HDL-Based System Engineering for Automotive Power Applications

Dieter Metzner, Georg Pelz, Jürgen Schäfer

Infineon Technologies AG
P.O. Box 800949 , D-81609 München, Germany
dieter.metzner@infineon.com, georg.pelz@infineon.com, juergen.schaefer@infineon.com

I. Abstract:

A Top Down Design Aproach using VHDL-AMS as HDL is outlined using a current control IC as an example.
An essential part of it is the concept of block specifications as a starting point for the implementation process. Behaviour modelling concepts of parts of the architecture are shown, in order to illustrate the ease of using the language standard.

II. Introduction:

In automotive electronics,a lot of good examples for mixed-signal designs on semiconductor component-, PCB- and system level can be found. These designs can benefit from the top-down methodology as have the digital designs in the last 15 years. This is even the case, though one key feature of the digital top-down flow – the circuit and layout synthesis – is not available in mixed-signal design. Still, the top-down flow for mixed-signal designs (in contrast to bottom-up) offers at lot of advantages:

- Executable, i.e. simulatable, specification through behavioural models
- Early assessment of the system functionality
- Continuous refinement of the design, while still the complete system can be simulated at any time
- Availability of a simulatable reference for all design steps
- Early evaluation of testability

On the other hand, the bottom-up flow allows to assess the influence of non-ideal implementation earlier than the top-down flow. All in all, the advantages outweigh the disadvantages. This paper is meant to illustrate this kind of top-down design.
First of all, the circuit description deals with three layers:

- Functional level, e.g. on s or z plane
- Block level, e.g. using units like Op-Amps, ADC, DAC, charge pump etc.
- Circuit level, using classical schematics

Ideally, the development goal is formulated as a modular description on functional level. A first implementation concept is then put together on block level. Here a lot of concept alternatives can be assessed quickly. Finally, the third step transforms the block description to a circuit of schematic level. By the way, the functional description may be omitted without sacrificing the advantages of the further steps.

Similar to the evolution of design methods for digital ICs, Analogue Circuit Designers have followed, adopting CAE tools for Design, Layout and Verification [Sommer], [Bjornsen]]. However, the analogy is being limited by the absence of general methods for synthetization of circuits. The main reason for this fact in turn is the absence of a complete library of reusable basic blocks, which are in the analog world much more difficult to define as in the digital case. The high number of degrees of freedom implies a considerable increase of chip size which is seldom tolerable.
A common misunderstanding of designers concerning a potential analog synthetization process is, that there will be no algorithms „inventing“ new circuit blocks but „just“ a top down approach, starting from a given , modular (!) specification descending into the hierarchy and finally placing a library element for a basic reusable function. In analogy to digital reuse-blocks like buffers, gates, counters etc., in the mixed signal domain charge pumps, drivers, comparators, measurement blocks and so on have to be added to a component library.
Now some words on the languages and tools necessary to implement the top-down flow:
Like in the digital world, the basis must be a standardized HDL which is suitable not only for simulation and verification but also for specification. Some first approaches in the automotive arena already use this concept of executable specifications was.[sae02], [sae03].
HDLs that support such a Mixed Signal Design Flow are available: VHDL-AMS which has been standardized by the IEEE in 1999 and Verilog-AMS, for which the standardization is under way. Driven by European carmakers and suppliers, a tendency towards VHDL-AMS has been established. Moreover, SAE (Society of Automotive Engineers) and VDA (Associa-

tion of German Automotive Industry) have adopted VHDL-AMS (besides MAST) as their specified modeling language.
In *[bmas02]*, a preliminary status of tools supporting the language was given.
At present, it can be stated that apart from details, the basis on the tool side has been created to enable a top down design flow with system perspective.
However a few more items have to addressed by Analog Chip designers before a top down design flow can be implemented:

- the generation of reusable module libraries for basic functions
- training to consider circuits on a higher abstraction level and
- exercise the usage of HDLs.

In *[sae03],* a new possibility for knowledge exchange to achieve system understanding on both sides – component supplier and system manufacturer – has been described using the concept of executable specifications: By exchanging models instead of implementation details, both partners can agree on a verified system architecture before entering the component design flow.
On the side of the semiconductor development this leads to an implementation based on circuit blocks, on the system side, software implementation is gaining more and more importance. The overall motivation is of course the idea of "First-Time-Right", since innovation cycles are constantly shortened and the semiconductor component is becoming the bottleneck of the system development.

