CAN FD open-source IP core

The Faculty of Electrical Engineering (FEE) at Czech Technical University in Prague (CTU) reached another milestone in July 2022. Their CAN FD IP offer is fully supported by a mainline Linux kernel. According drivers, emulation, and applications are available.

(Source: Adobe Stock)

Now, the CTU FEE team offers CAN FD solutions fully supported by the mainline Linux kernel version 5.19 after four years of the out-of-tree support. The VHDL (very high-speed integrated circuits hardware description language) design integration is available for PCI-express Cyclone FPGA-based cards, Xilinx Zynq, and Intel SoC (system on chip) systems. The work is ongoing on optional extension with the parity bits support for fault-tolerant space applications. The core functional emulation is available in the QEMU (open-source system and user-level emulator) emulator out of the box from the year 2020. This enables continuous integration testing against the actual Linux kernel and provides a valuable tool for the driver-porting to other operating systems.

Historical background

The CTU CAN FD design started at the CTU FEE Department of Measurement under the lead of Jiří Novák to extend their long-term CAN testing support for Volkswagen and Skoda Auto by CAN FD.

CTU activities in the area of real-time distributed control and communications (fieldbus area) started in the early 1990s when the CTU experts were invited to participate in the joint project with PTB Berlin (Physicalisch-Technische Bundesanstalt, German Metrology Institute) focused on electromagnetic susceptibility of fieldbus technologies in harsh industrial environments. Simultaneously the university cooperated with the Czech company Unicontrols on the implementation of an extensive CANopen design and development system. Within this cooperation, the team has focused on a dual-CAN interface implementation. This included the channel redundancy, message queuing, and timestamping in both directions. The developers also have worked on the interface's driver support for several hardware platforms as well as real-time operating systems (e.g. OS9, VxWorks) and generic OS systems (e.g. GNU/Linux, Windows).

In the middle 1990s, the CTU team was contacted by Volkswagen to design and develop an in-vehicle system for identification of the CAN network error sources within the car. On this basis, a long-term cooperation with Škoda Auto began, which is still ongoing. Within the 20 years, the engineers designed, developed, and deployed many technologies that were used for testing during the vehicle development. Here are two examples. The first is an automated test system for CAN/LIN ECUs (electronic control unit). It implements a list of tests that ECUs should pass at the physical, data-link, and application layers. The tests are focused on the behavior within the networked system, not on the individual ECU functionality. The second example is a HIL (hardware in the loop) system for overall vehicle integration tests. The Skoda Fabia and Skoda Scala vehicles were completely tested using this system.

Pavel Píša (FEE Department of Control Engineering at CTU) and his students from the department also contributed to the open-source technologies as well as the real-time and control-related projects starting in the 1990s. Drivers for data acquisition, control cards, and devices have been developed during the years. Investments into the generic Linux CAN (LinCAN) support preceded even ▷

the SocketCAN sub-system. When the community chose SocketCAN as the preferred solution, the knowledge and some card supports were reused. An updated bit-timing-parameter computation algorithm has been developed and is the base for the generic CAN and CAN FD bit-rate setup till now. The need for a common platform for the CAN driver development and testing emerged during the work on the RTEMS system (real-time executive for multi-processor systems). The QEMU emulator CAN sub-system has been designed based on previous experience with the Humusoft data acquisition cards emulation. The work, initially supporting SJA1000 CAN controller (NXP) only, has been accepted for QEMU mainline in 2018 and is a base for the Xilinx controllers and CTU CAN FD emulation support.

