Revision	Date	Author	Comments
1E	2021-11-18	Tim S. timothystotts08@gmail.com	Initial draft of introduction document to fpga-serial-acl-tester-3 project, starting from documentation and sources of both the fpga-serial-acl-tester-1 and fpga-serial-acl-tester-2 documents.
2E	2021-11-19	Tim S. timothystotts08@gmail.com	Merged-in the details of the ACL-Tester-Design-Zynq project from fpga-serial-acl-tester-2. Updated document formatting.

https://github.com/timothystotts/fpga-serial-acl-tester-3

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-SV (Vivado 2021.2)	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 SystemVerilog RTL 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. The RTL is a minor modification of a copy from the Verilog project. The test-bench requires a mixed-language HDL simulator that supports Verilog and VHDL, and is not currently maintained.
ACL-Tester-Design- Verilog (Vivado 2021.2)	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-2001 RTL 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. The test-bench requires a mixed-language HDL simulator that supports Verilog and VHDL, and is not currently maintained.
ACL-Tester-Design-VHDL (Vivado 2021.2)	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-2002 and VHDL-2008 RTL without a soft processor, and the beginning of an OS-VVM VHDL test-bench that performs automated testing of the

	Measurement Mode of the Pmod ACL2, including text displayed on the Pmod LCD and UART Terminal. 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. The test-bench was tested with this RTL variant, and is anticipated to work with no modification when simulating with RivieraPRO or GHDL, separate from the Xilinx Vivado.
ACL-Tester-Design-MB (Vivado 2021.2 and Vitis 2021.2)	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 FreeRTOS C language program executing on the Microblaze soft processor.
ACL-Tester-Design-Zynq (Vivado 2021.2 and Vitis 2021.2)	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 Zynq-7000 AXI subsystem with standard Xilinx IP Integrator components plus Digilent Inc. User IP, and FreeRTOS C language program executing on the Zynq ARM processor.

To successfully open the MB, SV, Verilog, or VHDL project, it is necessary to create an empty Xilinx Vivado project, select the Part from Boards and click Refresh, then download the Arty-A7-100 board. Then you can delete the new project as the board files are now installed. To successfully open the Zynq project, it is necessary to do the same, instead selecting the Zybo-Z7-20 board. It is no longer necessary to install board_files in the data/boards/board_files folder of the Vivado install.

Serial ACL Readings-Tester Experiment: Arty-A7: Methods of Operation

The purpose of the MB, SV, Verilog, or VHDL project design is to boot a Digilent Inc. Arty-A7-100T (Artix-7) development board with PMOD CLS, PMOD ACL2, and PMOD SSD peripheral boards, which are a 16x2 Character dot-matrix LCD display, a 3-axis MEMS Accelerometer, and a two-digit 7-segment display, respectively. The PMOD CLS and PMOD ACL2 each connect 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 PMOD SSD connects to board PMOD port JA. The use of extension cables makes, (a) the PMOD CLS able to connect to only one 2x6 PMOD port, (b) the PMOD SSD able to connect to only one 2x6 PMOD port, (c) the limited ability to move the PMOD ACL2 without requiring the movement of the Arty-A7 board or the Pmod CLS display or the Pmod SSD display. See Figure 1: Arty-A7-100T Assembled with Pmod CLS, Pmod ACL2, Pmod SSD.

