Contribution ID: **56** Type: **not specified**

BoF: upstream drivers for open source FPGA SoC peripherals

Tuesday 25 August 2020 07:00 (45 minutes)

There are active open source projects such as LiteX which have developed IP (e.g. chip-level hardware design) needed for building an open source SoC. The common workflow is that this SoC would be synthesized into a bitstream and loaded into a FPGA. (Aside: there is also the possibility of using these IP modules in an ASIC, but the scenario of supporting fixed-in-silicon hardware peripherals is already well-established in Linux).

The scenario of an open source SoC in a FPGA raises a question:

What is the best trade-off between complexity in the hardware peripheral IP and the software drivers?

Open source SoC design is done in a Hardware Description Language (HDL) with Verilog, VHDL, SystemVerilog or even newer languages (Chisel, SpinalHDL, Migen). This means we have the source and toolchain necessary to regenerate the design.

LiteX [1] is a good example of an open source SoC framework where it provides IP for common peripherals like DRAM controller, Ethernet, PCIe, SATA, SD Card, Video and more. A key design decision for these peripherals are Control and Status Registers (CSR). The hardware design and the software drivers must agree on the structure of these CSRs.

The Linux kernel drivers for LiteX are currently being developed out-of-tree [2]. A sub-project called Linux-on-LiteX-Vexriscv [3] combines the Vexrisv core (32-bit RISC-V), LiteX modules, and a build system which results in a FPGA bitstream, kernel and rootfs.

There is an long-running effort led by Mateusz Holenko of Antmicro to land the LiteX drivers upstream starting with the LiteX SoC controller and LiteUART serial driver [4]. Recently, support for Microwatt, a POWER-based core from IBM, was been added to LiteX and Benjamin Herrenschmidt has rekindled discussion [5] of how best structure the LiteX CSRs and driver code for upstream. In addition, an experienced Linux graphics developer, Marin Perens, has jumped into the scene with a LiteDIP [6]: "Plug-and-play LiteX-based IP blocks enabling the creation of generic Linux drivers. Design your FPGA-based SoC with them and get a (potentially upstream-able) driver for it instantly!"

Martin has blog posts that dives further into the issues I've tried to describe above: "FPGA: Why So Few Open Source Drivers for Open Hardware?" [7]

I think this BoF will be useful in accelerating the discussion that is happening on different mailing lists and hopefully bringing us closer to consensus.

[1] https://github.com/enjoy-digital/litex
[2] https://github.com/litex-hub/linux/commits/litex-vexriscv-rebase/drivers
[3] https://github.com/enjoy-digital/litex
[4] https://lkml.org/lkml/2020/6/4/303
[5] https://groups.google.com/d/msg/linux-litex/fJLlcsuBibY/3vP8_7nGAwAJ
[6] https://gitlab.freedesktop.org/mupuf/litedip/
[7] https://mupuf.org/blog/2020/06/09/FPGA-why-so-few-drivers/

I agree to abide by the anti-harassment policy

I agree

Primary author: Mr FUSTINI, Drew (BeagleBoard.org Foundation)

Presenter: Mr FUSTINI, Drew (BeagleBoard.org Foundation)

Session Classification: BOFs Session

Track Classification: Birds of a Feather (BoF)