IEEE VHDL subPAR 1076.1

Analog Extensions to VHDL

Design Objective Document (DOD)

Version 1.0
November 30, 1992

Document history

1.0 30 nov 92 Alain Vachoux
Revised following meetings at DAC (Anaheim, June 92), EuroDAC (Hamburg, September 92) and IEEE DASS (Washington DC, October 1992).
New document structure to highlight and to number the design objectives.

0.2 11 may 92 Robert Cottrell
Revised following meeting at European VHDL Forum in Santander, Spain, 28 apr 92.

0.1 2 apr 92 Hazem El Tahawy
Initial version prepared by Bergé, El Tahawy, Rodriguez and Rouquier.

Table of contents

1. Introduction

The Design Objective Document (DOD) is a working document that aims to serve as the basis on which the design of analog extensions to VHDL is performed. This work is done within the subPAR 1076.1 standardisation activities and it will produce a separate VHDL norm, called VHDL-A, until the merge with VHDL 92 in the next standardisation of 1997.

The DOD is build from a list of design requirements submitted to the subPAR 1076.1 for analog extensions to VHDL. The complete list of these requirements was collected by Mark Altmae and Kevin Nolan from separate European and North American requirements. The list of relevant requirements, further called ***Design Requirements*** (DR) in this document, is recalled below. Design requirements should be met by VHDL-A and are the focus of the language design activity.

The DOD contains a list of numbered ***Design Objectives*** (DO) along with some comments to explain or to highlight specific aspects of them. A design objective has to be studied and possibly implemented in order to verify or improve its functionality.

The DOD contains as few syntactic examples as possible. Some ideas are explicitly supported by one or several design requirements. The corresponding requirement numbers are listed after the corresponding design objective. Other ideas are not explicitly supported by any requirement, but appeared during analysis, or during discussions.

The definitions of VHDL-A elements and mechanisms will appear in a separate glossary document.

2. Design requirements

The first numbering (DR1 to DR64) was indicated in a list of requirements prepared by Wolfgang Nebel in March 1991. A new set of requirements were gathered by Mart Altmae and numbered sequentially (DR65 to DR180). Some “holes” occur in numbering due to historical reasons. In the DOD, we retain the numbers, but we omit the origin of the requirement (North American or European) and we prefix the number with DR.

Number	Author(s)	Brief title
DR1	Mark Brown	Continuous time.
DR2	Mark Brown	Conversion between continuous and discrete time.
DR3	Mark Brown	Network of interconnected elements.
DR4	Mark Brown	Port behavior.
DR5	Mark Brown	Operators.
DR6	Mark Brown	Error tolerance.
DR7	Mark Brown	Miscellaneous requirements.
DR12	C. Le Faou et al	New key-words and electrical objects.
DR13	C. Le Faou et al	Input parameter for entities.
DR14	C. Le Faou et al	Package and user library for electrical modelling.
DR16	D. Rouquier	New key-words and underlying semantics for electrical modelling
DR17	A.S. Gilman	Analog modelling.
DR18	A.S. Gilman	Physical quantities.
DR19	A.S. Gilman	Baseline character set.
DR46	M. Altmae	Correspondence between VHDL objects and alternative representations in other netlist formats.
DR47	M. Altmae	Modify the VHDL initialization to include DC analysis for analog parts.
DR48	M. Altmae	Specified interface semantics between digital and analog parts of a design.
DR62	D. Rodriguez et al	Dimensional equations.
DR63	D. Borrionne et al	New key-words and electrical objects. Modification and review of CLF+1 (= DR12).
DR64	D. Borrionne et al	Access to the current value crossing an electrical component.
DR65	D. Rouquier et al	At least two labels.
DR66	D. Rouquier et al	Re-use digital constructs.
DR67	D. Rouquier et al	Definition of analog time.
DR68	D. Rouquier et al	Time synchronisation.
DR69	D. Rouquier	Portability of analog models.
DR70	D. Rouquier	Use of existing methods.
DR71	D. Rouquier et al	User control over digital to analog interactions.
DR72	D. Rouquier et al	User control over analog to digital interactions.
DR73	N. Whitaker	Dimensional analysis of physical types.
DR74	D. Rouquier	Domain of application.
DR75	D. Rouquier	Several modelling techniques.
DR76	T. Rahkonen	Consistency checks for excitations and results in different analog simulators.
DR77	A. Patterson	Analog modelling requirements in VHDL.
DR78	D. Rouquier	Mixed entities required.
DR79	D. Rouquier	Implementation issues.
DR176	C. Ussery	Single timing semantics.
DR177	C. Ussery	Pass-through of analog values.
DR178	C. Ussery	Fully intermixed designs.
DR179	K. Nolan	Specification of initial analog conditions.
DR180	K. Nolan	Specification of non-conserved analog systems.

