Revision	Date	Author	Comments
1A	2020-06-22	Tim S. timothystotts08@gmail.com	First publishable draft of the serial accelerometer readings tester
2A	2020-06-24	Tim S. timothystotts08@gmail.com	Additional design operation details.
3A	2020-07-02	Tim S. timothystotts08@gmail.com	Additional design operation details after addition of LED pulsing effects, button depress holding alternate text, and additional modularization of the VHDL and Verilog sources. Differences between the HDL and IPI-BD designs is mentioned. Clerical updates were made.

https://github.com/timothystotts/fpga-serial-acl-tester-1

Serial ACL Readings-Tester Experiment

Serial ACL Readings-Tester Experiment: Folder Structure

ACL Readings designs with equivalent function of performing a three-axis reading and displaying them on an LCD and USB terminal.

Project Folder	Project Description
ACL-Tester-Design-Single-Clock-Verilog (Vivado 2019.1)	A utility designed for a custom operation of a serial accelerometer and displaying milli-g-force readings on both an LCD and a serial terminal. The design is completely in Verilog RTL and a simple stimulus-only visual VHDL test-bench without a soft processor. 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 enable pulse instead of clock division. Clock dividers are only used for the generation of an output clock that outputs at a bus port on the FPGA.
ACL-Tester-Design-Single-Clock-VHDL (Vivado 2019.1)	A utility designed for a custom operation of a serial accelerometer and displaying milli-g-force readings on both an LCD and a serial terminal. The design is completely in VHDL RTL and a simple stimulus-only visual VHDL test-bench without a soft processor. 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 enable pulse instead of clock division. Clock dividers are only used for the generation of an output clock that outputs at a bus port on the FPGA.
ACL-Tester-Design-AXI (Vivado 2019.1 and SDK 2019.1)	A utility designed for custom operation of a serial accelerometer and displaying milli-g-force readings on both an LCD and a serial terminal. The design is completely in Microblaze AXI subsystem with standard Xilinx IP Integrator components plus Digilent Inc. User IP, and C language program executing on the Microblaze soft processor.

For the ACL-Tester-Design-AXI project, the BSP of the bootloader and the BSP of the application require that the flash family parameter be set in the xilisf configuration on the BSP configure page. The default value is 1, but must be changed to 5 as the Arty-A7-100T board has a Spansion 128Mbit serial flash for booting the FPGA.

To successfully open the project, it is necessary to add the directory arty-a7-100 or arty-a7-35 from the directory board_files/ to the installation directory of Vivado 2019.1 and SDK 2019.1. For example:

```
$ which vivado
/opt/Xilinx/Vivado/2019.1/bin/vivado
$ cd ./board_files
$ sudo cp -R ./arty-a7-100 ./arty-a7-35 /opt/Xilinx/Vivado/2019.1/data/boards/board_files/
$ sudo cp -R ./arty-a7-100 ./arty-a7-35 /opt/Xilinx/SDK/2019.1/data/boards/board_files/
```

Note that the Digilent Guide at https://reference.digilentinc.com/vivado/installing-vivado/2018.2 indicates that an initialization script can be executed in the user's profile to set up a path to additional board files. It is the experience of this author that the TCL initialization script provides an intermittent or non-functional detection of the board files in the user's home folder. By copying to the install directory of the tool, the board files are always found. Otherwise, the following TCL command is supposed to instruct Vivado to locate the board files:

```
set_param board.repoPaths [list "<extracted path>/Vivado/board_files"]
```

Serial ACL Readings-Tester Experiment: Methods of Operation

The purpose of the design is to boot a Digilent Inc. Arty-A7-100T (Artix-7) development board with PMOD CLS and PMOD ACL peripheral boards, which are a 16x2 Character dot-matrix LCD display, and a 3-axis MEMS Accelerometer, respectively. Each PMOD connects to the FPGA with its own dedicated SPI bus via a higher-speed plug and jack. The PMOD CLS connects to board

PMOD port JB. The PMOD ACL2 connects to board PMOD port JC. The use of extension cables makes, (a) the PMOD CLS able to connect to only one 2x6 PMOD port, and (b) the limited ability to move the PMOD ACL2 without requiring the movement of the Arty-A7 board or the Pmod CLS display. See Figure 1: Arty-A7-100T Assembled with Pmod CLS and Pmod ACL2.

