Experiment 0
Self-Test
on HaHa v3.0 Board

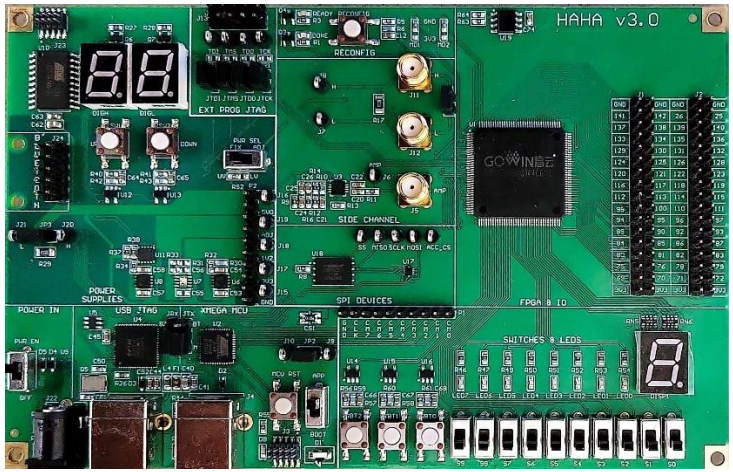
In this experiment, you will go through a complete test of the Hardware Hacking (HaHa) v3.0 Board.

Instructor: Dr. Swarup Bhunia
Co-Instructor: Reiner Dizon-Paradis

Experiment Set-up: Configuration

The hardware and software needed for this experiment include:

1. The HaHa v3.0 Board
2. A USB A-to-B cable
3. An oscilloscope or Analog Discovery (AD) Board
    a. *Optional* – Waveforms 2015 to view AD board signals [6].



Instructions and Questions

NOTE: If you have any questions for parts II-V, please refer to the FAQs at the end of this document or the HaHa Board Manual.

PART I: Test the Power Supply and Voltage Regulators

1. Connect the HaHa v3.0 board to the USB Power supply (J14) with a USB A-to-B cable to your Windows machine. Turn on the power to see if the power light is on.

2. Using an oscilloscope or Analog Discovery board, check the voltages in the pins along header (P2) on the board. There should be ten pins on this header along with five test pins/points to the right. Starting from the top pin:
    a. Pins 1-2 probe the input voltage.
    b. Pins 3-4 output the adjustable voltage. Use button SW1 and SW2 to change the voltage.
        i. Answer this question: what is the adjustable voltage range?
        ii. Take a picture showing the minimum voltage of the 3.3A test point.
    c. Pins 7-8 have a constant 3.3 V output.
    d. Pins 9-10 are ground pins.

NOTE: After doing voltage measurement, remember to set "PWR SEL" to "FIX" to make sure you can program both chips successfully.

PART II: Test the FPGA and the Peripherals

1. Make sure the HaHa v3.0 board is still connected to the USB Power supply (J14) Open GOWIN FPGA Designer and navigate to **Tools > Programmer** in the menu bar. The *Programmer* window will appear with another window called *Cable Setting*. Make sure the *Port* is set to "Gowin USB Cable(FT2CH)/0/…" Click on the **Find Device** button (first button below menu bar that has a magnifying glass). Double click on **GW1N-9C**. A new entry will now show up on the first row. Double click on *Read Device Codes* and change the following settings and then Press **Save**:
    a. Access Mode = *Embedded Flash Mode*
    b. Operation = *embFlash Erase,Program,Verify*
    c. File name = <location/of/self_test_fpga_haha3.fs/file>

    You are now ready to program the board. Navigate to **Edit > Program/Configure**. Wait until the programming is complete. For more information, you can refer to the *GOWIN FPGA Guide for HaHa v3.0 Board* document.

    If this is successful, the 8 LEDs on the board should be blinking, and the 7-segment display a hexadecimal digit counting from 0 to F. To check the function of the 10 user switches, just switch any of them and you will see the 8 LEDs will change their pattern of blinking. There are two different patterns of blinking. Whenever any of the switches are flipped, the pattern should change. If not, please let the instructor or TAs know.

    To check the function of the pushbuttons, just push any of them, and you should see the 7-segment display stops counting and start to repeat blinking with a fixed number. If not, please let the instructor or TAs know.

    a. **Answer these questions**:
        a. What are the two LED patterns occurring in the board?
        b. Which patterns of switches trigger either of the LED patterns?
    b. **Answer this question**: What happens to the digit on the 7-segment display when you keep pressing any of the key buttons?
    c. **Deliverable**: Attach a picture of the board with one of the LED patterns and a hexadecimal digit displayed in 7-segment display.

