Behavioural Model

of a DCDC Converter

written in VHDL-AMS
Contents

1.) Introduction: Behavioural Models / VDL-AMS

2.) DCDC Converter: Specification
   Realization

3.) Model Description: Graphical Structure
   Behavioural Models

4.) Simulation Results

5.) Conclusion
1.) Introduction: Behavioural Models / VHDL-AMS

Types of Models:

- physical models -> design, analysis, optimization of details
- behavioural or experimental model -> system analysis

VHDL-AMS:

- toolindependent standard -> IEEE 1076.1
- different labels of modeling -> structure, behaviour (equations, procedures)
- suited for model exchange -> supplier <-> car manufacturer
2.) DCDC Converter: Specification (Main Points)

Future cars: 2 voltages for board net: 42V, 14V -> DCDC Converter for both directions

<table>
<thead>
<tr>
<th>Buck Mode</th>
<th>42 V → 14 V</th>
</tr>
</thead>
<tbody>
<tr>
<td>- input voltage</td>
<td>30V ... 50V</td>
</tr>
<tr>
<td>- output voltage</td>
<td>14V (12V ... 16V)</td>
</tr>
<tr>
<td>- output current limit</td>
<td>72A</td>
</tr>
<tr>
<td>- voltage drop at max. current</td>
<td>0.4V</td>
</tr>
<tr>
<td>- settling time</td>
<td>&lt;2ms</td>
</tr>
<tr>
<td>- ripples</td>
<td>&lt;100mV</td>
</tr>
</tbody>
</table>
| - efficiency       | 95% (300 – 700W)             
                       | 93.5% (700 – 1000W)          |

<table>
<thead>
<tr>
<th>Boost Mode</th>
<th>42 V ← 14 V</th>
</tr>
</thead>
<tbody>
<tr>
<td>- input voltage</td>
<td>9V ... 16V</td>
</tr>
<tr>
<td>- output voltage</td>
<td>42V (36V ... 48V)</td>
</tr>
<tr>
<td>- output current limit</td>
<td>24A</td>
</tr>
<tr>
<td>- voltage drop at max. current</td>
<td>0.4V</td>
</tr>
<tr>
<td>- settling time</td>
<td>&lt;2ms</td>
</tr>
<tr>
<td>- ripples</td>
<td>&lt;100mV</td>
</tr>
</tbody>
</table>
| - efficiency       | 95% (300 – 700W)             
                       | 93.5% (700 – 1000W)          |
2) DCDC Converter: Specification (realization)

Physical Structure

- very complex structure with heterogeneous parts: uC, power, analog, digital, HF
  - different methods for design and simulation
  - different tools
3.) Model Description : System in Buck Mode (42V → 14V)
3.) Model Description : Graphical Structure of Behavioural Model
### 3.) Model Description: Model of a Switch (VHDL-AMS)

#### Ideal Switch

```vhdl
library ieee; use ieee.math_real.all;
USE work.energy_systems.ALL;
USE work.electrical_systems.ALL;
ENTITY switch IS
  GENERIC ( r_on: real := 1.0e-3;
             r_off: real := 1.0e6 );
  PORT   ( TERMINAL p,m: electrical;
           SIGNAL ctrl: IN boolean);
END entity;

ARCHITECTURE ideal OF switch IS
QUANTITY v ACROSS i THROUGH p TO m;
BEGIN
  IF ctrl USE v == 0.0;            -- switch closed
  ELSE i == 0.0;                  -- switch open
  END USE;
BREAK ON ctrl;
END architecture ideal;
```

#### Real Switch

```vhdl
ARCHITECTURE real OF switch IS
QUANTITY v ACROSS i THROUGH p TO m;
BEGIN
  IF ctrl USE v == i * r_on;       -- switch closed
  ELSE v== i * r_off;             -- switch open
  END USE;
BREAK ON ctrl;
END architecture real;
```
3.) Model Description: Graphical Structure of Behavioural Model
3.) Model Description: Converter for Buck Mode (logical structure)

- **input voltage:** $30 < v_{42} < 50$ ?
- **resistor with current limitation**
- **dynamical part**
- **dynamical load**

