Architecture Development of Mixed Signal ICs for Automotive Application with VHDL-AMS

Dieter Metzner, Jürgen Schäfer

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

Abstract:
A Methodology for the specification, design and verification of Application Specific Standard - IC -products (ASSP) is presented. The application of a "simulatable specification", written in the Hardware Description language VHDL-AMS, is the key feature of this method, which uses a modular approach with respect to specification, behavioural models and circuit topology.
The paper describes this process by an example of a Smart Power IC.

I. Summary:

Electronical Control Units (ECUs) for Cars are steadily replacing controlling functionalities in vehicles which have been recently fulfilled by mechanical or electromechanical components. Some well known examples are ignition and injection systems substituting camshafts and interrupters, anti blocking systems or comfort functions.
Associated with enhanced functionalities are increasing requirements on data transfer and signal processing capabilities.

Infineon´s strategy in the automotive market segment is to define products in terms of target applications rather than individual customers, indicated by the keyword Application Specific Standard - IC -products (ASSP – in contrast to ASIC)

Generally the system environment is of a mixed signal nature (figure 1) , dominated by analogue behaviour of actors and loads. Such a mixed signal system is controlled by a microcontroller which transmits and receives digital data from Smart Power ICs and sensors, typically input-, status- and sensor-signals. The analogue/digital interface is located in the Smart Power IC and the sensor. The smart power IC converts the digital input signals to analogue current, voltage characteristics and simultaneously sends digital status information to the microcontroller. On the other hand the sensor converts physical parameters like pressure, distance and temperature into digital information. An electromechanical transformer and a mechanical load close the feedback loop by delivering physical quantities to the sensor.

The future of the development process of ASSPs for Applications in Automotive ECUs has to face several challenges with impact on development tools:

a) The increasing functionality and hence complexity requires advanced simulation tools: In the past, the most commonly used approach to simulate mixed signal systems was "Co-Simulation", where digital and analogue parts are simulated by individual simulators, linked by data exchange and command interfaces. A clear disadvantage are the inherent incompatible modelling languages and the low simulation speed.
b) Harsher environments require a deep understanding of Safe Operating Area (SOA) limits and failure mechanisms: the constraint of cost reduction implies higher utilisation of PCB area which in turn means smaller chip area, higher peak temperatures and a reduced safety margin. In the EMI (Electromagnetic Interference) domain, efforts for disturbance filtering have to be traded off with signal quality. Experimental verification of such effects is indispensable, though time-consuming and expensive. Simulation can help to guide efficient test series.
c) The optimisation of an IC has to be done from a system perspective, integrated in the ECU application, similar like the latter is integrated into a vehicle network. A comprehensible example is the question of optimal partitioning between data processing- and controlling/driving functionalities: A true comparison of alternatives implies that both, component- and system level - can be simulated with the same tool.
d) reduced time to market

- The consequences of these challenges for IC suppliers are among others:
- An early discussion of the target application is necessary with key customers
- supplier sets up the IC specification in joint discussions Key customers and
- A high time pressure exists on Engineering samples in order to verify the system

- The analogue “design gap” for analogue ICs grows: System simulation on transistor level is no longer feasible with increasing complexity. A high risk of IC redesigns exists because of missing top-level simulations.

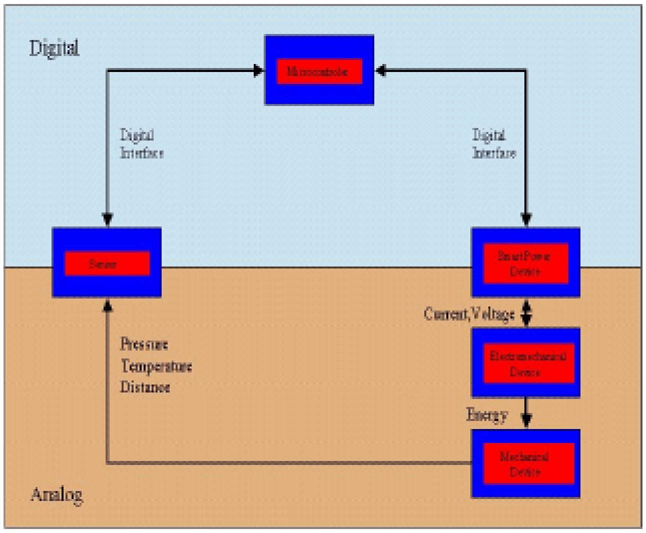


Figure 1: System representation of a typical automotive application

II. Architecture development with executable specifications in VHDL-AMS

In the following, a modular approach as a solution is illustrated, identified by:

- The development process shows a modularity with respect to. :
  a) The specification,
  b) the structure of behavioural Models (which are of a mixed–signal nature) and
  c) the circuit topology.
- A library of basic functionalities in different levels of abstraction and complexity is built
- A simulatable specification enables a check of important functionalities in the application environment (Figure 2.)
- The understanding of interactions between individual specification items and between the modules is improved
- An early detection of weak points in the concept is supported.

