GRI postpartum

UNITED STATES AIR FORCE ACADEMY

ECE 383 - Embedded Computer Systems II

Lecture 17 - Soft CPU - "MicroBlaze"

UNITED STATES AIR FORCE ACADEMY

Lesson Outline

- [redacted]
- HW# 9 BOC Next Lesson!
- Soft CPU – MicroBlaze
  - MicroBlaze Intro
  - MicroBlaze Tutorial
  - Adding LEDs GPIO

- While I talk, go ahead and do step 1 of
  - MicroBlaze_Install_short_version.pdf

- Warning: short path name!
- Got SDK installed?

UNITED STATES AIR FORCE ACADEMY

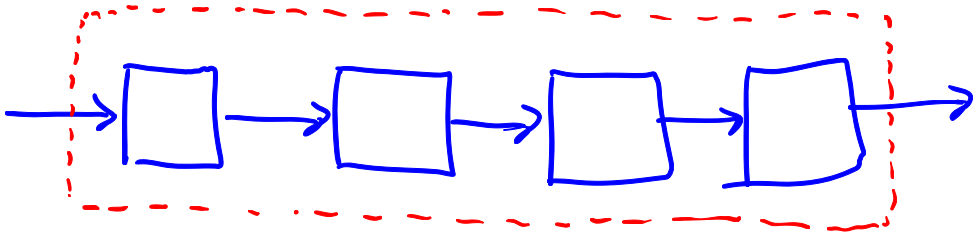
Lesson #s off one lesson (due to extra lab2 day)

L17	Lab2 - Data acquisition, storage and display		Lab2 Functionality	COB L17
L18	Soft CPU		Lab2 Write-up HW #9	COB L18 BOC L19
L19	Soft CPU		HW #10	BOC L20
L20	Soft CPU		HW #11	BOC L21
L21	Lab3 - O'scope control	Final Project Ideas		
L22	Lab3 - O'scope control		Gate Check 1	End of L22
L23	Lab3 - O'scope control		Gate Check 2	End of L23
L24	Lab3 - O'scope control		~~Lab3 Functionality~~ GC3	COB L24
L25	Lab3 - O'scope control		Lab 3 Functionality	COB L25
L26	Direct Digital Synthesis	Final Project Ideas	Lab3 Write-up Final Project Proposal Section 2	COB L26 BOC L27

UNITED STATES AIR FORCE ACADEMY

Debugging?

- Understand the lab
- Babysteps vs Homerun
- Divide-n-Conquer
- Methods
  1. Code Walk-through
  2. Testbench
     - Small “play space”
     - Large lab2



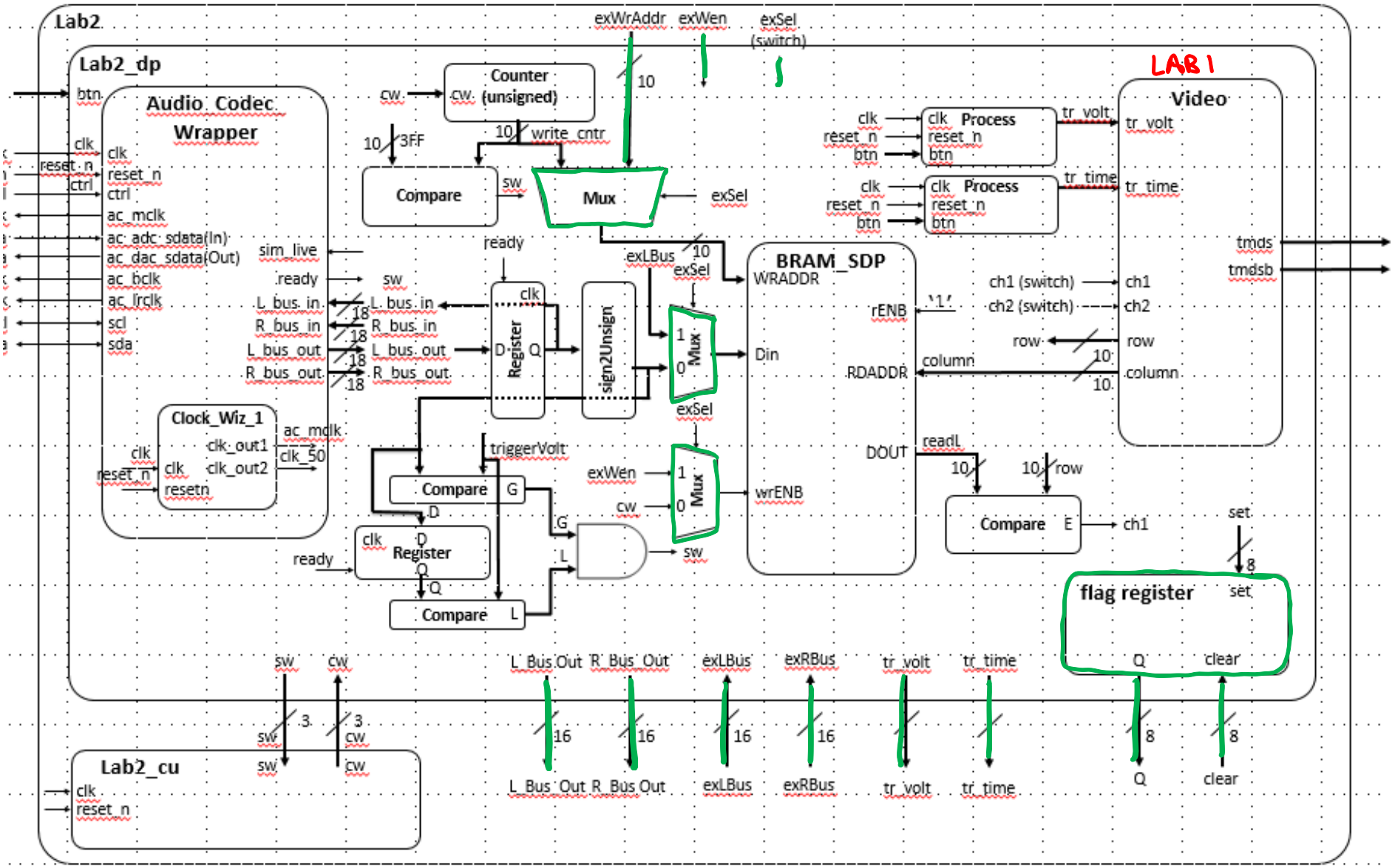
count is 10-bits < 1023 (0 11 1111 1111 / 00 0000 0000)

sw <= '0' when count < 1024 else '1'

sw is Never '1' ... why?

