Chip simulation of automotive ECUs

Jakob Mauss, QTronic GmbH
Matthias Simons, Daimler AG

9. Symposium
Steuerungssysteme für automobile Antriebe
Outline of the talk

Chip simulation of automotive ECUs

1. Motivation
2. Setting up a simulation
3. Performance
4. Limitations
5. Conclusion
Motivation

ECU: more than 30,000 software parameter
Example: $16 \times 10$ map

ECU memory dump
Motivation

**Engine calibration**
- tune more than 30,000 ECU parameter
- done by the OEM, not by the supplier of the ECU

**Process today**
- automated optimization of stationary states
- real-time test rig or vehicle: based on the real ECU
- PC based: engine and ECU both simulated, e.g. in Simulink

**Problems**
- real-time test rig:
  - limited reproducibility
  - expensive (invest, operation)
  - slow (real time)

- PC: reverse engineering of ECU is
  - time consuming
  - complex
  - error prone
Motivation

Idea
move engine calibration (and other development tasks) from test rig to PC

Benefit
- simulation runs much faster than real time
- enables use of mathematical optimization
Motivation

Simulation of ECUs on PC:

Problem:
How to simulate ECU if no C source or model is available?

Ideas:
- Simulate the CPU based on the hex file
- Integrate this feature into MATLAB and QTronic Silver
Example - TCU Control Software in Silver

Setting up a TriCore simulation

1. write spec.txt to specify what functions to run
2. step and debug the simulation in Silver debug mode
3. generate fast running SFunction or Silver module: runs without a2l and hex
Setting up a TriCore simulation

1. write spec.txt to specify what functions to run
2. step and debug the simulation in Silver debug mode
3. generate fast running SFunction or Silver module: runs without a2l and hex

```
# specification of sfunction or Silver module
02 hex_file (m12345.hex, TriCore_1.3.1)
03 a2l_file (m12345.a2l)
04 map_file (m12345.map)  # a TASKING or GNU map file
05 frame_file (frame.s)    # assembler code to emulate RTOS
06 frame_set (STEP_SIZE, 10)  # Silver step size in ms
07 frame_set (TEXT_START, 0xa0000000)  # location of frame code

# functions to be simulated, in order of execution
10 task_initial (ABCDE_ini)
11 task_initial (ABCDE_inisyn)
12 task_triggered (ABCDE_syn, trigger_ABCDE_syn)
13 task_periodic (ABCDE_20ms, 20, 0)
14 task_periodic (ABCDE_200ms, 200, 0)

# interface of the generated sfunction or Silver module
17 a2l_function_inputs (ABCDE)
18 a2l_function_outputs (ABCDE)
19 a2l_function_parameters_defined (ABCDE)
```
generated SFunction in MATLAB/Simulink

-特性转为MATLAB工作空间变量
  - 由S-function读取
  - 可由脚本修改

- 使用默认值
  - 从HEX文件作为m脚本
  - S函数块面罩和其他Simulink片段

. spec.txt
.hex .map
.a2l .frame.s

tcbuild

MATLAB/Simulink
S-function
40 MIPS

.default values for
characteristics from
HEX file as m script,
mask for S-function
block and similar
Simulink snippets

.characteristics turned into
MATLAB workspace variables
- 读取S-function
- 可由脚本修改
generated virtual ECU in Silver

**virtual ECU**
- ECU functions
- RTOS emulation
- 4GB virtual memory
- A2L conversion
- XCP

**INCA CANape**
- on-line calibration: measure and tune running simulation

**TCB build code generator**
- spec.txt
- .hex
- .map
- .a2l
- frame.s

**tcbuild**
- .mexw32 MATLAB/Simulink S-function

**TriCore**
- emulsion
- 40 MIPS

**vehicle simulation or measurements**

**SILVER**

**TriCore debug mode**
- 0.4 MIPS

**TriCore debug mode**
- 0.4 MIPS

**Simulink Enabled**
- MathWorks Partner
Virtual ECU running in Silver: MED17
Performance and Limitations

Run complex function for a measured scenario, 3.5 minutes

<table>
<thead>
<tr>
<th>target</th>
<th>execution time</th>
<th>MIPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Silver in debug mode</td>
<td>919.15 sec</td>
<td>0.41</td>
</tr>
<tr>
<td>generated Silver module or MATLAB/Simulink SFunction</td>
<td>9.30 sec</td>
<td>40.80</td>
</tr>
<tr>
<td>MED17 with TC1797, 180 Mhz</td>
<td>210.00 sec</td>
<td>270</td>
</tr>
</tbody>
</table>

Limitations:
- instruction accurate, but not cycle accurate
- based on TriCore specification: 'silicon bugs' are not simulated
- PCP, CAN controllers and other on chip devices not modeled
ECU simulation on Windows PC

- without expensive reverse engineering
- without access to ECU source files
- based on HEX, MAP and A2L file
- low work effort for modeling
- high accuracy of model
- application example: automated calibration

- works for TriCore processors: TC1796, TC1797, TC1798, ...
- performance: 40 MIPS