At the moment the functionality of Smart Power ICs are gaining more and more complexity. As a consequence, analogue and digital circuits are increasing in size. Especially the integration of digital control structures becomes more and more important in future. For these aspects a fully-integrated top-down development method is necessary. Up to now such a development process was impossible because there were no efficient mixed signal HDLs with single kernel network simulators.

In the case of products with low complexity, a bottom-up design method is generally justified. With increasing complexity, however, the architecture of a SoC (System on Chip) IC should be defined by the top-down method.
At the beginning the architecture is described by all sub-functions of the product. These subfunctions are classified according to their nature:

- Digital functions, like I/O interfaces and signal processing,
- Conversion blocks A/D and D/A
- Analogue functions like gate driver and protection and diagnostics and
- Power output stages (Figure 2).

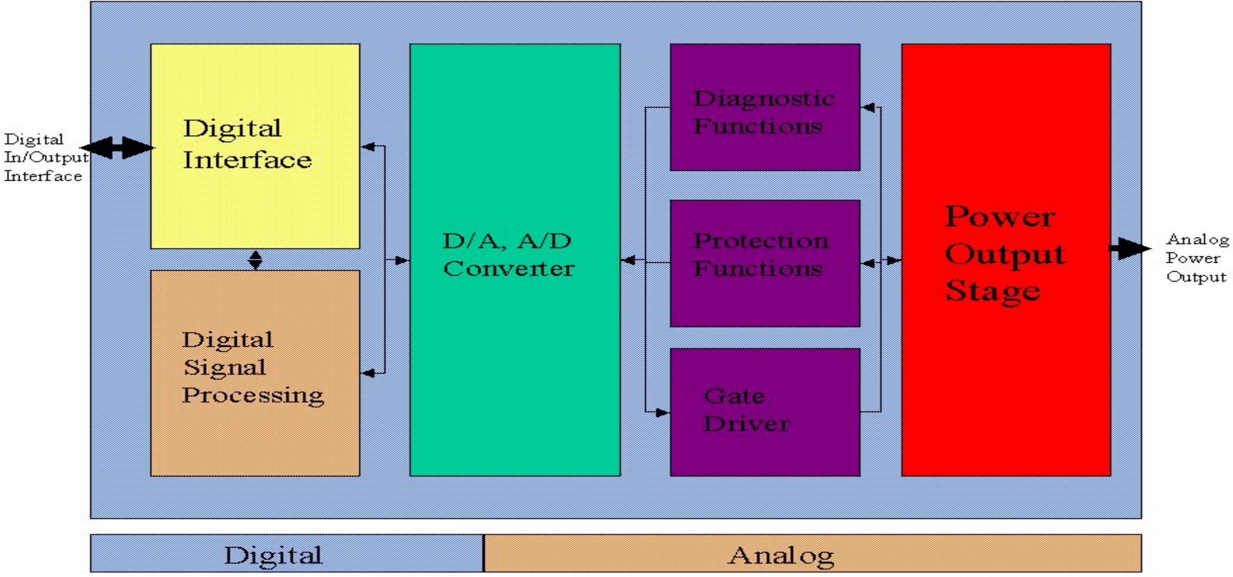


Figure 2 Functional block diagram of Smart Power IC

Presently the specification of such an IC was done by means of a list containing ratings and characteristic values at predefined boundary conditions.

This form of description is inherently insufficient, since it is never complete and accurate.

As a consequence there are a lot of misunderstandings between the semiconductor supplier and the customer which often causes redesigns.

An executable (simulatable) specification which contains the behaviour models of all sub-functions can improve the situation considerably:

Ideally it can answer all relevant questions of the application in the predefined operation range by a behaviour model, simulated in the target application

In the second step this simulation model of the IC architecture can be used for the concept verification.