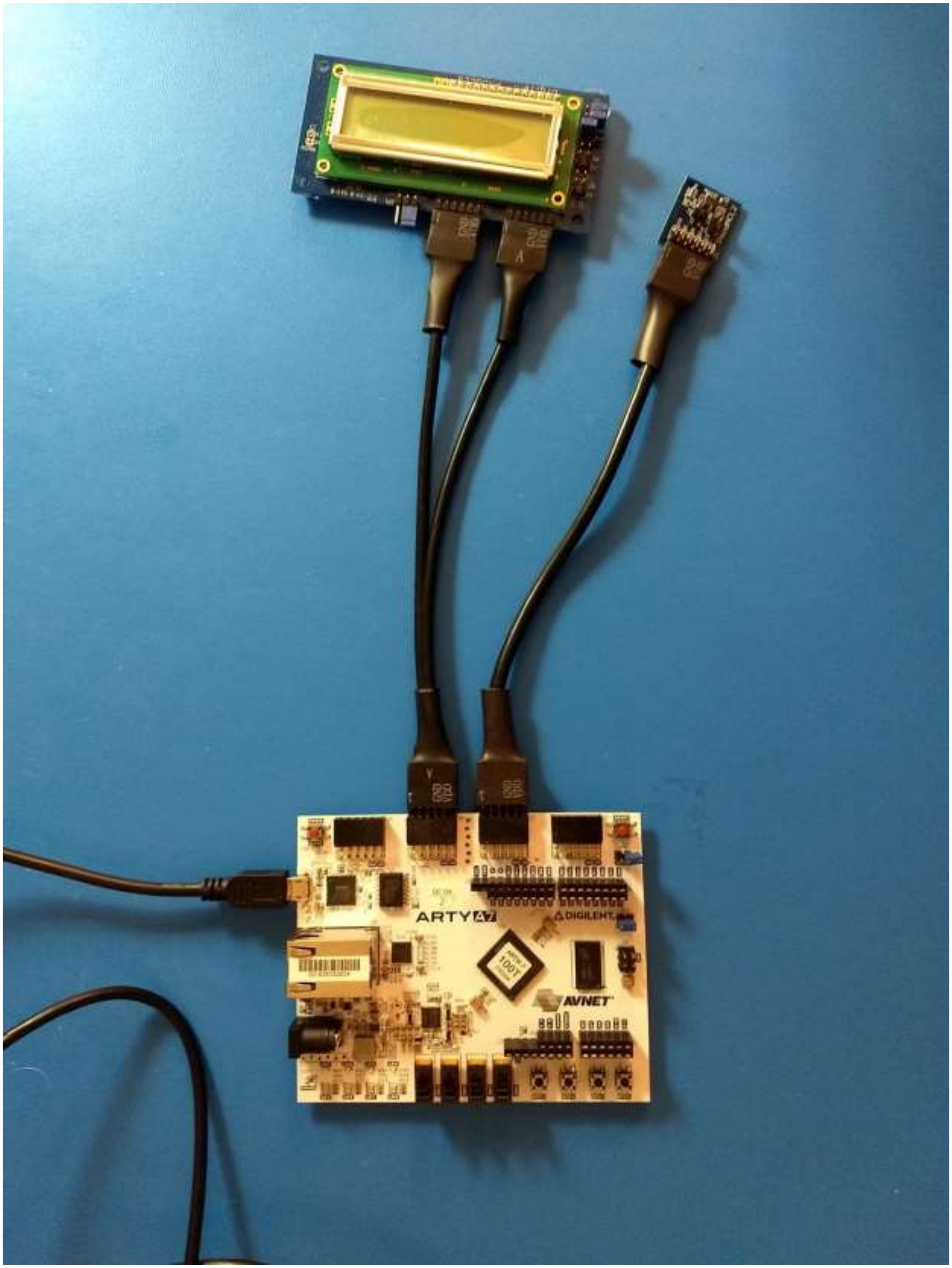

Figure 1: Arty-A7-100T Assembled with Pmod CLS and Pmod ACL2.

Serial ACL Readings-Tester Experiment: Method of Operation: streaming three-axis readings and displaying them, with alternate mode of activity detection

Serial ACL Readings-Tester Experiment: Design Operation

In the HDL implementations (Verilog, VHDL), the four switches sw0, sw1, sw2, sw3, are debounced and processed as mutually exclusive inputs. When switch 0 is exclusively selected to the ON position, the Tester design controls the PMOD ACL2 to stream 3-axis and compensating temperature readings at a rate of 100 Hz, and display readings at a rate of under 5 Hz. When switch 1 is exclusively selected to the ON position, the Tester design controls the PMOD ACL2 to detect motion activity and inactivity by thresholds. Each time there is an activity event, LD2 is lit green instead of red momentarily and a single reading from the PMOD ACL is displayed. Each time there is an inactivity event, LD3 is lit green instead of red momentarily and a single reading from the PMOD ACL is displayed.