Lab2 → Lab3

exSel and Flag Register?



MicroBlaze Intro

Pay attention:

- **Install microblaze 3 times over the next 3 lessons**
  - **(save vs redo?)**
- **HW 9, 10, and 11 crucial to lab3**

UNITED STATES AIR FORCE ACADEMY

Not a "Toy" Processor

picoblaze?

Microblaze Intro

- The goal of today's class is to bring you up to speed on how to instantiate a microBlaze processor on our Artix 7, integrate a custom piece of VHDL code to the processor, and then to write some C code to run on the microBlaze to control the custom VHDL module. Here are some specifications on the microBlaze processor:
- It has thirty-two 32-bit general purpose registers.
- It uses a 32-bit instruction word with three operands, and has two addressing modes.
- It has a 32-bit address bus.
- It uses a single issue (3 or 5)-stage pipeline.

Microblaze Intro

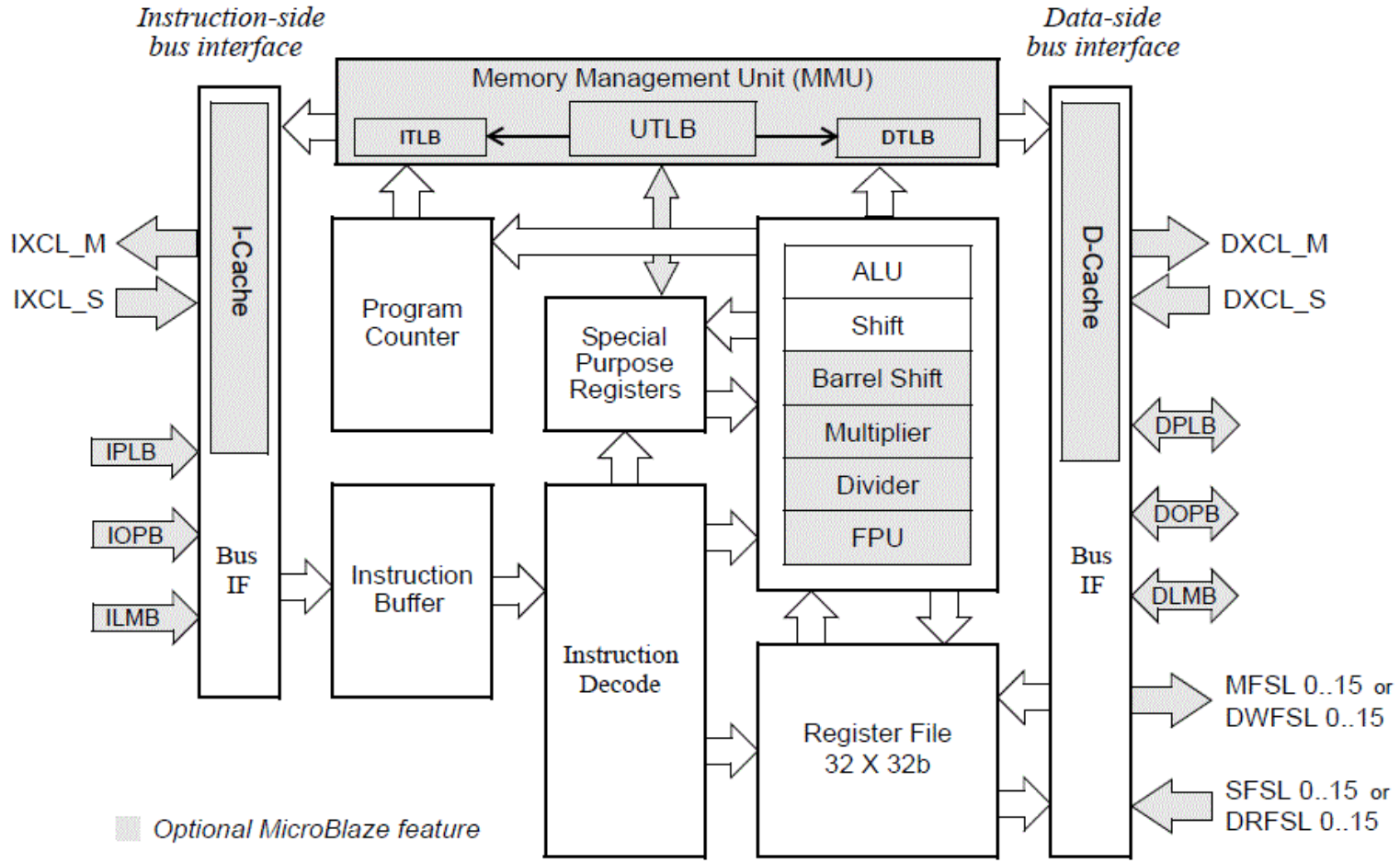


Figure 1-1: **MicroBlaze Core Block Diagram**

MicroBlaze Tutorial

History:

- MicroBlaze Tutorial **<- Digilent Website**
- **Nexys_Video_MicroBlaze_Tutorial.pdf** ← 46 pages
- **MicroBlaze_Install_short_version.pdf** ← 3 pages

Lecture:	17
Homework	HW #9
Status	Complete
MicroBlaze Tutorial on Digilent Website	MicroBlaze Tutorial
MicroBlaze Tutorial with ECE 383 Deviations	MicroBlaze Tutorial MicroBlaze Tutorial Short Version
Supplemental Lesson Slides	ECE_383_Lec17.pptx
Code	Lec17.c (Initial Hello World) Lec17_v2.c (Example code to interface with GPIO LEDs

UNITED STATES AIR FORCE ACADEMY

Microblaze - Tutorial Overview

- In this tutorial, you will be introduced to the tool flow for simple MicroBlaze designs. Specifically, you will create a design that continuously reads the input from UART and writes that value to the LEDs. The UART will be connected from the FPGA to your computer via a micro USB cable.
- You will follow the tutorial here step by step.