3. General guidelines

This section provides general guidelines for the design of VHDL-A.

3.1. Scope of the language

DO1 VHDL-A should be suitable for the description and simulation of mixed, analog-digital, systems. No synthesis semantics is taken into account.

3.2. Digital aspects

DO2 VHDL-A will be a "super-set" of VHDL-92; any description that is valid in VHDL 92 will also be valid in VHDL-A, ***with the same simulation results***.

The only permitted exception to this is that new keywords introduced into the language may conflict with identifiers used in a VHDL 92 description.

From DR17.

3.3. Analog aspects

DO3 The analog part of VHDL-A should be targeted primarily towards the following applications:
- DC and transient analyses
- electronic circuits (OpAmps, PLLs, comparators, ...)
- lumped element systems.

From DR74.

DO4 Attention should be paid to ensure that frequency analysis is possible (for purely analog descriptions), provided that this does not have too great an impact on the language.

From DR3, DR5.

DO5 Mixed, time-frequency analysis (MFTA) could be considered.

From DR17.

DO6 The other domains (mechanical, thermal, ...) could be easily introduced as they are analogous to the electrical domain (definition of *across* and *through* quantities).

DO7 The micro-wave domain will not be taken into account.

This is contradiction with part of DR17. Contacts should be taken with MHDL.

DO8 The semantics of analog simulations will obviously be different from that of digital simulations; the syntax of analog descriptions should also be specific (to clearly identify the analog parts). Analog descriptions ought to re-use existing (VHDL 92) syntax where appropriate, for such constructs as expressions, function calls, if-then-else statements, ...

From DR66.

3.4. Mixed aspects

DO9 A few, basic, mechanisms will be provided to allow analog parts and digital parts of a mixed circuit to interact.

From DR2, DR3, DR5.

DO10 The places where such interactions occur will be explicitly described by the user.

From DR71, DR72.

DO11 The basic A/D and D/A interaction mechanisms of VHDL-A will be completely defined, and will make no use of the "foreign interface" of VHDL-92.

This not in contradiction with section 4. In other words, analog will not be "foreign" to VHDL-A.

DO12 The guidelines contained in this paragraph should ensure a reasonable degree of portability, while giving enough freedom to the implementation. The intention is to maintain portability without defining the simulation algorithm.

Further study is required on the meaning of portability. This aspect is also closely related to the time synchronisation issues (see 8.1 below).

DO13 Further study is required on the restrictions which may be required on the combination of analog and digital constructs (but note DR178).

DO14 The nature (analog OR digital OR mixed) of any part of a mixed circuit is contained in its VHDL-A description.

For example, a component configured as a digital entity-architecture pair cannot be dynamically re-configured during simulation to become analog.

4. Analog structures

DO15 Using VHDL-A, it should be possible to describe the structural composition of analog sub-circuits connected by analog wires.

This implies to determine clearly what aspects may already be found in the digital domain and what aspects are specific to the analog domain.

From DR3, DR12, DR14, DR63.

4.1. Similarities with the digital domain

DO16 The following aspects are very similar to the corresponding ones in the digital domain:
- blocks
- components
- configuration of a component onto an entity-architecture pair
- generate statements.

From DR3, DR14.

DO17 For these aspects, it should be possible to re-use the existing syntax.

From DR66.

4.2. Specific aspects of analog domain

DO18 The analog ports (provisional name : PINs), or terminals (see the glossary for the definition of a terminal), of an analog component are characterised by a voltage across the PIN and a current flowing through the PIN.

It will be necessary to specify a clear convention concerning the direction of currents (entering or leaving the components).

From DR4, DR12.

DO19 The interactions between an analog component and its environment are always "two-way" (not "in" or "out" ; the "inout" digital mode does not seem appropriate either). The term "binding" has been used.

