2nd SoC workshop CERN - Minutes – June 8, 2021 – Projects1

https://indico.cern.ch/event/996093

Notes:

- These minutes were collected by J. Hegeman.
- They provide a record of the questions and answers following the presentations. For the presentations themselves including recordings, please, refer to the indico timetable.

Tuesday – June, 8th:

- T. Mkrtchyan, SoC-based Monitoring and Configuration of the ATLAS L1Calo TREX Module:
    - Q: You showed a fallback boot solution in case the SD card gets corrupted. How do you detect a failed boot?
      A: There are no active checks to verify successful boot. The main indicator of boot failure is the absence of ssh connectivity with the board. One could check for activity in the PS via VME access to the PL. This is not implemented at the moment.
    - Q: You are using Python3, with /dev/mem, isn't it? Does that provide sufficient performance?
      A: Yes, all monitoring software uses Python3 via /dev/mem. The slow-control is slow enough that this does not present a bottleneck. Some indications of a performance bottleneck have been observed when using the Xilinx Virtual Cable (which is written in C). This is slower than a Digilent FTDI connection.
    - Q: On slide 13, you mention that you tested booting from nfs. In the end you chose a local boot approach. What was the main problem/obstacle you faced with nfs booting?
      A: We want the board to immediately start sending environmental monitoring data to the DCS system. In case of network problems, network booting could be obstructed/delayed. This is avoided when using local boot. The main issue/worry is not so much technical as it is about the reliability of the network boot procedure.
- M. G. D. Gololo, SoC Developments for the Detector Control System of ATLAS Tile Calorimeter at the HL-LHC:
    - Q: More a comment than a question. Some of the comments in this presentation referring to XADC and QUASAR are not completely accurate. Some of these points may be addressed in more detail by Paris Moschovakos in his presentation later.
      A: Thank you for the feedback. I will update the slides where appropriate.
    - Q: Is the idea to, in the end, have one or several OPC-UA servers, and to always use Ironman for concurrent IPBus access?
      A: Yes. The idea is to have two OPC-UA servers: one for the generic XADC access and one for TileCal-specific tasks, like reading the CPMs. All running concurrently using Ironman.

- E. A. Smith, The Zynq MPSoC in the gFEX Hardware Trigger in ATLAS:
    - Q: Can Pynq and Yocto run in parallel?
      A: Yes. There is a Xilinx recipe that includes the Pynq libraries in Yocto. There are specific 'Pynq boards' for which the overlays arr already provided. For a custom board one simply has to provide the overlays and then include the necessary libraries.
    - Q: Does the Pynq case also work with Jupyter Notebook?
      A: Yes. We have not tried this with our custom hardware, but it does work with the Pynq boards. This should be no problem.
    - Q: How much of the Zynq PS is used?
      A: We have not done any CPU usage measurements recently. When we last did, the CPU usage was low.
    - Q: How large is the gFEX group?
      A: Five institutes are involved in this project, including physicists, engineers, and students.
- P. Moschovakos, SoCs for Detector Controls and their Applications:
    - Q: How much is the overhead of qemu compared to running native?
      A: Roughly five times slower than native (x86). The good point, though, is that this penalty is not so important for things like nightlies and gitlab access.
- Z. Xu, SoC-based DAQ for the Pixel Readout Chip RD53a/b:
    - Q: How did you aproach the performance optimisation for SIMD? Do automated tools exist, or was this done manually?
      A: Our data format is very straightforward, so we implemented this manually.
- I. Mostafanzhad (Nalu Scientific LLC), Waveform Digitizing and Processing Front-end Microelectronics for large Experiments:
    - Q: When you did the integration of the Aardvarc, you do have another FPGA+SoC in the middle, that does the aggregation of the data, and then you send the results out over 10GbE, right?
      A: Yes. The FPGA is essentially pass-through in this case.
    - Q: You implemented the SoC directly on the chip. What architecture is this?
      A: We implemented our own, homebrew, basic processor. This runs basic bare-metal applications written in C. It does not run/require an OS.
    - Q: What does the SoC do exactly? It gathers the data, formats them, and sends them out, right?
      A: To first order yes. We also benefit from the fast on-chip communication with the electronics for mundane tasks like calibration. It avoids having to ship all data out of the chip via massive parallel buses and then process them in an external FPGA.