CAN FD core as an open source project

The Czech Technical University of Prague is developing a CAN FD core. It is intended to provide the core under open source conditions, but the implementer needs to negotiate IP rights in particular with Bosch.

The Faculty of Electrical Engineering (Department of Measurement) has designed in VHDL (Very High Speed Integrated Circuit Hardware Description Language) a CAN FD (flexible datarate) core. This development is intended to be shared under open source conditions. Volkswagen sponsors this project. The design has been tested on Zylinx Zync and Altera FPGAs (field programmable gate array).

Once tested according to ISO 16845-1:2016, the VHDL model will be offered free-of-charge under MIT license. Nevertheless, the implementer has to negotiate IP rights. Especially, Bosch has some IP rights on the CAN FD protocol and its implementation. Students started this project. The current status of the IP core development is documented on Github.

Modular IP core design

Each of the IP core modules has a unique functionality. The number of dependencies between modules is minimal, thus keeping modularity design rule. The whole system is implemented as synchronous design with asynchronous reset (assumed to be connected to an input pin of FPGA or ASIC). The CAN FD IP core is a memory-mapped peripheral. In order to be compatible with the university's Flexray IP core, the Avalon bus interface is used. The whole IP core is implemented with VHDL 2008 version of the language (it also complies with 2002 version).

The core is accessible via the Avalon parallel bus by Altera. The bus consists of separate write/read data lines. Read and write cycles are distinguished via dedicated signals. Avalon address bits 19 to 16 must be matching the core-ID value during the access. With this architecture, it is possible to run up to 16 instances of the IP core on a single Avalon bus.

The whole core is synchronized to one clock signal. Any other time-period is derived from this clock. Every register has an asynchronous reset by default. The design is intended to be latch-free. Input signals of the Avalon bus interface and the time-stamp value are expected to be synchronous to the clock signal and no clock synchronization is implemented on these signals. The CAN_RX signal is synchronized by simple synchronization chain with two flip-flops. This synchronization chain is optional, but it is recommended to use it, unless synthesis tools automatically insert synchronization chain.

The functional blocks shown in Figure 1 are separate VHDL entities and are implemented in stand-alone files. The CAN controller core consists of several sub-blocks. The CAN FD protocol core shown in Figure 2 covers the functionality of serial data transmission according to ISO 11898-1:2015. Storing frames to be transmitted, storing received frames, transmission, reception, arbitration, bit stuffing, bit de-stuffing, CRC calculation, error handling, and fault confinement are implemented in this module. Furthermore, valid CRC selection, transmit trigger, and receive trigger multiplexing, status bus assignment and bus traffic measurement are implemented in this core.

There are by default three filter masks or one range filter to select, if this CAN FD data frame is to be stored or not. The receive buffer is a FIFO memory. The receive buffer can store received data frames with a length of 5 to 20 (32 bit) words. Therefore, if size of less than 32 words is used, long CAN FD frames won't be stored and data overrun will appear, even if the buffer is empty. Data overrun occurs always when there is less free memory in the buffer as size of the received data frame.

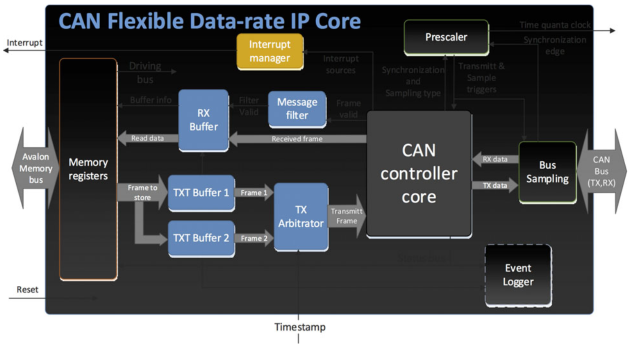

Figure 1: Block diagram of the CAN FD IP core (Source: Czech Technical University of Prague)

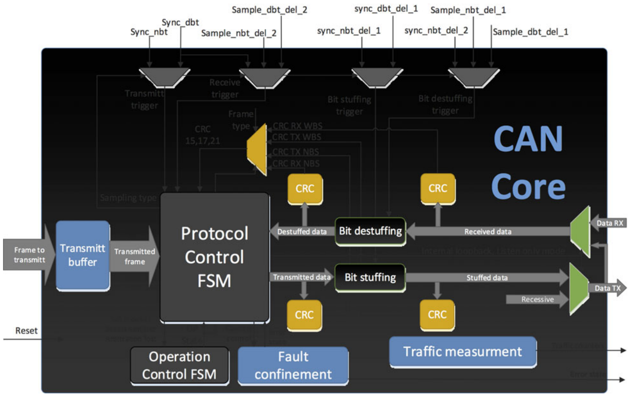

Figure 2: Block diagram of the CAN FD protocol core (Source: Czech Technical University of Prague)

In the transmission path, there is a TX arbitration circuitry. It manages the two implemented transmit buffers. Additionally, it implements the functionality of propagating a frame to the CAN FD protocol core at a specific time. If external time-stamp value is lower than time-stamp specified in a frame, then a data frame is propagated to CAN Core. If time-stamps in both buffers are lower than external time stamp, then the one with the lower time-stamp is propagated to the core for transmission. When both time-stamps are the same and lower than external time-stamp, the frame with lower CAN-ID is propagated to the output. If both timestamps are equal and both CAN-IDs are equal, then frame from TX buffer 1 is propagated. Additionally, the circuitry can be configured to forbid propagation from each of the TX buffers.

The CAN FD IP core provides an interrupt manager. The interrupt sources are configurable. Every interrupt is marked into an interrupt vector. If another interrupt source is activated when an interrupt is active, no further interrupts are produced on the output, but the interrupt source is accumulated into interrupt vector. Thus one active period, there can be managed a superposition of up to 11 interrupt sources going active. It is recommended that the low-lever driver always reads interrupt vector after an interrupt is detected, to determine the events that caused the interrupt. Interrupt manager can be configured to forbid or allow various interrupt sources.

Summary and outlook

The project at the Czech university is still under development. In particular, additional testing of the IP core is necessary. This includes implementing event logger and error detection feature tests.

The current CAN FD IP core has many features, which are not required by CAN FD standard. However, there is room for improvement. Following list names just a few of the possible improvements:

- Automatic bit rate detection, either as part of core or separate module. Once this unit is turned on, it measures bus timing (e.g. over several data frames) and provides the results, or sets the bus timing registers.
- Additional state machine, which decides about usage of bit rate based on relative error rate of two bit rates.
- Optimization of TX buffers to be synthesized into RAM elements, not to LUTs.

Of course, the project team appreciates further sponsors as well as partners for conformance and interoperability testing. ◀

Author

Holger Zeltwanger
CAN Newsletter
pr@can-cia.org
www.can-newsletter.org