In an early phase of the development process it is the basis of feasibility studies, investigations of interactions between features, worst case simulations, sensitivity on manufacturing processes as well as SOA considerations.

Smart Power ICs are used in many different applications like mechatronic systems, light modules, injector valves, ignition systems and so on with complex load characteristics

In order to include such complicated interactions, the sub-functions should be modelled in two different approaches using different abstraction levels:.

- Features which influence the output characteristics of the IC, like the power output stage, should be modelled with relevant physical equations because it is responsible for the exact representation of the switching behaviour (EMI, timing analyses) and the self-heating process.
- Features which describe the internal signal processing, like analogue protection- and diagnostic functions; digital signal processing- and interface functions should be modelled in behavioural parts only.

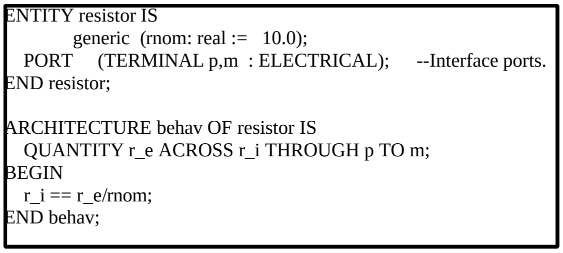
The modular behavioural description of such a SoC comprises many different domains:

System of Differential equations in various physical fields (electromagnetic, Thermal, mechanical, etc….), Conservative systems / Non Conservative Systems, Digital / analogue.

In [1], a detailed description of this modelling approach is given.

III. Basic VHDL- AMS Models

The ideal HDL for this description process is VHDL-AMS, since it fulfils all these requirements in a single-kernel mixed-signal simulator. In 1999 VHDL-AMS was standardised by the IEEE as VHDL 1076.1. Therefore a VHDL-AMS model can be developed independently of the simulator. In contrast to other HDL languages like MAST or Verilog-A, a VHDL-AMS Models consists of an entity and at least one architecture (Box 1). The entity describes the connection points, their nature and the adjustable parameters. The architecture on the other hand contains the functional description of the model. Since each entity can refer to several architectures, different abstraction levels can be defined in a model.



```
ENTITY resistor IS
          generic  (rnom: real :=   10.0);
   PORT     (TERMINAL p,m  : ELECTRICAL);     --Interface ports.
END resistor;

ARCHITECTURE behav OF resistor IS
   QUANTITY r_e ACROSS r_i THROUGH p TO m;
BEGIN
   r_i == r_e/rnom;
END behav;
```

Box 1: VHDL-AMS model of a resistor

Digital integrated circuits have been developed with VHDL for ten years. Previous mixed-signal HDLs like SABER MAST or Verilog-A allow digital behaviour description on a low level. These HDLs have not implemented features like bit vectors, wait statements, processes, etc. and the synchronisation is not ideally integrated. Therefore the description of complex digital circuits has been impossible or has needed a lot of effort. Since VHDL-AMS is the analogue extension of VHDL (IEEE 1076), all digital VHDL statements are allowed.

Box 2 shows an example of a VHDL-AMS model of an SPI input shift register. This model has tree input signals (**clk**, **cs**, **sdi**) and one output signal vector (**shreg_out**).

The functionality of the shift register is implemented in the process. The process is active if the signal **clk** or **cs** is changed. The signal **clk** determined the system clock and **cs** determined the chip select signal. The Information on the **sdi** (serial data input) signal are accepted if the **cs** signal is low and the **clk** signal is on the rising edge. The source code inside the process is handled sequentially and the VHDL code outside of the process is handled parallel (concurrent statements).

In Figure 3 the timing diagram of the described signal processing is shown.

```
entity shreg is
        generic(n:integer:=15);
        port( clk, cs, sdi : in std_logic;
             shreg_out : out std_logic_vector(n downto 0))
end entity;


architecture behav of shreg is

        signal out_in : std_logic_vector(n downto 0);


        begin
        process(cs,clk)
                begin

                if (clk 'event and clk = '1' and cs= '0') then
                for i in n downto 1 loop
                shreg_out <= out_in;
                out_in(i) <= out_in(i-1);
                end loop;
                out_in(0) <= sdi;
                shreg_out <=out_in;
                end if;
                end process;
end architecture;
```

