CAN IP core with DMU and TSU

Bosch has improved its M-CAN. The added functions include a DMA interface unit (DMU) and a time stamping unit (TSU).

The DMU add-on allows the reduction of the host controller load by off-loading the transport of CAN frames to a DMA controller. The TSU expansion module enables hardware-based and Autosar-compatible time synchronization.

When exchanging CAN data frames between the host controller and the M_CAN protocol controller, there are some issues to consider, especially for complex SoCs (system-on-chip). Due to the high complexity of modern SoC architectures, the on-chip communication paths are divided into several domains of different performance. The bridging between domains additionally slows down performance, e.g. due to clock-domain crossings.

The heat-map (red = fast, blue = slow) in Figure 1 illustrates the speed of data transfers initiated by the host controller in the processor domain. Thus, the connection of the host controller core to the dedicated caches is the fastest, followed by the TCM within the cluster. When creating the software, it must be ensured that these memories can be used efficiently.

In extreme contrast to this, single accesses to components in the peripheral domain can be up to 30 times slower. If, for example, the continuous exchange of CAN frames between the host controller and a M_CAN unit is considered, the following interactions are typically required:

- Check the status register of M_CAN when asserting an interrupt
- Optionally transfer the CAN data frames
- Optionally signal to the M_CAN the completion of frame transfers

If the cores in the processor domain would perform these interactions, they would be significantly slowed down by the NoC (network on chip). For example, such interactions can also be done via a processor core within the peripheral domain, if available. However, this article focuses on a different approach that does not allocate computational resources of any processor core.

DMU functionality

Bosch offers an add-on for the M_CAN called DMU. With this, the continuous exchange of CAN data frames can be completely outsourced to a DMA controller. The add-on unit is based on the concept of virtualizing the FIFO (first-in, first-out) head elements (Figure 2).

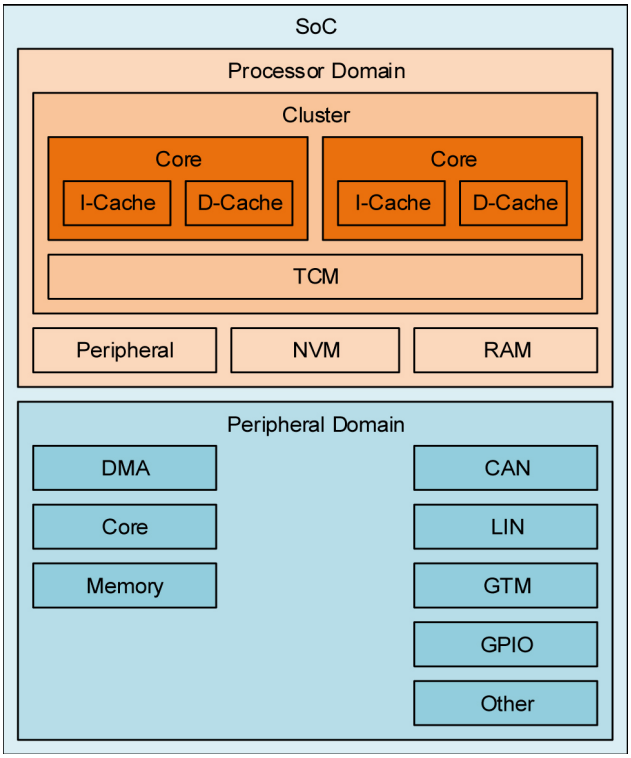

Figure 1: Domains of a SoC (Source: Bosch)

The M_CAN has an associated message RAM (MRAM), which i.a. contains the elements (CAN data frames) organized in FIFOs. To access the memory segment of the current message (head element) within these FIFOs by the host controller, the respective pointers (read/write pointer) from the M_CAN must previously be queried. To avoid this, accesses to fixed address areas are virtualized. The DMU dynamically redirects these accesses to the head elements in the MRAM. The redirection is controlled invisibly within the DMU by the FIFO pointers in the M_CAN. The size of the reserved areas corresponds to the largest possible frame elements, which are 18 words (32-bit) for the TX, RX0, and RX1 elements and two words (32-bit) for the TX Event element (three words, if the TSU timestamp is also transferred). The transfer of a last element word activates a process in the DMU, in which for TX elements the transmit request is set in M_CAN, or for the other elements (RX0, RX1, TXE) the dedicated FIFO acknowledge is set in the M_CAN unit. Thus, writing or reading the CAN data frames via DMU elements completes the whole queuing/de-queuing process in the M_CAN unit. The DMU supports data frame transfers ▷