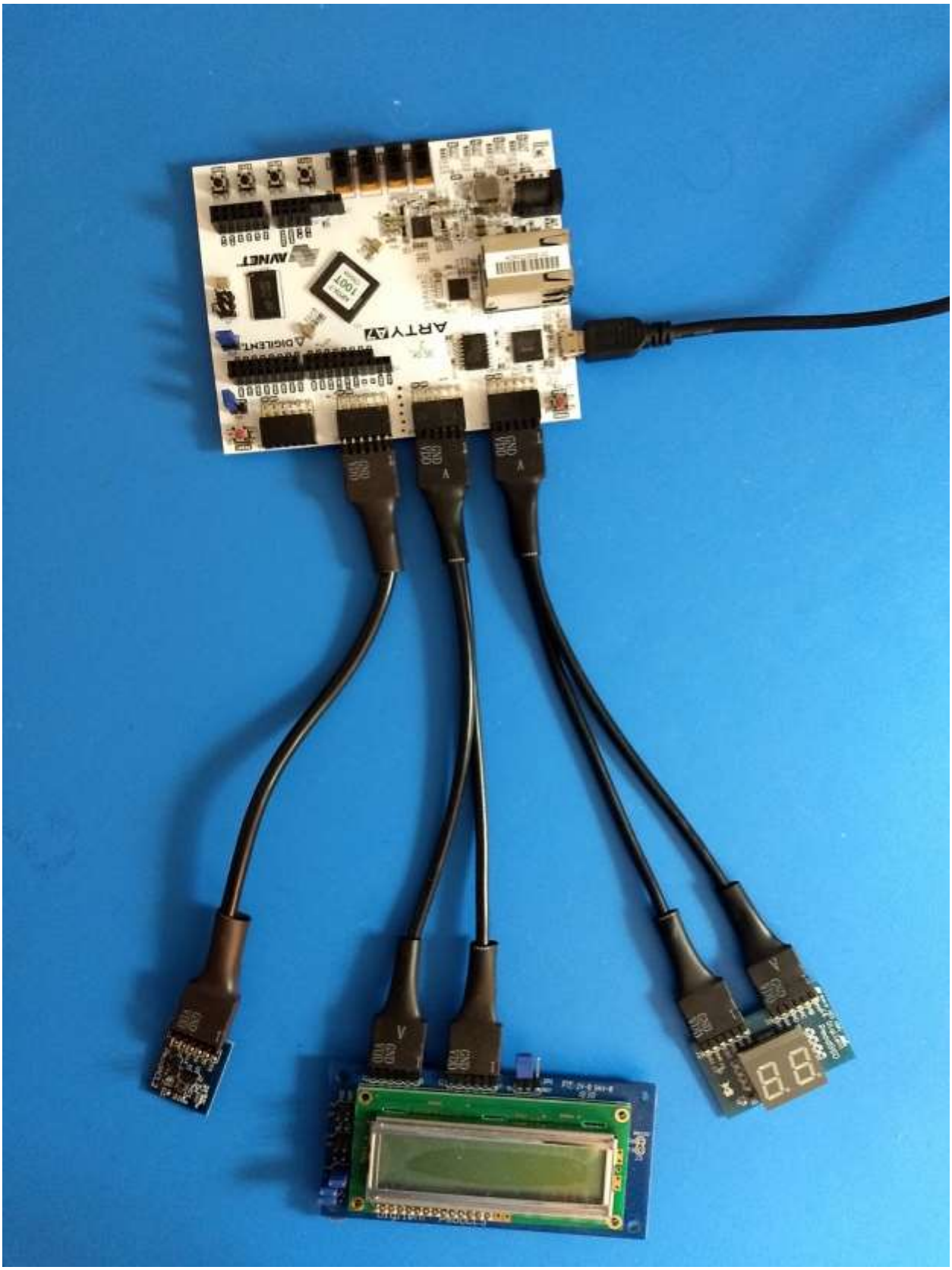

Figure 1: Arty-A7-100T Assembled with Pmod CLS, Pmod ACL2, Pmod SSD.

Serial ACL Readings-Tester Experiment: Zybo-Z7: Methods of Operation

The purpose of the design is to boot a Digilent Inc. Zybo-Z7-20 (Zynq-7000) development board with PMOD CLS, PMOD ACL2, and PMOD SSD peripheral boards, which are a 16x2 Character dot-matrix LCD display, a 3-axis MEMS Accelerometer, and a two-digit 7-segment display, respectively. The PMOD CLS and PMOD ACL2 each connect to the Zynq PL 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 PMOD SSD connects to board PMOD port JE. The use of extension cables makes, (a) the PMOD CLS able to connect to only one 2x6 PMOD port, (b) the PMOD SSD able to connect to only one 2x6 PMOD port, (c) the limited ability to move the PMOD ACL2 without requiring the movement of the Zybo-Z7-20 board or the Pmod CLS display or the Pmod SSD display. See Figure 2: Zybo-Z7-20 Assembled with Pmod CLS, Pmod ACL2, Pmod SSD.

