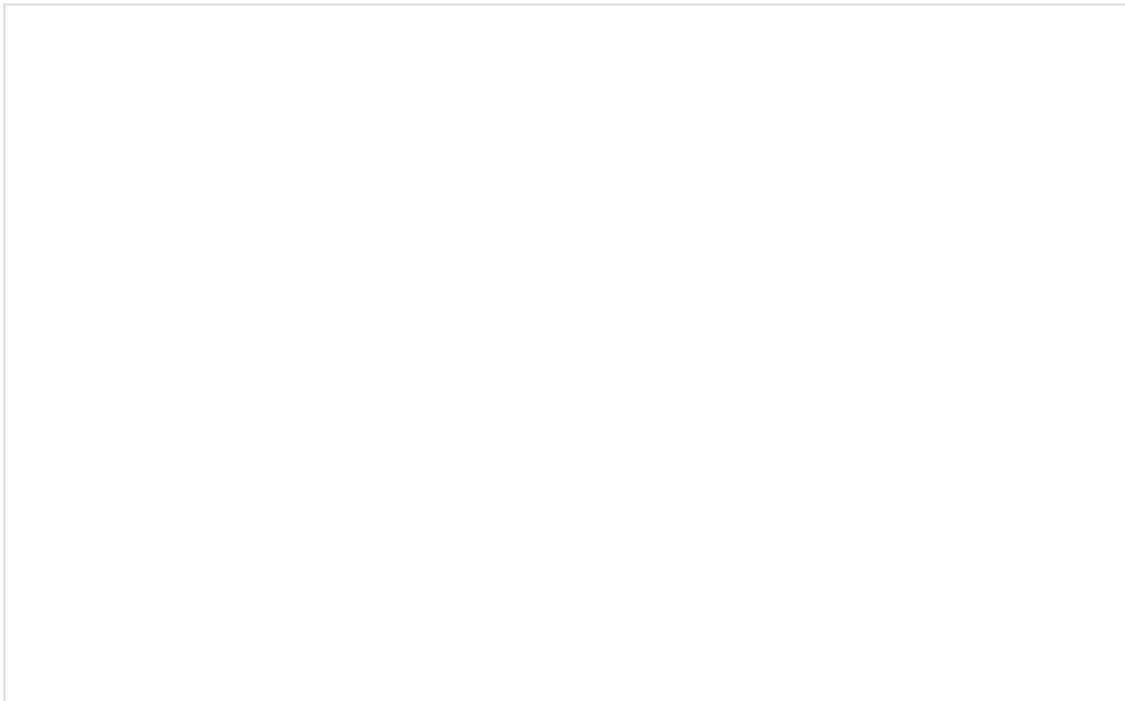


# Compliant with ISO CAN FD

The Czech Technical University (CTU) in Prague has developed an open-source CAN FD core and appropriate development tools. The core complies with ISO 11898-1:2015.



Block diagram of the CAN FD core (Source: CTU)

The IP core is available under MIT license conditions. Besides the ISO CAN FD specification (ISO 11898-1:2015), it also supports the non-ISO specification (Bosch CAN FD version 1.0). The basic features include a RX first-in, first-out (Fifo) buffer of 32 to 4096 words equivalent to one to 204 CAN FD data frames and four TX buffers for one CAN FD data frame each. Time stamping of frames is supported as well as time-triggered transmission of data frames.

The CAN FD core source code is available in [CTU FEE Gitlab repository](#). The IP core is accessible as a slave device via three memory networks: RAM-like interface, APB, and AHB. Each interface can be used via dedicated wrapper. The core shall not be accessed sooner than two clock cycles after external reset was released (due to reset synchronization). If the core is accessed earlier, writes have no effect and reads will return zeros. If an external reset is executed via a software driver (e.g. at driver load time), it is recommended to add corresponding delay before the driver executes any access to the device.

The IP core comes with Xilinx Vivado component for integration in FPGAs (field-programmable gate array). It can also be integrated in Asics (application-specific integrated circuit). After reset, the CAN FD core is disabled: it does not take part in network communication (no transmission, no reception, and no monitoring). Before it is enabled, it must be configured. Once it is configured, it can be enabled. After the enabling, the CAN FD core starts bus integration and joins the bus communication after receiving 11 consecutive recessive bits. When the core joins the bus, it becomes error active (during integration it was bus-off state). At this moment, the core starts communicating on the bus.

[hz](#)