The SOK Zynq® Target for MATLAB® Simulink® provides a new interface between Zynq hardware and the development level in MATLAB Simulink. The features of the platform, which are

- operating system integration (e.g. Debian, freeRTOS, baremetal),
- Inter-Core Communication protocol,
- external communication protocols and
- real-time task scheduling

are presented in this document. Here the focus is on the platform and firmware side. The software side (i.e. the combination of the target with MATLAB Simulink) can be found in the 'SOK Zynq Target for MATLAB Simulink'-document.
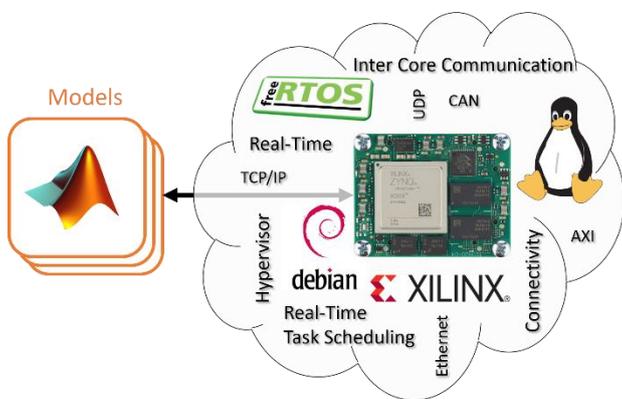

Characteristics of the SOK Zynq Target Platform.

In general, this platform supports the entire portfolio of Zynq UltraScale+ MPSoCs in the three variants CG, EG and EV. It is recommended to start using the platform with a 4 x 5 cm MPSoC module from Trenz Electronic which are compatible with many carrier boards, such as

- Trenz TE0701 Board and
- Trenz TE0703 Board.

Different **operating systems**, like Debian Linux and freeRTOS, can be used on the individual cores. In addition, baremetal code can also be executed. When Debian Linux is used on a core, a partitioning hypervisor must manage the hardware accesses for the Cortex-A Cores. For this case, the Target Platform supports Jailhous as a hypervisor.
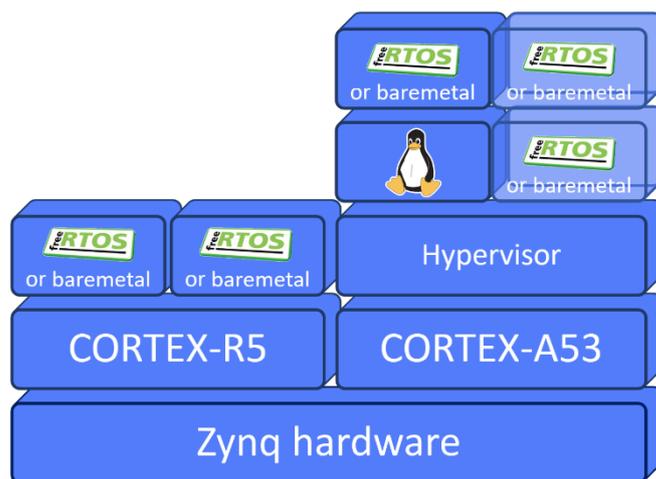
Processor-core schema with Linux and freeRTOS/baremetal Cores. Since Linux is used in this example, a hypervisor must distribute the resources.

An **Inter-Core Communication** (ICC) library is provided to communicate between different applications. These applications can either be written by hand in C/C++ or generated from MATLAB Simulink models and can be executed on any core of the target. ICC supports communication between applications on one core in addition to communication across multiple cores. The on-chip memory, a memory connected to all cores with a high-speed connection, is used for data exchange. For larger amounts of data, the SDRAM can be used. By using double buffers, the ICC communication becomes thread-safe and non-blocking. It also ensures that data is exchanged consistently. ICC is available under freeRTOS and Debian Linux.
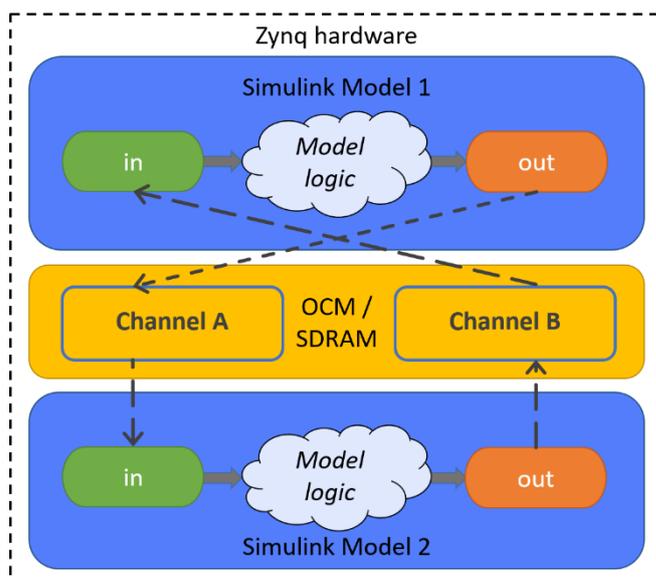
Inter-Core Communication over on-chip memory (OCM).

The Target Platform includes drivers that can be used for various **communication protocols**, for example

- UDP,
- TCP,
- Raw Ethernet and
- CAN.

These protocols are available in Simulink. Additional protocols, such as

- PROFINET over TSN,
- OPC-UA (server-side) or
- C8Y Agent

can be provided upon request.

Using freeRTOS, the SOK Zynq Target for Simulink provides a **real-time task scheduling** that the users configures in the Simulink models. By doing so, models can be split into multiple tasks. For example, parts of a model can be executed in different frequencies that can be divided into different tasks. This has the advantage that tasks that can be processed in a slow frequency do not fill up a faster time slice.

Furthermore, the transmit and receive functions can be moved to one or two separate tasks. This has the advantage that the core processes the important read and write accesses to interfaces in a higher priority and is therefore less likely to be interrupted. As seen in the figure below, there are several task configuration options.
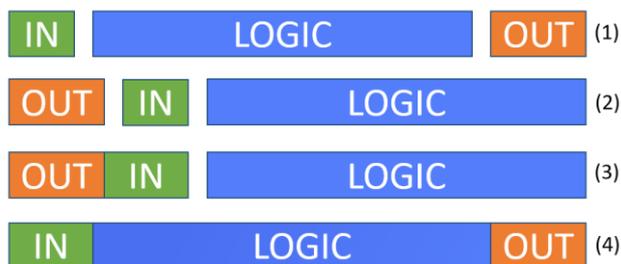
Different options for task configuration.

In addition to the possibility of executing all parts of an application in one task (4), there are three other alternatives. In two options, the models can be divided into three tasks (1,2). "IN" for reading and receiving, "LOGIC" for the model or application logic and "OUT" for writing and sending. It can be subdivided whether the 'OUT' task is executed at the beginning (2,3) or directly after the 'LOGIC' part (1,4). If the sequence 'OUT'-'IN'-'LOGIC' is selected (2), the outgoing data is always sent first and the timing can be relied upon. If 'IN' and 'OUT' do not require different priority, these parts can be combined into one task (3).

Links:

Produktseite
https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html
https://www.debian.org/
https://www.freertos.org/
https://github.com/siemens/jailhouse
https://linutronix.de/industrial_linux/hypervisor.php