Revision	Date	Author	Comments
1C	2023-04-23	Tim S. timothystotts08@gmail.com	Partial draft of the HYGRO Sensor Readings Tester Version 3.
2C	2023-04-29	Tim S. timothystotts08@gmail.com	Partial draft of the HYGRO Sensor Readings Tester Version 3.
C	2023-04-30	Tim S. timothystotts08@gmail.com	First publishable draft of the HYGRO Sensor Readings Tester Version 3.
D	2023-05-04	Tim S. timothystotts08@gmail.com	Added Appendix A to describe TCL commands to generate project and Block Design per each design folder.

https://github.com/timothystotts/fpga-iic-hygro-tester-3/

HYGRO Sensor Readings-Tester Experiment - Refreshed

HYGRO Sensor Readings-Tester Experiment: Folder Structure

Four HYGRO Sensor Readings Testing designs with equivalent function of polling a HDC1080 sensor via IIC bus; displaying temperature and relative humidity readings in multiple formats on a 16x2 character LCD and a UART Terminal; and displaying sensor reading display selection on a two-digit seven-segment display.

Project Folder	**Project Description**
HYGRO-Tester-Design-Verilog (Vivado 2021.2)	A utility designed for polling a temperature and relative humidity sensor once per second and displaying the readings in Celsius, Fahrenheit, and RH Percent. Two clock domains exist; 20. MHz and 7.37 MHz, together controlled by a MMCM; and each slower domain is controlled by a clock enable divider RTL dividing a MMCM clock by clock by clock enable pulse, or a clock divider, generating slower clocks that can be in a process' sensitivity list or drive an output clock port. This design targets either a Digilent Inc. Arty A7-100 board or a Digilent Inc. Arty A7-100 board.
HYGRO-Tester-Design-MB-A7 (Vivado 2021.2 and Vitis 2021.2)	A utility designed for polling a temperature and relative humidity sensor once per second and displaying the readings in Celsius, Fahrenheit, and RH Percent. The design is completely in Microblaze AXI subsystem with standard Xilinx IP Integrator components, vendor Xilinx IP Integrator components, a custom 7SD IP block, and Standalone C language program executing on the Microblaze soft processor. This design targets a Digilent Inc. Arty A7-100 board.

HYGRO-Tester-Design-MB-S7 (Vivado 2021.2 and Vitis 2021.2)	A utility designed for polling a temperature and relative humidity sensor once per second and displaying the readings in Celsius, Fahrenheit, and RH Percent. The design is completely in Microblaze AXI subsystem with standard Xilinx IP Integrator components, vendor Xilinx IP Integrator components, a custom 7SD IP block, and Standalone C language program executing on the Microblaze soft processor. This design targets a Digilent Inc. Arty S7-25 board.
HYGRO-Tester-Design-Zynq (Vivado 2021.2 and Vitis 2021.2)	A utility designed for polling a temperature and relative humidity sensor once per second and displaying the readings in Celsius, Fahrenheit, and RH Percent. The design is completely in Zynq-7000 AXI subsystem with standard Xilinx IP Integrator components, vendor Xilinx IP Integrator components, a custom 7SD IP block, and Standalone C language program executing on the first ARM A9 hard processor. This design targets a Digilent Inc. Zybo Z7-20 board.

To successfully open the MB-A7/MB-S7, SV, or VHDL project, it can be necessary to create an empty Xilinx Vivado project, select the Part from Boards and click Refresh, then download the Arty A7-100 and/or Arty S7-25 board. To successfully open the Zynq project, it can be necessary to do the same, instead selecting the Zybo Z7-20 board. It is no longer necessary to manually install board_files in the data/boards/board_files folder of the Vivado install, or use a TCL init script to point to a custom folder containing board files.

Note that each of the three MicroBlaze/Zynq-7000 projects mentioned has a <project>/IP/vivado-library path that is a GIT submodule. It is necessary to update this path with GIT prior to creating one of the IPI Block Design projects, to have the appropriate fork of the Digilent Inc. Vivado User IP (open source) in a path where Vivado can locate it.

For the MB-A7/S7 projects, the SPI Flash bootloader is not created as it a Xilinx program that can be generated within the Xilinx Vitis IDE. To demo one of these projects, the default is to execute the program via JTAG without external power. The USB serial Terminal should be connected at 115200 baud prior to programming the FPGA via JTAG. This is necessary to prevent an internal reset of the design if later connecting the USB serial Terminal to the USB UART.

