BoF: upstream drivers for open source FPGA SoC peripherals

- open source projects such as LiteX which have developed IP (e.g. chip-level hardware design) needed for building an open source SoC
- common workflow is that this SoC would be synthesized into a bitstream and loaded into a FPGA
- 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.

BoF: upstream drivers for open source FPGA SoC peripherals

- 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.

- 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.

BoF: upstream drivers for open source FPGA SoC peripherals

- Linux-on-LiteX-Vexriscv [2] combines the Vexrisv core (32-bit RISC-V), LiteX modules, and a build system which results in a FPGA bitstream, kernel and rootfs
- Linux kernel drivers for LiteX are currently being developed out-of-tree [3]
- LiteX SoC driver and LiteUART driver patch series from Mateusz Holenko of AntMicro is currently in v10 [4]

BoF: upstream drivers for open source FPGA SoC peripherals

- Earlier this year, support for Microwatt, a POWER-based core from IBM, was been added to LiteX by Ben Herrenschmidt
- Ben started a linux-litex thread of how best structure the LiteX CSRs and driver code for upstream [5]
- **Issue:** should drivers use CSR accessors or normal MMIO?
- **Issue:** should Linux drivers support 8-bit CSR or just 32-bit?
- **Issue:** how to describe what exists? Device tree overlays?

BoF: upstream drivers for open source FPGA SoC peripherals

- Marin Perens is developing 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 wrote a blog post: "FPGA: Why So Few Open Source Drivers for Open Hardware?" [7]