Linux device driver for a RISC-V System-on-a-Chip (SoC)

June 2020

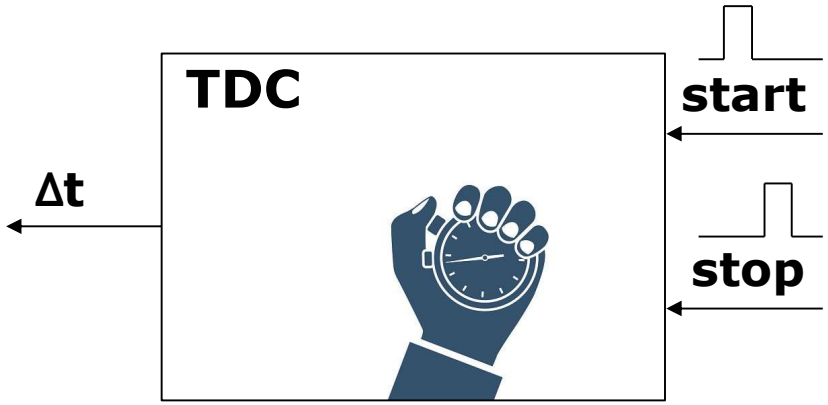
Felix Arnold | felix.arnold@gmail.com

Agenda

- Goals
- System
  - Hardware
  - FPGA / CPU
  - Software
- TDC
  - FPGA Module
  - Device Driver
- Tools
- Demonstration
- Conclusions

Goals

- Learn about
  - open-source SoC FPGA tools
  - embedded Linux
- Build
  - a simple Linux device driver for time-to-digital converter (TDC) core

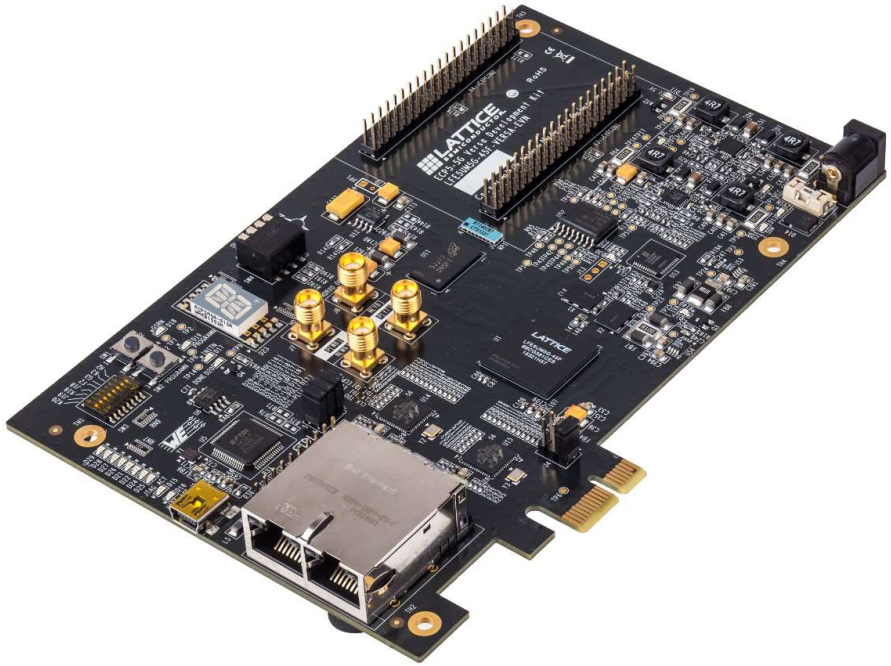


Agenda

- Goals
- System
  - Hardware
  - FPGA / CPU
  - Software
- TDC
  - FPGA Module
  - Device Driver
- Tools
- Demonstration
- Conclusions