$i_{42} = f(v_{42}, i_{14}, v_{14}, \text{efficiency})$

$\text{efficiency} = f(i_{14}) \rightarrow \text{table / polynom. function}$

$v_{14}_{\text{ref}} \rightarrow v_{14}_{0}$ (delay)

SYNOPSYS User Group Saber (SNUG Saber), October, 8th, 2002, Munich
3.) Model Description: Converter for Buck Mode (VHDL-AMS) [1]

```vhdl
Library IEEE;
use ieee.math_real.all;
USE work.electrical_systems.ALL;
ENTITY dcdc_ua14 IS
   GENERIC (
      imax_14   : REAL := 72.0; -- max. current
      ud : REAL := 0.4; -- voltage drop
      tt : REAL := 0.5e-3; -- time constant
      us : REAL := 0.5; -- dyn. voltage drop
      r_on : REAL := 1.0e-4; -- switch on
      r_off : REAL := 1.0e9; -- switch off
   )
   PORT (TERMINAL u42, u14_ref, u14 : electrical );
END ENTITY dcdc_ua14;
ARCHITECTURE simple OF dcdc_ua14 IS
   TERMINAL u14_0,u14_1,u14_2 : ELECTRICAL;
   QUANTITY v42 ACROSS i42 THROUGH u42 TO electrical_ref;
   QUANTITY v14_ref ACROSS u14_ref TO electrical_ref;
   QUANTITY v14_0 ACROSS i14_0 THROUGH u14_0 TO electrical_ref;
   QUANTITY vr14 ACROSS ir14 THROUGH u14_0 to u14_1;
   QUANTITY vl ACROSS il THROUGH  u14_1 TO u14_2;
   QUANTITY ir THROUGH u14_1 TO u14_2;
   QUANTITY vd ACROSS id THROUGH u14_2 TO u14;
   QUANTITY efficiency : REAL;
```

SYNOPSYS User Group Saber (SNUG Saber),
October,8th,2002,Munich
3.) Model Description: Converter for Buck Mode (VHDL-AMS) [2]

BEGIN

efficiency == 88.07 + 1.4319*ir14 - 104.28625e-3*ir14**2.0 + 3.788357e-3*ir14**3.0 - 73.44230e-6*ir14**4.0 + 719.9188e-9*ir14**5.0 -2.80269e-9*ir14**6.0;

i42 * v42 * efficiency/100.0 == v14 * ir14;

IF v42 <= 30.0 USE v14_0 == 0.0;
ELSE IF v42 > 50.0 USE v14_0 == 0.0;
ELSE v14_0 + tt * v14_0'dot == v14_ref;
END USE;

BREAK WHEN v42 >= 50.0;
BREAK WHEN v42 <= 30.0;

IF ir14 <= imax_14 USE vr14 == (ud / imax_14 ) * ir14;
ELSE vr14 == ud + (ir14-imax_14) * 100.0e6;
END USE;

vl == tt/3.0 * 1.0/(imax_14*(1.0/ud + 1.0/us)) * il'dot;

ir == vl / (us/imax_14);

IF vd >= 0.0 USE id == vd/r_on;
ELSE id == vd/r_off;
END USE;
END ARCHITECTURE simple;
4.) Simulation Results (current limitation)
4.) Simulation Results (switched load)
5.) Conclusion: Behavioural Modeling

Benefits:

- simple model structure (independent of physical structure)
- black box modeling possible
- small models compared to physical models (better performance)
- easy use in upper system shells
- use of one universal simulation language for heterogeneous systems

Problems:

- none or only little relation to physical system
- all effects of interest must be considered and modeled separately
- limitation of use because of limited model behaviour
- great experience needed for modeling
5.) Conclusion: VHDL-AMS

**Benefits**

- toolindependent language (IEEE 1076.1)
- different tools available
- different modeling levels for same entity
- multiple use of models
- simple model exchange
- standardized (public) model libraries possible

**Problems**

- only a few tools available today
- actually none tool supports the full standard
- no standardized packages for natures
- no standardized fundamental models