UNITED STATES AIR FORCE ACADEMY

Microblaze - Tutorial Deviations

- **Deviations from the Tutorial**
  - Call your project "L17" or "Lesson17".
  - Keep in mind that you can zoom in and out on your block diagram.
  - The AXI bus is the bus the Microblaze uses, similar to a PCI bus in normal PC's.
  - What does the UART actually do? Although you learned about it in ECE382, you can read more about UART's for a refresher.
  - UART Controller?
  - Ignore step 6.3 completely. When you do step 6.2, just check the "reset" box in the automatic connections dialogue under clock wizard. You do not need to make the connection manually.
  - The Memory Interface Generator (MIG) is used essentially add BRAM (it is SDRAM in this case).
  - Although you didn't need to make the first connection manually, you do need to make the 2nd connection manually (with the RAM).

UNITED STATES AIR FORCE ACADEMY

Microblaze - Tutorial

- https://reference.digilentinc.com/learn/programmable-logic/tutorials/nexys-video-getting-started-with-microblaze/start
- https://reference.digilentinc.com/learn/programmable-logic/tutorials/pmod-ips/start
- https://reference.digilentinc.com/nexys/nexysvideo/gsmb?s[]=ip&s[]=integrator
- https://reference.digilentinc.com/learn/programmable-logic/tutorials/zedboard-creating-custom-ip-cores/start

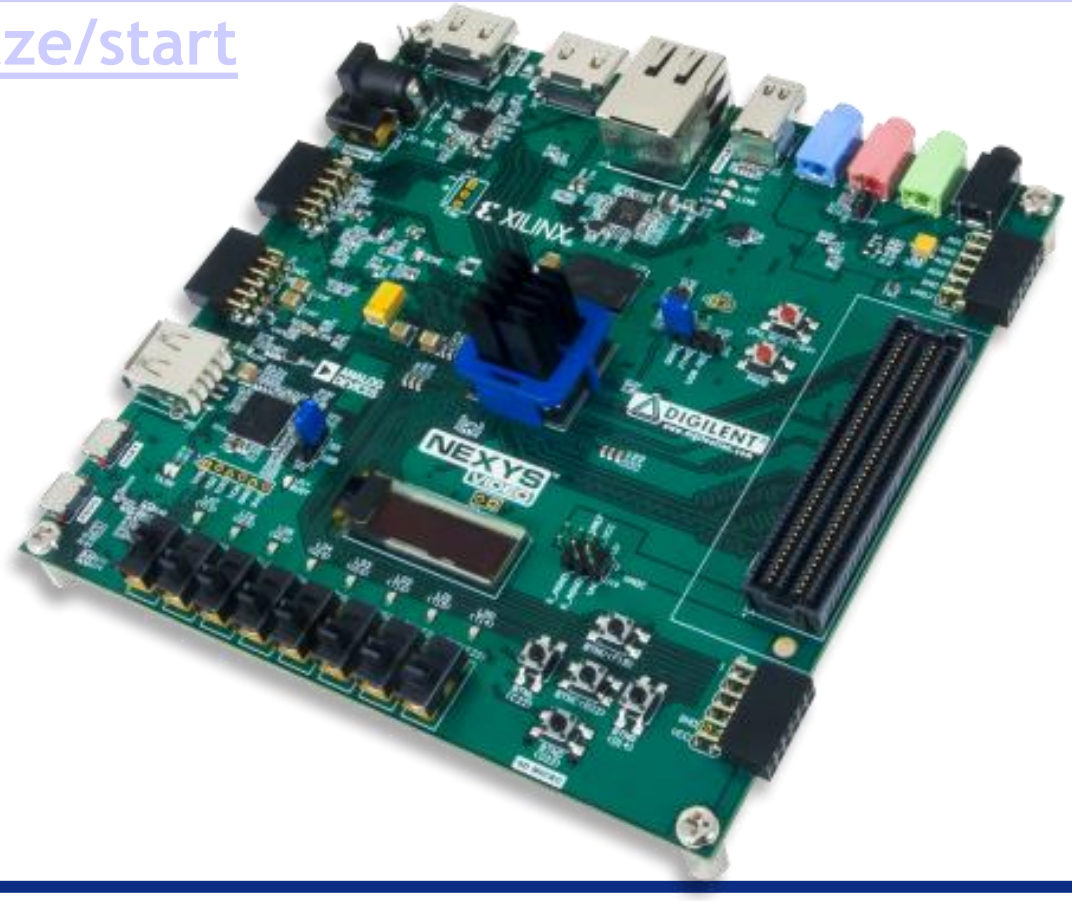
UNITED STATES AIR FORCE ACADEMY

Microblaze - Getting Started

- What you need before proceeding with this guide
- Software
  - Xilinx Vivado with the SDK package.
    - Follow this Wiki guide (Installing Vivado) on how to install and activate Vivado 2018.2
- Board Support Files
  - Board Support Files. These files will describe GPIO interfaces on your board and make it easier to select your FPGA board and add GPIO IP blocks.
    - Follow this Wiki guide (Vivado Board Files for Digilent 7-Series FPGA Boards) on how to install Board Support Files for Vivado 2018.2
- Hardware
  - Digilent Nexys Video FPGA Board and Micro USB Cable for UART communication and JTAG programming

Microblaze

- https://reference.digilentinc.com/learn/programmable-logic/tutorials/nexys-video-getting-started-with-microblaze/start



UNITED STATES AIR FORCE ACADEMY

Microblaze - Introduction

- Microblaze is a soft IP core from Xilinx that will implement a microprocessor entirely within the Xilinx FPGA general purpose memory and logic fabric. For this tutorial, we are going to add a Microblaze IP block using the Vivado IP Integrator tool.
- In addition to the Microblaze IP block, we would also like to make use of the DDR3 SDRAM component on the Nexys Video. Therefore a MIG (Memory Interface Generator) IP block will be added to our design.
- Finally, a UART (Universal Asynchronous Receiver/Transmitter) IP block will be added to communicate between the host PC and the soft processor core running on the Nexys Video.

