Revision	Date	Author	Comments
1B	2022-10-01	Tim S. timothystotts08@gmail.com	First partial draft of the serial flash sector tester, refreshed.
2B	2022-11-13	Tim S. timothystotts08@gmail.com	Completed first full draft of Serial Flash Sector Tester Refreshed.
B	2022-11-13	Tim S. timothystotts08@gmail.com	First release of fpga-serial-mem-tester-3 .

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

Serial Flash Sector-Tester Experiment

Serial Flash Sector-Tester Experiment: Folder Structure

Serial Flash Sector Testing designs with equivalent function of performing an erase/program/read cycle in 1/32 address steps of a 256 Mbit Serial Flash. Note that the Xilinx MicroBlaze and Xilinx Zynq-7000 designs can only address the first 128 Mbit of serial flash due to a limitation of the Xilinx SPI IP block.

Project Folder	**Project Description**
SF-Tester-Design-SV (Vivado 2021.2)	A utility designed for a custom operation of a serial flash and displaying erase/program/read-back byte error count on both an LCD and a serial terminal. The design is completely in SystemVerilog RTL without a soft processor and includes a simple stimulus-only visual VHDL test-bench. Two clock domains exist; 40. 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. Note that the full 256 Mbits of the SPI flash are addressed by design of the RTL SPI Flash serial driver using a 32-bit address when communicating, instead of a 24-bit address.
SF-Tester-Design-VHDL (Vivado 2021.2)	A utility designed for a custom operation of a serial flash and displaying erase/program/read-back byte error count on both an LCD and a serial terminal. The design is completely in VHDL RTL without a soft processor and includes a simple stimulus-only visual VHDL test-bench. Two clock domains exist; 40. 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 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. Note that the full 256 Mbits of the SPI flash are addressed by design of the RTL SPI Flash serial driver using a 32-bit address when communicating, instead of a 24-bit address.
SF-Tester-Design-MB-A7 (Vivado 2021.2 and Vitis 2021.2)	A utility designed for custom operation of a serial flash and displaying erase/program/read-back byte error count on both an LCD and a serial terminal. The design is completely in Microblaze AXI subsystem with standard Xilinx IP Integrator components, and FreeRTOS C language program executing on the Microblaze soft processor. Xilinx Vivado User IP from Digilent Inc. is used to control the Pmod peripherals. Note that only the first 128 Mbits of the SPI Flash are addressable with the provided 24-bit address protocol. This design targets a Digilent Inc. Arty A7-100 board.
SF-Tester-Design-MB-S7 (Vivado 2021.2 and Vitis 2021.2)	A utility designed for custom operation of a serial flash and displaying erase/program/read-back byte error count on both an LCD and a serial terminal. The design is completely in Microblaze AXI subsystem with standard Xilinx IP Integrator components, and FreeRTOS C language program executing on the Microblaze soft processor. Xilinx Vivado User IP from Digilent Inc. is used to control the Pmod peripherals. Note that only the first 128 Mbits of the SPI Flash are addressable with the provided 24-bit address protocol. This design targets a Digilent Inc. Arty S7-25 board.
SF-Tester-Design-Zynq (Vivado 2021.2 and Vitis 2021.2)	A utility designed for custom operation of a serial flash and displaying erase/program/read-back byte error count 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 #0 processor. Note that only the first 128 Mbits of the SPI Flash are addressable with the Xilinx-provided 24-bit address protocol. 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 SF-Tester-Design-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. The 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.

Serial Flash Sector-Tester Experiment: Methods of Operation

The purpose of the MB/MB-S7, SV, Verilog, or VHDL project design is to boot either a Digilent Inc. Arty A7-100 (Artix-7) or a Digilent Inc. Arty S7-25 (Spartan-7) development board with PMOD CLS and PMOD SF3 peripheral boards, which are a 16x2 Character dot-matrix LCD display, and a 256 Mbit N25Q serial flash, respectively. The PMOD CLS and PMOD SF3 each connect to the FPGA with its own dedicated SPI bus via a higher-speed plug and jack. On the Arty A7-100 the PMOD CLS connects to board PMOD port JB, and the PMOD SF3 connects to board PMOD port JC. On the Arty S7-25 the PMOD CLS connects to board PMOD port JA, and the PMOD SF3 connects to board PMOD port JB. The use of an extension cable makes the PMOD CLS able to connect to only one 2x6 PMOD port.