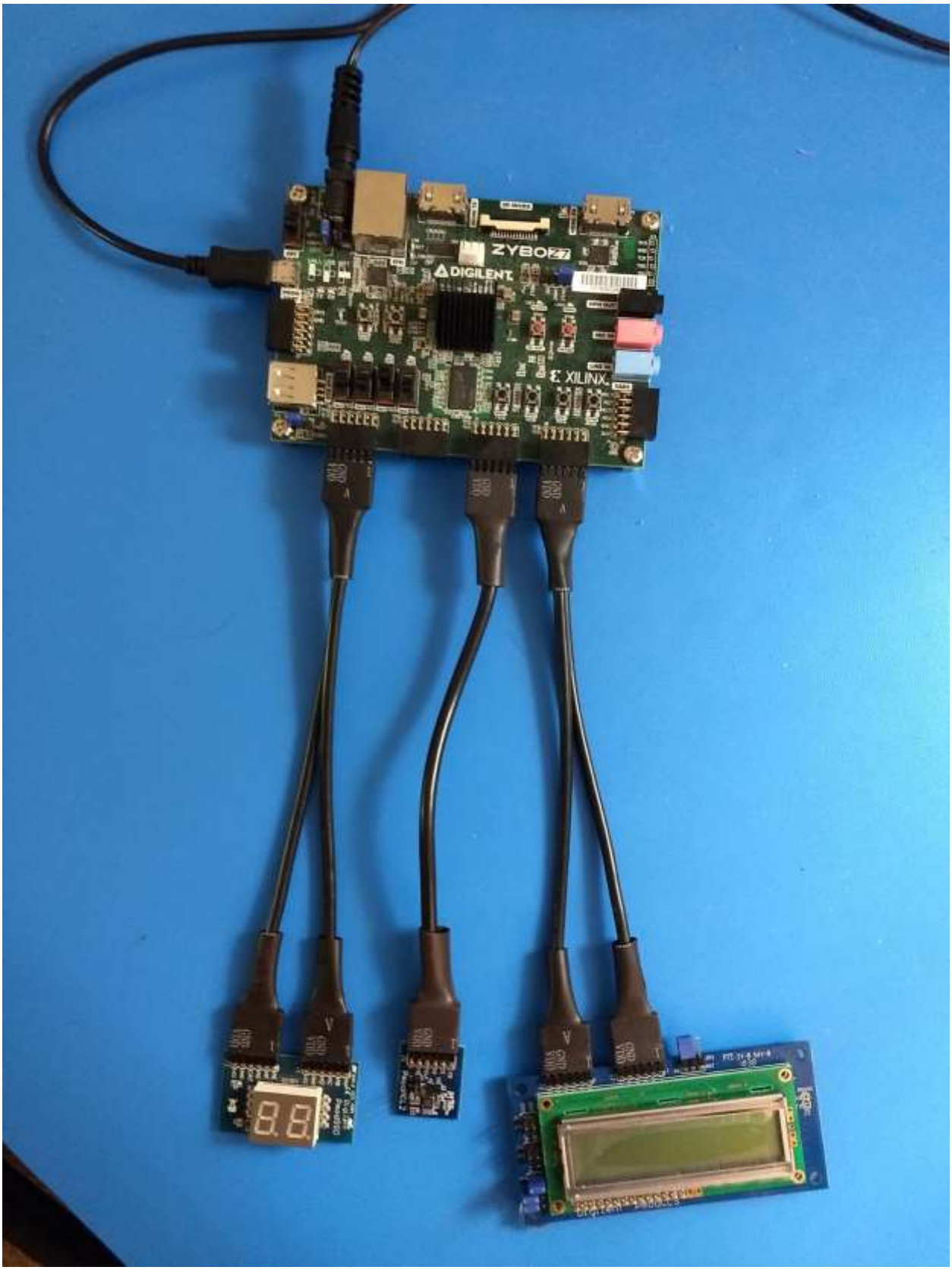

Figure 2: Zybo-Z7-20 Assembled with Pmod CLS, Pmod ACL2, Pmod SSD.