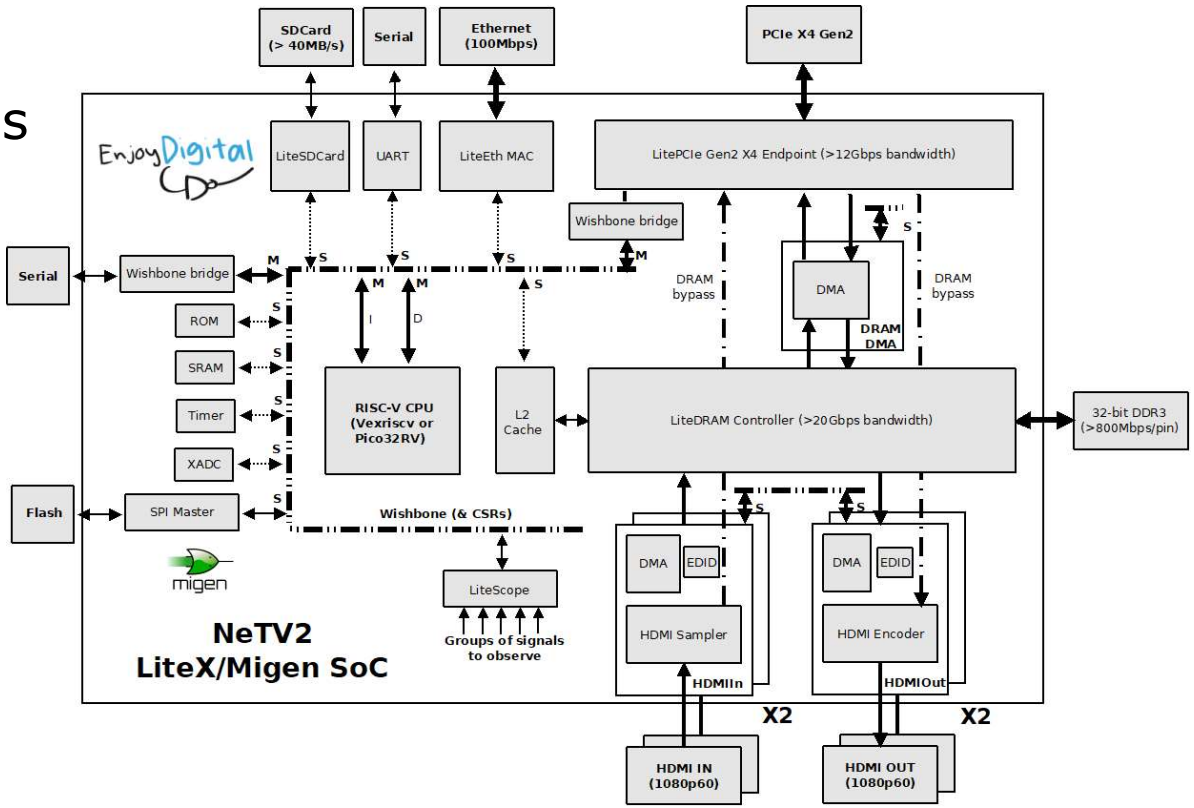
Hardware

- Versa Development Kit
  - Lattice ECP5 FPGA
  - SDRAM
  - UART/JTAG (USB FTDI)
  - Ethernet



FPGA SoC

- Litex Soc Builder
  - Written in Python
  - Configures SoC (CPU, Cores, IOs, etc)
  - Open source synthesis tools
- TDC core is added to wishbone bus
- CPU running at 75 MHz



CPU

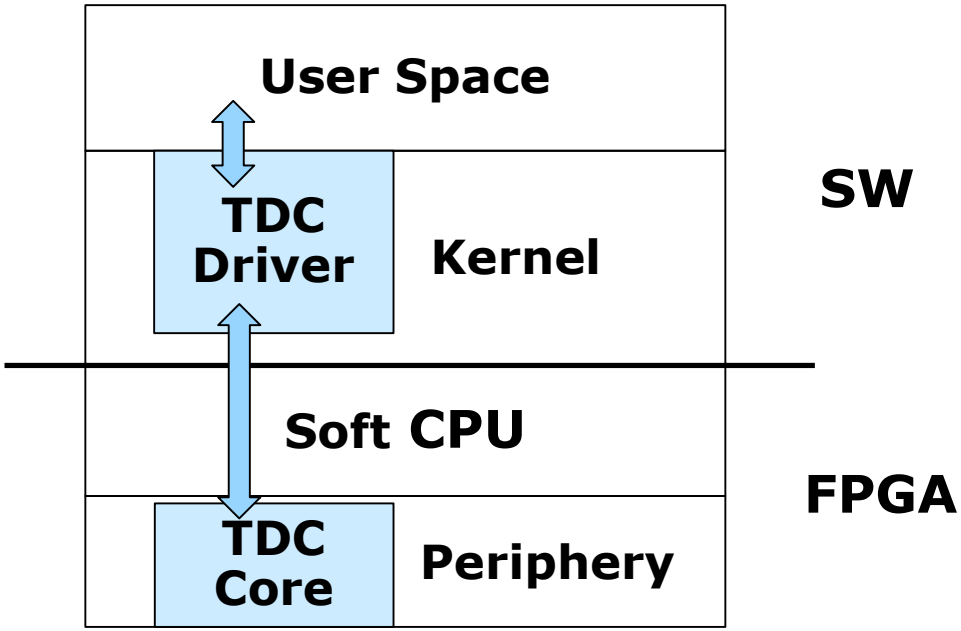
- VexRiscV processor
- RISC-V architecture
- Soft Core
- Highly configurable
- Linux capable
- Written in SpinalHDL language

Agenda

- Goals
- System
  - Hardware
  - FPGA / CPU
- TDC
  - FPGA Module
  - Device Driver
- Tools
- Demonstration
- Conclusions