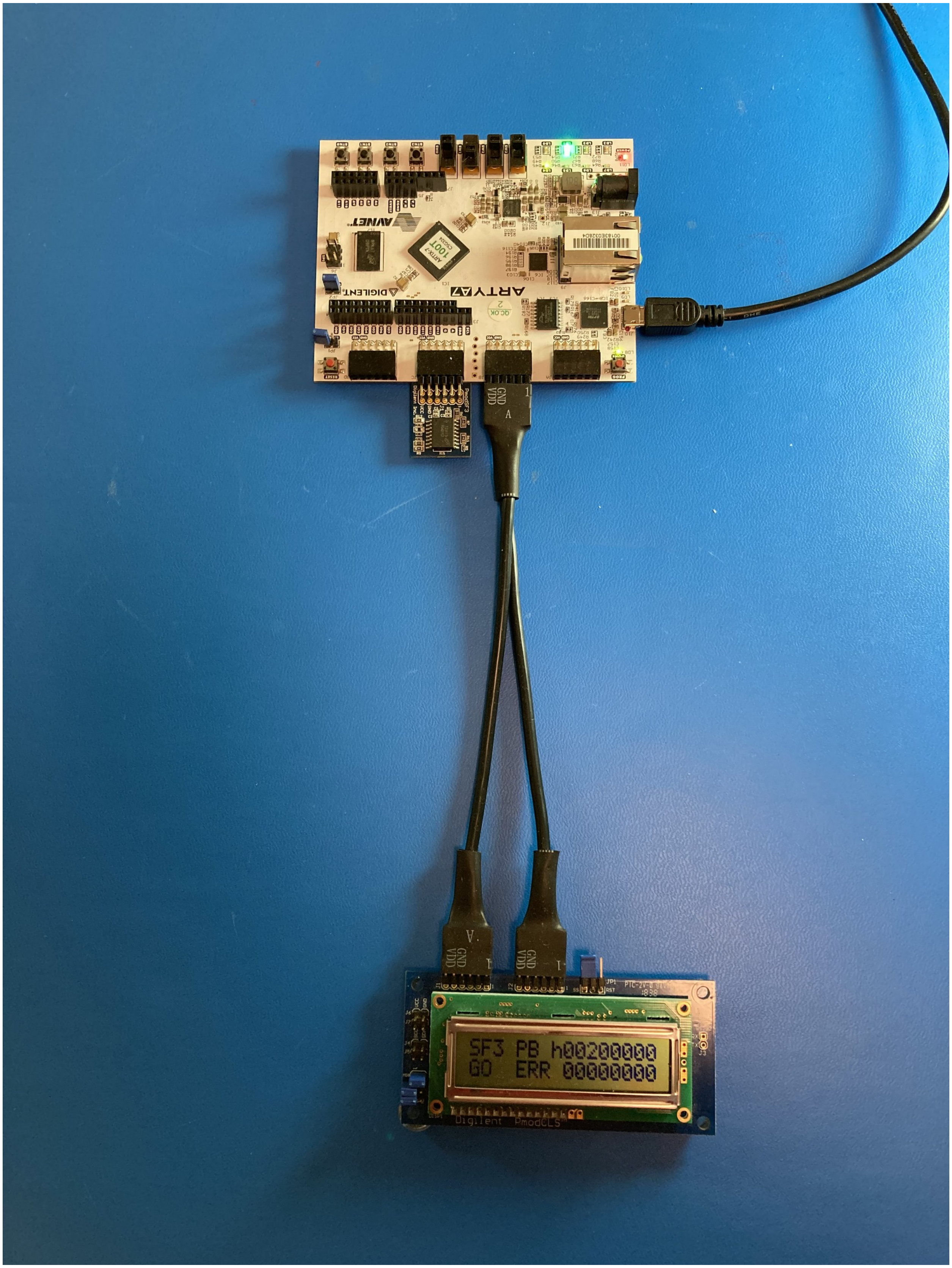


Figure 1: Arty A7-100 