Microblaze

- General Design Flow - Vivado
  - Open Vivado and select Nexys Video board
  - Create an new Vivado Project
  - Create empty block design workspace inside the new project
  - Add required IP blocks using the IP integrator tool and build Hardware Design
  - Validate and save block design
  - Create HDL system wrapper
  - Run design Synthesis and Implementation
  - Generate Bit File
  - Export Hardware Design including the generated bit stream file to SDK tool
  - Launch SDK → then do C programming!

HW

SW

Microblaze

- Now the Hardware design is exported to the SDK tool. The Vivado to SDK hand-off is done internally through Vivado. We will use SDK to create a Software application that will use the customized board interface data and FPGA hardware configuration by importing the hardware design information from Vivado.
- General Design Flow - SDK
  - Create new application project and select default Hello World template
  - Program FPGA
  - Run configuration by selecting the correct UART COM Port and Baud Rate 9600

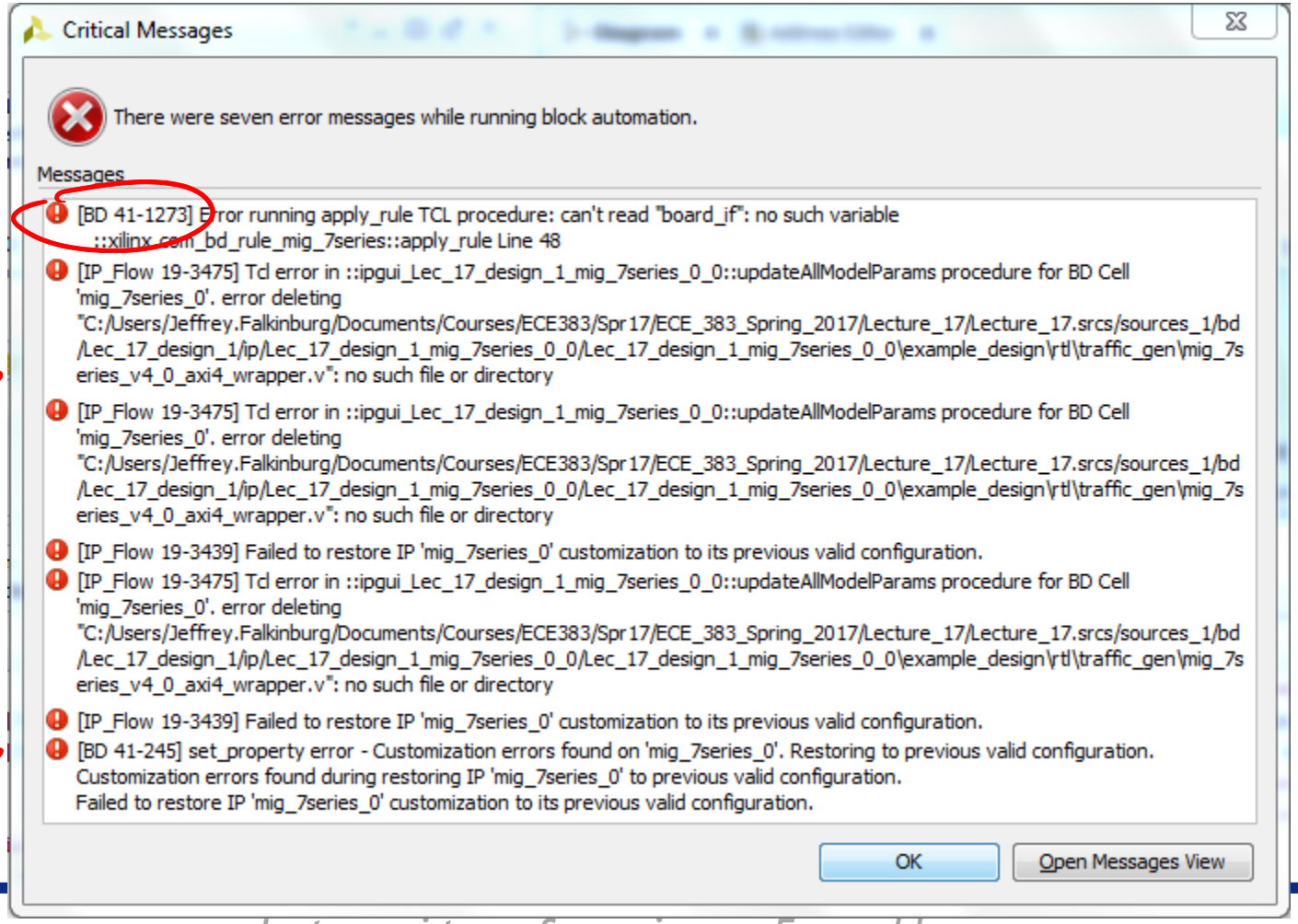
UNITED STATES AIR FORCE ACADEMY

Microblaze - Issues

will for HW 9-n-10. Not HW 11 and Lab3

- Ensure at this time don't use the MicroBlaze Interrupt Controller on the MicroBlaze Block Automation.
- Trouble with too many errors with MIG Block Automation. Should only have the error message [BD 41-1273] ← the "good" error
  - Design won't validate or build with more error messages.
- Block Design name may need to start with "design"