HYGRO Sensor Readings-Tester Experiment: Methods of Operation

The purpose of the design is to boot a Digilent Inc. Arty A7-100 (Artix-7) or a Digilent Inc. Arty S7-25 (Spartan-7) development board with Pmod CLS, Pmod HYGRO, and Pmod 7SD peripheral boards, which are a 16x2 Character dot-matrix LCD display, an IIC bus temperature and relative humidity sensor, and a two-digit seven-segment display, respectively. The PMOD CLS connects to the FPGA with its own dedicated SPI bus via a higher-speed plug and jack.

- On the Arty A7-100, the Pmod 7SD connects to board Pmod port JA, the Pmod CLS connects to board Pmod port JB, the Pmod HYGRO connects to board Pmod port JC.

- On the Arty S7-25, the Pmod CLS connects to board Pmod port JA, the Pmod HYGRO connects to Pmod port JC, and the Pmod 7SD connects to Pmod port JD.
- On the Zybo Z7-20, the Pmod CLS connects to board Pmod port JB, the Pmod HYGRO connects to Pmod port JC, and the Pmod 7sD connects to Pmod port JE.
- The use of an extension cable makes the Pmod CLS able to connect to only one 2x6 Pmod port and the Pmod 7SD able to connect to only one 2x6 Pmod port. The 2x6 to 2x1x6 Pmod extension cables enable the Pmod CLS and the Pmod 7SD to connect to the development board and use only one Pmod port.
- See figures 1, 2, 3, on the next three pages.