In this paper, we illustrate our design approach containing system definition, architecture development, block specification and circuit implementation in the system context by an example of a current controller IC. These are typical representatives of mixed signal ICs that are applied in various automotive applications like:
- valve coils for engine management, braking or stability control systems,
- Stepper motor drivers for headlight control systems
- Induction motors for steering or
- alternator control.

III. System Definition and Requirements

The general task for the controller IC is to force a time dependent current defined by a reference signal given by a higher hierarchy level into a load.
This load has generally mainly resistive/ inductive characteristics; additionally these can be nonlinear (valve coils) and in the case of motor loads can also contain back EMF (Induced Voltages).
One important specification item, which will be considered, is the current control accuracy. Depending on the particular application the

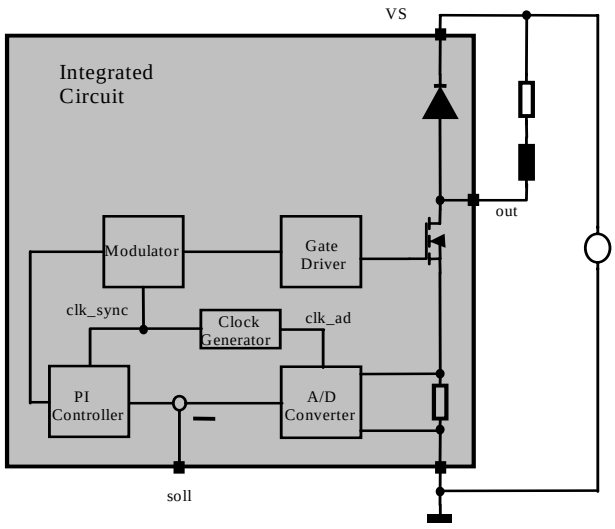

Figure 1: Block description of a control system

requirements for it can vary from 5 to 20% of the nominal value.
Since the typical current levels of such loads can reach from 1 to 500A, PWM switched mode regulators have to be used instead of linear regulation concepts.
Figure 1 shows a first architectural approach of the IC under definition:
In the selected example the required current value is small enough to integrate the MOSFET switches and the corresponding freewheeling diodes on one chip, therefore also the measurement system can be integrated.
Besides this interface to the analogue power side of the system the first architecture contains:

- the driver stage for the Mos Transistor with protection for over current and over temperature
- Open load and Under voltage detection
- PWM Modulator
- A time-discrete Controller
- A/D Converter for the current measurement
- SPI – Interface for diagnosis output and input of the required current value.

1. General control goal

In a PWM control concept, the desired current can only be specified by means of an average value, practically over one switching period. The most important specification item is of course the control accuracy.

The dynamics of the output current is another item, however basic limits are set by the limited output voltage and the fact that an actual impact of the controller is possible only once in each switching cycle.
Time-continuous and linear descriptions of the system are therefore inadequate.

IV. Behaviour modelling of the components

1. The modeling philosophy for the Power–DMOS and its driver Stage

The modelling philosophy for the Power–DMOS and its driver stage has been described in detail in [SAE2002] and [BMAS2001] and is therefore only treated here with respect to the specification process. The sizing criteria are mainly given by:

- the load current / voltage requirements and the load impedance
- thermal aspects and
- short circuit considerations

With respect to the control accuracy, mainly the delay time from the digital switching command to the voltage transient at the load output is of importance as a specification item (cf. Figure 2) . This can be fulfilled in a straightforward manner by adjusting the charging and discharging paths for the MOS Gate.

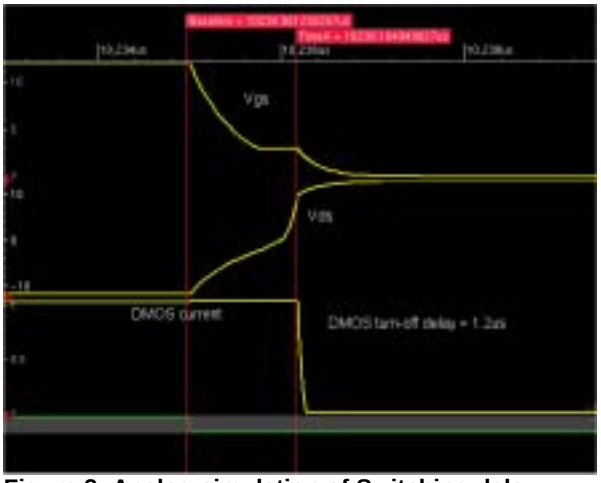

Figure 2: Analog simulation of Switching delay

2. Specification of the Measurement Block

When using PWM with a chopper (step-down) configuration, usually shunt resistors in the current path or in a portion of it are applied (Figure 1) . There are three possible locations where current sensing can be implemented:

