Development of an AMC Module Management Controller

Peter Kämmerling, Matthias Drochner, Harald Kleines, Stefan van Waassen, Michael Ramm, Axel Ackens

June 2012 Forschungszentrum Jülich, Germany

THE MMC (Module Management Controller) of an AMC (Advanced Mezzanine Card) communicates with the Carrier IPMC (Intelligent Platform Management Controller) and the ShMC (Shelf Management Controller) of an ATCA crate (Advanced Telecommunications Computing Architecture). The MMC handles the hardware of the AMC and communicates to the Carrier IPMC by the management bus (IPMB). The MMC handles registration in and out to the shelf-management, inventory, payload, status indications and optional internal measurements like temperature, voltage and current. The AMCs fit into ATCA blades (carrier card) or into µTCA crates.

MMC

- microchip PIC32MX460
- MIPI-CSI2 M4k-Core 80MHz 125DMIPS
- 70 variants with 28 36 44 64 100 pins
- very fast I/O, e.g., for JTAG-Master + SelectMAP for Xilinx FPGAs
- in-circuit-debugger and -programmer, JTAG-interface
- recently introduced, promises long family duration

coreIPM Firmware

- baseboard management controller (BMC) software
- open source
- runs on microcontroller on field replaceable unit (FRU)
- for ATCA carrier boards and µTCA boards
- modified for our µTCA TDC board

Experience

- assembly of the FPGA to PCB, switch on payload and further hardware testing
- tests with the ipmi_tool based test framework developed in FZJ / ZEL
- MC is fast enough for JTAG-master or Xilinx-SelectMAP
- extract IPMI protocol machine completely to an extra library to advance modularization

AMC developed by FZJ ZEL

- 4x GPX ASIC TDC
- Virtex 5
- PCIe 4 lanes
- MMC
- partially tested
- design in revision

Toolchain

- gcc-toolchain
- IDE under WIN, Linux, MOX
- Standard-C-Libraries
- DSP-Libraries
- I/O-Libraries
  - Stack-Libraries:
    - USB, CAN, TCP/IP
    - audio / speech
    - WiFi, ZigBee

Outlook

- hardware-related code was closely bound to the characteristics of the original MC, hence we extracted it into a library
- we implemented extensions like a reliable I2C-stack
- further modularization of the code:
  - board hardware related lib
  - microcontroller related libs: timers, interrupts, stacks for peripherals
  - protocol lib: I2C, IPMI