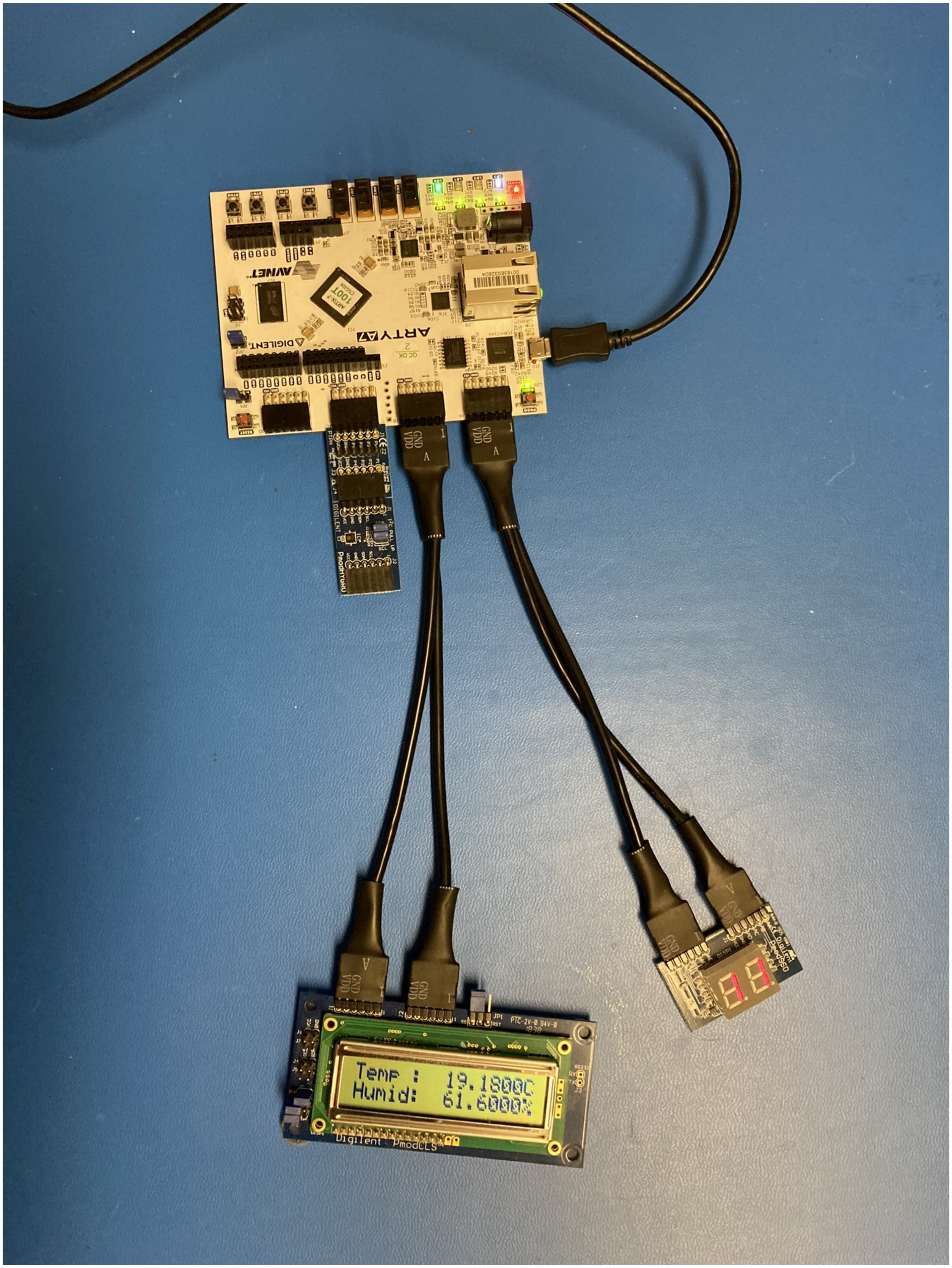


Figure 1: Arty A7-100 Executing with Pmod HYGRO, Pmod CLS, Pmod 7SD