PART III: Test the Microcontroller and the Accelerometer

1. Connect the HaHa v3.0 board to the USB Power supply (J4) with a USB A-to-B cable to your Windows machine. In the HaHa v3.0 board, place the following switch (see yellow rectangle below) to *BOOT* (switch down). Double tap **MCU RST** button.

Follow the instructions in the Linux installation guide (in the docs folder) to program the provided file (U_ACC.hex).

Read the *XMega Microcontroller Guide for HaHa v3.0 Board* document for instructions in programming the microcontroller.

To run the programmed executable, move the switch from BOOT to APP and press the MCU RST button a couple times.

The microcontroller will read the acceleration data from the accelerometer and send it to the FPGA. The FPGA will show the value in binary on the eight LEDs if you place the most significant four switches to "1001". Please rotate the board and see if the values are showing on the LEDs are changing. NOTE: You may need to tap MCU RST button multiple times before the LEDs show the correct value and change as you move the board.

a. **Deliverable**: Attach a photo of the board showing the acceleration value on the LEDs in your report.

Open GOWIN FPGA Designer. Go to **File > New**. Select **FPGA Design Project**. Name the project *haha3_self_test* with the directory of your choosing. In the device selection window, use the following filters:

- Series: *GW1N*
- Device: *GW1N-9*
- Device Version: *C*
- Package: *LQFP144*
- Speed: *C6/I5*

Select the Part Number: **GW1N-UV9LQ144C6/I5**. The project creation process is just needed to unlock the GOWIN Analyzer Oscilloscope (GAO) tool. Go to **Tools > Gowin Analyzer Oscilloscope**. Press the folder button. Navigate to the location of the provided file (self_test_fpga_haha3.rao) and click **Open**. Check **Enable Programmer**. Press the button with the magnifying glass. Select *GW1N-9C*. Check the box below *Enable*. Under Fs File, click the empty field. Navigate to the location of the provided file (self_test_fpga_haha3.fs). Press the small green play button (under *Enable Programmer*) to program the FPGA. After successfully programming, press the **F2 key**. You should see the two signals.

a. **Deliverable**: Attach a screenshot of the GOWIN Analyzer Oscilloscope window showing the aforementioned signals.
b. **Answer the following**:
   i. What is the first signal in the oscilloscope? Where is this signal coming from on the board?
   ii. What is the second signal in the oscilloscope? Where is this signal coming from on the board?

PART IV: Test the Temperature Sensor

1. Finish Part III before proceeding. Refer to Part III for microcontroller programming steps. Program the microcontroller with the provided file (U_TEMP.hex). The microcontroller will read the data from its internal temperature sensor and send it to the FPGA. Put switches S9 – S6 to "1001" to see the values on the LEDs.

    a. **Deliverable**: Attach a photo of the board showing the microcontroller temperature value in your report.
    b. **Deliverable**: Attach a screenshot of the GOWIN Analyzer Oscilloscope window showing the aforementioned signals.

PART V: Test the Flash Memory

1. Finish Part III before proceeding. Refer to Part III for microcontroller programming steps. Program the microcontroller with the provided file (U_FLASH.hex). The microcontroller will read a signature of the EEPROM and send it to the FPGA. Put switches S9 – S6 to "1001" to see the values on the LEDs. The LEDs should be showing 00101001 (0x29).

    a. **Deliverable**: Attach a photo of the board showing the value 0x29 on the LEDs in your report.
    b. **Deliverable**: Attach a screenshot of the GOWIN Analyzer Oscilloscope window, showing 0x29 for the second signal.