Assembled and Running with Pmod CLS and Pmod SF3

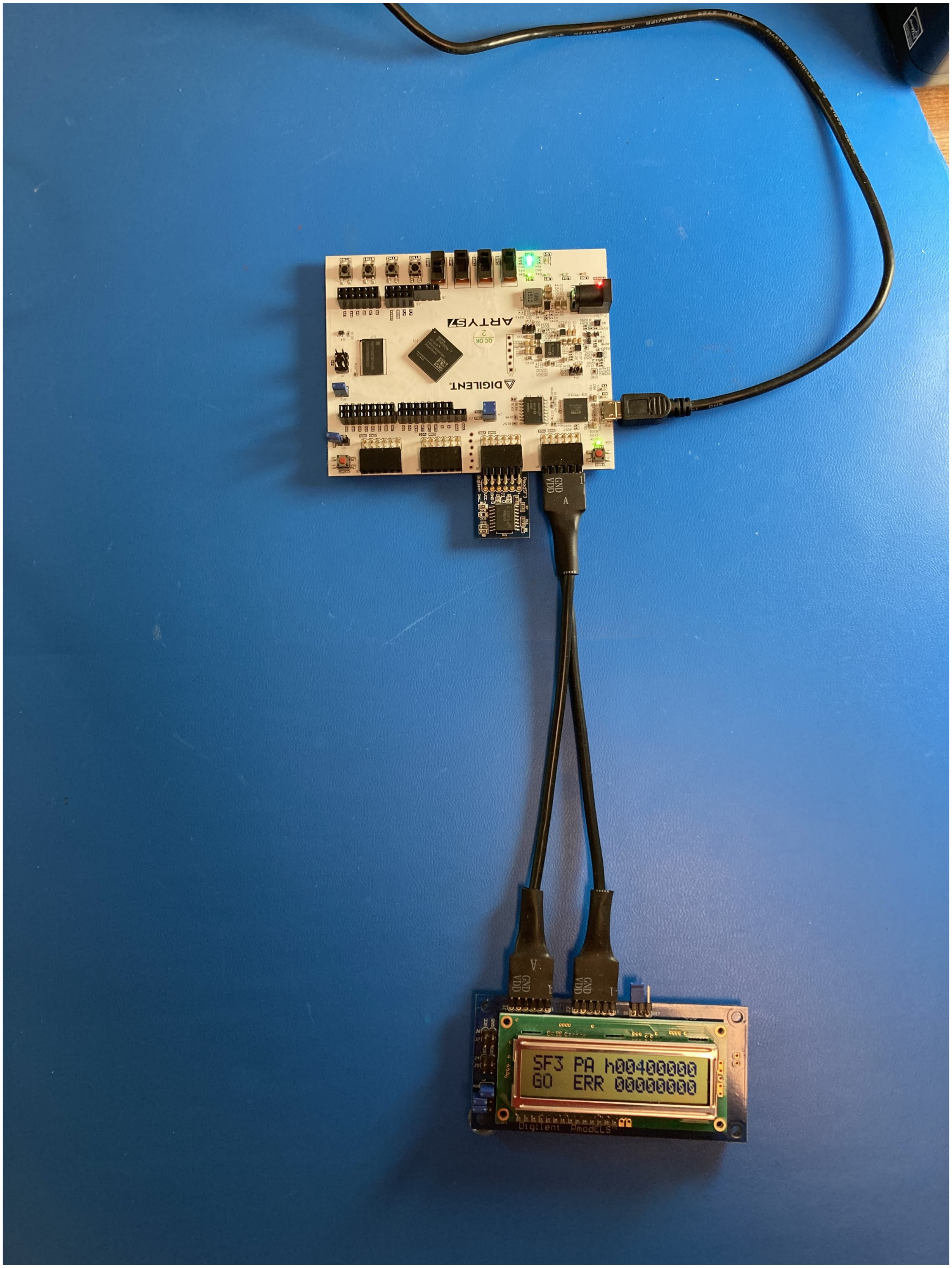
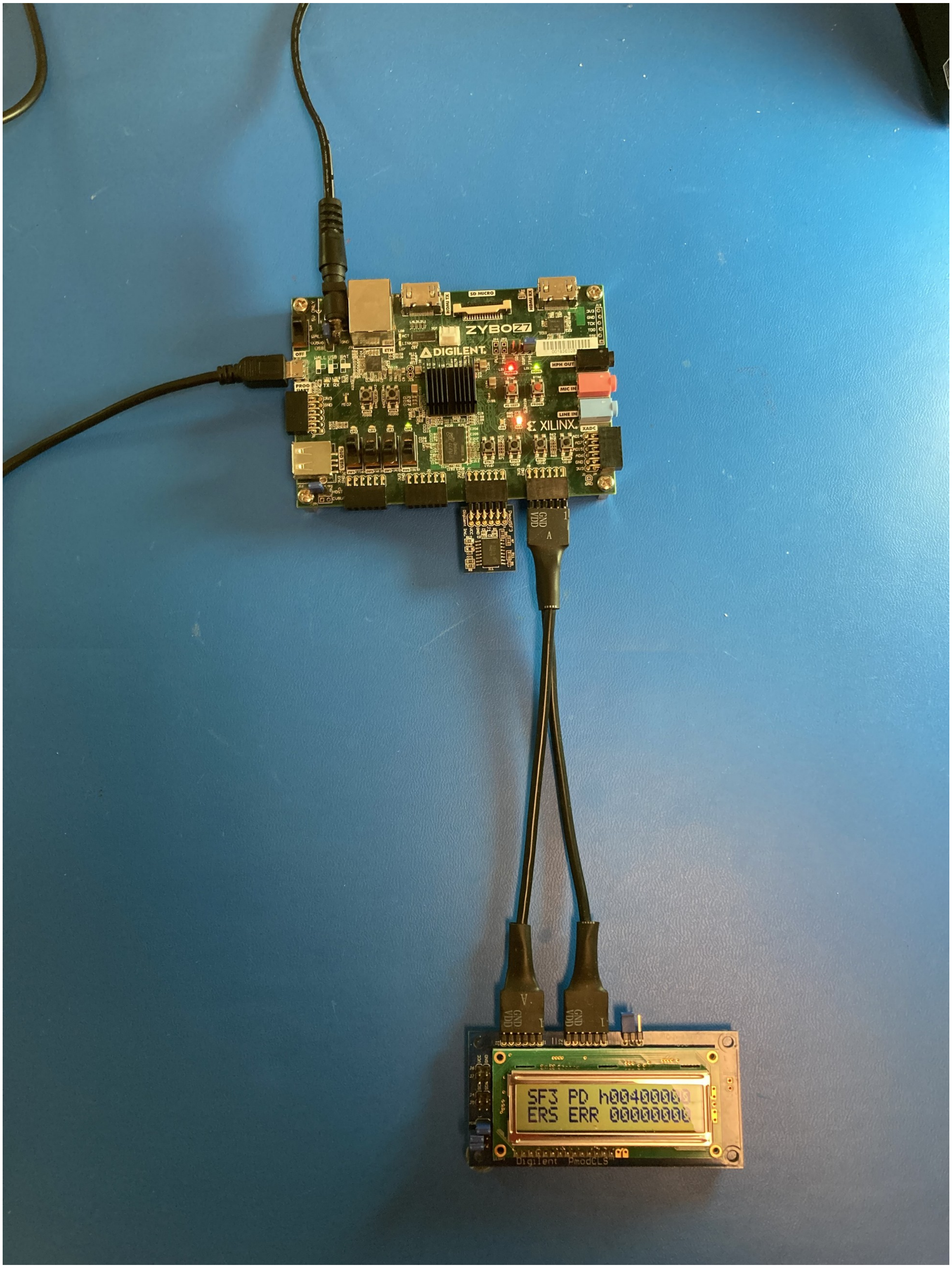


