RISC-V Secure Hardware Video Output

ECE Capstone

Sponsored by: Dan Zimmerman and Michal Podhradsky, Galois Inc., galois.com

Background and Motivation

Galois is participating[1] in the DARPA MTO SSITH program, and has developed novel RISC-V systems-on-chip (SoCs) with certain security properties. The SoCs run Linux and FreeBSD on a Xilinx VCU118 FPGA development board, and developers are able to interact with and test them via Ethernet and UART connections. However, the SoCs cannot currently drive displays, which makes it difficult to develop compelling demonstrations of the technology.

The hardware runs too slowly (approximately 100MHz) to support a full-fledged graphical user interface as found in modern Linux and FreeBSD distributions; however, even a simple graphical interface (whether text-based or generated by directly pushing images to a framebuffer) would enable us to demonstrate a much wider range of applications for secure hardware. Thus, we would like to add video output capability—which may require both hardware and software changes—to the SoCs.

Milestones

Our goals with this capstone are as follows:

1. **Select the best approach to add video output capability.** In our estimation, there are three options to choose from:
    a. Use VCU118's PMOD connector with VGA adapter
    b. Use VCU118's PMOD connector with DVI adapter
    c. Use VCU118's PCIe expansion and a regular graphics card
2. **Modify Xilinx Vivado project to enable video hardware.** Depending on the chosen option, it might be necessary to modify the SoC hardware itself to add the video capability. This might range from a simple tweak in the design to the addition of more complex IP blocks.
3. **Modify Linux/FreeBSD drivers to enable video output.** Once hardware changes are made, it will also be necessary to modify the software stack. If the video output is

[1] Project page: https://galois.com/project/besspin/

correctly implemented in the hardware, software changes should be minimal as Linux and FreeBSD both support generic frame buffers.

4. **Demonstrate video output on Linux/FreeBSD.** The last step is to demonstrate new video output capabilities - either showing an image on a small screen, showing a live terminal window, or even using something such as Qt for embedded Linux.[2]

Learning Outcomes

The team will gain expertise in the following:

- Hardware design of RISC-V processors and SoCs
- Hardware and software involved in video output on embedded platforms
- Linux/FreeBSD drivers
- Embedded video and graphics

Student skills

Previous experience with FPGAs, SoC design, and video preferred. Some knowledge of Linux/FreeBSD will be useful as well.

The work will be closed-source, and signing a Non-Disclosure Agreement (NDA) will be required.

[2] https://doc.qt.io/qt-5/embedded-linux.html