From DR3, DR63.

DO20 The analog wires (provisional name : NODEs) of an analog structure are characterised by a voltage. The NODE object should be able to provide the values of all its contributing currents from its incidental branches (see the definition of branch in the glossary).

This assumption might not hold in micro-wave circuits, but see DO7.

From DR12.

DO21 The connection of several PINs to a NODE means that :

- all PINs must have the same voltage
- the sum of PINs currents must be zero (Kirchhoff current law, or KCL).

This assumption might not hold in micro-wave circuits, but see DO7.

From DR1, DR4, DR12, DR63, DR77.

DO22 Analog components may also need "parameters" ; parameters may be defined as generics whose value may vary during simulation.

From DR13.

DO23 Analog components could embody SPICE netlists since the analog simulator should understand this de facto standard description.

Communication between top-level (VHDL-A) description and SPICE netlist should be carefully studied (node correspondence, generics, parameters, …).

From DR46.

DO24 By changing the definition of the "across" and "through" quantities, and by keeping KCL and KVL (equal "across" quantity on a "node", null sum of "through" quantities on a node), other domains (mechanical, thermal, ...) can easily be introduced.

DO25 For these aspects, new syntax has to be created.

From DR66.

5. Analog behaviors

DO26 Using VHDL-A, it should be possible to describe the behavior of analog circuits. Two styles should be provided:
• relations or equations
• procedural models.

These two styles will form the “core" of the analog part of VHDL-A, and should be designed with great care. It might be interesting to allow mixing both styles within the same analog part, like in FAS or MAST.

From DR4, DR12, DR75, DR79.

5.1. Relations

DO27 The behavior of an analog part may be described by an un-ordered set of differential equations.

Notes:
• Each one of these equations is merged into the set of equations to be solved to perform analog simulation.
• Other equations come from the structural description.
• Adding an equation to the equation set should be complemented by adding an independent variable to the set of variables.

From DR4, DR12, DR14, DR70, DR75, DR79.

5.2. Procedural models

DO28 The behavior of an analog part may be described by a procedure, returning branch currents (*through* quantities) as a function of node voltages (*across* quantities). Each of these procedures is called, by the simulator during analog simulation, to find node voltages satisfying KCL.

It might be useful to allow a procedural model to return also voltages as functions of currents (e.g. for an inductance).

From DR4, DR70, DR75, DR79.

5.3. Access to voltages and currents

DO29 For analog behavioral modelling (and for A-to-D interactions), it is necessary to designate the voltage or the current of a PIN (seen from within a component). It is also useful to designate a branch current or a node voltage (seen from outside components).

The following notations have been proposed:

PIN.V	access to the voltage of a pin
(PIN1,PIN2).V	access to the voltage difference between PIN1 and PIN2
PIN.I	access to the current (entering or leaving) a pin
NODE.V	access to the voltage of a node
COMP1.PIN2.I	access to the current (entering or leaving) the pin of an outside component
(PIN1,PIN2).I	access to the branch current of the branch flowing from PIN1 to PIN2 (see the definition of a branch in the glossary)
(PIN1,NODE).I	access to the branch current flowing from PIN1 to NODE (see the definition of a branch in the glossary).

The above notations are provisional and purely syntactic, and do not mean that a PIN or NODE is represented by a "record" (the exact representation is left to the implementation). They may be used to allow reading, or writing, or within an equation (relation).

The notation COMP1.PIN2.I is intended to designate the current flowing into the component instance COMP1 coming from the (unique) node connected to the pin PIN2 of that instance and assuming that the corresponding model have a pin named PIN2. This current is “seen from the outside” of component COMP1 and may be read (but not written) by some other element of the enclosing architecture (port map, pin map, parameter map, or direct access by a process or an “analog black-box”).

From DR14, DR64.

6. Interactions between analog and digital domains

DO30 Interactions between the analog and digital domains address both the way the model is described and the way it is simulated.

From DR2, DR69, DR79.

DO31 The user must be given control over the exact transformations used between the analog and digital domains of description.

From DR2, DR48, DR71, DR72.

6.1. Analog to digital interactions

DO32 A user-supplied thresholding function is required to ensure that digital events are triggered only when “significant changes” occur in the analog part.

