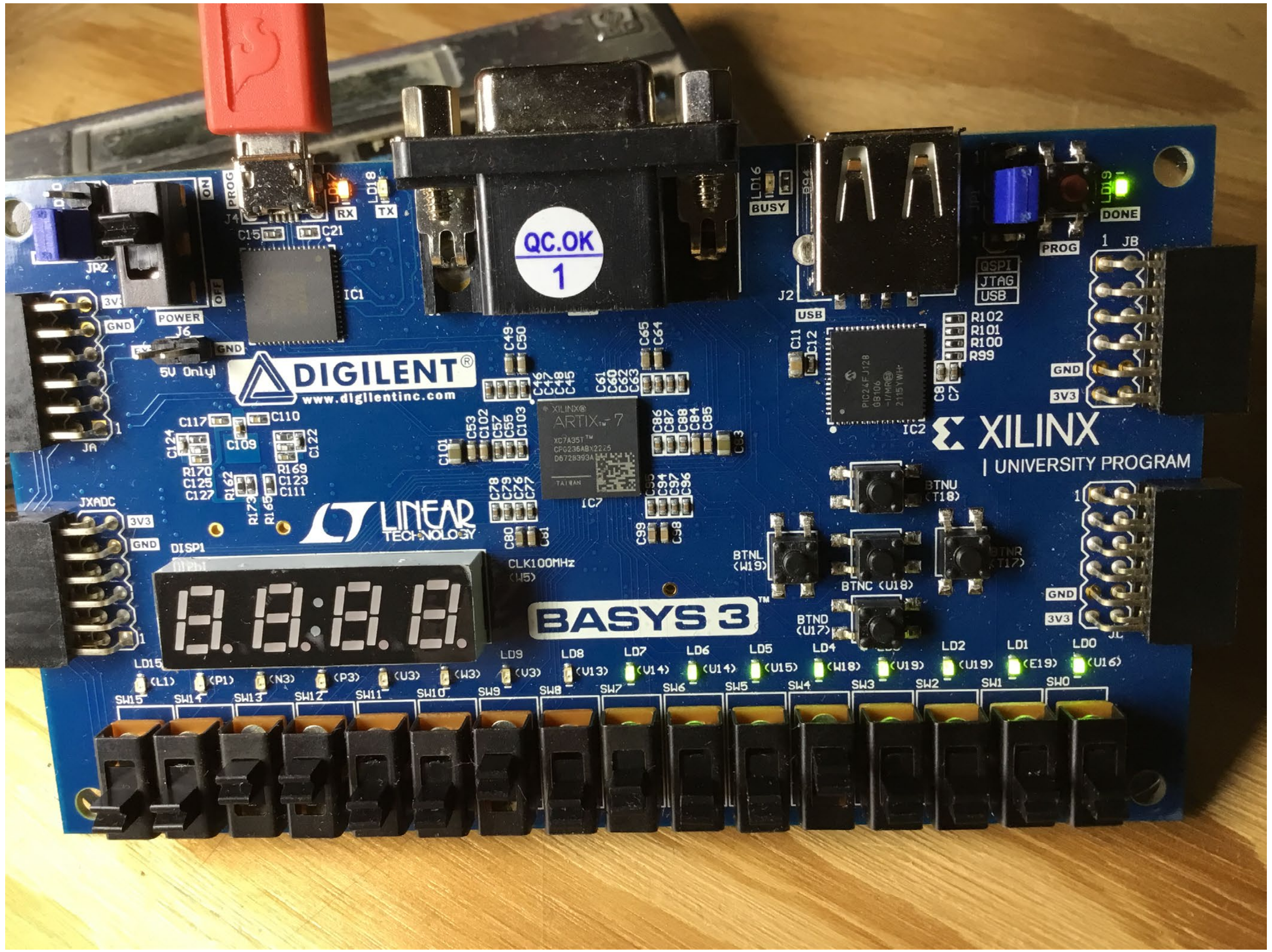
- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

Fred Eady

Visit ‘Lecturer Profile’ in your console for more details.

AGENDA

- **Create a Block Design with Vivado**
- **Finish the Design with Vitis**

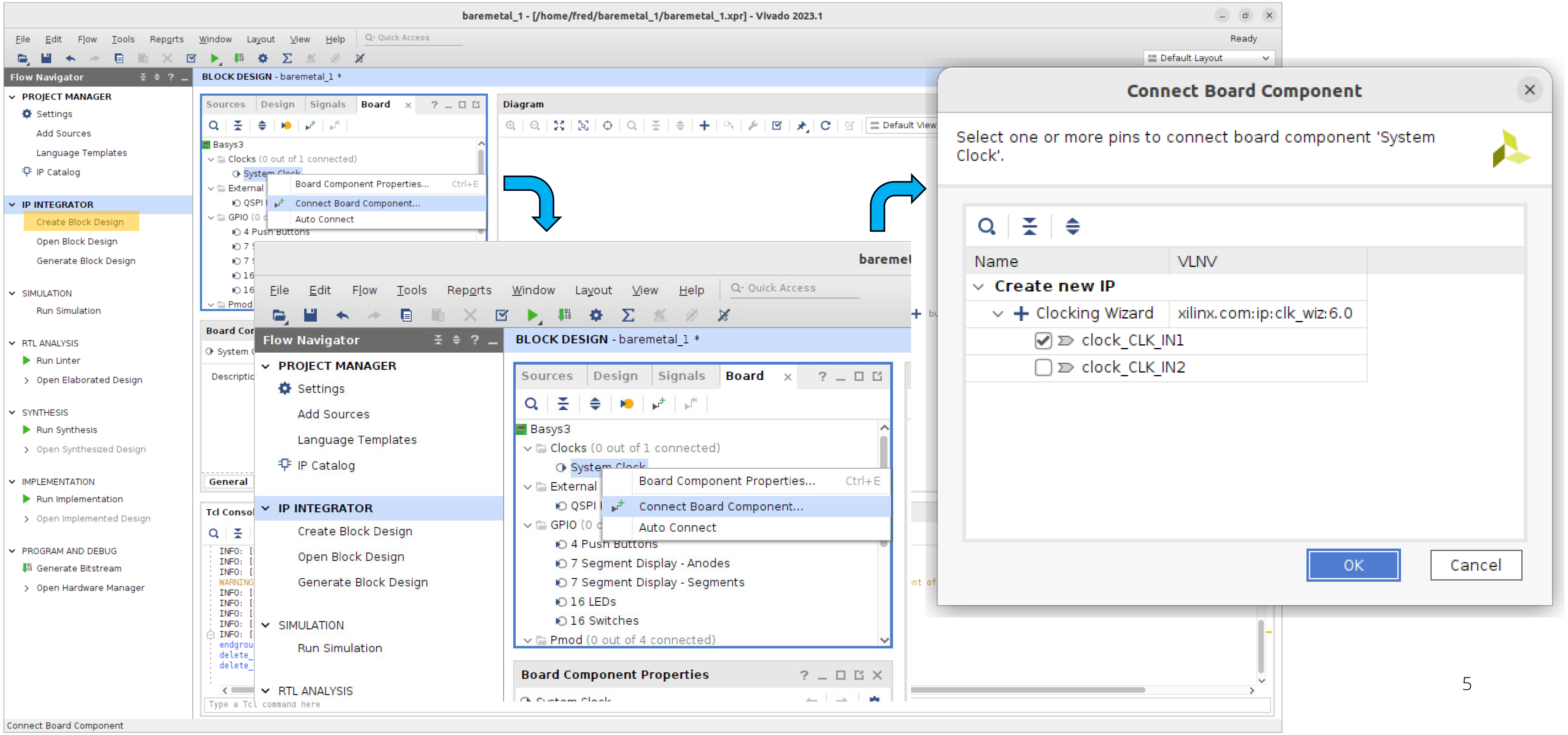


CEC Continuing Education Center

Field-Programmable Gate Array (FPGA) Primer
Hardware and Software Design with Vivado and Vitis
Create a Block Design with Vivado

Sponsored By Digi-Key Corporation

Create Block Design and Add System Clock

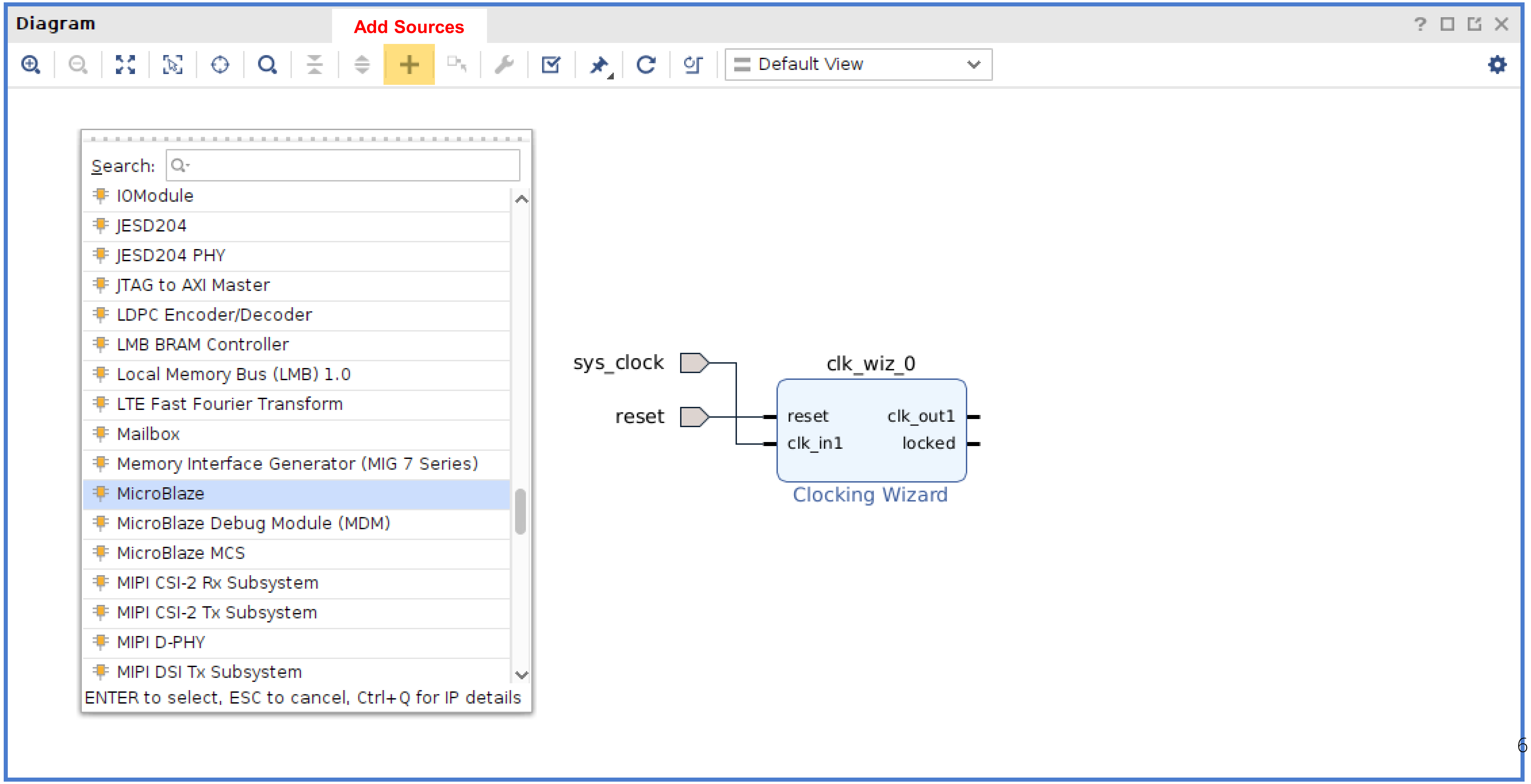


Connect Board Component

Select one or more pins to connect board component 'System Clock'.