CTU CAN FD IP core

Figure 1 shows the CAN FD IP core structure with Rx and Tx paths. The IP core provides the following features:

- VHDL design with no vendor-specific libraries required, yet RAM for buffers and Rx FIFO automatically inferred by Xilinx and Intel tools
- Compliant with ISO 11898-1:2015
- Rx buffer FIFO with 32 to 4096 words (1 to 204 CAN FD frames with 64 bytes of data)
- 2 to 8 TXT buffers (one CAN FD frame in each TXT buffer)
- 32-bit device memory interface (APB, AHB, RAM-like interface)
- Support of “ISO and non-ISO” conform CAN FD implementations
- Time-stamping and time-triggered transmission
- Interrupts
- Loop-back mode, bus monitoring mode, ACK forbidden mode, self-test mode, restricted operation mode

The CTU CAN FD IP core is optimized for 32-bit register access from the main CPU (central processing unit) connected over APB, Avalon, AXI bridge, etc. The core provides a synthesis-time-configurable depth FIFO on the reception side and four (optionally two or eight) transmission buffers, of which the transmission order can be controlled by individually assigned priority. Priority assignment is controlled by up to eight four-bit fields in one register, which allows maintaining the Tx FIFO order: a requirement for standard SocketCAN setup. Still, it allows a division of the Tx buffers into multiple priority queues or can even be combined with the time-triggered postponed release of selected buffer(s).

The design allows precise hardware time-stamping of the received frames. A time-stamp counter common for multiple CAN channels can be integrated into FPGA or SoC design with a resolution of up to 64 bit. Hardware time-stamp support for the corresponding SocketCAN driver is already developed and will be contributed to the 5.20 or following Linux kernel releases. Precise (10 ns in CTU’s design) time-stamping and deep Rx FIFO make the core suitable for latency evaluation and continuous integration checks of operating systems, different CAN controllers, and software performance. Such quality-assuring setup ensures that the kernel update is safe and will not cause problems for a given project or third-party controller in the future.

To prevent regressions in their own CTU CAN FD development process, the developers have set up complete systems, which allow analyzing of multiple CAN cores connected with a parametrized connection jitter and topology. More than 160 unit-tests and complex tests have run, as well as almost 200 another tests were precisely reimplemented according to the ISO 16845-1:2016 standard. All these pipelines have run in the environment based on the open-source GHDL GCC frontend for each design change. The complete synthesis of the Xilinx Zynq SoC design and the build driver have run nightly after each CTU CAN FD core repository

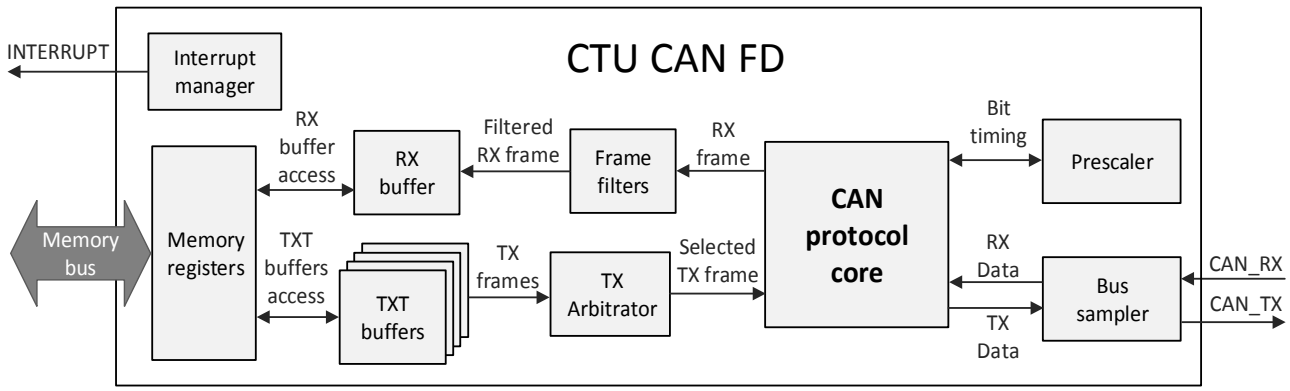

Figure 1: CTU CAN FD IP core structure (Source: CTU)

Ondrej Ille is a digital design expert working for several chip design centers. The initial goal of his work on a new core VHDL design was a specialized controller for CAN FD Skoda Auto analyzer update with support for time-based transmission of messages. In 2018, it was extended into development of a generic CAN FD controller to support the needs of the Volkswagen subsidiary to replace the commercial core. The latter did not fit in a required count into their Xilinx Zynq based project limited by the already produced hardware.

update. Generated artifacts were copied to Xilinx Zynq based MZ_APO education kit, where multiple tests, including interoperability with the Opencores SJA1000 controllers, have run.