Serial ACL Readings-Tester Experiment: Method of Operation: Arty-A7: streaming three-axis readings and displaying them, with alternate mode of activity detection

Serial ACL Readings-Tester Experiment: Arty-A7: Design Operation

In the HDL implementations (SV, 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 ACL2 is displayed. Each time there is an inactivity event, LD3 is lit green instead of red momentarily and a single reading from the PMOD ACL2 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.

LD0 boots as Blue. Upon selecting a switch, SW0 or SW1, the internal FSM quickly transitions to Red to indicate display configuration is in process. After this it transitions to White or Purple to indicate the display running mode selected. Due to sub-second transition, the LD0 is only seen as red if the FSM were to stick at display initialization and not transition to the display running mode. (The FSM of the ACL-Tester-Design-MB project skips displaying Red on LD0 and transitions directly from Blue to Green. This keeps the software FSM simpler.)

For example, upon positioning switch 0 alone to ON, the LD0 transitions from Blue to Red 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 fixed-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 remains Red, and LD3 remains Red.

Upon positioning switch 1 alone to ON, the LD0 transitions from Blue to Red 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 moderate 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 all implementations, 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 UART Terminal will display fixed-point 3-axis milli-g readings and temperature reading decimal value to match the Pmod CLS display. When button 2 is released, the UART 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 UART Terminal.

If button 0 is pressed momentarily, the left digit of the Pmod SSD increments from zero to nine, and back to zero. Internally, the design selects a different Activity Threshold and Timer preset value for the next time the switch 1 is deselected and reselected for executing Linked Mode with Activity Events. If button 1 is pressed momentarily, the

right digit of the Pmod SSD increments from zero to nine, and back to zero. Internally, the design selects a different Inactivity Threshold and Timer preset value for the next time the switch 1 is deselected and reselected for executing Linked Mode with Activity Events.

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. (The color mixing technique was based upon the Digilent Inc. Arty-A7-100's BIST software using pulse duty cycles to mix LED colors as is not a representation of true or calibrated color mixing found in consumer or industrial products.)

Serial ACL Readings-Tester Experiment: Arty-A7: HDL and MB AXI: Design Theory

Conceptual-only FSM diagrams are also included in the PDF document `ACL-Design-Documents/ACL-Tester-Design-Diagrams.pdf`. 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. More complete FSM diagrams are also included in the document, following after the simpler FSM diagram page, for each major FSM designed in the HDL designs.

Note that in the IPI-BD (called MB) Design, drivers downloaded from Digilent Inc. for the PMOD ACL2, PMOD CLS, and a generic AXI PWM, 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 Pmod drivers are repackaged to target the Arty-A7 instead of the Arty, and to be used in a newer version of Xilinx Vivado and Vitis. The Pmod ACL2 User software driver source code module was copied, renamed, and expanded in the Xilinx Vitis project, allowing for switching the ADXL326 between Measurement Mode and Linked Mode. The AXI design integrates the vendor components plus adds 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; thus, 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: Method of Operation: Zybo-Z7: streaming three-axis readings and displaying them, with alternate mode of activity detection

Serial ACL Readings-Tester Experiment: Zybo-Z7: Design Operation

In the IPI block design and SDK design, 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 take 3-axis and compensating temperature readings at a rate of 100 Hz, poll these readings at an approximated rate of under 5 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, LD6 is lit green instead of red momentarily and a single reading from the PMOD ACL2 is displayed. Each time there is an inactivity event, LD5 is lit green instead of red momentarily and a single reading from the PMOD ACL is displayed.

LD0 through LD3 are used to display operational statuses. The LD0 is on when switch 0 is positioned to on, LD1 is on when switch 1 is positioned to on, LD2 is on when the accelerometer readings are being taken, and LD3 is on when the status register of the accelerometer indicates that it is in the AWAKE state. (Refer to the ADXL362 datasheet.) 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 LD2 is lit to indicate readings are being taken. The Pmod ACL2's ADXL362 chip is operating in Measurement Mode and that acceleration readings are polled 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 fixed-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. LD5 remains Red, and LD6 remains Red.