Figure 2: Arty S7-25 Assembled and Running with Pmod CLS and Pmod SF3



```
SF3 PD h00400000
ERS ERR 00000000
```

Figure 3: Zybo Z7-20 Assembled and Running with Pmod CLS and Pmod SF3

Serial Flash Sector-Tester Experiment: Method of Operation: iterative erase/program/read-back byte memory test from beginning to end of the serial flash

Serial Flash Sector-Tester Experiment: Design Operation

The purpose of the MB-A7/MB-S7, SV, or VHDL project design is to boot either a Digilent Inc. Arty A7-100 (Artix-7) or a Digilent Inc. Arty S7-25 (Spartan-7) development board with PMOD CLS and PMOD SF3 peripheral boards, which are a 16x2 Character dot-matrix LCD display and 256 Mbit N25Q serial flash, respectively. The PMOD CLS and PMOD SF3 each connect to the FPGA with its own dedicated SPI bus via a higher-speed plug and jack.

In the HDL implementations (SV, VHDL), the four switches sw0, sw1, sw2, sw3, are debounced and processed as mutually exclusive inputs. Each time one of the four switches is exclusively selected to the ON position, or one of the four buttons is depressed momentarily, the Tester design controls the PMOD SF3 to write and then read-back the first or next 1/32nd of the serial flash memory. If a switch is positioned to ON, the Tester will continue testing until the entire flash has been tested for a pattern write and read-back. If no switch is positioned to ON, one of the buttons can be depressed momentarily to continue the testing to only the next 1/32nd of the flash memory. Note that each switch selects a different pattern to written and read-back from the flash memory. The same is true for the buttons. Button 0 and Switch 0 select test pattern A. Button 1 and Switch 1 select test pattern B. Button 2 and Switch 2 select test pattern C. Button 3 and Switch 3 select test pattern D. Colors are displayed on the color LEDs to indicate progress and step being executed. Two of the basic LEDs display status, Test Done, Test with no Errors. The pattern name, step name, and error count are also displayed in text on the 16x2 LCD display. The same text displayed on the LCD is also output in plain text on the USB Terminal.

For the Arty A7-100:

Note that this board has four (4) color LEDs.

When the tester program first starts, all four color LEDs are lit Red, and all four basic LEDs are unlit. This indicates that the program is idle. The LCD displays text of:

```
SF3 PA h00000000
GO  ERR 00000000
```

The first line states that the program is “SF3” tester, with Pattern A, and starting address of 0x00000000. The second line states that the program is at step GO (waiting on a button press or switch), and that the byte Error Count is zero.

For example, pressing Button 2 will cause the program to execute testing of the first 1/32nd address space of the Pmod SF3 serial flash memory. First, at step GO, three of the color LEDs will turn off and LED 2 will turn Green to indicate that Pattern C is starting. The LCD will also display PC for Pattern C. The color LEDs will cycle from 0 to 3 displaying white, then orange, then the next color LED, indicating each of the four steps that the Tester executes through. When the LED is lit White the step is executing, and then when the LED is lit Orange there is a pause in processing. Matching the White LED, the LCD will cycle through the following 5 steps:

GO – pausing for button press or switch position
ERS – erasing 1/32nd of the flash memory space
PRO – programming a byte pattern to the flash memory space
TST – reading-back and checkign the byte pattern in the flash memory space
END – pausing to indicate that this 1/32nd iteration is completed

After the first iteration, the basic LED 4 is lit to indicate that no byte errors have occurred in the testing. This LED will remain lit as long as no read-back errors occur.

Continuing the example, moving switch 1 to ON will cause the program to execute testing of the next 1/32 address space of the Pmod SF3. The steps iterated are the same, with the same LED statuses. The LCD is updated as to which address space is currently being tested, or was tested (when idle). Because a switch is positioned instead of a button press, the Tester will automatically test each next 1/32nd of the flash memory until testing of all 32 address spans is completed. At that time, if there are no errors, the LCD and USB terminal will still display zero after the text “ERR”, and basic LED 5 will also light to indicate Test Done. Also, as the SF3 contians a 256 Mbit flash, the final 1/32nd starting address is 0x01F00000. The Tester will light all four color LEDs as Red again, when the testing is completed. If successful, the error count will be zero, and basic LEDs 4 and 5 will be lit.

For the Arty S7-25:

Note that this board has two (2) color LEDs.

When the tester program first starts, both color LEDs are lit Red, and all four basic LEDs are unlit. This indicates that the program is idle. The LCD displays text of:

```
SF3 PA h00000000
GO  ERR 00000000
```

The first line states that the program is “SF3” tester, with Pattern A, and starting address of 0x00000000. The second line states that the program is at step GO (waiting on a button press or switch), and that the byte Error Count is zero.

The operation of the SF3 Tester on the Arty S7-25 is the same as on the Arty A7-100, with only a difference that there are 2 colors LEDs. When a pattern is selected by a button press or switch position, LED0 will be Green (Pattern A) or Blue (Pattern C), or LED1 will be Green (Pattern B) or Blue (Pattern D). Basic LEDs 2,3,4,5 have the same display as basic LEDs 4,5,6,7 did on the Arty A7-100.