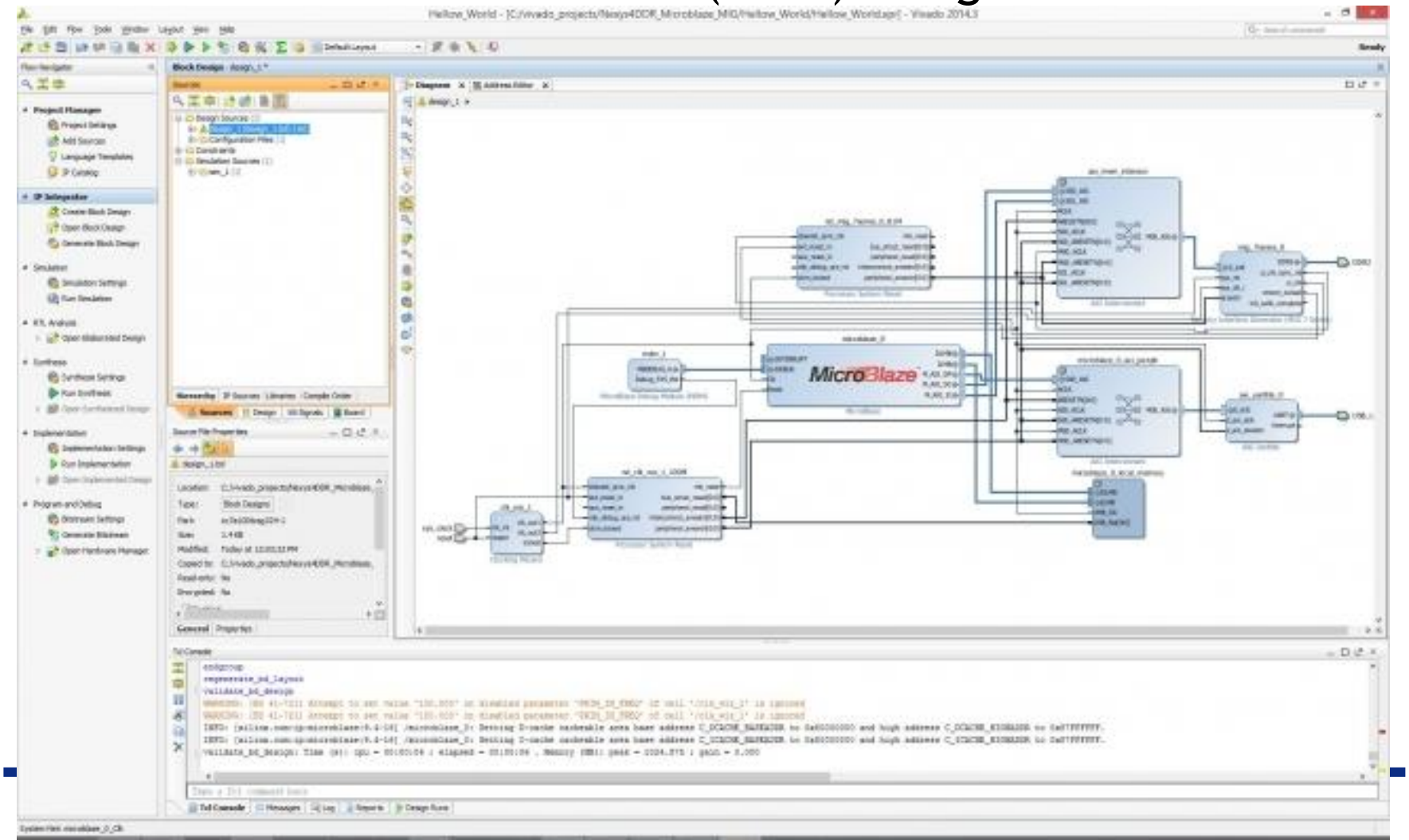
- Errors after MIG Block Automation Run



UNITED STATES AIR FORCE ACADEMY

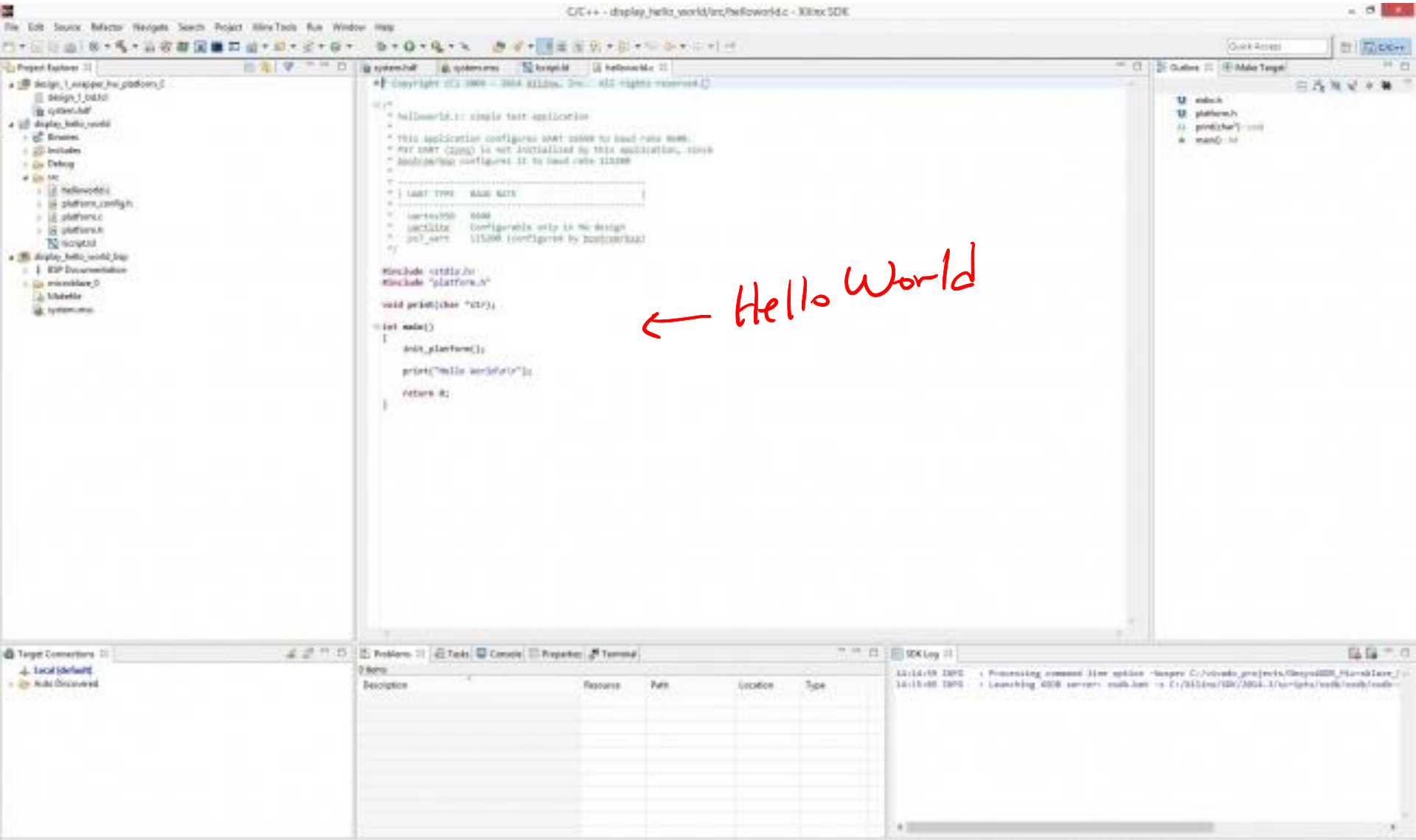
Microblaze - Tutorial

- Microblaze based hardware (HW) design in Xilinx Vivado



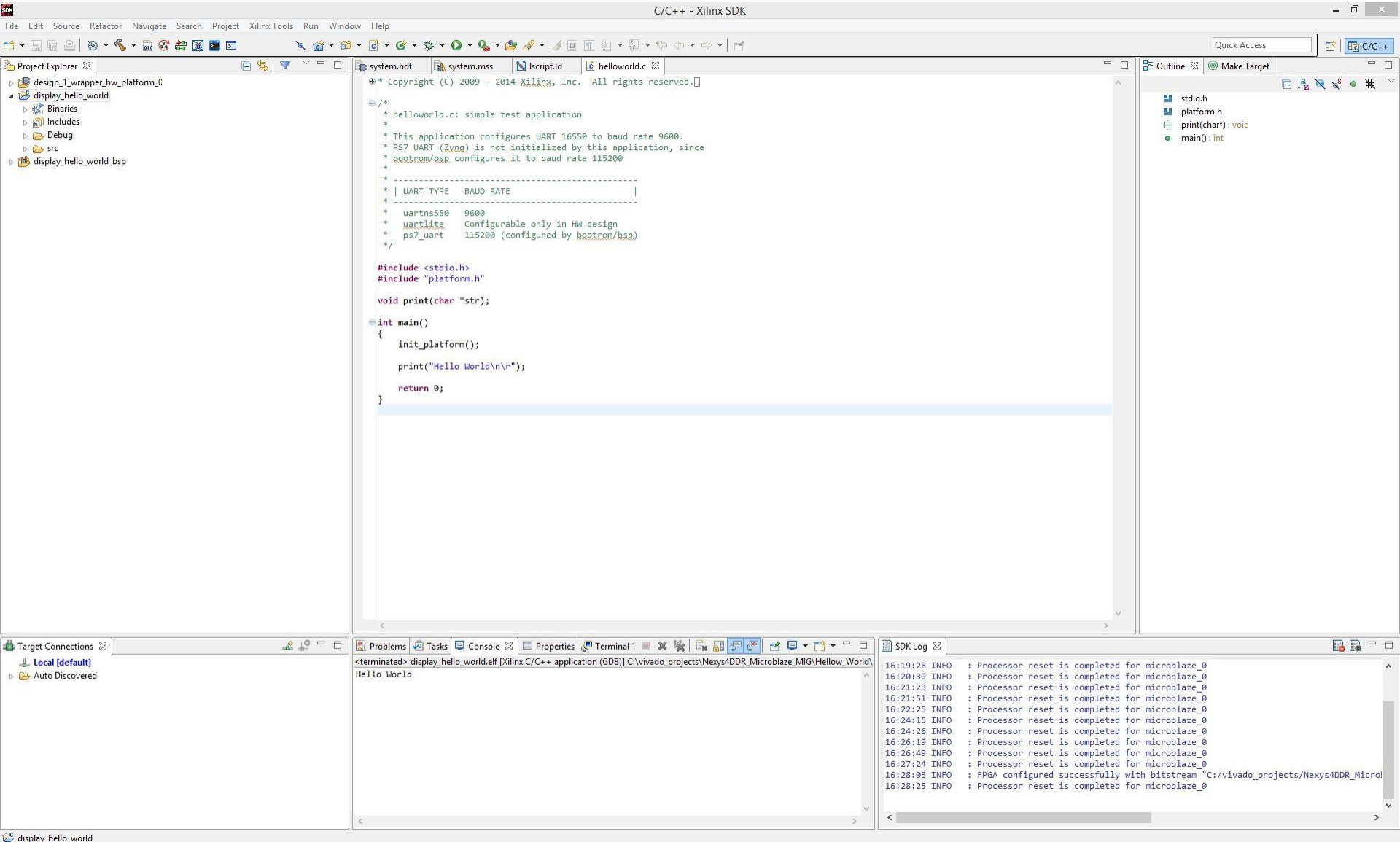
UNITED STATES AIR FORCE ACADEMY

