

ANALOG HARDWARE DESCRIPTION LANGUAGE AND ITS RELATIONS TO VHDL

Karel Vlček and Jan Popelek
Dept. of Electronics and Telecommunication
Faculty of Electrical Engng. and Computer Science
Technical University Ostrava, Czech Republic
karel.vlcek@vsb.cz, jan.popelek@vsb.cz

Abstract

Primary motivations for analogue hardware description language (VHDL-A) is to support the modelling of physical systems. The VHDL-A must therefore allow to model the physical conservation laws, such as the energy conservation law, which states that energy can neither be created nor destroyed, but it can only change its form.

Keywords

VHDL, VHDL-A, Circuit description, Circuit simulation, Analog, Digital, Mixed D/A

1. Analogue Hardware Modelling Basis

The VHDL-A is a description and simulation language. Design objectives are related to the scope of VHDL-A, also the relations with VHDL. The VHDL-A must be suitable for the description and simulation of digital, analogue, and mixed digital/analogue systems at several abstraction levels (behavioural, macromodel, functional and circuit levels for analogue systems).

The VHDL-A must be able to support any design methodology and be technology independent. It must provide mechanism that allow the analogue and digital behavioural parts of a mixed description to interact. The model of the interface between the analogue and digital descriptions should be customisable by the user. There are two fundamental mechanisms for modelling physical systems: behavioural modelling and structural modelling.

2. Design from Behavioural Modelling

Behavioural modelling requires the language constructs to describe the physical laws of a system, in particular basic components of the systems, by mathematical equations or assignments. This modelling

concept is based on equations that describe the relations of constituent components. The distinction between analogue and digital systems occurs whenever the designers selects how the system or part of it will work. The digital behaviour is event-driven and therefore takes care of a limited number of well defined states, while analogue behaviour continuous and take care of entire waveforms.

The of a large system may involve more then one team, and each team may base their design on different methodologies. The design of large systems may involve use of existing design units. This is the key to managing the complexity of large design. The design may use top-down and bottom-up hierarchical design methodologies. The top-down design starts from abstract descriptions that intend to capture essential aspects of the design and ignores implementation details. These abstract descriptions are progressively refined until they reach a point where one implementation that satisfies the design requirements may be directly inferred from the description.

The bottom-up approach takes the opposite way, by starting from a set of descriptions of pre-defined components and assembling them to form more complex components. The top-down design is usually associated with synthesis, while the bottom-up design is usually associated with verification. In the real design the both top-down and bottom-up approaches are used in mixture. The modelling concept can be based on the hierarchical assembly of low level, circuit components, such as controlled sources. Only a very necessary number of these components are used to approximate the input/output behaviour. The structure of macromodel is usually simpler than the underlying hardware.

3. Structural Modelling

The structural modelling require the language mechanisms to support hierarchical composition of basic components in a way to obey the physical laws. The macromodel level may not be considered as an abstraction level as such, but rather as a specific style of description. The description of the system structure is usually built from a connected network of components. Finally, components have to encapsulate some behaviour if the description is intended to be simulated.

Connection points in the network of an analogue system are called nodes. Terminals of components are called pins. There are two types of the networks: *conservation-law* and *signal-flow*, which are dependent on the schematic of connection. In a conservation-law network there is a so-called *through quantity* (current in

electronic circuits), associated with each branch and so-called *across quantity* (voltage in electronic circuits), associated with each node.

Conservation-law networks are supported from the circuit abstraction level up to behavioural abstraction level. Conservation-law networks are also directionless: connecting two components together implies a mutually-shared contribution to each other's behaviour. For each node in electrical and electronic circuits network is the algebraic sum of through quantities associated with all the branches entering that node must be zero (Kirchhoff's Current Law). Analogously: for each loop, the sum of across quantity drops along the loop is zero (Kirchhoff's Voltage Law).

The product of across quantity and through quantity is a power. This will depend on the modelling requirements. For example, a model of a mechanical system may be written using through quantity to be the force and across quantity to be the position. The product across-by-through is work in this case.

The *signal-flow* networks can be found in early stage of an electronic system design or in a control system design. In a signal-flow network, quantities are associated with nodes only. Signal-flow networks are useful to desirable a linear systems. Signal-flow semantics provides the simplest binding rule for analogue system specification. The signal-flow semantics into VHDL-A will not complicate the language, but give rise the power.

4. VHDL-A Relations to VHDL

It is not supported as a generally available structure mechanism in for example SPICE simulator, which is originally developed with the emphasis on integrated circuits consisting of a limited number of primitive components. More than 80% of printed circuit boards, and 60% of custom integrated circuits and nearly all system-level designs involve both analogue and digital elements. The system design must support the both approaches. Many high-speed digital application specific integrated circuits (ASICs) use PLLs for clock synchronisation and frequency multiplication. Many digital ASICs and systems incorporate digital to analogue converters, analogue to digital converters.