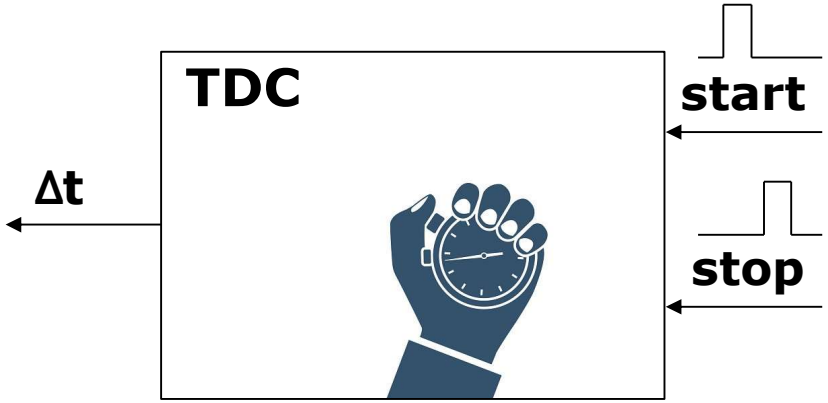
SW/FPGA Stack

- Blue: implemented within this project



TDC FPGA Core

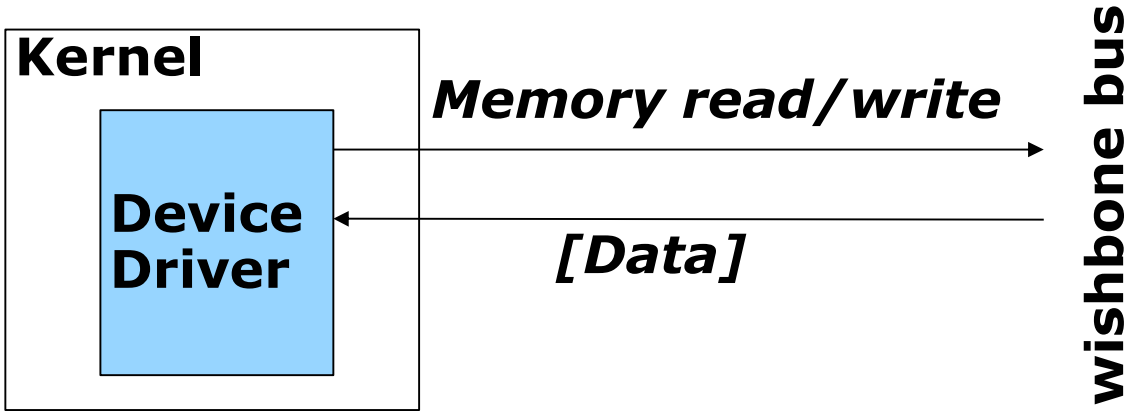
- Measures time between two events
- Implemented as a counter
- 75MHz → 13.3ns resolution
- Written in Python! (Migen)



- *Shortcut:*
  start & stop signals come from device driver,
  not from external inputs!

TDC Device Driver

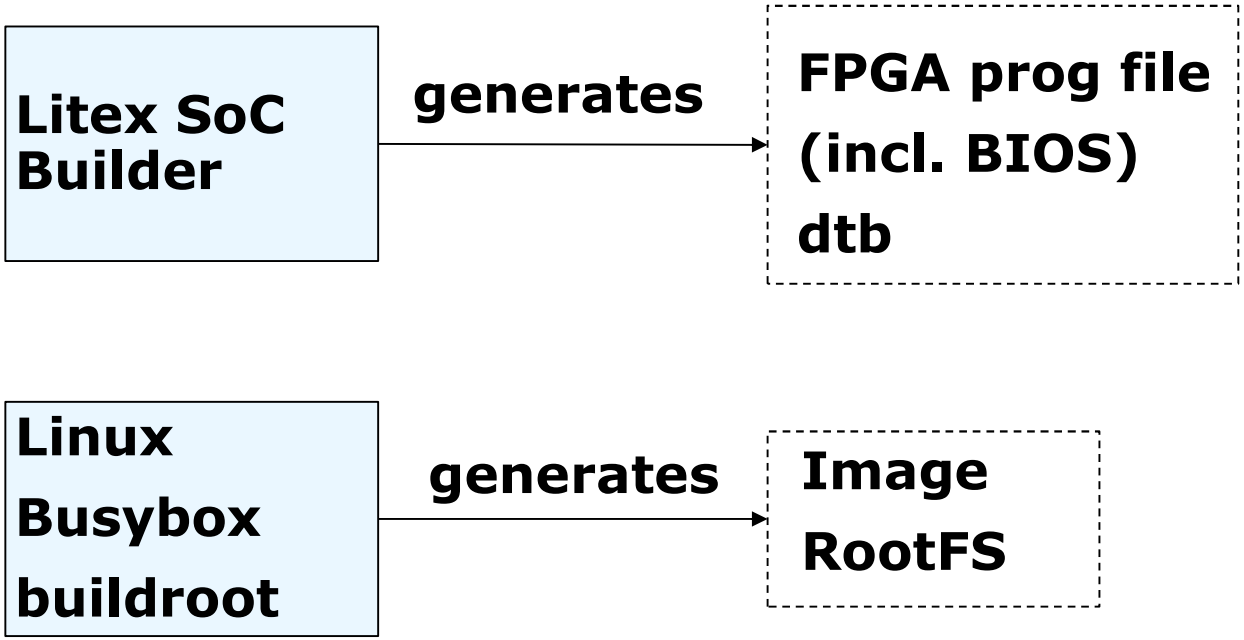
- Implemented as a dynamic kernel module
- Character device: `/dev/mod_tdc`
- Interfaces to TDC FPGA core via system bus