Name	VLNV
Create new IP	
+ Clocking Wizard	xilinx.com:ip:clk_wiz:6.0
☑ clock_CLK_IN1	
☐ clock_CLK_IN2	

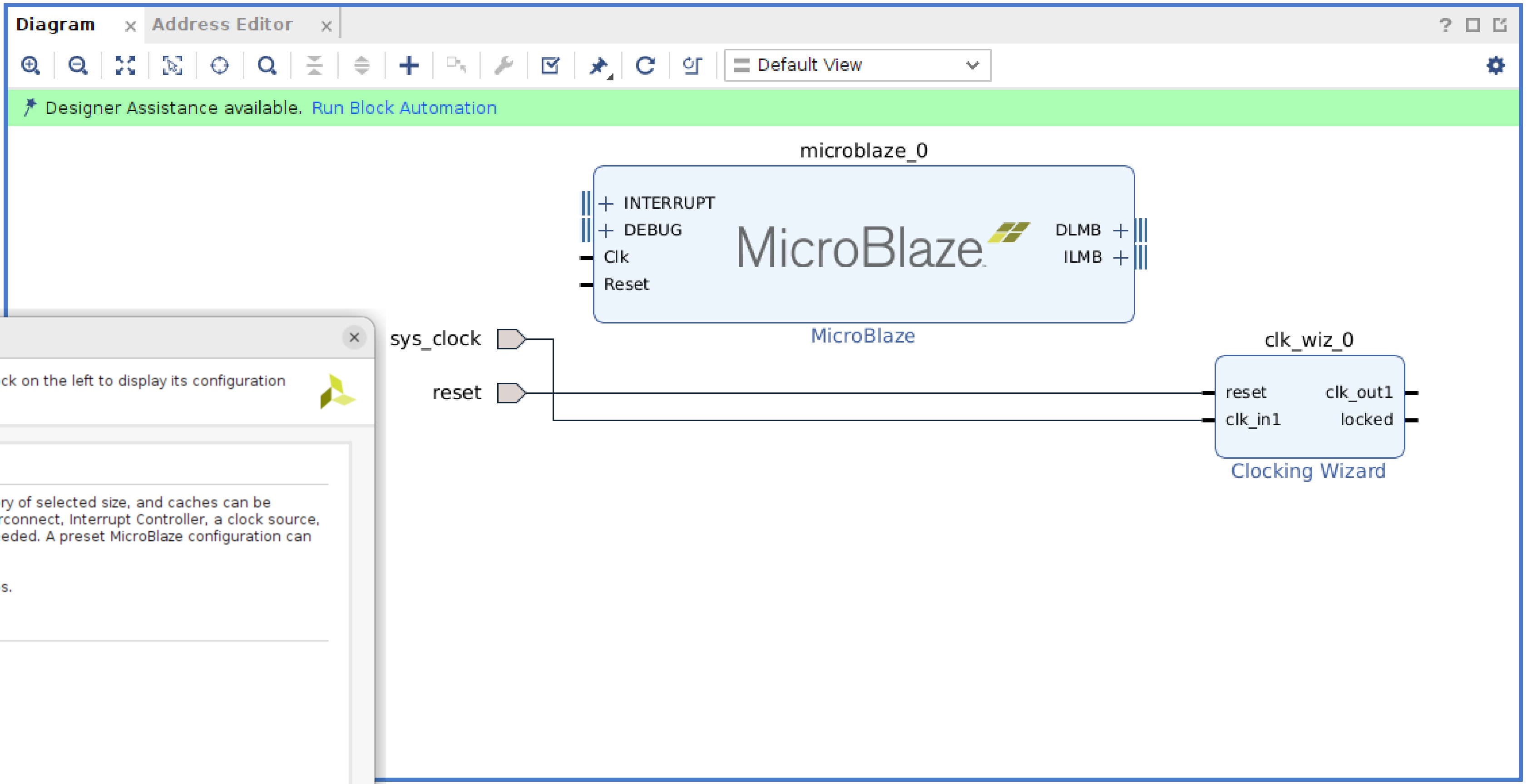
Add MicroBlaze

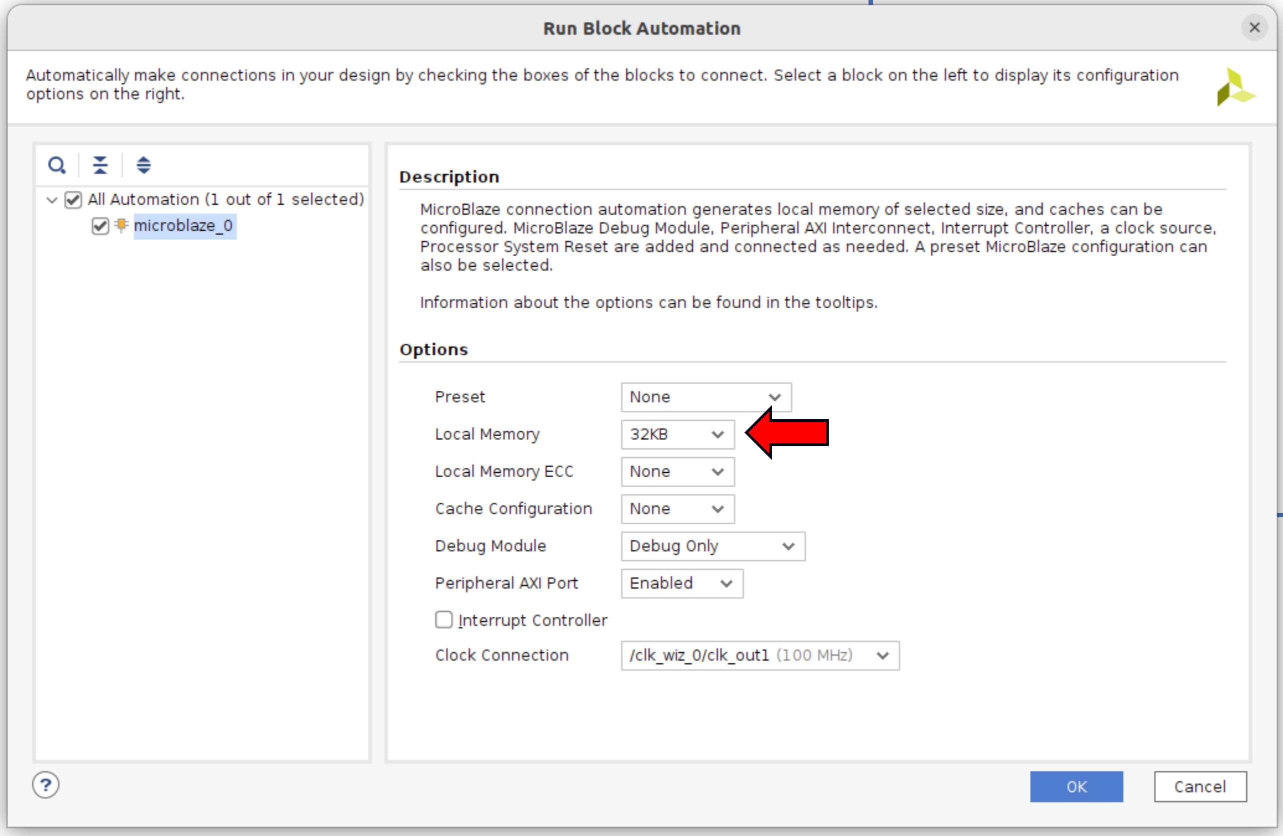


IOModule
JESD204
JESD204 PHY
JTAG to AXI Master
LDPC Encoder/Decoder
LMB BRAM Controller
Local Memory Bus (LMB) 1.0
LTE Fast Fourier Transform
Mailbox
Memory Interface Generator (MIG 7 Series)
MicroBlaze
MicroBlaze Debug Module (MDM)
MicroBlaze MCS
MIPI CSI-2 Rx Subsystem
MIPI CSI-2 Tx Subsystem
MIPI D-PHY
MIPI DSI Tx Subsystem

ENTER to select, ESC to cancel, Ctrl+Q for IP details

Specify 32KB of Local Memory




Run Block Automation

Automatically make connections in your design by checking the boxes of the blocks to connect. Select a block on the left to display its configuration options on the right.

- ☑ All Automation (1 out of 1 selected)
  - ☑ microblaze_0

Description

MicroBlaze connection automation generates local memory of selected size, and caches can be configured. MicroBlaze Debug Module, Peripheral AXI Interconnect, Interrupt Controller, a clock source, Processor System Reset are added and connected as needed. A preset MicroBlaze configuration can also be selected.

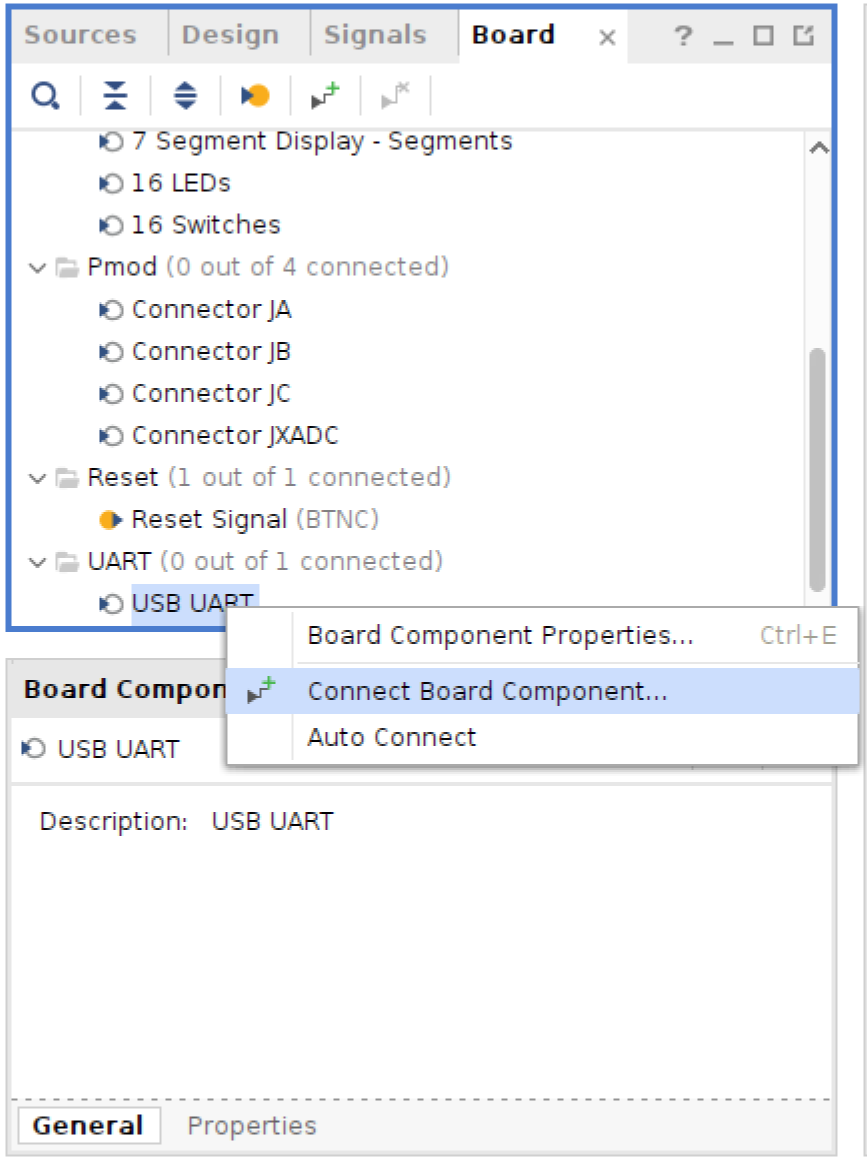
Information about the options can be found in the tooltips.