CAN latency testing

One of the CTU CAN FD IP core applications is the latency testing of a CAN gateway based on the team’s own or on third-party controllers. Figure 2 shows the testing setup. ▷

There are two MZAPO boards based on the Xilinx Microzed boards with Zynq 7000 SoC. The bottom board does the benchmarking, and the top board is a device under test (DUT). Any CAN gateway device can be used as a DUT. The FPGA in Xilinx Zynq SoC is loaded with a design composed of four CTU IP cores and a software-controlled 4 x 4 x 2 CAN cross-bar. The goal of the latency testing is to continuously evaluate new versions of the Linux kernel, specifically the latencies of CAN gateways in the kernel and the user space.

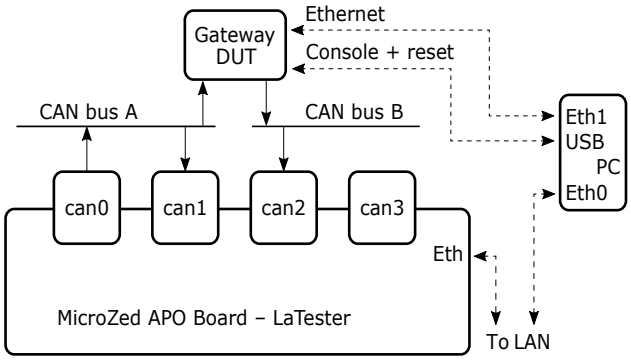

Figure 2: Testing setup for latency testing (Source: TCU)

The kernel is compiled for every new kernel version, booted on the DUT board, and tests are started on the Latester board (Latency Tester) equipped with CTU CAN FD. During the tests, a CAN message is repeatedly sent from the Latester interface can0 and immediately received on can1 to obtain a precise hardware time-stamp. And, after the gateway (DUT) copies the message from CAN bus A to CAN bus B, the message is time-stamped again when received on the Latester can2 interface. The gateway latency is obtained by simply computing the difference of the time-stamps. Then, the distribution of latencies can be plotted and analyzed. Integration of this latency testing into OSADL's QA farm is in progress.

The system has been successfully used even during the development of CAN NuttX RTOS driver for Espressif ESP32C3 RISC-V based micro-controller.

The Latester setup uses a 64-bit time-counter, and the CTU CAN FD design is running on 100 MHz, which provides an excellent 10-ns time-stamp resolution. Linux driver for CTU CAN FD IP core has been already mainlined, and the time-stamping support in the driver is on the way.

References

[1] CTU CAN FD IP core project
[2] CTU CAN FD IP CORE Datasheet, Czech Technical University in Prague, Faculty of Electrical Engineering, Department of Measurement
[3] CAN bus related projects list and CTU CAN FD IP core automatic tests results at Czech Technical University in Prague, Faculty of Electrical Engineering
[4] CTU CAN FD Driver Linux kernel mainline documentation
[5] QEMU CAN emulation and subsystem documentation
[6] Martin Jeřábek: FPGA Based CAN Bus Channels Mutual Latency Tester and Evaluation, Bachelor Thesis, CTU FEE, 2016
[7] Martin Jeřábek: Open-source and Open-hardware CAN FD Protocol Support, Master Thesis, CTU FEE, 2019
[8] Matěj Vasilevski: CAN Bus Latency Test Automation for Continuous Testing and Evaluation, Master Thesis, CTU FEE, 2022

Authors

Ondrej Ille
Czech Technical University in Prague (graduate)
ondrej.ille@gmail.com
www.cvut.cz

Jiří Novák
Czech Technical University in Prague
FEE, Department of Measurement
jnovak@fel.cvut.cz
meas.fel.cvut.cz

Pavel Píša
Czech Technical University in Prague
FEE, Department of Control Engineering
pisa@fel.cvut.cz
control.fel.cvut.cz

Matěj Vasilevski
Czech Technical University in Prague (graduate)
matej.vasilevski@gmail.com
www.cvut.cz