Bachelor-Thesis Idea: P2P-Network on FPGAs

Elena Frank

December 2020

My basic idea for this bachelor thesis is the development of a peer-to-peer network on FPGAs, since this is something that I am personally interested in and after my first quick search, I did not really find existing open-source projects for this. FPGAs are quite useful for the IoT because they are general-purposed and can easily be reprogrammed, but I think the peer-to-peer network idea is something theoretical and its purpose is research rather than being useful for real-world applications.

1 Hardware, Firmware, Tools, Libraries, etc.

For easier prototyping and debugging, and since a Peer-to-Peer-Network obviously requires a network connection, I want to combine the FPGA with an Raspberry Pi. It would also have been a option to directly buy the FPGA with a board like the Spartan-7 SP701 FPGA Evaluation Kit, but I personally prefer the benefit of broad documentation, many projects, and open source software that can be used with Raspberry Pis.
After briefly considering the ZynqBerry, which is a Xilinx Zynq FPGA Soc based Raspberry Pi on one board, I decided to use the IOTA ICCFPGA instead, where the dev board with the FPGA has a can be used as a hat for a Raspberry Pi 3 or 4 and use JTAG, UART and Ethernet over the RPi. Even though the Zynqberry itself is quite an interesting project, there is not too much documentation about it and I want to avoid that the hardware ends up consuming all the time before I even get to the actual project. I am aware though that this can also happen with the ICCFPGA, but the chances are much lower since the ICCFPGA project itself already uses the VexRiscV Soft-CPU that I also plan to use. This VexRiscV Project provides a RISC-V Soft-CPU implementation for FPGAs written in SpinalHDL with a plugin-architecture and claims to not use any vendor specific IP block / primitive.
As already mentioned, the ICCFPGA board project was developed by the IOTA Foundation and will have an important role in my thesis because I am using it as a basis for my work. It already provides, among others, the following tools:

- VexRiscV firmware that can be uploaded to the FPGA with a provided script
- Raspberry Pi 3/4 compatible connector
- Debugging tools like the OpenOCD debugger for VexRiscV, serial terminal, virtual cable server, ...
- Jumpers to configure the board e.g. to select TXD and RXD from RISC-V JTAG, USB, Raspberry Pi GPIO or pin-header J5

The ICCFPGA project itself with its FPGA CryptoCore actually has a security focus and was designed for *"applications that need fast, dedicated proof of work and a secure memory"*, but in my first steps I won't make use of the security features.
For the P2P Network itself, I am planning to program it in Rust and use different libraries, depending whether no_std support is necessary or not. Since VexRiscV provides among other plugins also the option to run linux, I hope I will be able to use the Rust std lib and thus can use Libp2p.
Libp2p is a peer-to-peer framework that was developed by the IPFS community and I already worked with rust-libp2p, my first draft of the communication feature within the IOTA Stronghold project is something that I might also use for this thesis.
Regarding the whole technology, this thesis focuses more on deploying, modifying and combining different existing solutions, instead of writing completely new software. It is important for me to note, that everything that I develop and publish within any IOTA project, even if it is on the subject of a peer-to-peer-network, will not count as part of my work for the thesis since it is property of IOTA and I can only make use of it, but not claim it as part of my thesis.

2 Planned Steps and Milestones

The following steps are roughly what I have planned, but for each milestone there is the chance that it could either be quickly solved, or end up taking much more time then expected. Therefore the focus of my thesis can shift within these milestones depending on how difficult they turn out to be.

1. Getting Started with Xilinx Spartan-7 FPGA:
   (a) Research the FPGA itself and existing projects
   (b) Connect to the FPGA with the proper tools from Xilinx, maybe write very simple test program
2. Run VexRiscV on the FPGA
   (a) Follow the ICCFPGA quickstart Guide and scripts
   (b) Research the VexRiscV Project itself and especially the provided plugins
   (c) Extract the VexRiscV part of the ICCFPGA and modify it if necessary
3. Implement the peer-to-peer network with Rust
   (a) Load and run a test rust program on the FPGA
   (b) Research useful Rust crates, if necessary with no_std support
   (c) Implement the p2p-communication between peers that can be reached directly

Possible extensions:

1. P2p-communication between peers behind a firewall/ NAT traversal, i.e. by using a relay peer
2. Utilize the ICCFPGA security features for e.g. encryption of the communication
3. Connect with an E-Ink device to display incoming messages / pictures
4. Deploy and test the p2p-network on actual Risc-V microcontroller

3 Challenges and Alternatives

As already mentioned, each step can create major Problems; the following list only briefly touches a few ones that I consider critical at this point in time.

- The documentation on the Spartan-7 FPGA might be almost useless because I am not using any official development board. This is not necessarily a problem, but would mean that I heavily depend on the ICC-FPGA Project
- The VexRiscV part can not easily be separated from the ICCFPGA:
  This would shift my focus more onto VexRiscV itself, its plugins and how to deploy it on the FPGA. Since ICCFPGA rather documents the CryptoCore and not so much its implementation of VexRiscV, I might have to "reinvent the weel" and/ or get stuck.
- The VexRiscV Soft-CPU requires no_std for Rust, but Libp2p does not support this. This would either mean that I would have to use a different library with considerably less features/ provided protocols, or that I would have to write a pull-request to rust-libp2p that modifies the project to enable no_std, but this would be a lot of work and only possible, if the preceding steps did not consume much time.

In general, the time is probably the most critical factor and my first approach will be to use as much existing solutions as possible and later on reduce it to what's really necessary.

4 Useful other Projects

- ICCFPGA: `https://gitlab.com/iccfpga-rv/iccfpga-manual/-/blob/master/iccfpga.pdf`
- Rust on VexRiscV: `https://craigjb.com/2020/01/22/ecp5/`
- Linux on VexRiscV: `https://github.com/litex-hub/linux-on-litex-vexriscv`
- Rust-libp2p examples: `https://github.com/libp2p/rust-libp2p/tree/master/examples`