- In the lowside path of the halfbridge, i.e. in series with the Power DMOS Transistor
- In the highside path (in series with the freewheeling diode
- In the load connection

All of these alternatives have their pros and cons; generally a measurement in the load path shows the disadvantage of fast transients on the reference potential (out), therefore as a start a sensor in the lowside path (active switch) is considered in detail.
It is characterized by the fact that a valid signal is only available during the on state of the DMOS switch . For this reason, a hysteresis control scheme cannot be directly applied and a fixed frequency modulation scheme has to be used.

For the analog measurement block, accuracy and dynamic characterisitics are the key specification items. In order to verify those, process parameter statics and temperature dependent models have to be used for transistor level simulations of the measurement principle.

3. A2D conversion

In a digital implementation of the control loop, the discretization of the current signal is a key item. In the block specification we define a value range (Imax), a discretization interval (dI) and a maximum Slope of the signal (dI_by_dt_max) which the converter can follow.
In the Top Down Design Approach, next a conversion algorithm has to be selected, in our example a counting method (c.f. Box 1)

Then main part of the converter model is made up by two processes, ***p1*** which performs the counting of the output variable (count) and ***p2*** which controls the counting direction (***v_a***: analog input, quantity across p to m).

In Figure 3 a and b the typical situation is shown in a close-up of one switching cycle in a stationary state:
During the on phase of the switch, the counter follows the rising signal with the typical bit-oscillations. When the en signal is switched to low, no valid measurement is available and the counter is stopped. At the next turn on, the current signal starts at a lowered value (decay during the freewheeling phase) and the converter counts down at its maximum speed given by the clock frequency, which results in a certain delay (~20 μs).

```
p1:process(clk,en,reset,direct)
  begin
    if clk 'event and clk 'last_value='0'
        and en='1'  and direct='1' and reset='1' then
    count<=count+1;
    elsif clk 'event and clk 'last_value='1' and
en='1'
          and  direct='0'       and reset='1' then
   count<=count-1;
   elsif en='0' then
   count<=count;
   elsif reset='0' then
   count<=0;
   end if;
end process;

p2:process(count)
        variable quant: real:=0.0;
        begin
        quant:=v_a/v_quant;
        if real(count) > quant then
        direct<='0';
        else
        direct<='1';
        end if;
end process;
```

Box 1: processes of the counting A/D - Converter

Figure 3 a: ADC holds output during off-state, Counting down at maximum speed after turn on

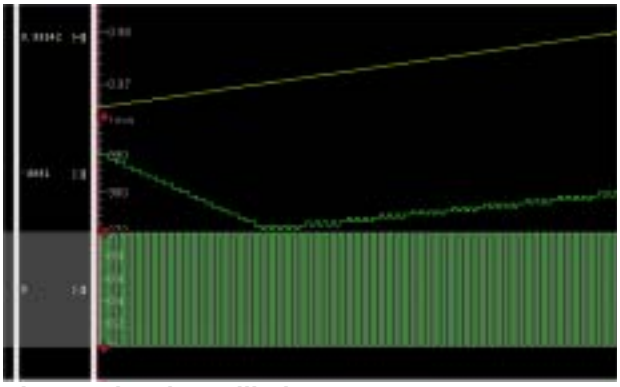

Figure 3 b: Bit oscillations at current ramp

A more intelligent approach for a dedicated A/D converter could use an additional register which stores the out put signal some delay time after each turn on event. This value can then be used to initialize the counter to a value closer to the next valid measurement signal.

The necessary changes in VHDL-AMS Code are minimal, and the impact on total system behaviour can be easily demonstrated (Figure 4)

Figure 4: modified ADC counter with initialization

4. Averaging of the current

Besides the sensor (shunt) signal A2D conversion , the description of the average current also includes an algorithm to estimate the average value of the load current without having information during the freewheeling phase. The simplest strategy is to determine the end of the pulse by the instantaneous current value, which means a peak current regulation method.