Microblaze - Xilinx Vivado SDK



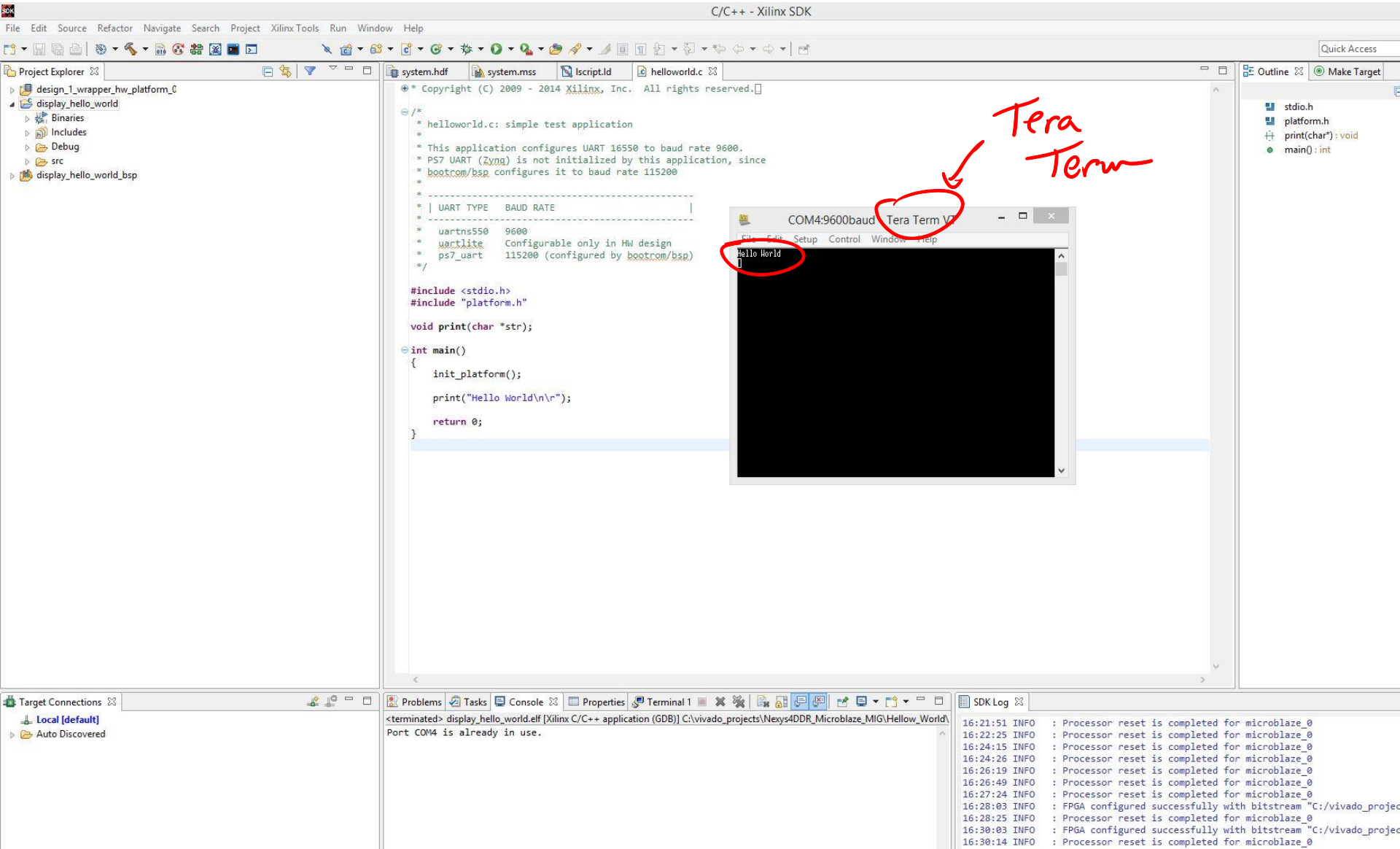
UNITED STATES AIR FORCE ACADEMY

Microblaze - Xilinx Vivado SDK



UNITED STATES AIR FORCE ACADEMY

Microblaze - Xilinx Vivado SDK



UNITED STATES AIR FORCE ACADEMY

Microblaze – Issues with SDK