Options

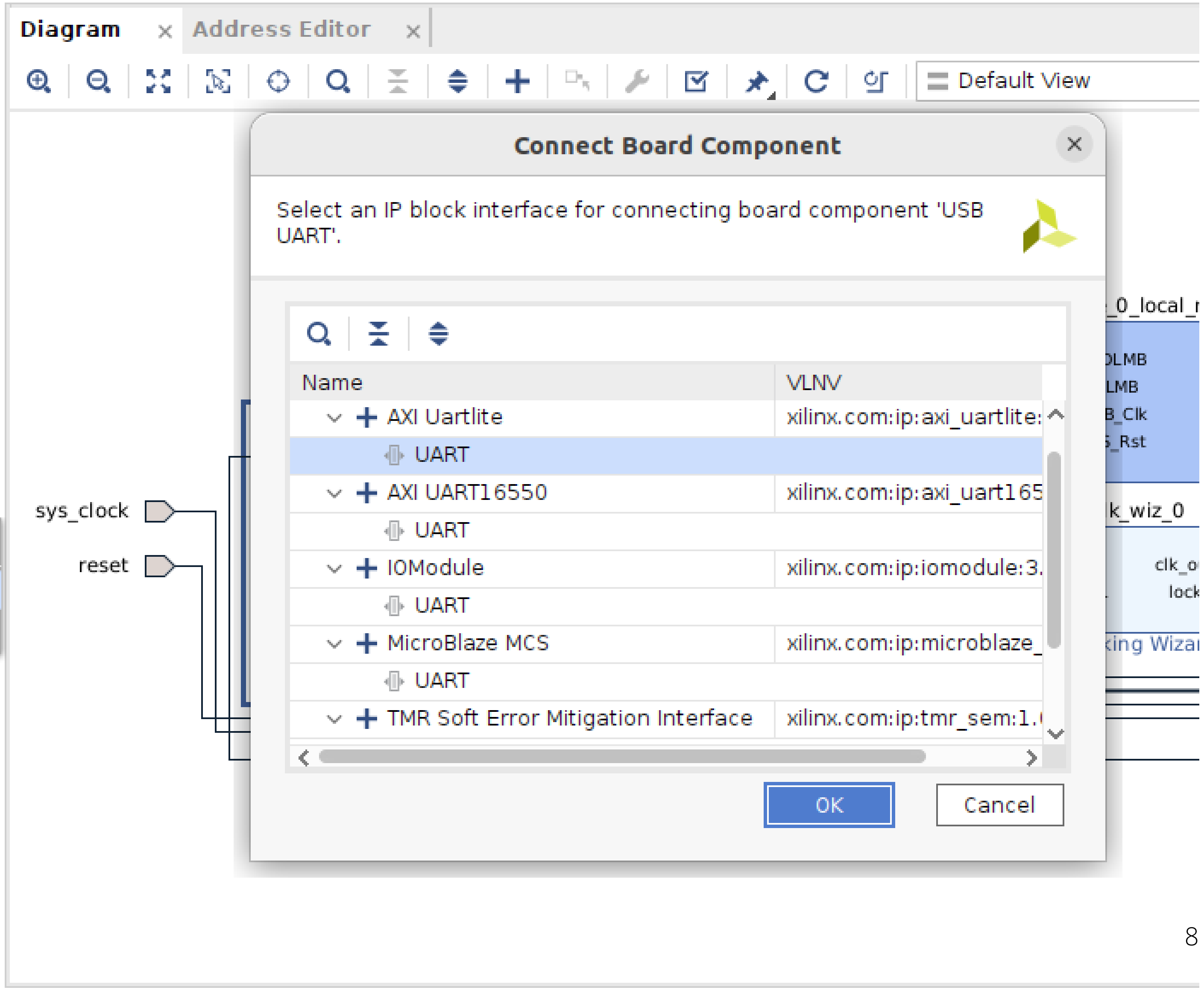
Preset	None
Local Memory	32KB
Local Memory ECC	None
Cache Configuration	None
Debug Module	Debug Only
Peripheral AXI Port	Enabled
☐ Interrupt Controller	
Clock Connection	/clk_wiz_0/clk_out1 (100 MHz)

OK Cancel

Add USB UART


Sources
Design
Signals
Board
7 Segment Display - Segments
16 LEDs
16 Switches
Pmod (0 out of 4 connected)
Connector JA
Connector JB
Connector JC
Connector JXADC
Reset (1 out of 1 connected)
Reset Signal (BTNC)
UART (0 out of 1 connected)
USB UART
Board Component Properties... Ctrl+E
Connect Board Component...
Auto Connect
Board Compon
USB UART
General
Properties

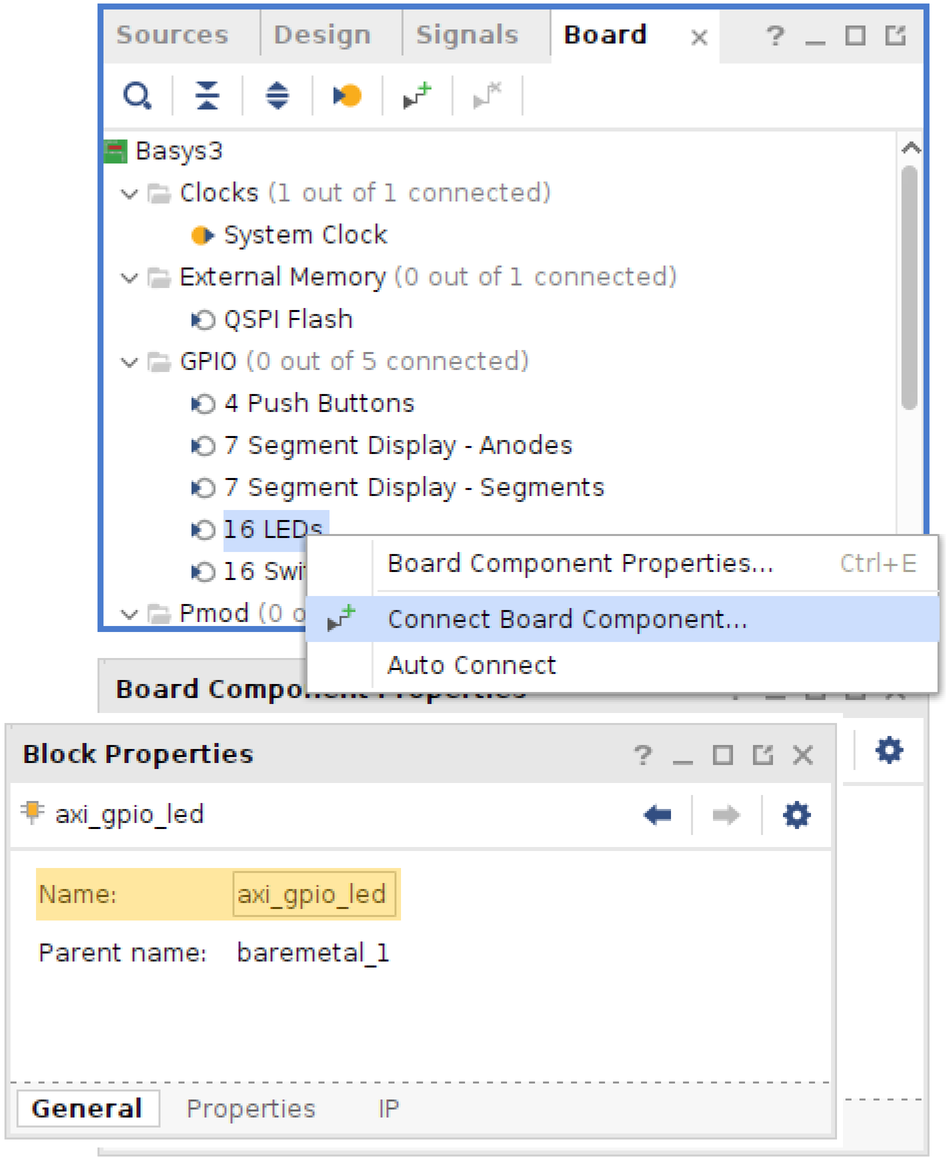
Description: USB UART



Select an IP block interface for connecting board component 'USB UART'.

Name	VLNV
AXI Uartlite	xilinx.com:ip:axi_uartlite:
UART	
AXI UART16550	xilinx.com:ip:axi_uart165
UART	
IOModule	xilinx.com:ip:iomodule:3.
UART	
MicroBlaze MCS	xilinx.com:ip:microblaze_
UART	
TMR Soft Error Mitigation Interface	xilinx.com:ip:tmr_sem:1.

Add Bank of 16 LEDs – Rename LED Block



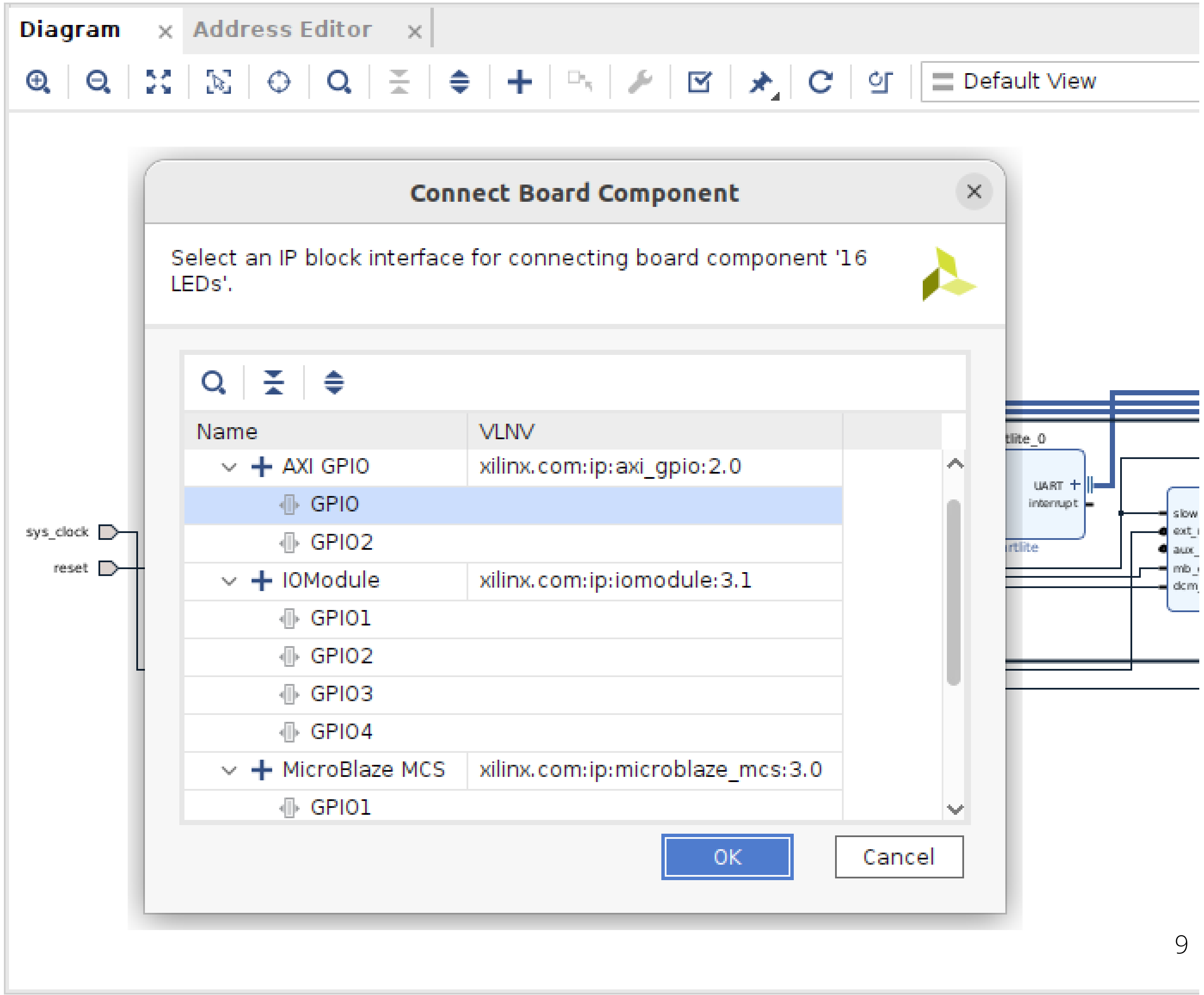
Basys3
- Clocks (1 out of 1 connected)
  - System Clock
- External Memory (0 out of 1 connected)
  - QSPI Flash
- GPIO (0 out of 5 connected)
  - 4 Push Buttons
  - 7 Segment Display - Anodes
  - 7 Segment Display - Segments
  - 16 LEDs
  - 16 Swi
- Pmod (0 o