Note that in the MicroBlaze AXI designs, drivers downloaded from Digilent Inc. for the PMOD SF3 and PMOD CLS are used in the block design with some minimal modification and update to Xilinx Vivado and Vitis release 2021.2. The AXI implementation integrates vendor components plus adding additional C code. Note that the AXI implementation is much slower to execute each of the 1/32 cycles, and it only executes internally with a 24-bit flash address, limiting the testing to the first 128 Mbits of the 256 Mbits of the SF3 flash memory.

Note that the Zynq AXI design that targets the Zybo Z7-20 is functionally equivalent to the MicroBlaze AXI design of either Arty target.

Serial Flash Sector-Tester Experiment: Design Theory

Conceptual-only FSM diagrams are also included in this project in the PDF document <project>/SF3-Design-Documents/SF3-Experiment-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 are also be included in the document, following the simpler FSM diagram page, for each major FSM designed in the HDL designs.

Note that in the IPI-BD (called AXI) Design, drivers downloaded from Digilent Inc. for the PMOD SF3 and PMOD CLS are used in the block design with some minimal modification to target the Arty-A7-100. The drivers for Arty A7 and Arty S7 target the Arty A7-100 instead of the older Arty. The drivers for Zybo Z7-20 target that board. The Make files and SPI sources were updated to match usage with Xilinx Vitis 2021.2 . 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 SystemVerilog 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. (Some logic was also borrowed from the fpga-serial-acl-tester-3 design.) The first draft 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.

The led_pwm_driver.vhdl module was also refactored to infer a DSP48E1 unit without DRC errors, one unit per LED emitter. The Arty A7-100 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. The Arty S7-25 has 2 3-emitter color LEDs and 4 basic LEDs, totaling at 10 DSP48E1 being inferred to manage PWM period and duty cycle control of the six LEDs.

Serial Flash Sector-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

How To Store Your SDK Project in SPI Flash
https://reference.digilentinc.com/learn/programmable-logic/tutorials/htsspisf/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 (SV, VHDL) and IPI Block Design (MB-A7, MB-S7, Zynq) projects.

In projects fpga-serial-mem-tester-1 and fpga-serial-mem-tester-2 the author chose to commit all generated sources. This created a bulky Git project. To solve this, the fpga-serial-mem-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-mem-tester-3 folder. Replace the path shown with the path of fpga-serial-mem-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**

SF-Tester-Design-SV
(Arty A7-100)
	`cd {G:/wa/fpga-serial-mem-tester-3/SF-Tester-Design-SV/Work_Dir/}`
	`source ./init_project_SF-SV-A7-100.tcl`

SF-Tester-Design-SV
(Arty S7-25)
	`cd {G:/wa/fpga-serial-mem-tester-3/SF-Tester-Design-SV/Work_Dir/}`
	`source ./init_project_SF-SV-S7-25.tcl`

SF-Tester-Design-VHDL
(Arty A7-100)
	`cd {G:/wa/fpga-serial-mem-tester-3/SF-Tester-Design-VHDL/Work_Dir/}`
	`source ./init_project_SF-VHDL-A7-100.tcl`

SF-Tester-Design-VHDL
(Arty S7-25)
	`cd {G:/wa/fpga-serial-mem-tester-3/SF-Tester-Design-VHDL/Work_Dir/}`
	`source ./init_project_SF-VHDL-S7-25.tcl`

SF-Tester-Design-MB-A7
(Arty A7-100)
	`cd {G:/wa/fpga-serial-mem-tester-3/SF-Tester-Design-MB-A7/Work_Dir/}`
	`source ./init_project_SF-Tester-MB-A7.tcl`
	`cd {G:/wa/fpga-serial-mem-tester-3/SF-Tester-Design-MB-A7/}`
	`source ./IPI-BDs/system_mb_a7.tcl`

SF-Tester-Design-MB-S7
(Arty S7-25)
	`cd {G:/wa/fpga-serial-mem-tester-3/SF-Tester-Design-MB-S7/Work_Dir/}`
	`source ./init_project_SF-Tester-MB-S7.tcl`
	`cd {G:/wa/fpga-serial-mem-tester-3/SF-Tester-Design-MB-S7}`
	`source ./IPI-BDs/system_mb_s7.tcl`

SF-Tester-Design-Zynq
(Zybo Z7-20)
	`cd {G:/wa/fpga-serial-mem-tester-3/SF-Tester-Design-Zynq/Work_Dir/}`
	`source ./init_project_SF-Tester-Zynq.tcl`
	`cd {G:/wa/fpga-serial-mem-tester-3/SF-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.