- ERROR: Specified device 'Digilent Nexys Video 210276723218B/1-xc7a200t' is not found on the board
  - This essentially means that the .bit file you created was for a different board and the Artix 7 is rejecting it.
  - Solution: Regenerate .bit file in Vivado and re-export to SDK

Adding LEDs GPIO

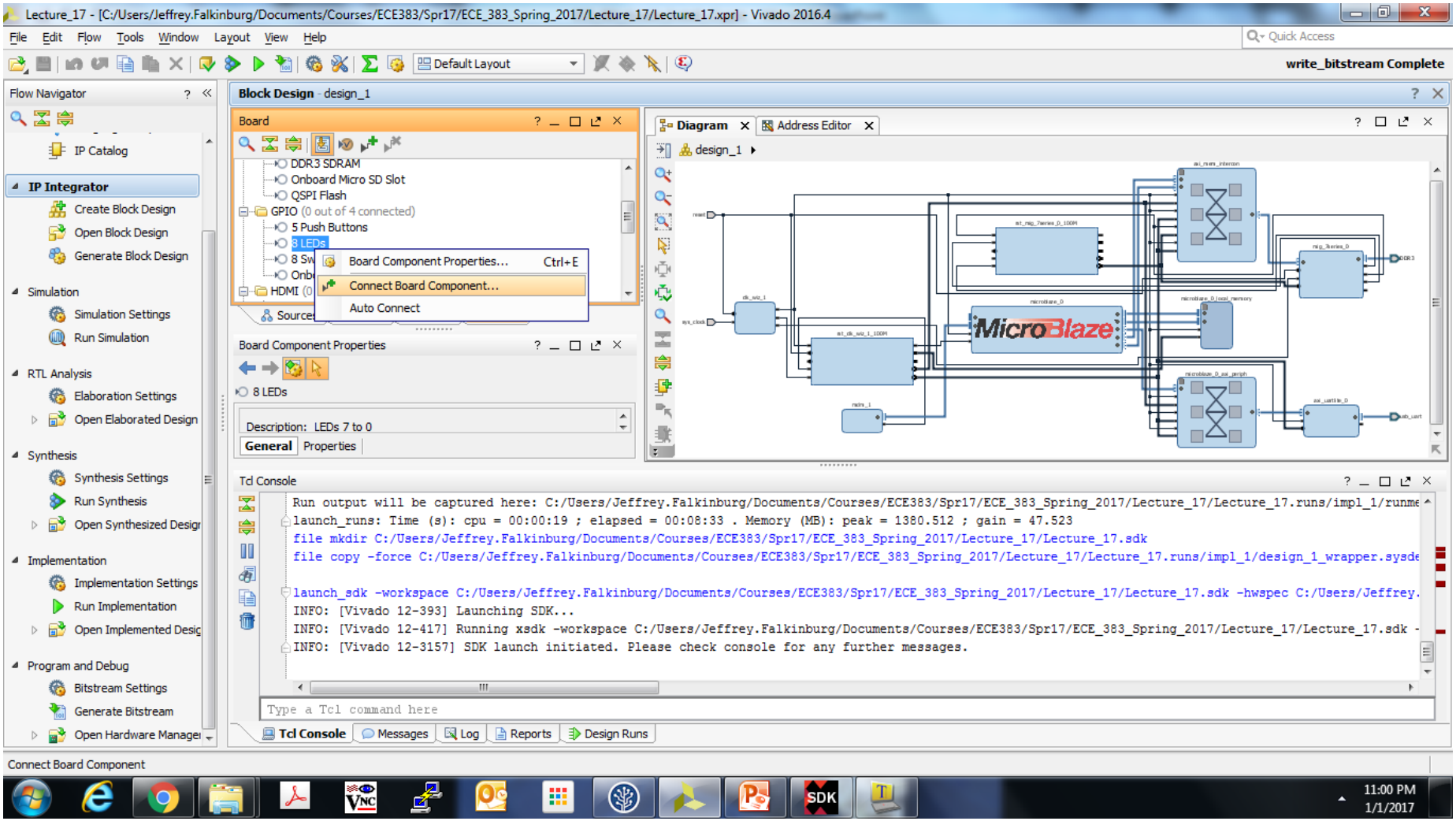
(was part of HW09)