Board Component Properties

Block Properties: axi_gpio_led

Name: axi_gpio_led

Parent name: baremetal_1

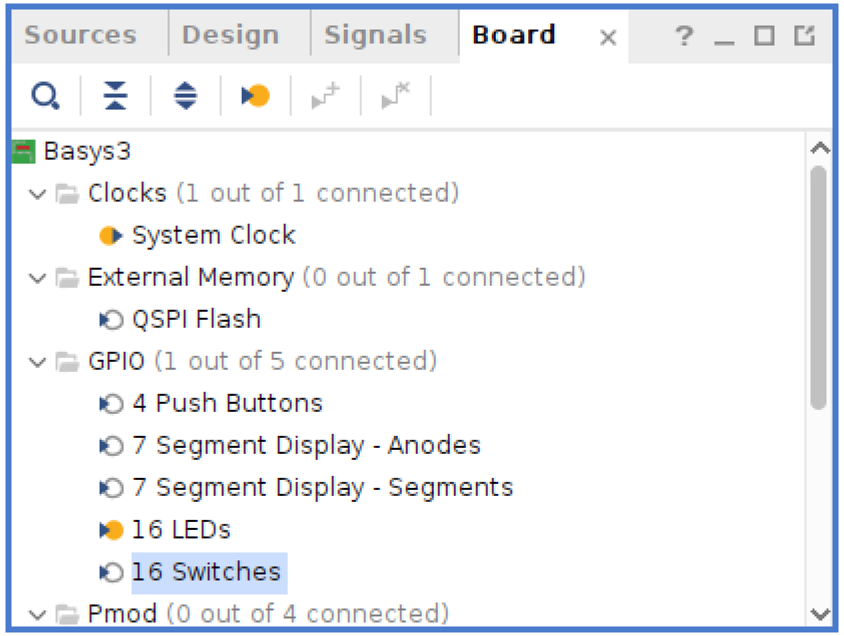


Connect Board Component

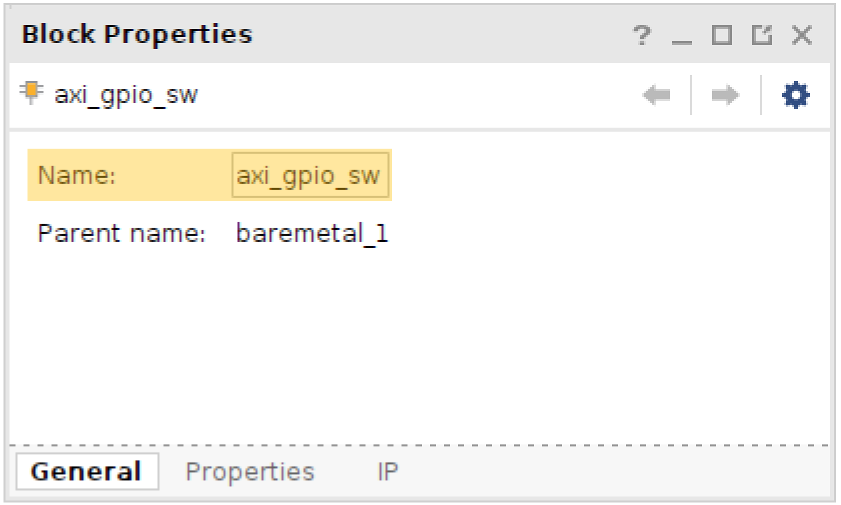
Select an IP block interface for connecting board component '16 LEDs'.

Name	VLNV
AXI GPIO	xilinx.com:ip:axi_gpio:2.0
GPIO	
GPIO2	
IOModule	xilinx.com:ip:iomodule:3.1
GPIO1	
GPIO2	
GPIO3	
GPIO4	
MicroBlaze MCS	xilinx.com:ip:microblaze_mcs:3.0
GPIO1	

Add Bank of 16 Switches – Rename Switch Block



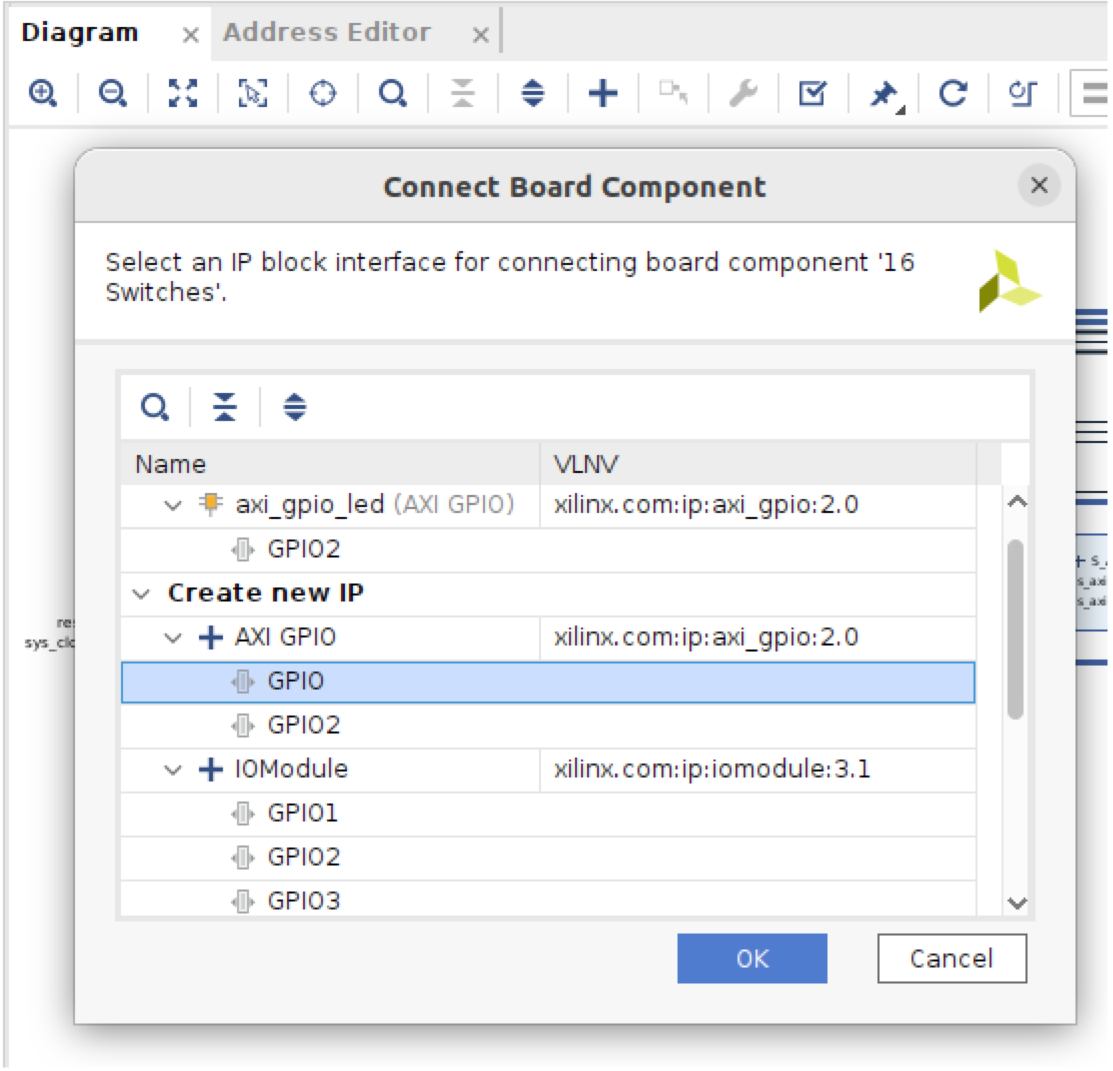
Basys3
- Clocks (1 out of 1 connected)
  - System Clock
- External Memory (0 out of 1 connected)
  - QSPI Flash
- GPIO (1 out of 5 connected)
  - 4 Push Buttons
  - 7 Segment Display - Anodes
  - 7 Segment Display - Segments
  - 16 LEDs
  - 16 Switches
- Pmod (0 out of 4 connected)



axi_gpio_sw

Name: axi_gpio_sw

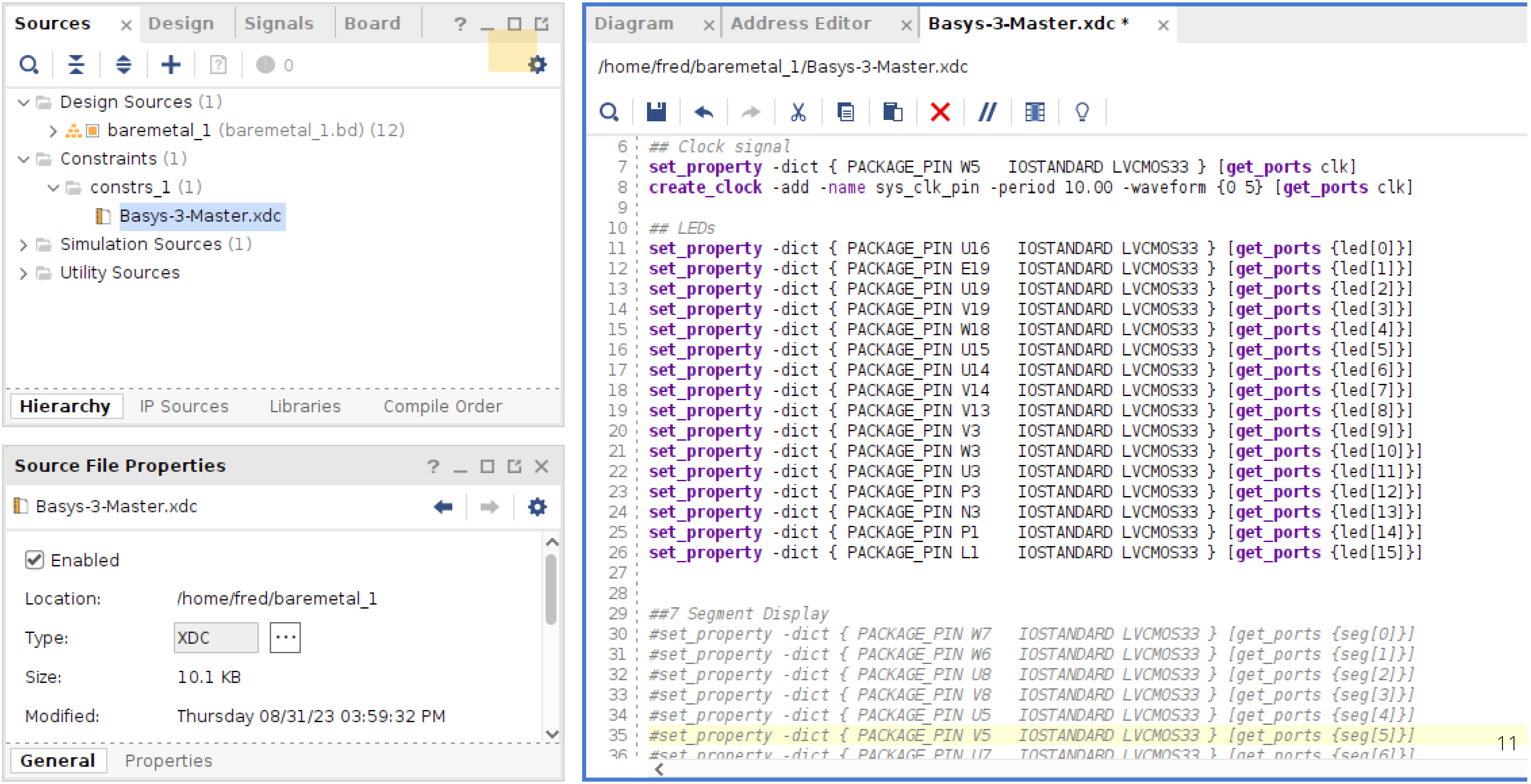
Parent name: baremetal_1



Select an IP block interface for connecting board component '16 Switches'.

Name	VLNV
axi_gpio_led (AXI GPIO)	xilinx.com:ip:axi_gpio:2.0
GPIO2	
Create new IP	
AXI GPIO	xilinx.com:ip:axi_gpio:2.0
GPIO	
GPIO2	
IOModule	xilinx.com:ip:iomodule:3.1
GPIO1	
GPIO2	
GPIO3	