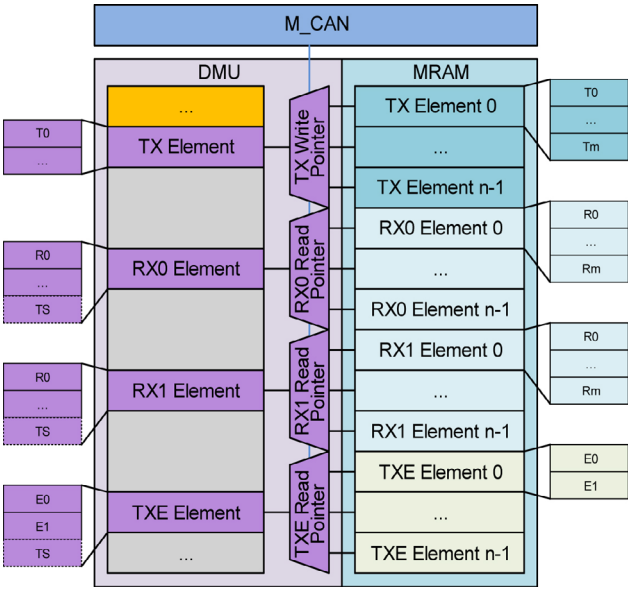

Figure 2: Functional block diagram of the DMU (Source: Bosch)

from the CRAM to the TX-FIFO/Queue and vice versa, from the RX-FIFOs respectively the TX-event FIFO to the CRAM. The block diagram in Figure 3 shows the M_CAN unit with the add-ons DMU and TSU. The host controller accesses to the M_CAN are routed through both add-on modules.

Figure 4 shows the memory map of the DMU. In the yellow marked memory area starting at address 0, the registers of the M_CAN unit and the TSU are memory-mapped. Afterwards, the purple coloured Virtual Buffers are shown, which are accessed by the DMA in order to transport the head elements of the M_CAN message FIFOs.

TX element

The CAN data frame elements are written by the DMA controller to be added in the TX-FIFO/Queue. When writing the last element word, the TX request is automatically set for this element, so that the M_CAN unit sends this. The DMU requests further CAN data frame elements from the DMA controller as long as the TX-FIFO/Queue is not full.

RX0 / RX1 elements

The CAN data frame elements are read by the DMA controller, which are located in the receive FIFO 0 or FIFO 1 of the M_CAN unit. When the last element word is read, the de-queuing is communicated to the M_CAN unit by setting the dedicated acknowledge index by the DMU. Optionally, the time-stamp of the TSU can also be transmitted. The DMU triggers the DMA to de-queue further CAN data frame elements as long as the RX-FIFO is not empty.

TX event element

Like the RX0 / RX1 elements, but here the TX events are read, optionally with the time stamp of the TSU.

DMU register

The DMU gets most of the configuration parameters from the M_CAN, only the transport of the hardware time-stamp of the TSU can be switched on/off. The status information provides feedback on whether the access to the virtual elements is correct or, if not, what problem occurred. This is particularly helpful when debugging the DMA routines, but should also be monitored during normal operation for reasons of functional safety.

DMU debug section

When debugging the software, the DMU elements can be accessed by reading without affecting the queuing or de-queuing of the DMA. For the TX element, the last element written is read, for the RX0, RX1, and TXE elements the current element is read. These accesses do not trigger automatisms of the DMU, like the acknowledgment in M_CAN core.

Data flow within the SoC

The following approach is recommended for the data flow within the SoC: A RAM has to be selected to which the desired processor core can access with the highest possible read/write performance and to which the DMA controller also has direct access. This CRAM is then used to exchange the CAN data frames, with the DMA controller taking over the slow transfers of the CAN data frames across large distances of the NoC and storing them in the CRAM close to the processor core, which then access without performance loss.

TSU add-on

For the Autosar-compatible (automotive open system architecture) synchronization of time bases between CAN nodes, only software implementations had been

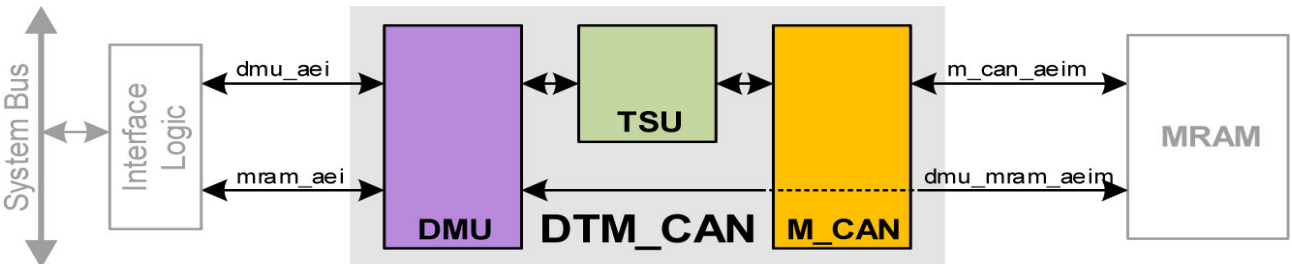

Figure 3: M_CAN unit with DMU and TSU (Source: Bosch)