Agenda

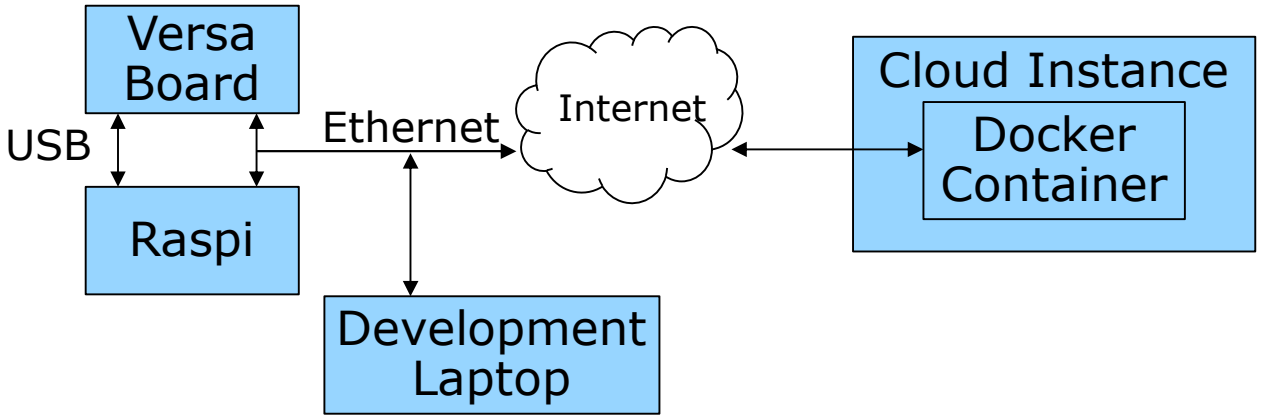
- Goals
- System
  - Hardware
  - FPGA / CPU
  - Software
- TDC
  - FPGA Module
  - Device Driver
- Tools
- Demonstration
- Conclusions

Generation Flow



Development Setup

- All tools are in docker container https://github.com/plex1/TrellisDev.git
  - Easy to be replicate
  - Compilation of Image takes multiple hours
  - Requires 50GB + of disk space
- Raspberry Pi for interfacing to HW: USB/JTAG & Ethernet/TFTP
- Complex setup



Conclusion

- Open source SoC with Linux running
  - Docker Container created

- Simple TDC FPGA core implemented
- Simple Linux device driver implemented

- Learned about Linux ecosystem
- Great open source tools for FPGA SoC available

Demonstration

```
root@buildroot:~# insmod mod_tdc.ko
[ 5548.627714] mod_tdc: Loading tdc module V1...
[ 5548.628412] mod_tdc: Hello universe!
[ 5548.629148] mod_tdc: read from test reg 0 ->          0
[ 5548.629939] mod_tdc: read from test reg 1 ->          1
[ 5548.631612] mod_tdc: read from test reg as uint32 -> 65538
root@buildroot:~# mknod -m 666 /dev/mod_tdc c 250 0
root@buildroot:~# echo "start" > /dev/mod_tdc
[ 5548.632606] mod_tdc: driver module is ready to be used.
[ 5574.475942] mod_tdc: Start pulse received
root@buildroot:~# echo "stop" > /dev/mod_tdc
[ 5576.897541] mod_tdc: Stop pulse received
root@buildroot:~# cat /dev/mod_tdc | head -n 1
Measured time: 181574632 clock cyles, this corresponds to 2420 ms
root@buildroot:~# _
```

Links

- This project:
  - https://github.com/plex1/tdc_kernel_driver
  - https://github.com/plex1/TrellisDev
- Referenced repos:
  - https://github.com/litex-hub/linux-on-litex-vexriscv
  - https://github.com/enjoy-digital/litex
  - https://github.com/m-labs/migen
  - https://github.com/SpinalHDL/VexRiscv