Add Basys 3 Constraint File

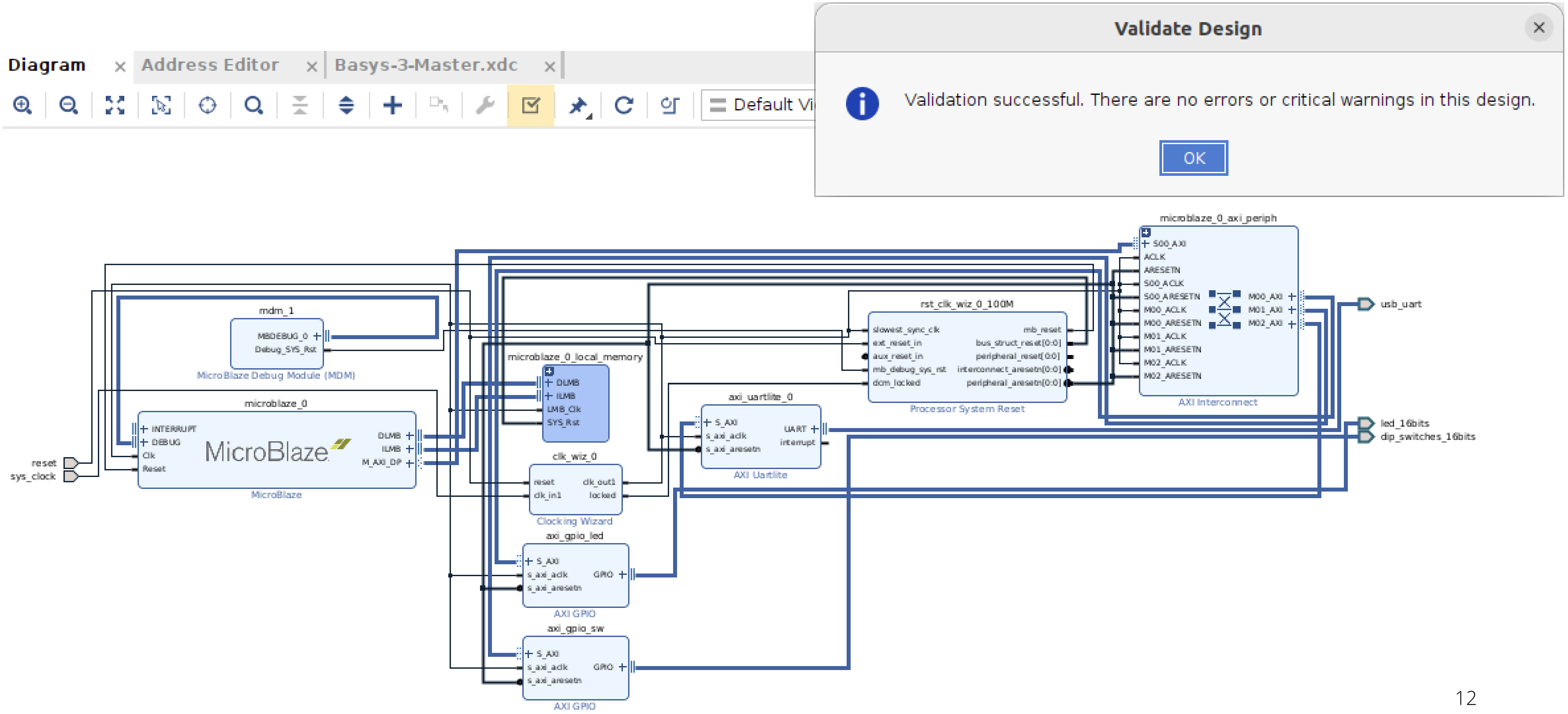


Enabled
Location: /home/fred/baremetal_1
Type: XDC
Size: 10.1 KB
Modified: Thursday 08/31/23 03:59:32 PM

/home/fred/baremetal_1/Basys-3-Master.xdc

```
## Clock signal
set_property -dict { PACKAGE_PIN W5   IOSTANDARD LVCMOS33 } [get_ports clk]
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]

## LEDs
set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports {led[0]}]
set_property -dict { PACKAGE_PIN E19   IOSTANDARD LVCMOS33 } [get_ports {led[1]}]
set_property -dict { PACKAGE_PIN U19   IOSTANDARD LVCMOS33 } [get_ports {led[2]}]
set_property -dict { PACKAGE_PIN V19   IOSTANDARD LVCMOS33 } [get_ports {led[3]}]
set_property -dict { PACKAGE_PIN W18   IOSTANDARD LVCMOS33 } [get_ports {led[4]}]
set_property -dict { PACKAGE_PIN U15   IOSTANDARD LVCMOS33 } [get_ports {led[5]}]
set_property -dict { PACKAGE_PIN U14   IOSTANDARD LVCMOS33 } [get_ports {led[6]}]
set_property -dict { PACKAGE_PIN V14   IOSTANDARD LVCMOS33 } [get_ports {led[7]}]
set_property -dict { PACKAGE_PIN V13   IOSTANDARD LVCMOS33 } [get_ports {led[8]}]
set_property -dict { PACKAGE_PIN V3    IOSTANDARD LVCMOS33 } [get_ports {led[9]}]
set_property -dict { PACKAGE_PIN W3    IOSTANDARD LVCMOS33 } [get_ports {led[10]}]
set_property -dict { PACKAGE_PIN U3    IOSTANDARD LVCMOS33 } [get_ports {led[11]}]
set_property -dict { PACKAGE_PIN P3    IOSTANDARD LVCMOS33 } [get_ports {led[12]}]
set_property -dict { PACKAGE_PIN N3    IOSTANDARD LVCMOS33 } [get_ports {led[13]}]
set_property -dict { PACKAGE_PIN P1    IOSTANDARD LVCMOS33 } [get_ports {led[14]}]
set_property -dict { PACKAGE_PIN L1    IOSTANDARD LVCMOS33 } [get_ports {led[15]}]


##7 Segment Display
#set_property -dict { PACKAGE_PIN W7   IOSTANDARD LVCMOS33 } [get_ports {seg[0]}]
#set_property -dict { PACKAGE_PIN W6   IOSTANDARD LVCMOS33 } [get_ports {seg[1]}]
#set_property -dict { PACKAGE_PIN U8   IOSTANDARD LVCMOS33 } [get_ports {seg[2]}]
#set_property -dict { PACKAGE_PIN V8   IOSTANDARD LVCMOS33 } [get_ports {seg[3]}]
#set_property -dict { PACKAGE_PIN U5   IOSTANDARD LVCMOS33 } [get_ports {seg[4]}]
#set_property -dict { PACKAGE_PIN V5   IOSTANDARD LVCMOS33 } [get_ports {seg[5]}]
#set_property -dict { PACKAGE_PIN U7   IOSTANDARD LVCMOS33 } [get_ports {seg[6]}]
```

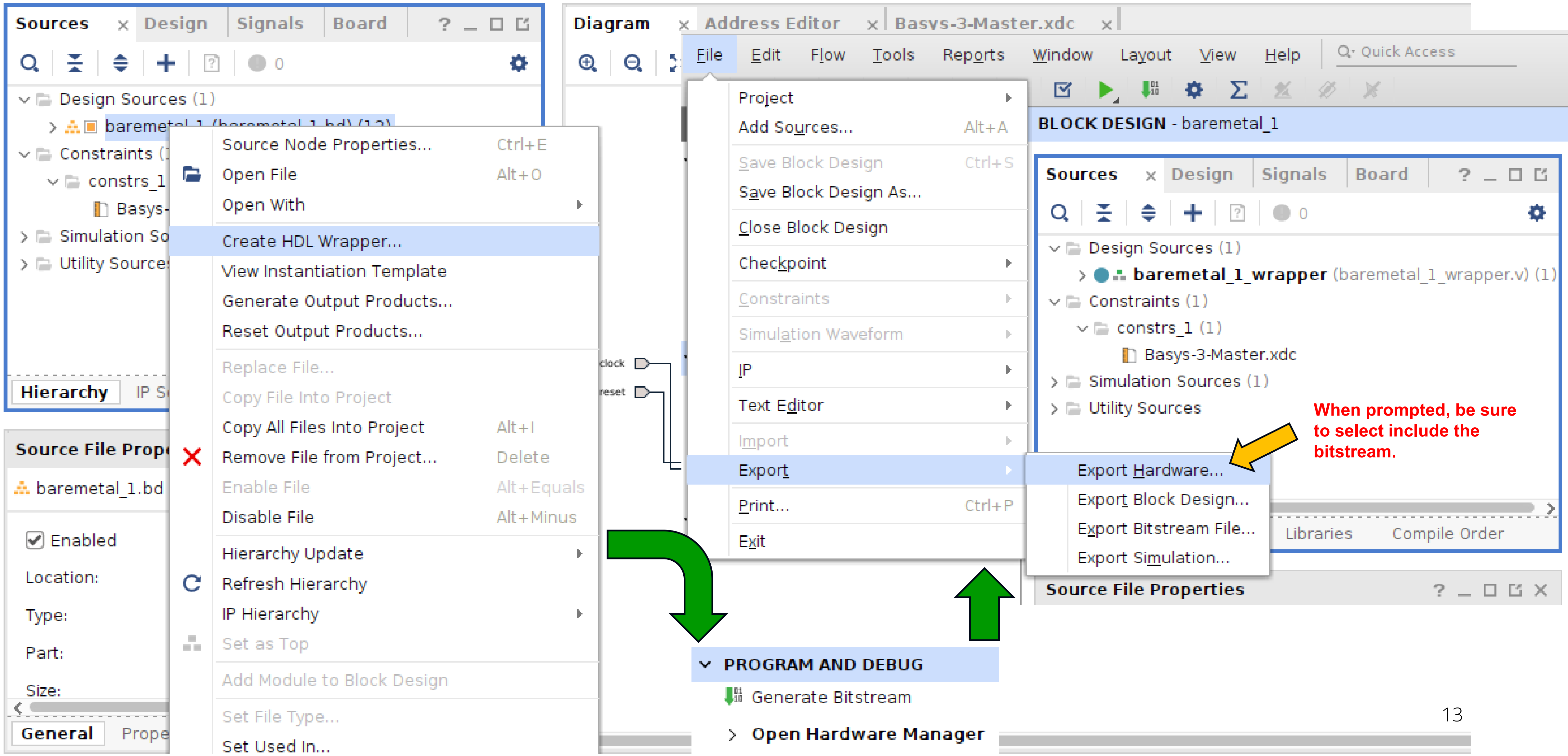
Validate the Design



Validate Design

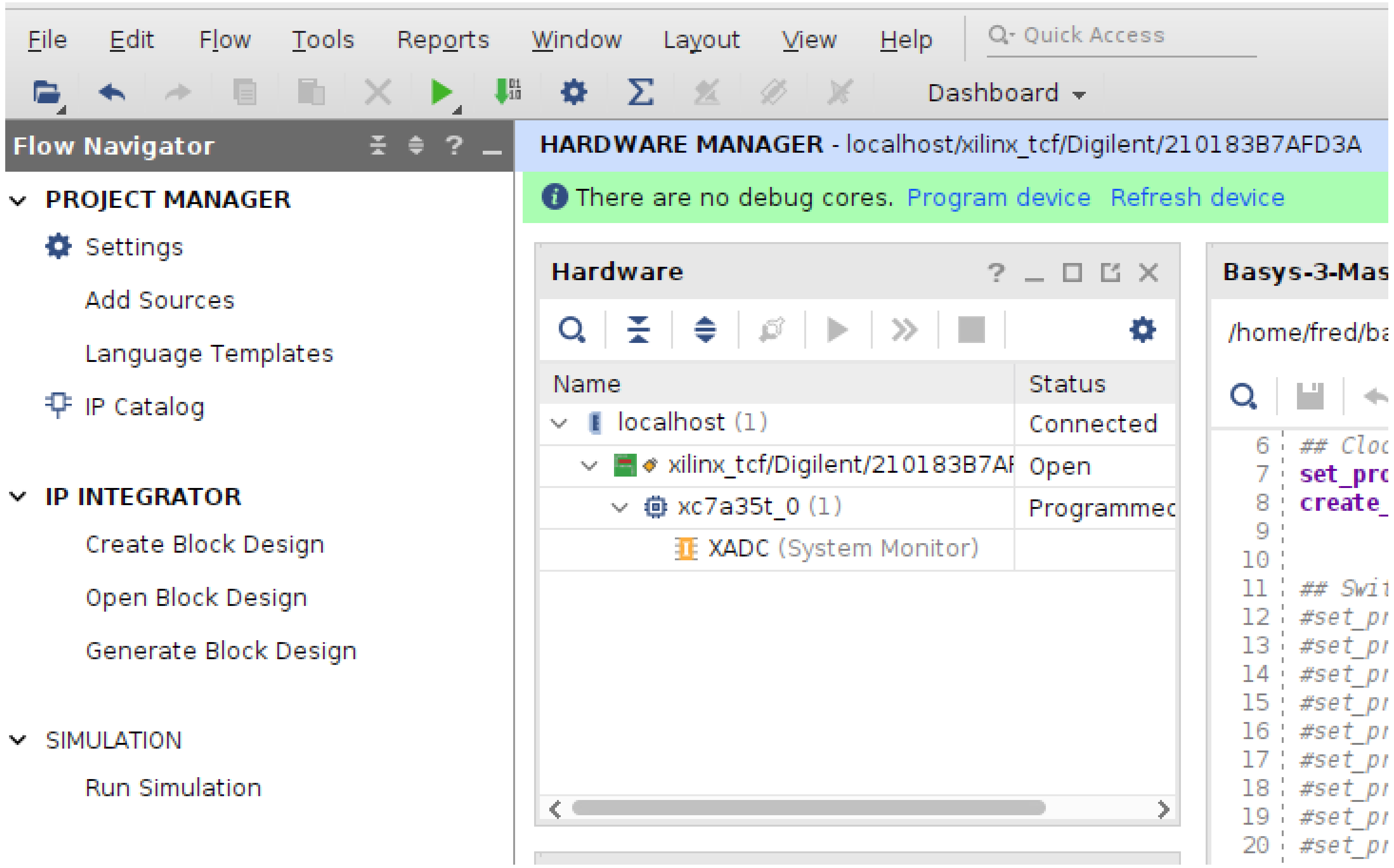
Validation successful. There are no errors or critical warnings in this design.