References

[1] https://cdn.gowinsemi.com.cn/Gowin_V1.9.8.11_win.zip

[2] https://www.gowinsemi.com/en/support/license/

[3] https://www.gowinsemi.com/en/support/download_eda/

[4] https://ww1.microchip.com/downloads/aemDocuments/documents/DEV/ProductDocuments/SoftwareTools/as-installer-7.0.2594-full.exe

[5] https://ww1.microchip.com/downloads/aemDocuments/documents/DEV/ProductDocuments/SoftwareTools/JRE-Flip-Installer-3.4.7.112.exe

[6] https://mautic.digilentinc.com/asset/110:waveforms-windows-64-bit-download

References

[1] https://www.gowinsemi.com/en/support/license/

[2] https://cdn.gowinsemi.com.cn/Gowin_V1.9.8.11_win.zip

[3] https://www.gowinsemi.com/en/support/download_eda/

[4] https://ww1.microchip.com/downloads/aemDocuments/documents/DEV/ProductDocuments/SoftwareTools/as-installer-7.0.2594-full.exe

[5] https://www.microchip.com/en-us/tools-resources/develop/microchip-studio

[6] https://ww1.microchip.com/downloads/aemDocuments/documents/DEV/ProductDocuments/SoftwareTools/JRE-Flip-Installer-3.4.7.112.exe

[7] https://www.microchip.com/en-us/development-tool/flip

[8] https://mautic.digilentinc.com/asset/110:waveforms-windows-64-bit-download

[9] https://mautic.digilentinc.com/waveforms-download

Frequently Asked Questions

FAQ about the Microcontroller

1. Q: After I program the Microcontroller and turn off the board, will the program remain in the Microcontroller?
   A: Yes.

2. Q: Why is my program for the Microcontroller not working as soon as I succeeded in programming it?
   A: Your program will not be running until you reboot the board.

3. Q: Why cannot I program the Microcontroller?
   A: You could have made these mistakes:
      You are using the wrong USB port. You must use J4, instead of J14.
      Or, you turned the switch next to MCU RST button to APP.

4. Q: Why is the Microcontroller not running the code I programmed after I reboot the board?
   A: To run the code, you should:
      a. Turn the switch next to MCU RST button to APP. Then, press the MCU RST button a couple of times.

5. Q: Why cannot I re-program the Microcontroller? It seems it is always running the code I programmed last time.
   A: To re-program the chip, you should:
      a. Turn the switch next to MCU RST button to BOOT.
      b. Press the MCU RST button a couple of times.

FAQ about the FPGA

1. Q: Which version of GOWIN FPGA Installer should I install to program the FPGA?
   A: Version 1.9.8 or higher. You can download it from GOWIN website.

2. Q: I created the bitstream file, but why I cannot program with it?
   A: You should make sure you chose the correct device part number when you were building the project. You can find the number on the marking of the chip.

3. Q: After I program the FPGA and turn off the board, will the program remain in the FPGA?
   A: It depends. If you programmed it to the SRAM, it would be lost after the power is off. If it is programmed to the embedded Flash, it would stay. Refer to the next answer for more details.

4. Q: Which kind of file is recommended to program the FPGA?
   A: GOWIN only has the FS file to program the FPGA. The GOWIN Programmer has settings that can send the bitstream either to SRAM (volatile) or embedded Flash (non-volatile, survives power down). You should only program to the embedded Flash if you are 100% sure your design will do no harm to the board. Each of the I/O ports of your design should be assigned to a pin number. It is highly recommended to use the gowin_pin_assignments.cst file to help assign the FPGA pins correctly.

5. Q: Why cannot I see any output from the FPGA?
   A: You may forget to assign signals to the pins.

6. Q: Why am I getting the following error?
   `Directory "<some/directory/to/file>" has null character.`
   A: The directory or path contains a space or dashes, which can stop any of the processes. Please move your project to conform to this naming standard.

FAQ about the Experiments

1. Q: For a variable in Verilog, where is the LSB?
   A: LSB means least significant bit. No matter how the variables are declared, the LSB is always on the right side. For example, the LSB for aaa [1:8] is aaa[8], and the LSB for bbb [63:0] is bbb[0].

2. Q: When I am trying to see the power up values in the SRAM of the Microcontroller, I find every time they stay the same. Why?
   A: You have to totally power off the board before you power up. Plug out both the USB cables.

3. Q: Why I cannot write or read from the EEPROM on the board.
   A: You may forget to configure the 'hold' pin of the EEPROM. Please refer to its datasheet for details.

4. Q: In the bus snooping experiment, why do I not see any signal in the Waveforms software?
   A: You may try zooming in with a higher time base.

5. Q: Why is the ADC on the Microcontroller not working well?
   A: Make sure the ADC is set up correctly on the Microcontroller. Refer to its datasheet for more information.

6. Q: Do I need to use the FPGA to implement the ADC?
   A: You do not. This is done in the microcontroller and then sent to the FPGA via the chip interconnection if needed.