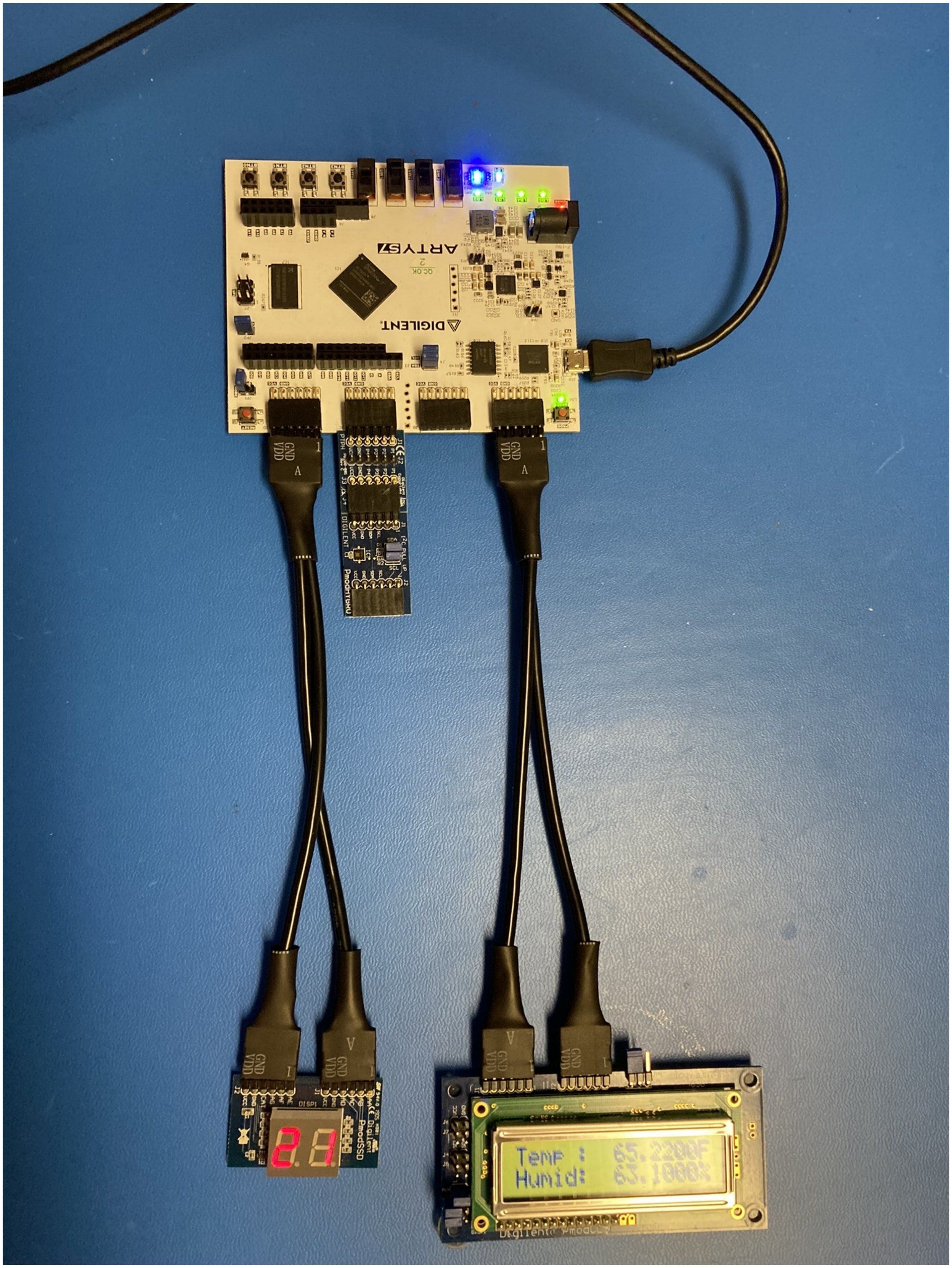
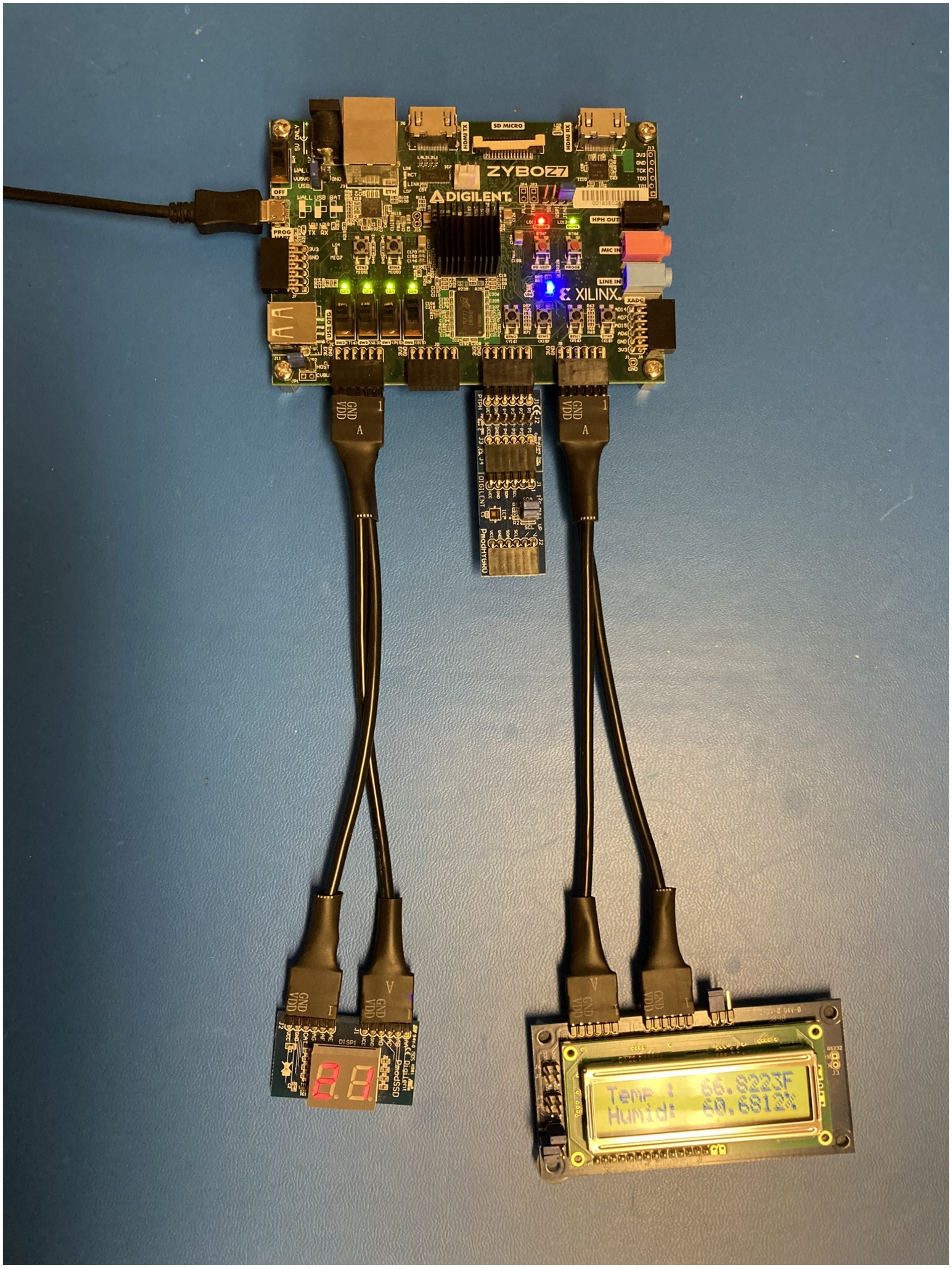


