Virtual ECUs with QEMU and SystemC TLM-2.0

Lukas Jünger\textsuperscript{1}, Jan Henrik Weinstock\textsuperscript{1}, Munish Jassi\textsuperscript{2}, Megumi Yoshinaga\textsuperscript{3}, Hitoshi Hamio\textsuperscript{3}, Koichi Sato\textsuperscript{3}

\textsuperscript{1}MachineWare GmbH, Aachen, Germany
\textsuperscript{2}Renesas Electronics Europe GmbH, Düsseldorf, Germany
\textsuperscript{3}Renesas Electronics Corporation, Tokyo, Japan
Motivation

Mercedes Recalls EQS And EQE EVs For Software Bug That Could Cause Sudden Power Loss

Mercedes must recall 8,291 EQ models as a result of a software glitch that has led two vehicles to lose propulsion in the U.S.

Tesla gets hacked – winning the hackers $100,000 and a Model 3

The automaker wants to speed up EV development.

VW Group CEO Admits Audi Lags Behind Rivals, Software Issues To Blame

The automaker wants to speed up EV development.

Volvo, Polestar delay flagship electric SUVs on software issues

Volvo CEO Jim Rowan blames "lumps and bumps" associated with rigorous testing of "mission critical software."
Motivation

- SW complexity is rising
  - Car: >100 Mio. lines of code
- Security & safety critical
- Quality is key

- Bad software is dangerous and expensive
  - Accidents, recalls, liability for hacks, maintenance
- Problem: Testing is
  - Hard to automate, hard to scale, limited by hardware resources

https://informationisbeautiful.net/visualizations/million-lines-of-code/
Virtual Platform / Level 4 vECU

- Virtual Platform: Full System Simulation
- Executes unmodified target binary
- Advantages over physical prototypes
  - Available earlier (shift-left methodology)
  - Full flexibility, deep introspection
  - Non-intrusive debug
  - Near endlessly scalable
vECU Architecture

- Goal: Fast VP for SW verification
- QEMU + SystemC co-simulation
- QEMU for fast CPU subsystem
- SystemC TLM-2.0 peripherals for maximum reuse (VCML-based)
- Host network integration
• Open-Source Full System
  (Sim-)/(Em-)ulator and Virtualization tool
• Fast’ish CPU and many peripheral models
• GPLv2 license
• Monolithic architecture
  • No standard interfaces, global memory view, …
• Written in C :(
QEMU SystemC Integration

- **QEMU in a SystemC Box**
- **QEMU + SystemC Co-Simulation**
- Modularize QEMU models for reuse in Virtual Platform
- **Add standard TLM-2.0 interfaces**
- **Integrates in MW VCML**
External Communication Interfaces

• Send / Receive data from host
• Integrate with Linux SocketCAN
  • Attach physical CAN device, restbus simulation, …
• Integrate with SLiRP, TAP devices
  • Access host network
• Open-Source bridges from VCML
Benchmark Results

**Linux Boot**

Time from power-off to command line (mttcg vs icount mode)

- mttcg: 2.80s
- icount: 6.34s

**Simulation Speed Performance (relative)**

- Dhrystone benchmark
  - SoC platform - QEMU and SC IP models: 0.77
  - QEMU CPU only (bare minimal): 1.0
- Coremark benchmark
  - SoC platform - QEMU and SC IP models: 0.64
  - QEMU CPU only (bare minimal): 1.0
Open-Source / Free Software

- Disclaimer: I am not a lawyer and this is not legal advice
- QEMU is (mostly considered) GPLv2 license
  - Free as in freedom, not free of charge
- GPLv2 license requires source code access for SW user on request
- User can publish, modify, copy, redistribute program under GPLv2
- GPLv2 is infectious -> simulator must be GPLv2 compatible
Summary

• QEMU CPU models deliver decent performance
• QEMU - SystemC integration is a challenge
• VCML enables modularization of SystemC and QEMU models
• License needs to be considered