Upon positioning switch 1 alone to ON, the LD2 is lit to indicate readings are being taken. The ADXL362 is operating in Linked Mode with activity detection. Every time the Pmod ACL2 is moved above a modest threshold of motion, the LD6 displays Green momentarily to indicate that the ADXL362 has determined an Activity event. When the Pmod ACL2 is left motionless on a desk, the LD5 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.

The display text of the USB-UART and the Pmod CLS can be changed for purposes of debugging. If button 2 is held depressed, the Terminal will display fixed-point 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 seven-segment display is used to indicate preset index for both activity events and inactivity events. If button 0 is pressed momentarily, the right digit of the Pmod SSD increments from zero to nine, and back to zero. Internally, the design selects a different Inactivity Threshold and Timer preset value for the next time the switch 0 is deselected and reselected for executing Linked Mode with Activity Events. If button 1 is pressed momentarily, the left digit of the Pmod SSD increments from zero to nine, and back to zero. Internally, the design selects a different Activity Threshold and Timer preset value for the next time the switch 1 is deselected and reselected for executing Linked Mode with Activity Events. (Refer to the ADXL362 datasheet and the "thresh_presets_include" sources.)

Serial ACL Readings-Tester Experiment: Zybo-Z7: Zynq AXI: Design Theory

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 Zybo-Z7-20 and support different modes of operating the Pmod ACL2. Both drivers target the Zybo-Z7-20 instead of the Arty; and the Pmod ACL2 User driver was copied, renamed, and expanded in the Xilinx Vitis project, allowing for switching the ADXL326 between Measurement Mode and Linked Mode. The IPI design integrates the vendor components plus adds 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

The led_pwm.[ch] module was also refactored from Arty-A7-100T definitions to Zybo-Z7-20 definitions of the quantity and silkscreen number of the LEDs.

Serial ACL 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/

Zybo Z7 Reference Manual
https://reference.digilentinc.com/reference/programmable-logic/zybo-z7/reference-manual

Textbook References

In the HDL sources and design diagrams document:

- Pulse Stretcher Synchronous, Textbook Figure 8.28a. quoted from,
- One Shot FSM, Textbook Figure 5.7c 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. Two figures reprinted courtesy of The MIT Press.

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

- Tutorials followed from text to understand IPI block design,

Use of IP Integrator to create the Zynq-7000 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.

Study of SystemVerilog HDL IEEE 1800-2012:

- FPGA-relevant reading studied in-part and applied for coding SystemVerilog designs,

Stuart Sutherland, *RTL Modeling with SystemVerilog ...* . USA: Sutherland HDL Inc., 2017.

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

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 (SV, Verilog, VHDL) and AXI (MB, Zynq) projects

In projects fpga-serial-acl-tester-1 and fpga-serial-acl-tester-2 the author chose to commit all generated sources. This created a bulky Git project. To solve this, the fpga-serial-acl-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-serial-acl-tester-3 folder. Replace the path shown with the path of fpga-serial-acl-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**
ACL-Tester-Design-SV	`cd {G:/wa/ACL-Tester-Design-SV/Work_Dir/}` `source ./init_project-ACL-SV.tcl`
ACL-Tester-Design-Verilog	`cd {G:/wa/ACL-Tester-Design-Verilog/Work_Dir/}` `source ./init_project-ACL-Verilog.tcl`
ACL-Tester-Design-VHDL	`cd {G:/wa/ACL-Tester-Design-VHDL/Work_Dir/}` `source ./init_project-ACL-VHDL.tcl`

ACL-Tester-Design-MB	`cd {G:/wa/fpga-serial-acl-tester-3/ACL-Tester-Design-MB/Work_Dir}`
	`source ./init_project_ACL-AXI-MB.tcl`
	`cd {G:/wa/fpga-serial-acl-tester-3/ACL-Tester-Design-MB}`
	`source ./IPI-BDs/system.tcl`

ACL-Tester-Design-Zynq	`cd {G:/wa/fpga-serial-acl-tester-3/ACL-Tester-Design-Zynq/Work_Dir}`
	`source ./init_project_ACL-AXI-Zynq.tcl`
	`source ../IPI-BDs/system.tcl`

After the commands of a specific project (in the above table) are executed successfully, the next step is to create an HDL Wrapper for the block design called "system.bd". 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 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.