Figure 2: Arty S7-25 Executing with Pmod 7SD, Pmod HYGRO, Pmod CLS



```
Temp :  66.8223F
Humid:  60.6812%
```

Figure 3: Zybo Z7-20 Executing with Pmod 7SD, Pmod HYGRO, Pmod CLS

HYGRO Sensor Readings-Tester Experiment: Method of Operation: iterative poll of sensor readings and display on LCD

HYGRO Sensor Readings-Tester Experiment: Design Operation

In the Verilog implementation, the four switches are debounced as mutually exclusive inputs, as are the four buttons. To operate the design, the LCD displays an instruction to the operator to put Switch Zero in the ON position. After that, the LCD displays an instruction to the operator to push one of the four buttons.

While Switch Zero is in the ON position, the design polls the Pmod HYGRO HDC1080 sensor approximately once per second. Each time a button is depressed and then released while switch zero is positioned as ON, the incoming Pmod HYGRO sensor readings are converted to human readable content which in turn is displayed on the Pmod CLS LCD display. Pressing a different button (or the same button if button zero) changes the text and/or number format units on the LCD display.

Note that in the AXI design, drivers downloaded from Digilent Inc. for the Pmod HYGRO and Pmod CLS are used in the block design with some minimal modification and update to target the Vitis 2021.2 and the Arty A7-100 board instead of the original Arty board, as well as configure the IIC bus for dividing down from the AXI speed of 83.33 MHz. The AXI implementation integrates vendor components plus adding additional C code.

Note that the Zynq design is an approximate equivalent operation to the two Arty designs, but executing on a Zynq-7000 APSoC instead of a straight FPGA.

HYGRO Sensor Readings -Tester Experiment: Design Theory

The Verilog design targets either of the two mentioned Arty boards. Its primary design feature is the implementation of interacting Finite State Machines (FSMs) in the Verilog-HDL 2001 hardware language. Conceptual-only FSM diagrams are included in this design repository in the PDF document HYGRO-Tester-Design-Documents/HYGRO-Tester-Design-Diagrams.pdf. The final FSM diagrams show the original FSM design prior to and during coding of the first draft; and the block descriptions assist with understanding the code architecture. More complete FSM diagrams will also be included in the document, following the simpler FSM diagram page, for each major FSM designed in the HDL designs. Also, the Verilog design outputs the same LCD display text contents to the USB UART at baud 115200. That functionality was not implemented with the three IPI-BD designs.

Note that in the IPI-BD design, drivers downloaded from Digilent Inc. for the Pmod HYGRO and Pmod CLS are used in the block design with some minimal modification to target the Arty A7-100. Both drivers target the Arty-A7 instead of the Arty. The Git repository contains a submodule that pulls from a branch of the author’s fork of the Digilent Vivado-library repository on GitHub.

Coding style and choices of block design

Software design practices were used to author the Verilog-HDL 2001 sources. After the sources were drafted with a large top-level module and cohesive modules for drivers, a large self-instruction homework experiment was converted into a standalone design.

HYGRO Sensor Readings-Tester Experiment: 3rd-party references:

Digilent Inc. References

Arty – Getting Started with Microblaze Servers
https://reference.digilentinc.com/learn/programmable-logic/tutorials/arty-getting-started-with-microblaze-servers/start

Vivado Board Files
https://github.com/digilent/vivado-boards

Master XDC files for all Digilent Inc. boards, including Arty-A7-100T
https://github.com/Digilent/digilent-xdc

Digilent Inc IP library for Xilinx Vivado
https://github.com/Digilent/vivado-library/

Textbook References

In the HDL sources and design diagrams document:

- Pulse Stretcher Synchronous, Textbook Figure 8.28a. quoted from,
- FSM design theory and methodology adapted by Tim S. and extended from,

Volnei A. Pedroni, *Finite State Machines in Hardware: Theory and Design (with VHDL and SystemVerilog)*. London: The MIT Press, 2013.