Box 2 VHDL Code of SPI input shift register

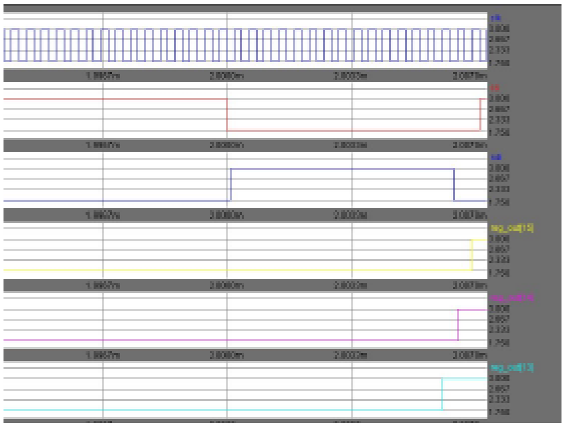


Figure 3: timing diagram of the SPI input shift register

Modelling the physical domain:

In Figure 4 a/b, an example of a lumped thermal model is given with its VHDL AMS implementation

Since thermal problems are subject to conservation laws, a netlist representation is most suitable.

Besides the thermal network, one basic element, a thermal capacitance is defined (thermal resistance not shown). Once the entity declaration is available, refinements starting from the most primitive case – the “linear” architecture are conceivable.

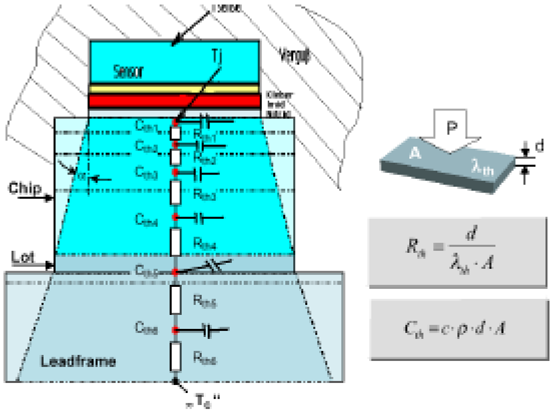


Figure 4a: lumped thermal model of a power device

```
 -- declaration of thermal capacitance

ENTITY cth IS
          generic (ct : real :=10.0);
          port (TERMINAL p, m : thermal);
end entity;

 ARCHITECTURE linear Of cth Is
          QUANTITY dT ACROSS pow THROUGH p To M;
          BEGIN
          pow == ct * dT'dot;
 END ARCHITECTURE;

-- thermal network
 ENTITY thermal_net IS
 END ENTITY thermal_net;

 ARCHITECTURE cauer OF thermal_net IS
          TERMINAL  tj, t1, t2, tamb            : thermal;
 BEGIN

 r1 :   ENTITY rth (linear)           GENERIC MAP (rt => 1.0)
                              PORT MAP ( p => tj, m => t1);
 c1 :   ENTITY cth (linear)           GENERIC MAP (ct => 1.0)
                             PORT MAP ( p => tj, m => thermal_ground);
 r2 :   ENTITY rth (linear)           GENERIC MAP (rt => 1.0)
                             PORT MAP ( p => t1, m => t2);
 c2 :   ENTITY cth (linear)           GENERIC MAP (ct => 1.0)
                             PORT MAP ( p => t1, m => thermal_ground);
 r3 :   ENTITY rth (linear)           GENERIC MAP (rt => 1.0)
                              PORT MAP ( p => t2, m => tamb);
 c3 :   ENTITY cth (linear)           GENERIC MAP (ct => 1.0)
                               PORT MAP ( p => t2, m => thermal_ground);
```

Box 3: implementation of a thermal model

Elements are connected to a network by port “maps” In the same manner all physical meaningful phenomena can be modelled with adequate simplifications

The most relevant domain for automotive electronics is of course the electrical “nature” followed by thermal, magnetic or mechanical, depend on the target application.

The power of this netlist formulation is that complex interactions between the elements are included automatically by terminal connections. A classical example in the field of power electronics is a Power MOS Transistor switching a resistive/inductive load.