Create the HDL Wrapper – Generate Bitstream - Export



When prompted, be sure to select include the bitstream.

Activate Hardware Manager

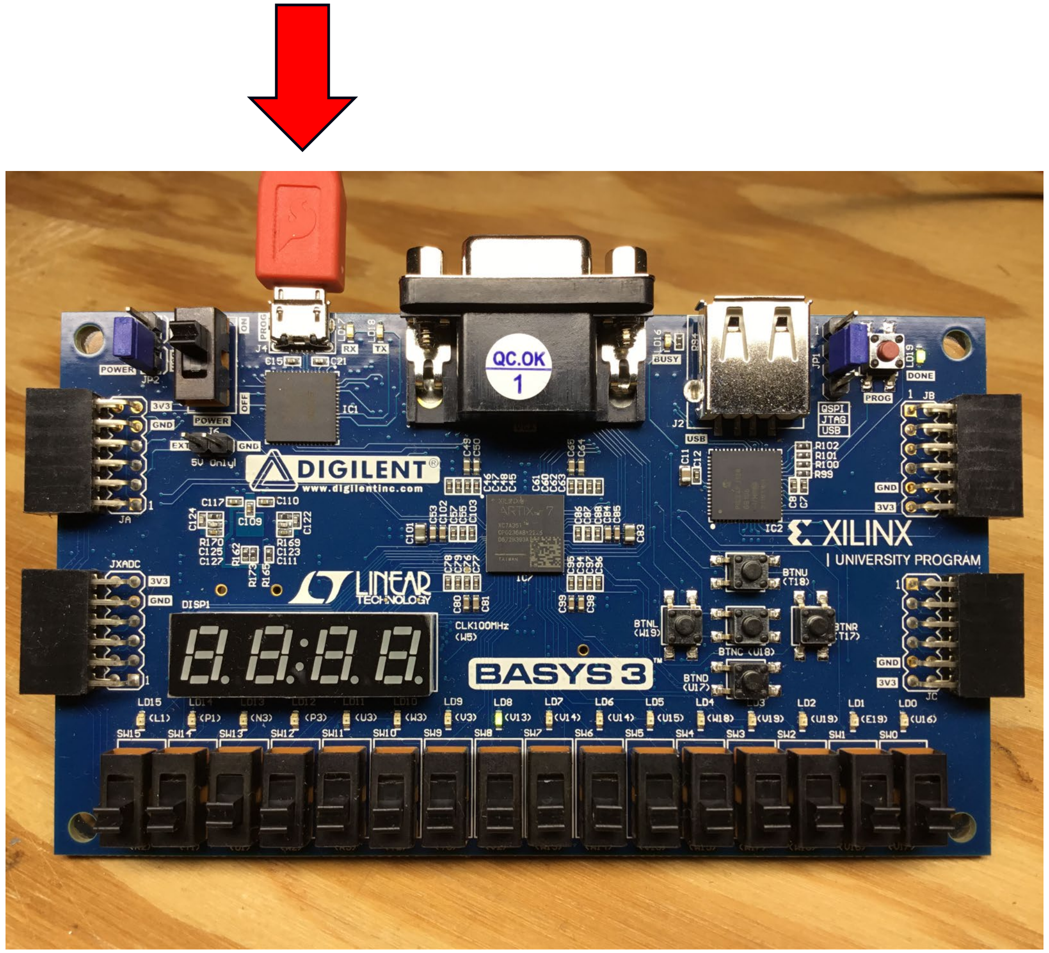


HARDWARE MANAGER - localhost/xilinx_tcf/Digilent/210183B7AFD3A

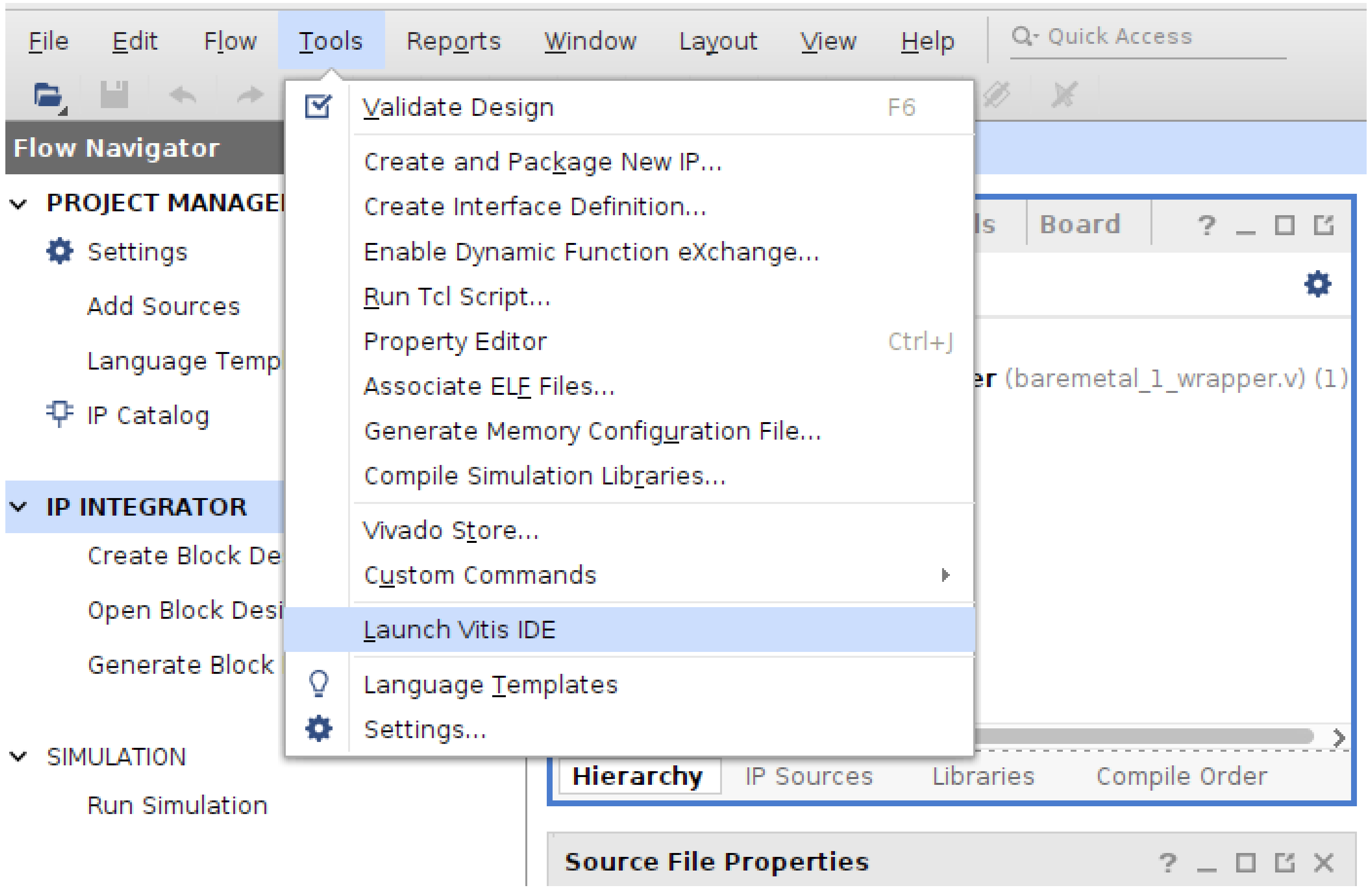
There are no debug cores. Program device Refresh device

Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210183B7AF	Open
xc7a35t_0 (1)	Programmed
XADC (System Monitor)	

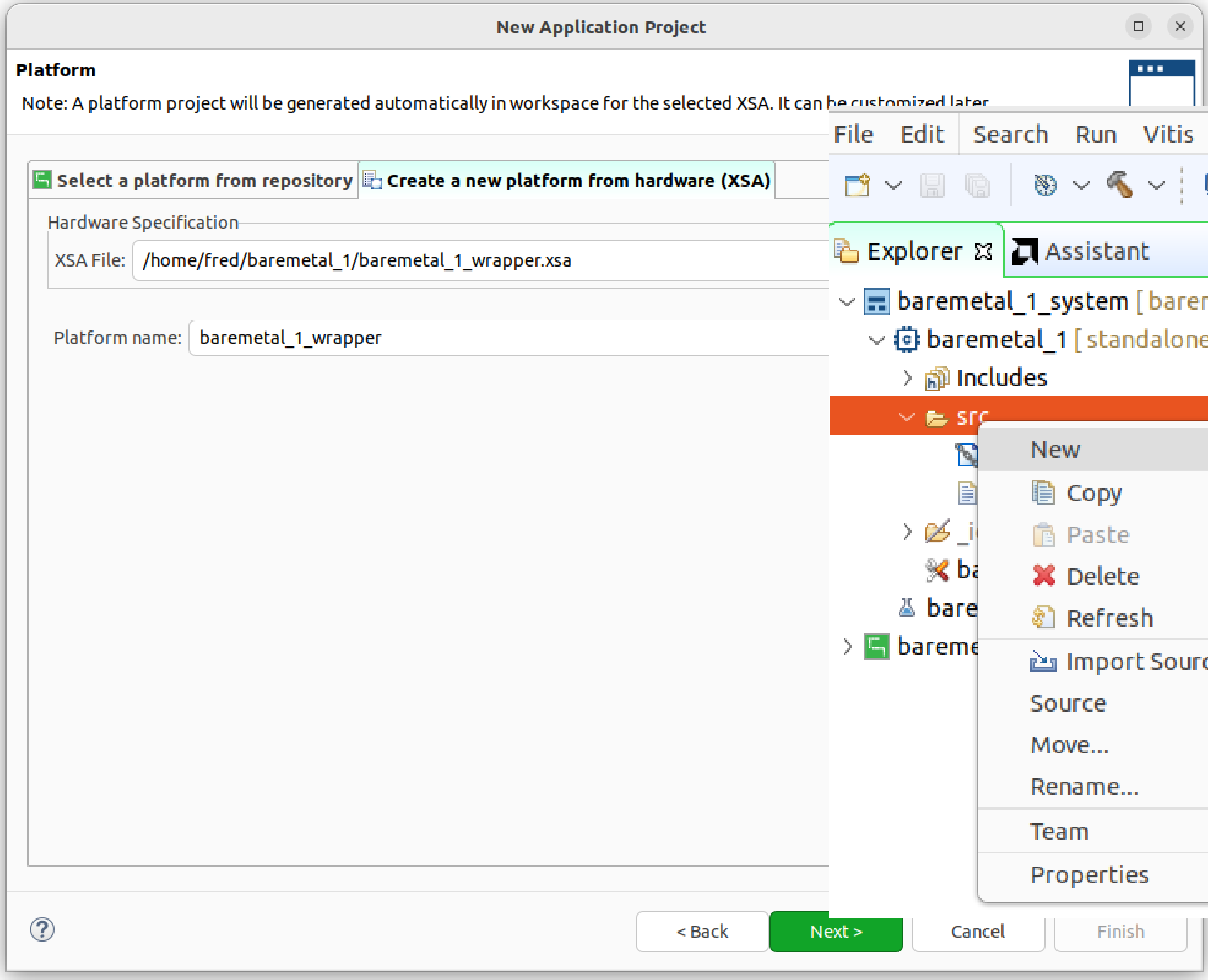
```
/home/fred/ba
## Cloc
set_pro
create_

## Swi
#set_pr
#set_pr
#set_pr
#set_pr
#set_pr
#set_pr
#set_pr
#set_pr
#set_pr
```



Enter… Vitis


File
Edit
Flow
Tools
Reports
Window
Layout
View
Help
Quick Access
Flow Navigator
PROJECT MANAGER
Settings
Add Sources
Language Temp
IP Catalog
IP INTEGRATOR
Create Block De
Open Block Desi
Generate Block
SIMULATION
Run Simulation
Validate Design F6
Create and Package New IP...
Create Interface Definition...
Enable Dynamic Function eXchange...
Run Tcl Script...
Property Editor Ctrl+J
Associate ELF Files...
Generate Memory Configuration File...
Compile Simulation Libraries...
Vivado Store...
Custom Commands
Launch Vitis IDE
Language Templates
Settings...
Board
er (baremetal_1_wrapper.v) (1)
Hierarchy
IP Sources
Libraries
Compile Order
Source File Properties