Use of IP Integrator to create the Microblaze AXI block diagram and synthesis:

- Tutorials followed from text to understand IPI block design,

L. H. Crockett, R. A Elliot, M. A. Enderwitz, and R. W. Stewart, *The Zynq Book: Embedded Processing with the ARM Cortex-A9 on the Xilinx Zynq-7000 All Programmable SoC*, First Edition, Strathclyde Academic Media, 2014.

Study of Verilog HDL IEEE 1364-2001:

- FPGA-relevant homework studied and applied for coding Verilog-2001 designs,

Samir Palnitkar, *Verilog HDL: A Guide to Digital Design and Synthesis*. 2nd ed., USA: SunSoft Press, 2003.

Suggestions for best practices when coding VHDL:

- Suggestion to code RTL design with as few MMCM/PLL generated clock domains as possible,
- Suggestion to exercise software design practices when coding VHDL,

Ricardo Jasinski, *Effective Coding with VHDL: Principles and Best Practice*. London: The MIT Press, 2016.

Appendix A: How to initialize and open the HDL (Verilog) and IPI Block Design (MB-A7, MB-S7, Zynq) projects.

In projects fpga-iic-hygro-tester-1 and fpga-iic-hygro-tester-2 the author chose to commit all generated sources. This created a bulky Git project. To solve this, the fpga-iic-hygro-tester-3 project utilizes TCL scripts to recreate each project. Examples are shown here where G:\wa\ is the Windows 10 work area folder containing the fpga-iic-hygro-3 folder. Replace the path shown with the path of fpga-iic-hygro-tester-3 on your computer. If you are not familiar with the Xilinx Vivado TCL command-line, TCL is an open source and documented command-based language, and Xilinx provides documentation on Vivado's specific TCL commands. Also, Digilent Inc. provides detailed tutorials on recreating their demo projects, and can be referred to for learning.

PROJECT NAME	**TCL COMMANDS**

`HYGRO-Tester-Design-Verilog`
(Arty A7-100)
	`cd {G:/wa/fpga-iic-hygro-tester-3/HYGRO-Tester-Design-Verilog/Work_Dir/}`
	`source ./init_project_HYGRO-Verilog-A7-100.tcl`

`HYGRO-Tester-Design-Verilog`
(Arty S7-25)
	`cd {G:/wa/fpga-iic-hygro-tester-3/HYGRO-Tester-Design-Verilog/Work_Dir/}`
	`source ./init_project_HYGRO-Verilog-S7-25.tcl`

`HYGRO-Tester-Design-MB-A7`
(Arty A7-100)
	`cd {G:/wa/fpga-iic-hygro-tester-3/HYGRO-Tester-Design-MB-A7/Work_Dir/}`
	`source ./init_project_HYGRO-Tester-MB-A7.tcl`
	`cd {G:/wa/fpga-iic-hygro-tester-3/HYGRO-Tester-Design-MB-A7/}`
	`source ./IPI-BDs/system_mb_a7.tcl`

`HYGRO-Tester-Design-MB-S7`
(Arty S7-25)
	`cd {G:/wa/fpga-iic-hygro-tester-3/HYGRO-Tester-Design-MB-S7/Work_Dir/}`
	`source ./init_project_HYGRO-Tester-MB-S7.tcl`
	`cd {G:/wa/fpga-iic-hygro-tester-3/HYGRO-Tester-Design-MB-S7/}`
	`source ./IPI-BDs/system_mb_s7.tcl`

`HYGRO-Tester-Design-Zynq`
(Zybo Z7-20)
	`cd {G:/wa/fpga-iic-hygro-tester-3/HYGRO-Tester-Design-Zynq/Work_Dir/}`
	`source ./init_project_HYGRO-Tester-Zynq7.tcl`
	`cd {G:/wa/fpga-iic-hygro-tester-3/HYGRO-Tester-Design-Zynq/}`
	`source ./IPI-BDs/system_z7.tcl`

After the commands of a specific HDL project (in the above table) are executed successfully, the project can be synthesized. After the commands of a specific MB or Zynq project (in the above table) are executed successfully, the next step is to create an HDL Wrapper for the block design called "system.bd" or similar. After this, Xilinx Vivado can be used to synthesize the block design of the project. For the MB and Zynq projects, it necessary to export the fixed hardware with bitstream, and import this into a new Vitis application design. Source code is provided to be reused within a freshly created Vitis project. Refer to Xilinx's documentation, Digilent Inc. tutorials, and other documentation available with a Google search.