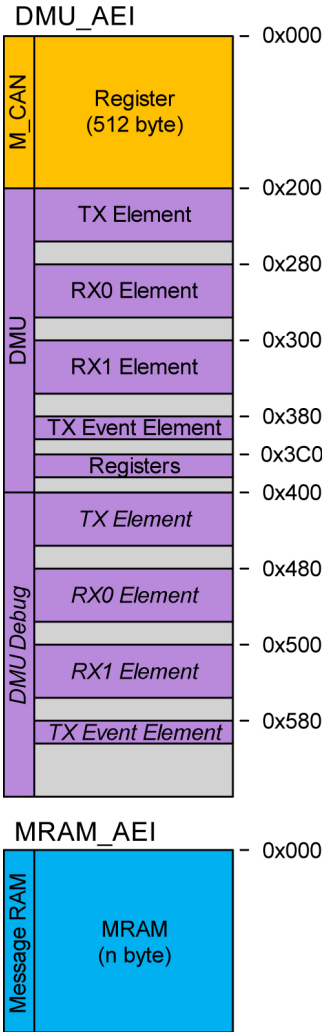

Figure 4: DMU address map (Source: Bosch)

used due to a lack of special hardware. To further increase the time accuracy, special hardware is required. The CiA 603 document specifies a hardware-based concept, which has been implemented in TSU. This approach is independent of interrupt response times and thus achieves the best possible accuracy. The TSU may operate on its own internal time-base, or uses an external time-base, e.g. a reference time base within the SoC.

Receipt of messages with timestamp

In order to receive a time-stamped message, a base or extended frame ID filter element must be configured accordingly, i.e. S0.SSYNC = 1 or F1.ESYNC = 1. Upon receipt of a valid data frame matching the filter, the time-stamp is stored in the TSU. Since the TSU stores several time-stamps, a pointer is written into the CAN data frame (R1.RXTSP), which points to the corres ponding entry in the TSU. When reading out such an RX data frame with the DMU, the TSU time-stamp can be automatically attached.

Sending messages with timestamp

If a time stamp has to be generated when a CAN data frame is sent, the following bits must be set in the message: T1.TSCE = 1 and T1.EFC = 1. Upon successful transmission, a time-stamp is stored in the TSU and a TX-Event message is generated, which refers to the corresponding entry in the TSU, i.e. field TXTSP in event message word E1 (E1.TXTSP). When using DMU, the time-stamp of the TSU can also be automatically attached.

Synchronization process after Autosar

After successful configuration (see above) of all participating CAN nodes, the timers of the time slaves can be corrected with a two-step synchronization process. In the first step, the current time T0 is latched in the time master, and the part of T0, which represents the seconds, is sent to the time slaves with the SYNC message. If the EoF is reached when sending this SYNC message, then timestamps are stored in the TSUs of all nodes, i.e. in the Time Master and all Time Slaves. This is the common point of time reference. In the second step, the time master adds its TSU time stamp to the nanosecond part of T0 and sends it to the time slaves via FollowUp message. From this, the time slaves can derive their own time base.

Saving time stamps at SoF

For non-Autosar applications, the TSU can also be configured to store time-stamps at the start of a CAN data frame (SoF bit). ◀

Literature

[1] Time-stamping of CAN frames, CAN Newsletter 2/2017[2] CiA 603, CAN frame time-stamping – Requirements for network time management, Nuremberg 2017

[2] CiA 603, CAN frame time-stamping – Requirements for network time management, Nuremberg 2017

Authors

Stefan Thiele
Robert Bosch
info@de.bosch.com
www.bosch.com

KVASER

YOUR TRUSTED **CAN** PARTNER

#CAN FD
#LIN #SCRIPTING
#LINUX #DATALOGGING
#ETHERNET #MAGISYNC #RUGGED
#SILENTMODE #EMBEDDED

www.kvaser.com
sales@kvaser.com

Advanced CAN solutions built by engineers, for engineers.

Kvaser CAN Interfaces & dataloggers are the reference for CAN and related bus protocol connection. Kvaser is committed to providing open solutions that ensure easy toolchain integration.

- Combine Kvaser hardware with software from our Technical Associates for a powerful, tailored solution.
- Free software includes Kvaser CANlib SDK, a one-stop development environment for our CAN and LIN interfaces, and Kvaser's free CanKing bus monitor. Windows and Linux versions available, accessible through Python.
- Free global support. Our International Support Team provide timely email and phone support to Kvaser customers throughout the world.
- Unique design: Lightweight, ergonomic, durable, and flexible.