Create main.c



New Application Project

Platform

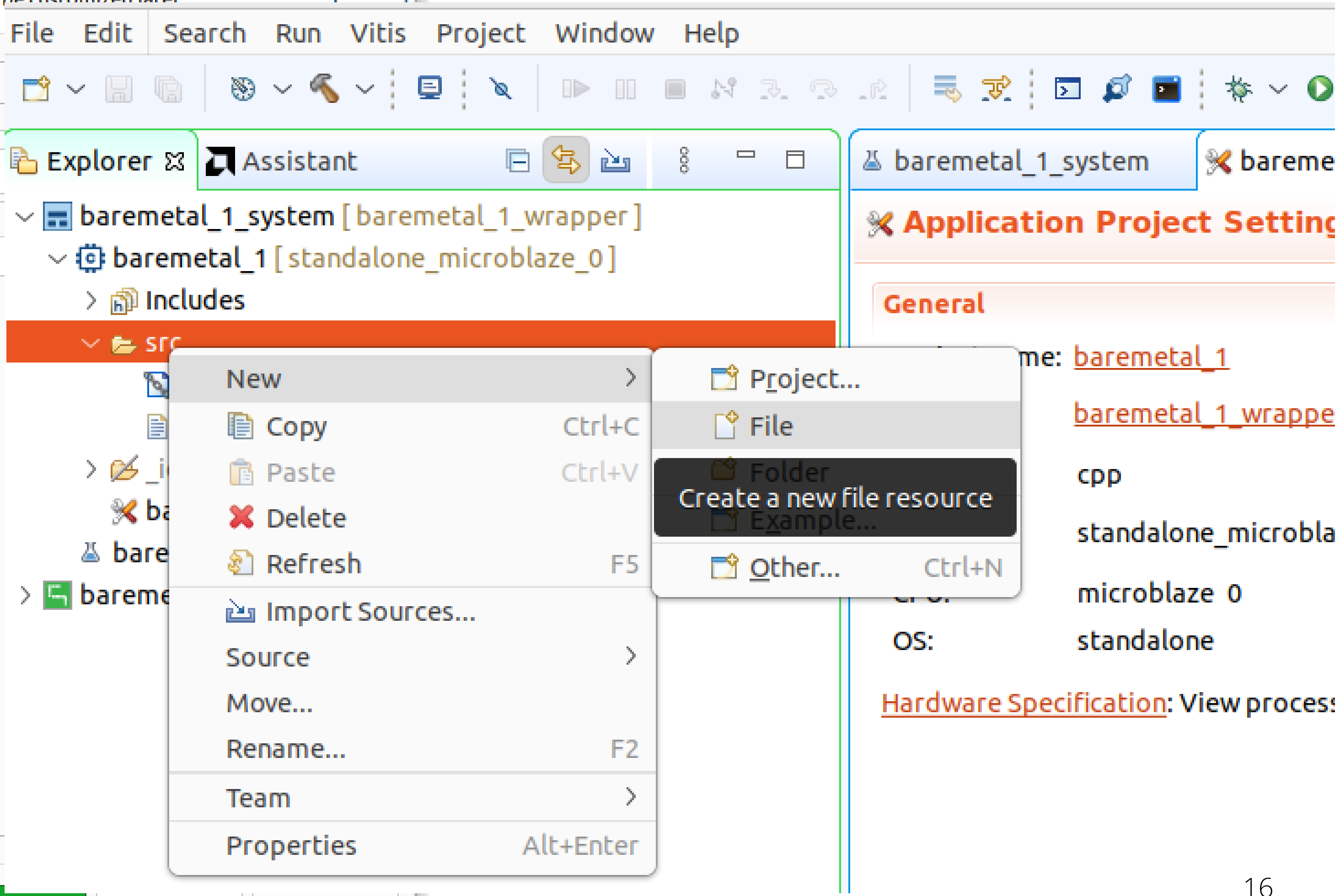
Note: A platform project will be generated automatically in workspace for the selected XSA. It can be customized later.

Select a platform from repository | Create a new platform from hardware (XSA)