Since the specification requires a control of the average current value, a better approach is to assume linear current ramps (which is fulfilled well for a high switching frequency) and use the mean value during the on-phase only.

A further improvement in case of highly nonlinear loads and/or low switching frequencies is conceivable with an observer approach, where the current during the “blind phase” is estimated by means of a self adjusting load model .

The explicit implementation of the averaging mechanism in our example is not necessary, since it is implicitly contained in the integral component of the subsequent controller.

5. Discrete Controller

The time discrete controller is a modification of a standard PI controller type with the adjustable (= to specify) parameters ***c1*** and ***c2*** and the ports ***signal output : out real;*** ***signal input : in real;***

The integral component of the output (***inte***) is controlled by an enable signal (***en***) in such a way, that it stops integrating as soon as the measured signal is absent in the off phase.
Thus, the controller will automatically adjust the average during the on-phase to the desired reference.

```
p1:process(clk)
 variable pro : real;
 variable inte :real;
  begin
   if clk 'event and clk 'last_value='0'
                and en='1' then
         integral<=integral+input*1.0/tp;
   end if;
   if clk 'event
             and clk 'last_value='0' then
             pro:=input*c1;
            inte:=input*c2*integral;
         output<=pro+inte;
   end if;
end process;
```

Box 2: process of time discrete PI - Controller

6. PWM Modulator

The principle of the PWM is based on a comparison of two signals (***process switch***): The required ***duty_cycle*** (Output of the controller) and a triangular source implemented as an integer signal ***v_tri***, counting from 0 to 1 in a specified increment (***process p1***) every switching period of the PWM frequency.
Besides this counting increment the block specification contains of course the switching frequency. The switch control signal is ***cmd***.

```
switch: process
        if (duty_cycle >= v_tri*fs/fsync) and
        (cmd = '0') and clk 'event and
        clk 'last_value='0' then
                        cmd <= '1';
        elsif (duty_cycle <v_tri*fs/fsync) and
         (cmd = '1') and clk 'event and
        clk 'last_value='0' then
                        cmd <= '0';
        end if;

        wait on clk;
end process;
p1:process(clk)
        begin
        if clk 'event and clk 'last_value='0' and
         v_tri<fsync/fs then
                v_tri<=v_tri+1.0;
        elsif clk 'event and clk 'last_value='0' and
         v_tri>=fsync/fs then
                v_tri<=0.0;
        end if;
end process;
```

Box 3: body of PWM – Modulator

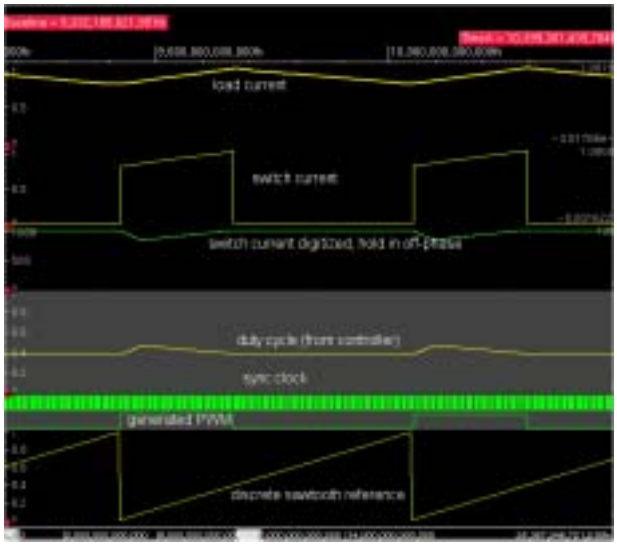

Fig. 5: principle of the discrete PWM modulator

8. Overall system performance

In Figure 6, a verification example of a complete control loop is given. At this point an investigation of the influence of different parameter is enabled in order to adjust component specifications.
Figure 7 gives just one example by varying the synchronization clock for the controller and the PWM – modulator, which has a direct impact on the accuracy for the duty cycle adjustment.
After reduction from 1MHz down to 100kHz the resulting load current shows typical aliasing effects.