skip for this year

What is HW09?

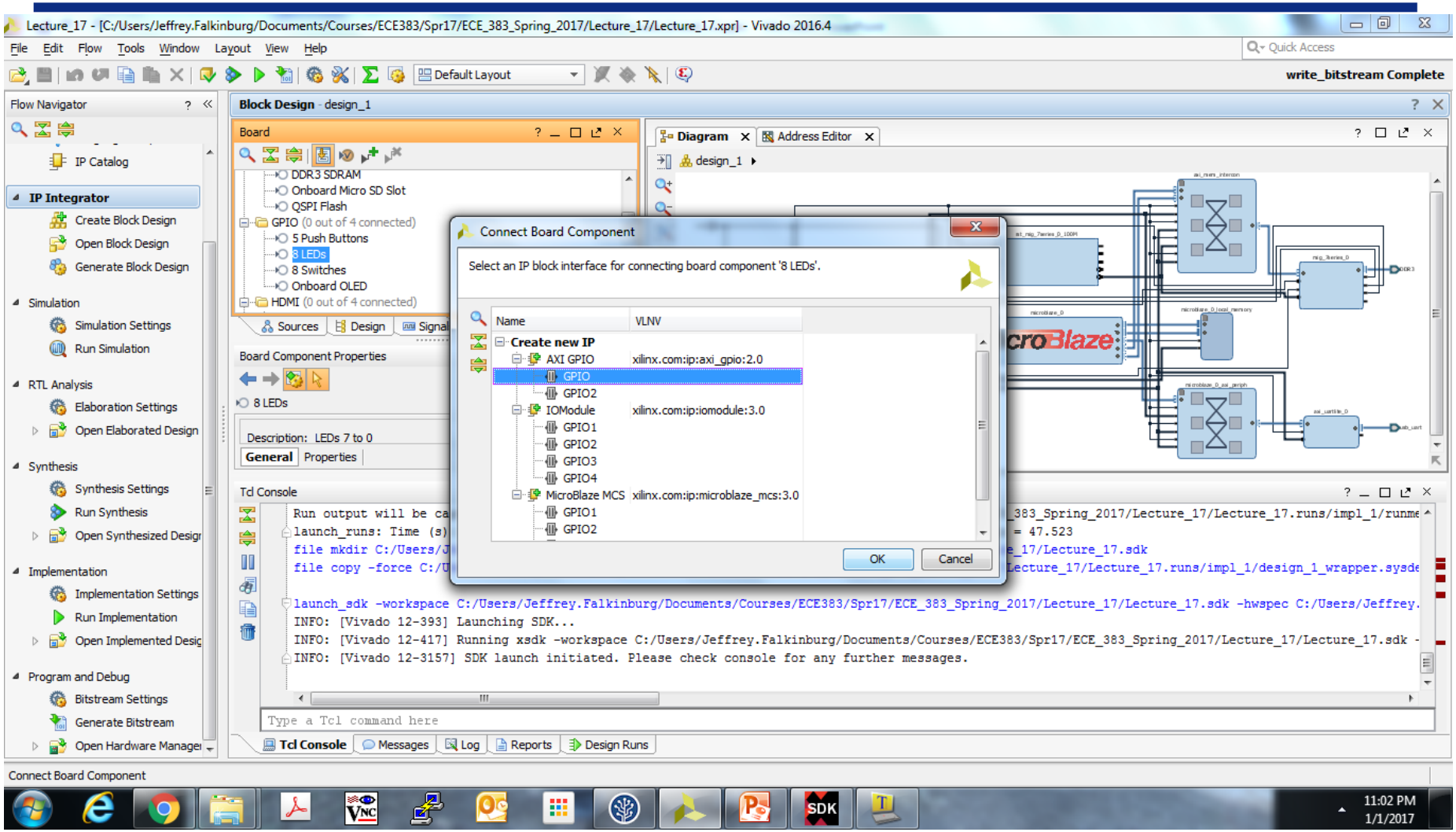
UNITED STATES AIR FORCE ACADEMY

Microblaze - Adding LEDs GPIO



UNITED STATES AIR FORCE ACADEMY

Microblaze - Adding LEDs GPIO



UNITED STATES AIR FORCE ACADEMY

Microblaze - Adding LEDs GPIO

- Run Connection Automation
-