Transient behaviour strongly depends on the load parameters, not only with respect to rise- and fall-time, but yields qualitatively different results if LC oscillations are generated (cf Fig.4)

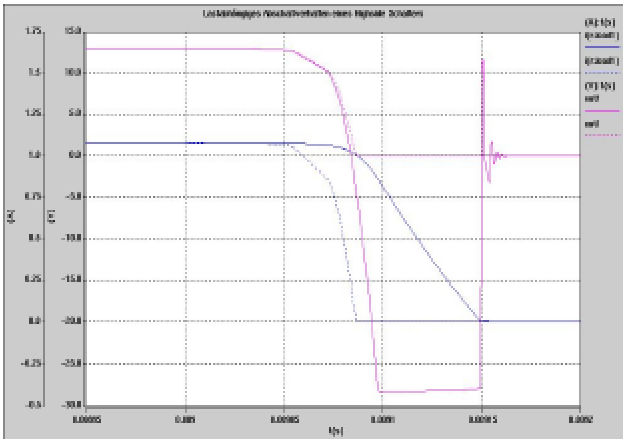


Figure 4: Load dependent switching transients of Power- MOS Transistor

Mixed Mode Models

An essential part of a behavioural model is the interface between the physical world and functional parts realised by digital behaviour. The corresponding hardware elements are e.g. D/A- or A/D converters.
One of the challenges for a simulation tool is the synchronisation of the analogue and digital solution algorithms. In this section we demonstrate two basic, frequently used elements in models of Smart Power Devices :

An input buffer with hysteresis (electrical terminal input, std_logic output) and
a digital controlled gate-driver switch. The latter comprises a std_logic input (en) and an electrical terminal output (gate).
The simple application example in figure 5 uses both sub-models to describe a voltage controlled switch with the digital intermediate variable <*en*> . In the case of Smart Power ICs such a digital control signal is the result of a number of more or less complex operations (eg. undervoltage, overtemperature, PWM current controller, etc.)

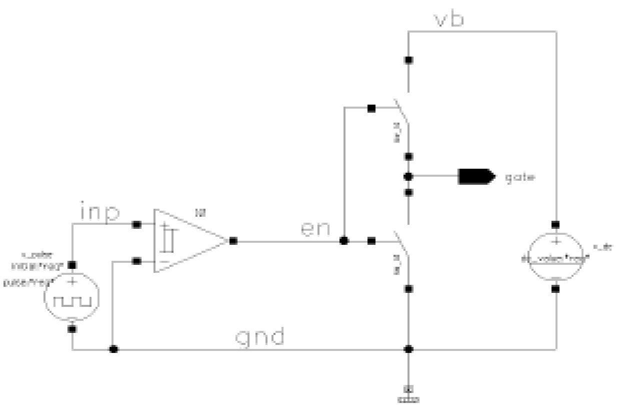


Figure. 5: basic structure of controlled switch.

```
entity input_buffer is
        generic(limith : real := 3.5;
        e         limitl : real :=2.0;
                  rin : real :=20.0e3);
        port(terminal inp, gnd : electrical;
                  outp : out std_ulogic);

end entity;

architecture behav of input_buffer is
        quantity vin across i_in through inp to gnd;

        signal outl : std_ulogic;
  signal outh : std_ulogic;

        begin
                process
                begin
                if vin 'above(limith) then
                        outh <= '1';
                        else
                        outh <= '0';
                        end if;
                        wait on vin'above(limith);

                end process;

                process
                begin
                if not vin 'above(limitl) then
                        outl <= '1';
                        else
                        outl <= '0';
                        end if;
                        wait on vin 'above(limitl);
                end process;

                process(outh,outl)
                begin
                if(outh= '1' and outl = '0') then
                        outp <= '1';
                        else
                        outp <= '0';
                        end if;
                end process;

                i_in==vin/rin;

        end architecture;
```

Box 4: VHDL-AMS Code of input buffer with hysteresis

The essential statements for this interface are “wait” and “break”-statements and the attribute “x‘above“.