The SPICE associated syntax becomes a *de facto* standard. The acceptance of VHDL-A will then depend on how the language will allow the SPICE users to re-use existing SPICE netlists. The VHDL-A should be powerful enough to allow the translation of a SPICE netlist into VHDL-A description, possibly through the use of automatic tools.

Some VHDL tools provide this feature to allow mixed mode modelling and simulation without any modification on the language. Many device simulators are built into simulators, because no standard procedural interface exists in SPICE-like simulators. Programming constructs of VHDL-A should allow these models to be

rewritten in VHDL-A. Behavioural modelling beyond SPICE receiving more attention. Increasing of demands are driven by mixed-signal designs, large analogue designs, and model development for new technologies. An industry standard analogue description language will accelerate development of analogue design automation.

A prospective user of VHDL-A is likely to have a large investment in SPICE libraries and netlists that they wont to re-use. This does not require SPICE syntax to be included in, or understood by, VHDL-A, but forces it to allow a mechanism dealing with backward compatibility of all the SPICE concepts can be translated into VHDL-A.

5. A/D and D/A Interfaces

An analogue to digital language interface is no converter model. Real based analogue time can be rounded or truncated to digital time. Digital processes execute in response to time-dependent conditions on quantities. Across quantity (voltage) of analogue connection point to be transformed into legal digital value. The through quantity (current) of analogue connection point may be included to get more specific digital value when multi-valued logic is used by IEEE 1164. An issue is whether VHDL-A supports its functionality between different simulation runs. An example illustrates the behavioural description of A to D event:

```
entity COMPARATOR is
  generic (LEVEL: float);
  pin (A, B: electrical);
  port (O: out std_logic);
end COMPARATOR;
architecture SIMPLE of COMPARATOR is
begin
  process
  begin
    if (A, B).v' rising (LEVEL) then
      O <= '1';
    else
      O <= '0';
    end if;
    wait on (A, B).v' crossing (LEVEL);
  end process;
end SIMPLE;
```

A digital to analogue language interface is realised by event on signal forces through break statement embedded in process. The digital value is translated into both across and through values in analogue part. In an electronic system a specific voltage-controlled switch may become invoked. The switch selects one of several loaded source subscripts for analogue input, according to the signal. The digital signal should be delayed, while analogue input may require smoothing to avoid discontinuities.

An example illustrates the digital to analogue interface on functional level:

$$O = \sin \left(2\pi \int_0^t (f_{ref} + \tanh(v_a - v_b - v_{ref})) dt \right)$$

```
entity VCO is
  generic (FREF, VREF: float);
  pin (A, B: electrical);
  coupling (O: analog);
end VCO;

architecture NON-LINEAR of VCO is
  quantity F, PHASE: analog;
  begin
  relation
  procedural =>
    F:= FREF + tanh((A,B).v - VREF);
    PHASE:= integ(2*PI*F);
    O %= sin (PHASE);
  end relation;
end NON-LINEAR;

end architecture;
```

All equations are simultaneous and must be gathered into one set before simulation. Both implicit and explicit forms must be supported in VHDL-A, because both may naturally occur when describing analogue behaviour. The resulting set of equations, which will be actually solved by the simulator, depends on the formulation method used by the simulator and on the underlying connection semantics.

6. Conclusion

Analogue to digital interactions involves the transformation of an analogue waveform description, which is continuous in amplitude and in time, into a digital signal description, which is quantized in amplitude and discrete in time. Digital to analogue interaction involves the transformation of a digital signal into an analogue waveform. The both functional conversions requires a value conversion and a time conversion. It must e precise defined in all the new concepts VHDL-A, which is bringing to VHDL and will ensure that the VHDL philosophy is not disturbed.

References

- [1] Shi, C.-J.R., Vachoux, A.: VHDL-A Design Objectives and Rationale. In Modeling in Analog Design. Kluwer Academic Publishers, (1995).
- [2] Patel, A., Ra, Ø.: VHDL-A in 15 minutes. DAK - Forum 95.

About authors...

Karel VLČEK was born in Zlin in 1948. He received the Msc degree at Technical University of Brno and PhD. degree at Czech Technical University of Prague. At the time beeing, he is at the Department of Electronics, Technical University of Ostrava. His main field of interests is investigated Digital Signal Processing, and Error Control Coding, Automation of Circuit Design, and Design for Test.

Jan POPELEK Was born in Opava, Czechoslovakia in 1966. He graduated in 1989 with a RNDr. (MSc. equivalent degree) in Optoelectronics from Olomouc University. From 1989 to 1991 he was a research assistant with Institute of Metallurgical Processes of Czechoslovak Academy of Sciences in Ostrava, working on visualisation (interferometry and optical tomography). From December 1991 to June 1992 he was a visiting researcher at Phillips International Technology Centre Leuven, Belgium, researching advanced optical link for digital audio. In 1993 he finished studies toward his Ph.D. degree in Optoelectronics at the Department of Microelectronics, Technical University of Brno. At the time beeing, he finished his Ph.D. thesis at the Department of Electronics, Technical University of Ostrava. His main field of interest behind playing violin is integrated optics, particularly the theory of diffractive optical elements design.