Hardware Specification

XSA File: /home/fred/baremetal_1/baremetal_1_wrapper.xsa

Platform name: baremetal_1_wrapper



baremetal_1_system

Application Project Settings

General

Name: baremetal_1

baremetal_1_wrappe

cpp

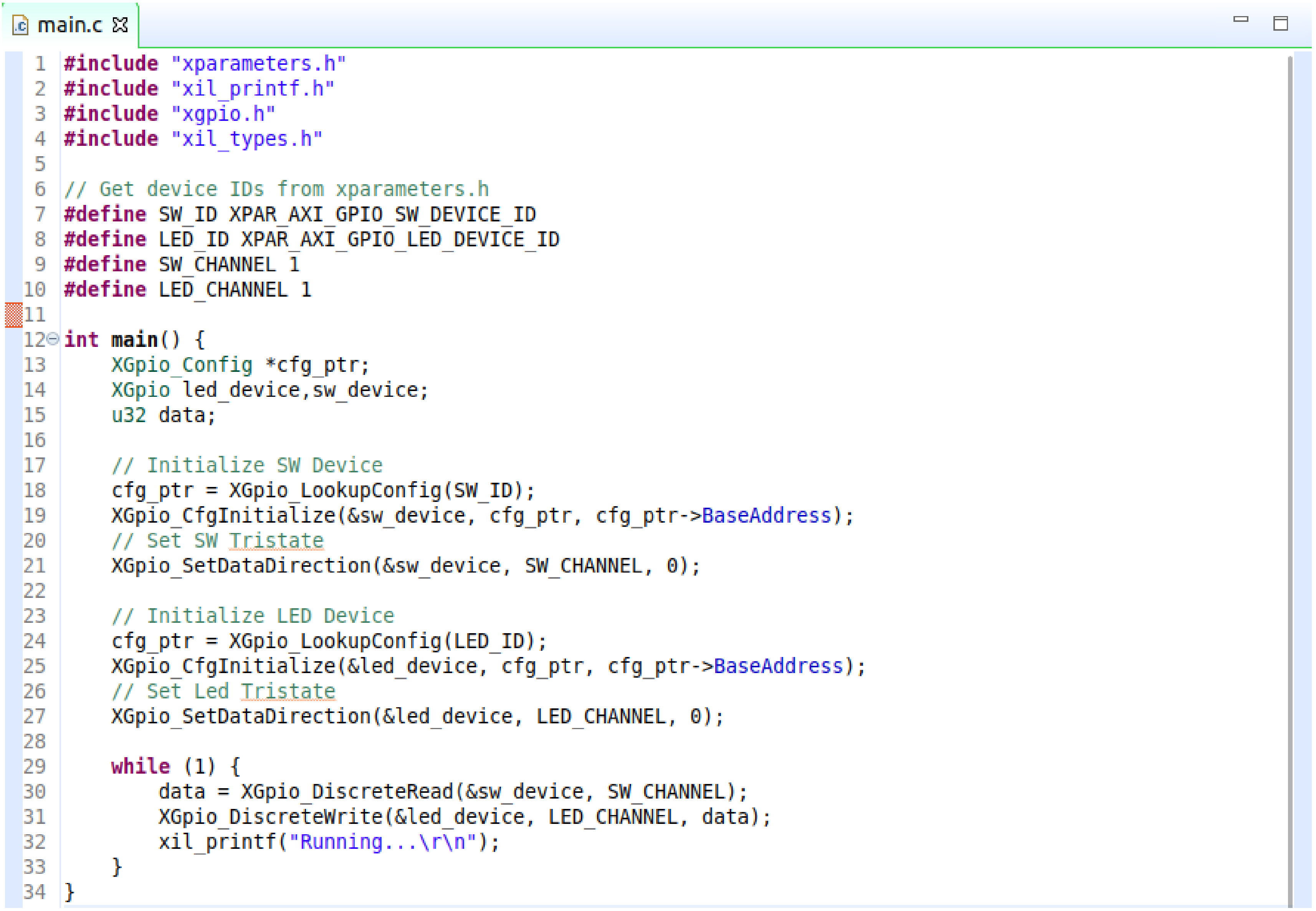
standalone_microbla

CPU: microblaze_0

OS: standalone

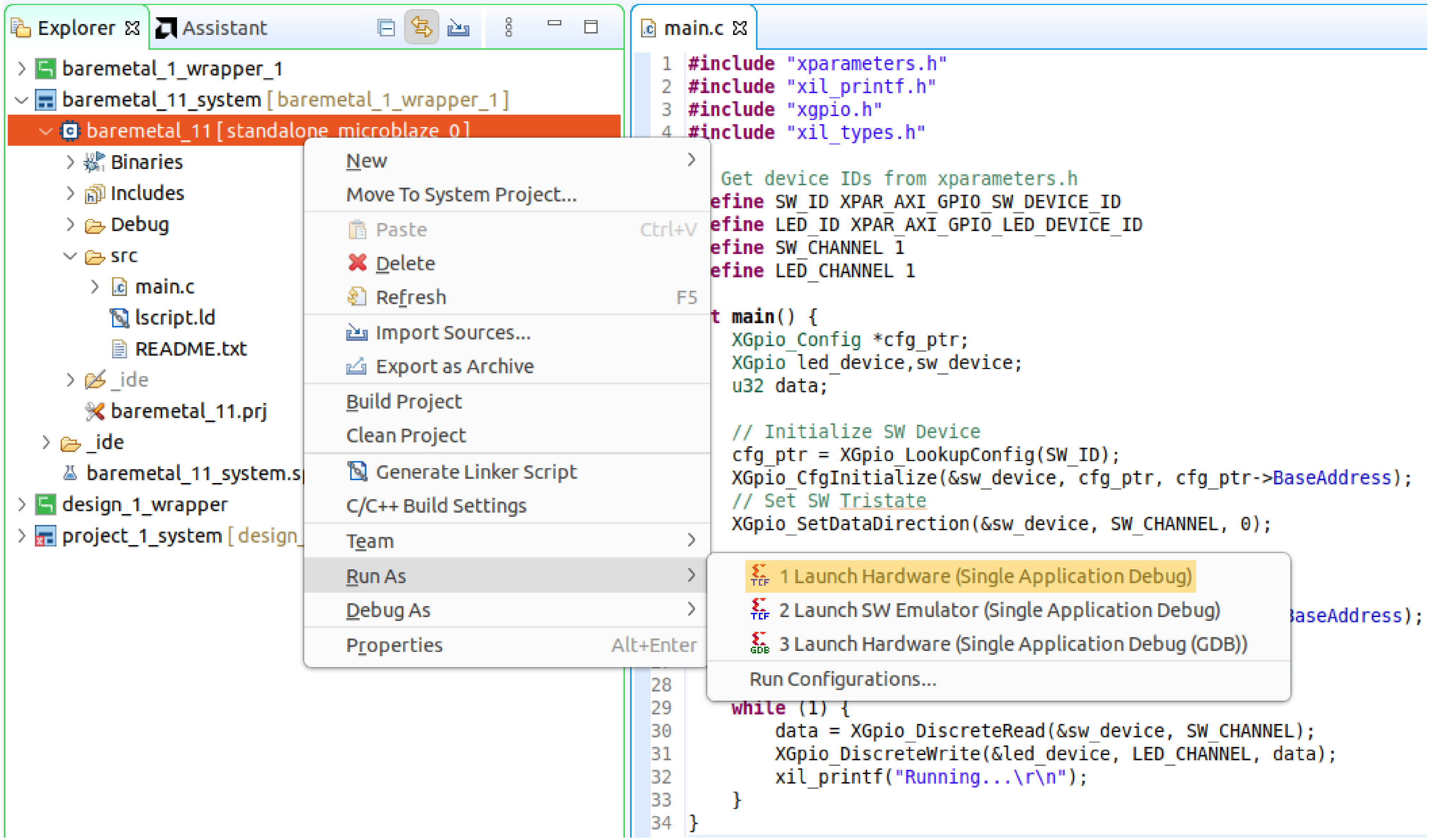
Hardware Specification: View process

Vitis Application Code


main.c

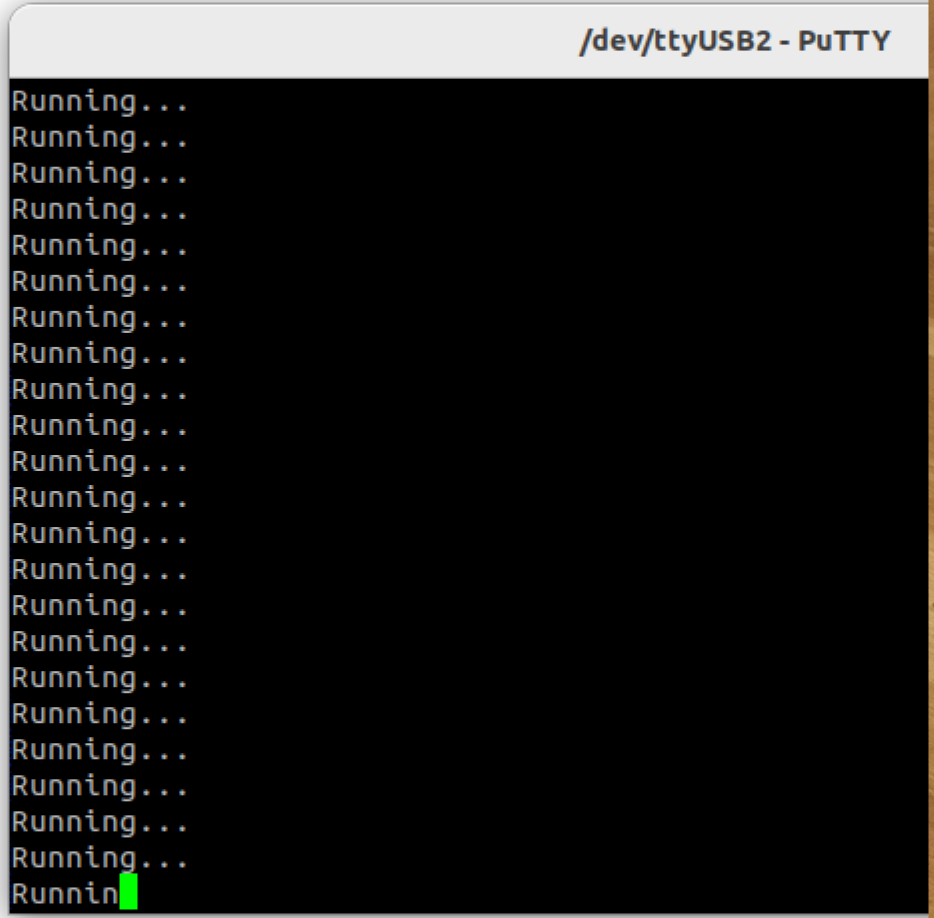
```c
#include "xparameters.h"
#include "xil_printf.h"
#include "xgpio.h"
#include "xil_types.h"

// Get device IDs from xparameters.h
#define SW_ID XPAR_AXI_GPIO_SW_DEVICE_ID
#define LED_ID XPAR_AXI_GPIO_LED_DEVICE_ID
#define SW_CHANNEL 1
#define LED_CHANNEL 1

int main() {
    XGpio_Config *cfg_ptr;
    XGpio led_device,sw_device;
    u32 data;

    // Initialize SW Device
    cfg_ptr = XGpio_LookupConfig(SW_ID);
    XGpio_CfgInitialize(&sw_device, cfg_ptr, cfg_ptr->BaseAddress);
    // Set SW Tristate
    XGpio_SetDataDirection(&sw_device, SW_CHANNEL, 0);

    // Initialize LED Device
    cfg_ptr = XGpio_LookupConfig(LED_ID);
    XGpio_CfgInitialize(&led_device, cfg_ptr, cfg_ptr->BaseAddress);
    // Set Led Tristate
    XGpio_SetDataDirection(&led_device, LED_CHANNEL, 0);

    while (1) {
        data = XGpio_DiscreteRead(&sw_device, SW_CHANNEL);
        XGpio_DiscreteWrite(&led_device, LED_CHANNEL, data);
        xil_printf("Running...\r\n");
    }
}
```

Run it

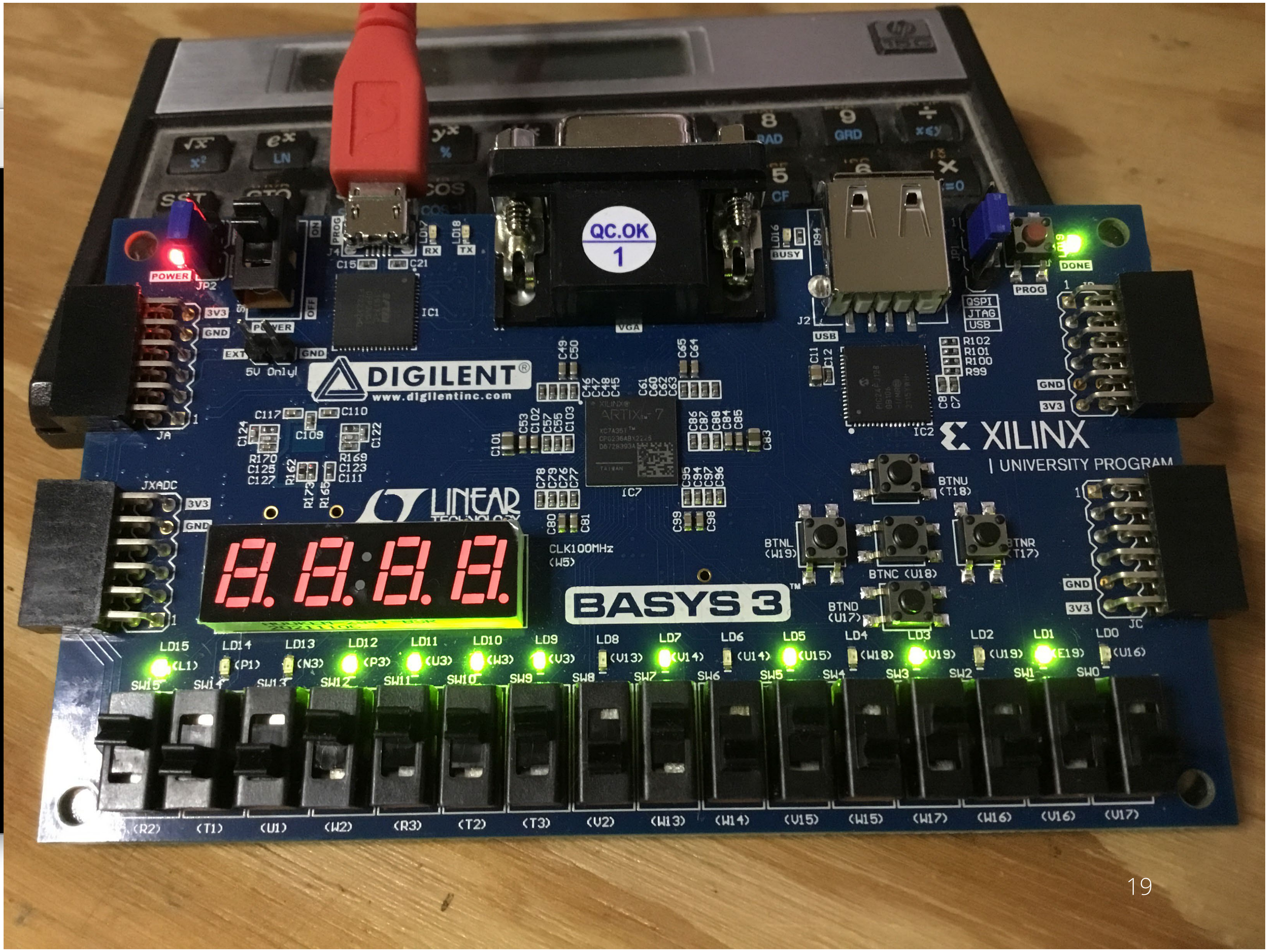


```c
#include "xparameters.h"
#include "xil_printf.h"
#include "xgpio.h"
#include "xil_types.h"

 Get device IDs from xparameters.h
efine SW_ID XPAR_AXI_GPIO_SW_DEVICE_ID
efine LED_ID XPAR_AXI_GPIO_LED_DEVICE_ID
efine SW_CHANNEL 1
efine LED_CHANNEL 1

t main() {
    XGpio_Config *cfg_ptr;
    XGpio led_device,sw_device;
    u32 data;

    // Initialize SW Device
    cfg_ptr = XGpio_LookupConfig(SW_ID);
    XGpio_CfgInitialize(&sw_device, cfg_ptr, cfg_ptr->BaseAddress);
    // Set SW Tristate
    XGpio_SetDataDirection(&sw_device, SW_CHANNEL, 0);

                                                                   BaseAddress);

    while (1) {
        data = XGpio_DiscreteRead(&sw_device, SW_CHANNEL);
        XGpio_DiscreteWrite(&led_device, LED_CHANNEL, data);
        xil_printf("Running...\r\n");
    }
}
```

Running...


/dev/ttyUSB2 - PuTTY

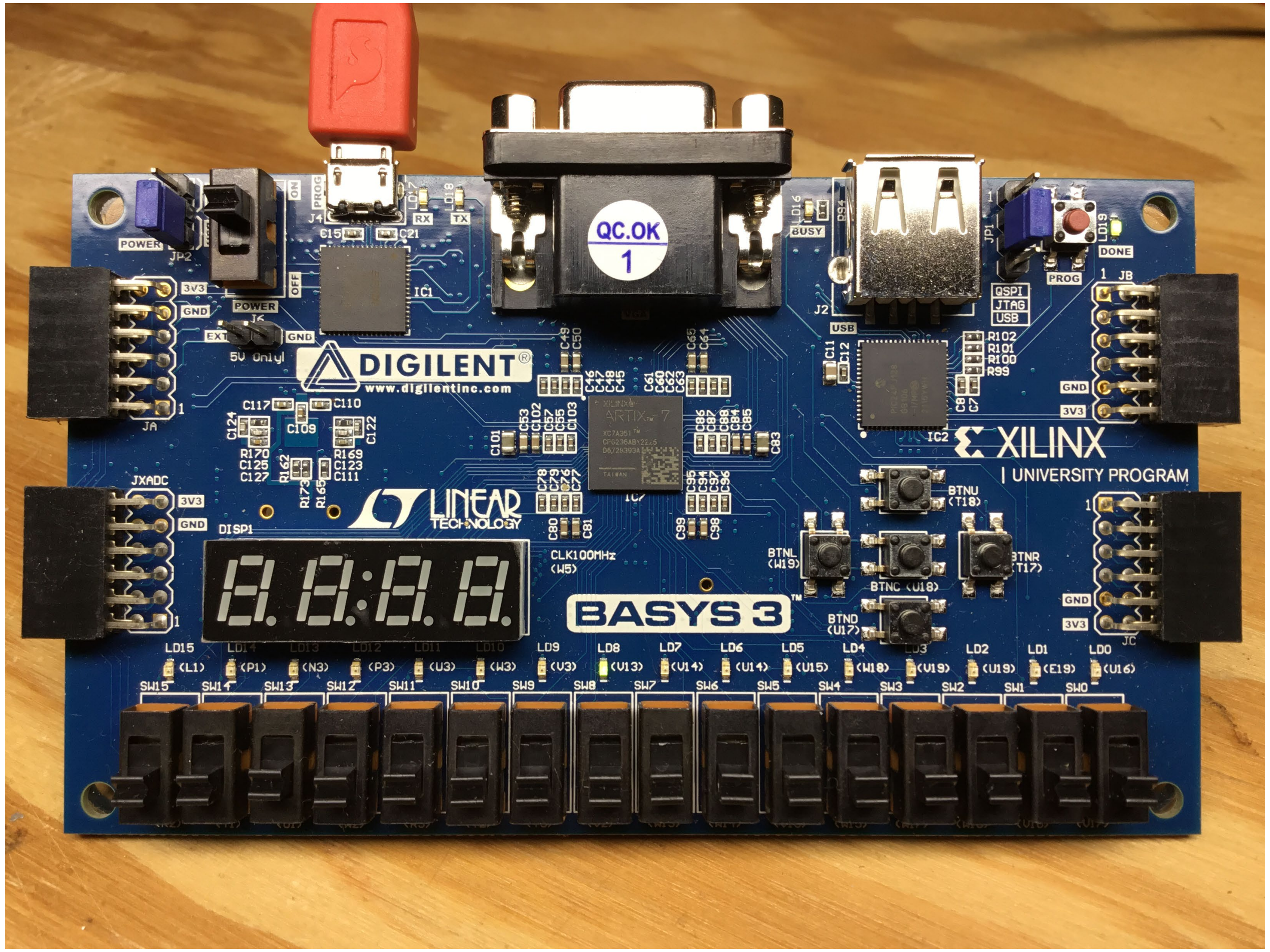
```
Running...
Running...
Running...
Running...
Running...
Running...
Running...
Running...
Running...
Running...
Running...
Running...
Running...
Running...
Running...
Running...
Running...
Running...
Running...
Running...
Running...
Running...
Runnin
```



Thank you for attending!!!

Please consider the resources below:

- xilinx.com
- digilent.com
- Basys 3 Reference Manual



CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

Digi-Key Electronics

informa markets