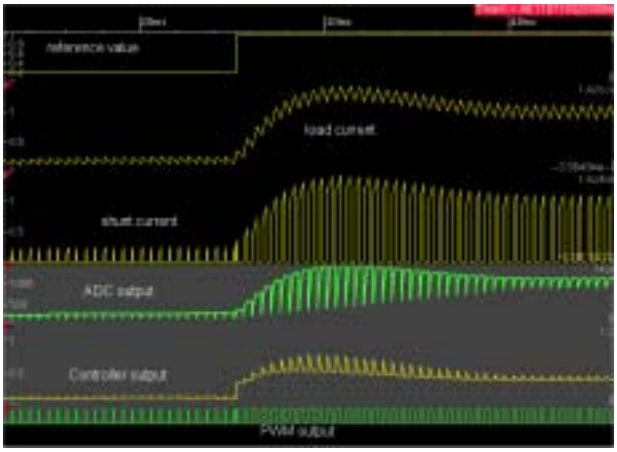

Figure 6: Step response of the control loop (0.2A => 1.0A)

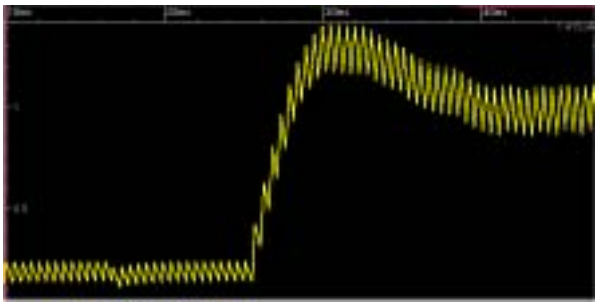

Figure 7: Aliasing effects on the load current with slower synchro-nization clock 1MHz => 100kHz

V. Conclusions

An example for concept development with the aid of VHDL-AMS behaviour models has been given. It has been confirmed that the maturity of simulations tools is sufficient to fully support a top down design flow.
The example also shows that behavioural models for particular modules cannot generally be provided by tool vendors since those are very soon reflecting application specific algorithms and concepts.
Modelling expertise has to be built up in design groups in order to take full advantage for the concept development; in the ideal case a concept engineer is able to generate VHDL- AMS models by himself.
What´s left to be done for the tool vendors in the next future, is to standardize and include libraries of basic elements into their tools which enable the exchange of testbenches and also gives a quick introduction to concept simulations for graphics-oriented circuit designers.
These basic elements are of the simplest kind out of electrical, thermal, mechanical and control system domains but are nevertheless essential.
Entity declarations, standardized by IEEE should fulfil these requirements.

Another topic of constantly rising importance is the inclusion of processor cores and/or firmware into the simulation of a mixed-signal circuit or system.
A lot of methods have been devised to cope with the problem in a purely digital context in the past. Most tool vendors have addressed it as a tool problem, which led to simulator backplane approaches. This in turn resulted in large simulator compounds, which are difficult to handle. Especially in the mixed-signal area, there is a need to hide the processor core or the firmware in a model. This is easily accomplished as the sequential nature of firmware execution perfectly fits to the sequential execution of C-routines attached to the mixed-signal simulator's C-interface.

References:

[sommer]:
Sommer et al:
“From System Specification to Layout: Seamless Top Down Design Methods for Analog and Mixed Signal applicatoions”
Proc. DATE 2002.

[bjornsen]:
Bjornsen et al:
“Behavioural modelling an d simulation of Mixed Signal System-On-A-Chip using System C”
Proc. Of Analog Int. Circuits and Signal Processing 2003, pp 25-38

[bmas01]:
Metzner, Schäfer, Xu:
“Multi Domain Behavioral Models of Smart-Power ICs for Design Integration in Automotive Applications”,
BMAS 2001, Santa Rosa CA, USA

[bmas02] *:*
Metzner, Schäfer:
“Architecture Development of Mixed Signal ICs for Automotive Application with VHDL-AMS”,
BMAS 2002, Santa Rosa CA, USA

[sae02]:
Metzner, Schäfer Langfermann, Büdding:
”Integrated Mechatronic Design and Simulation of a Door Soft Close Automatic with Behavioral Models of Smart Power ICs”, SAE Conference 2002

[sae03]:
Metzner, Schäfer, Genta, Manzone, Santero:
Investigations of a Direct Injection System with a “Simulatable Specification” of Smart Bridge Driver ICs. , SAE Conference 2003