Upon power-up, with the four switches not selected an operational mode, the LD0 displays Red to indicate that the measurements are not running. The LD1 displays Red to indicate that no operational mode is currently running. The LD2 and LD3 display Red to indicate Activity detection and Inactivity detection events, respectively, are not occurring.

Upon positioning switch 0 alone to ON, the LD0 transitions from Red to Blue to Green to indicate that measurements are running. The LD1 displays White to indicate that the Pmod ACL2's ADXL362 chip is operating in Measurement Mode and that acceleration readings are streaming from the ADXL362 to display in ASCII on the Pmod CLS and on the Digilent USB-UART at 115200 baud. The display on the Pmod CLS is in floating-point milli-g-force and raw compensating temperature; and the display on the Digilent USB-UART is four raw register readings per line for ability to be parsed by a desktop utility. LD2 may blink Green occasionally to indicate Activity events, and LD3 remains Red.

Upon positioning switch 1 alone to ON, the LD0 transitions from Red to Blue to Green to indicate that measurements are running. The LD1 displays Purple to indicate that the ADXL362 is operating in Linked Mode with activity detection. Every time the Pmod ACL2 is moved above a modest threshold of motion, the LD2 displays Green momentarily to indicate that the ADXL362 has determined an Activity event. When the Pmod ACL2 is left motionless on a desk, the LD3 displays Green momentarily to indicate that the ADXL362 has determined an Inactivity event. Note that if an Activity event does not display even with modest movement of the Pmod ACL2, it may be necessary to wait several seconds with the Pmod ACL2 at rest for an Inactivity event to occur. After this, the ADXL362 is ready to detect the next Activity event.

In the HDL implementations (Verilog, VHDL), the four buttons btn0, btn1, btn2, btn3, are debounced and processed as mutually exclusive inputs independent of the four switches. If button 2 is held depressed, the Terminal will display 3-axis milli-g readings and temperature reading decimal value to match the Pmod CLS display. When button 2 is released, the Terminal resumes displaying 16-bit hexadecimal readings from the four registers. If button 3 is held depressed, the Pmod CLS will display 16-bit hexadecimal readings from the four registers, matching the display of the Terminal.

The IP Integrator Block Design (IPI-BD) creates an equivalent design to the HDL designs; but the usage of the ADXL326 interrupts is substituted with polling the accelerometer; and LED color pulsing is not available, even though the LED colors are still managed by a simple form of 24-bit palette color selection based upon PWM period and duty cycles. Also, the IPI-BD design does not operate the temporary display text formatting by buttons 2 and 3.

Serial ACL Readings-Tester Experiment: Design Theory

Conceptual-only FSM diagrams are also included in the PDF document `ACL-Design-Documents/ACL-Tester-Design-Diagrams.pdf`. The intent is to update the block description and FSM state machine diagrams at a later point. The 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.

Note that in the IPI-BD (called AXI) Design, drivers downloaded from Digilent Inc. for the PMOD ACL2 and PMOD CLS are used in the block design with some minimal modification to target the Arty-A7-100T and support different modes of operating the Pmod ACL2. Both drivers target the Arty-A7 instead of the Arty; and the Pmod ACL2 User driver was copied, renamed, and expanded in the Xilinx SDK project, allowing for switching the ADXL326 between Measurement Mode and Linked Mode. The AXI example demonstrates integration of vendor components plus adding additional C code. 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 VHDL and Verilog 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. The top-level HDL sources were excessively large; so modules were created to contain top-level execution-procedure FSMs, data-to-ASCII conversion combinatorial, and the addition of LED color choice control with a "pulsing" effect for aesthetics.

The led_pwm_driver.v / led_pwm_driver.vhdl module was also refactored to infer a DSP48E1 unit without DRC errors, one unit per LED emitter. The Arty-A7 has 4 3-emitter color LEDs and 4 basic LEDs, totaling at 16 DSP48E1 being inferred to manage PWM period and duty cycle control of the eight LEDs.

Serial ACL Readings-Tester Experiment: 3rd-party references:

How To Store Your SDK Project in SPI Flash
https://reference.digilentinc.com/learn/programmable-logic/tutorials/htsspisf/start

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/