Next Gen Automotive Ethernet Functions and the Implementation in an Ethernet MAC

2017 IEEE-SA Ethernet & IP @ Automotive Technology Day
Harald Zweck, Infineon Technologies
Ethernet In-Vehicle Network Example

- Chip2Chip Safety
- Time Synchronization
- Real Time
- Freedom from Interference

Note: The picture shows an example for a part of an In-Vehicle Network, and does not represent a comprehensive architecture.
Vehicle Development Cycle

- **Application Use Case**
- **Solution Concept**
  - Scalable, flexible and customizable in-vehicle network
- **Product Verification**
- **Product Design**

Pictures by courtesy of AUDI

Copyright © Infineon Technologies AG 2017. All rights reserved.
## Vehicle Development Cycle

**X years to SOP**

<table>
<thead>
<tr>
<th>Concept Phase</th>
<th>Series Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Discussion</td>
<td>Coordination</td>
</tr>
<tr>
<td>Concept Discussion</td>
<td>Verification &amp; Confirmation</td>
</tr>
<tr>
<td>Preparation &amp; Planning</td>
<td>Final Preparation</td>
</tr>
</tbody>
</table>

**Picture by courtesy of AUDI**
First Generation In-Vehicle Ethernet

› System requirement
  Simple Data Transfers

› Example Ethernet for diagnosis
  – Vehicle is in the repair shop
  – Vehicle does not move
  – Connection is based on standard Ethernet technology
  – No real time requirements for transfers
Use Case Diagnostic

- Implementation
  - Buffered, DMA driven data transfer
  - Using IEEE 802.3
  - 802.1AS conforming Time Stamps
  - MAC address filter
Second Generation In-Vehicle Ethernet

Vehicle Control Loop

Real Time Communication

High Bandwidth Communication

Network

Computing -> internal Network <- Computing

Sensors

Vehicle Physics / Dynamics

Actuator

PHY

Connector

Copyright © Infineon Technologies AG 2017. All rights reserved.
Use Case Time Synchronization

› System requirement:
  Synchronized time base for the IVN*
  - Time stamping compatible with IEEE 802.1AS
  - Feature is implemented in HW
  - Supports the implementation of a common time base for the IVN*
  - Required e.g. for AVB protocols

*IVN = In-Vehicle Network
Use Case In-ECU Communication

Real Time Communication

High Bandwidth Communication

ECU

Computing

internal Network

Computing

Network

PHY

Connector

Sensors

Vehicle

Physics / Dynamics

Actuator
Use Case Memory-to-Memory Copy

- System requirement
  Synchronize content of variables
  - Data transfer between controllers (MCUs)
  - Data block copy process
  - The process is scheduled
  - The process execution time is constraint
  - MCU connection options
    - Directly coupled via xMII
    - Coupled via a switch
Use Case Memory-to-Memory Copy

› MCU level requirement
  - Data flow separation inside the MCU – no dependencies
  - Any receive flow can be directed to any CPU core (in case of multicore MCU)
  - Any transmit flow can be sent from any CPU core
  - Precise scheduling for data transmission
  - Low software load

Note:
Red arrows represent MCU to MCU flow
Yellow arrows represent other flows
Use Case Memory-to-Memory Copy

- Implementation
  Separation of data flows
  - IEEE 802.1Q VLAN separation
  - Introduction of several channels as “queues”
  - Each queue holds data independently
  - Each queue is emptied / filled by its own DMA
Use Case Memory-to-Memory Copy

- Implementation
  - Separation of data flows (inbound)
    - IEEE 802.3 frame header filter
    - Separates inbound traffic based on destination address
    - Forwarding to according queue
    - In turn selects destination CPU
Use Case Memory-to-Memory Copy

› Implementation
  Separation of data flows (outbound)
  -> re-merging of the flows
  - IEEE 802.1Q shaper
  - Separate outbound traffic is merged before the xMII interface
  - Merging is based on IEEE 802.1Q shaper rules
Use Case Memory-to-Memory Copy

Implementation
- Outbound data flow is scheduled
  - Using IEEE 802.1AS
  - 802.1AS drives a scheduler
  - Queue output is driven by the scheduler (in “shaper” block)

Outbound Transfers

Schedule defined by Time Stamp

Red arrows represent MCU to MCU flow
Yellow and grey arrows represent other flows
Use Case Memory-to-Memory Copy

- Implementation
  - Data flow separation in the MCU
  - Needs freedom from interference
  - Transfer paths of DMAs are safe guarded
  - Potential DMA write targets can be protected
Use Case Memory-to-Memory Copy

› Implementation
Data flow separation in the MCU
- Freedom from interference enables firewalling the flows
- Processes and their data flows can be separated to different CPUs
Summary

› The next gen Ethernet implementations will expand the support for Automotive application use cases

› There will be HW features for use cases requiring deterministic data transfer

› IEEE 802 offers the standards to make the step successful
Thank You!

- CAN-FD
- FlexRay
- IEEE Ethernet (1 Gbit/s)
- ASC LIN
- SENT
- PSI5

Copyright © Infineon Technologies AG 2017. All rights reserved.