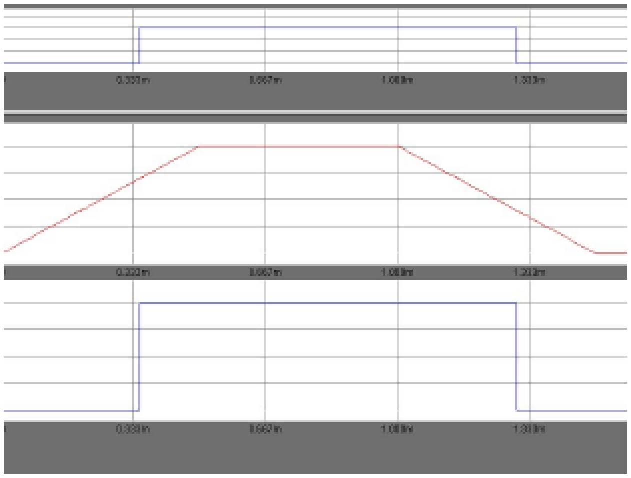


Figure. 6: Testbench result for input buffer.

```
entity gate_driver is
        generic(ron_c : real :=1.0e3;
            ron_dc : real :=1.0e3;
            roff : real :=10.0e6);
        port(terminal vs, gnd, gate : electrical;
             en : in std_logic);
end entity;



architecture behav of gate_driver is
    quantity vgs across idc through gate to gnd;
    quantity vbg across ic through vs to gate;
    signal res_c : real;
    signal res_dc : real;

    begin

        process(en)
            begin
                if(en='1') then
                res_c <= ron_c;
                res_dc <= roff;
                elsif(en='0') then
                res_c <= roff;
                res_dc <= ron_dc;
                end if;
        end process;

        break res_c => res_c;
        break res_dc => res_dc;

        ic == vbg/res_c;
        idc == vgs/res_dc;

end arcitecture;
```

Box 5: VHDL-AMS Code of digital controlled switch

Based on a modular description, different concepts of the IC can be tested in the application environment.

In order to support a continuous development process (resolution into transistor level), VHDL-AMS simulators should also be able to include SPICE – like standard libraries, because it seems devious to rewrite all existing model libraries

A few CAD Vendors have addressed this requirement by either generating intermediate description formats from various input formats or by including compiled VHDL-Code into the respective circuit simulator by standard modelling interfaces.

IV. Actual status about VHDL-AMS

At the moment there are different simulator suppliers which support the VHDL-AMS standard. But none of these suppliers are actually supporting the complete IEEE 1076.1 (VHDL-AMS) standard. The core competence of most simulator suppliers is analogue simulation.

Normally, in these simulators, the complete digital VHDL standard (IEEE 1076) like bit_vectors, configuration and so on has been implemented.

Another great problem of these simulators is the synchronisation between the analogue equations solver and the event controlled digital simulator core. A frequently observed problem is the feedback of digital signal events to analogue quantities. The reason for this is the non-ideal or missing implementation of the break statement.

The objective of the VHDL-AMS standardisation was a simulator-independent HDL. Therefore the use of a mixed-signal simulation model should be ensured on different simulators. But the reality is more difficult. In particular complex analogue models which were developed on a certain simulator don't necessarily work on the any other simulator. The obvious reason is the different software implementation of linearisation and numerical integration algorithms like "Newton-Raphson" and "Euler-Backward".

The exchange of simulation models between different VHDL-AMS simulators is complicated by the absence of standardised libraries and packages. At the moment each simulator supplier uses different analogue libraries and packages. Reference nodes such as ground are specified with different notations.

As a consequence, a VHDL-AMS model must be modified for the use in different simulators.

For these reasons a model coding by compiling is not possible at the moment and the model exchange is hampered.

An efficient use of VHDL-AMS is complicated by the absence of standard component libraries (SPICE Library) with elemental devices like voltage- and current sources, resistor, capacitor, inductor and so on.

Many simulator suppliers have integrated such libraries into their simulator.

Alternatively, the user has to develop such basic models by himself. The consequence of this is the model-incompatibility between simulators.

A possible solution to the problem could be VHDL-AMS packages standardised by the IEEE organisation with all natures and reference points. For a basic component library

(SPICE library) the entities with connection points, natures of the connection points and the adjustable model parameters should be defined by IEEE as well.

The simulator supplier in turn could concentrate on the development of architectures to these entities, with a model description optimised for the respective simulator core [2].

V. Conclusion

For a fully integrated design method for SoC there seems to be no alternative at the moment to a standardised, high abstraction level, mixed signal HDL .

Among the two candidates, VHDL-AMS and Verilog-AMS, the first is favoured by Infineon.

The reason is a strong emphasis on SABER (MAST™) models in the automotive industry, which is the presently used language for model exchange.

Although MAST™ is not suited for future requirements, there will be a synergy with existing libraries and modelling methods.

Despite all actual deficiencies of the implementation of VHDL-AMS, it can be stated that there is a great opportunity for a generally accepted HDL, both suited for architecture development as well as for model exchange between customers and semiconductor suppliers.

The purpose of this paper is to encourage potential users to experiment with existing tools, start pilot projects with VHDL-AMS models and assist CAD Vendors by feedback of experiences with the tools.

References:

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

[2] : Moser et al:
Automotive Power Net and Customer – Supplier Model Exchange,
to be published (IEEE journal of Solid-State Circits)