The idea behind the term “significant changes” is that at least some form of user controlled thresholding must be performed by the analog simulator in order to avoid unnecessary computing in the digital simulator. The exact sharing of work for analog to digital interactions between the analog and digital simulators is still to be defined.

One useful assumption to help produce a consistent definition of analog to digital interactions in VHDL-A would be: “The time unit used by the analog simulator when dating threshold crossings for the digital simulator must be equal to or greater than the secondary resolution time unit of the digital simulator”.

It was proposed that the analog simulator reports to the digital simulator the fact that an analog value has crossed a threshold along with the direction of the crossing (e.g. rising or falling).

From DR5, DR72.

DO33 The basic mechanisms could be restricted to :
• a digital process may read an analog value (voltage or current)
• a digital process may be made "sensitive" to an analog value (and in this case the thresholding function is used to decide when to trigger the process; this is similar to the "monitor" construct proposed by M.Brown and B.Hanna, but more general).

From DR72.

DO34 Default conversions between analog and digital domains may be provided. This would make the connection between PINs and PORTs legal without additional syntax. However, it should be illegal to connect a PIN to a SIGNAL.

From DR79.

6.2. Digital to analog interactions

DO35 A user-supplied "slope function" might be required, to avoid numerical problems due to discontinuities in the analog part. This "slope function" is similar to the "waveform generator" proposed in DR5, but less general.

The term “slope function” was considered inappropriate. The term “transition function” was proposed but does not seem to be convenient either because “transition” has a very specific semantics in VHDL. A more convenient name is still to be defined.

From DR5, DR71.

DO36 The basic mechanisms could be restricted to :
- an analog behavioral model may read a digital signal
- it would be possible to read a digital signal to drive a node by controlling either its voltage or (part of) its current, or a mix of both.

From DR71.

DO37 Default conversions between digital and analog domains may be provided. This would make the connection between PORTs and PINs legal without additional syntax. However, it should be illegal to connect a PORT to a NODE.

From DR79.

7. Units

7.1. Entities

DO38 Entities in VHDL-A may have PINs and PARAMETERs in addition to PORTS and GENERICs ; following this line, entities may be digital, analog, or mixed.

From DR78, DR177.

7.2. Architectures

DO39 Architectures in VHDL-A may contain NODEs and "Analog-Black-Boxes" (see 7.1) in addition to SIGNALs and PROCESSes ; following this line, architectures may be digital, analog, or mixed.

From DR78, DR177.

DO40 In addition to GENERIC MAP and PORT MAP, components will have "PIN MAP" and "PARAMETER MAP" ; same for blocks.

From DR63.

DO41 Analog-to-digital and digital-to-analog interactions will occur only within architectures containing PROCESSes and "Analog-Black-Boxes", and making use of one of the basic mechanisms described in 6.1 or 6.2.

From DR79.

7.3. Math package

DO42 For analog behavioral modelling a specific mathematical package is required, providing useful functions such as sine, cosine, exp, log, ln, complex numbers and associated operations.

Notes:
• Double precision (64 bits) is required.
• Piece-wise linear functions could be provided (such as arrays of value pairs).
• Matrix manipulations could be provided (such as FFT).

The design of the mathematical package may be addressed by another working group such as the VHDL Math Package Study Group.

From DR7, DR77.

DO42 Time derivative is necessary for analog behavioral modelling. It must be possible to apply it to any analog quantity:
• node voltage
• branch current
• pin voltage
• pin current
• user-defined extra independent variable (whether within a "relation" or within a "procedural model").

Derivatives with respect to other quantities than time were also considered but not retained.

From DR5, DR12, DR70, DR77.

DO43 Also an integral operator is useful (at least "from 0 to now").

7.4. Electrical package

DO44 For electrical modelling (i.e. to model analog structures), a specific electrical package is required, providing the declarations and the implementations of basic circuit elements (such as resistance, capacitance, inductance, sources, …).

Common models for semiconductors (diode, transistor) could also be provided. For transistors, we can mention SPICE level 1 to 3 and BSIM models.

From DR14.

7.5. Dimensional analysis package

DO45 The language should support dimensional analysis. To this aim, a separate package should be designed.

This objective requires to design a new syntax since the existing VHDL syntax for physical types does not seem to be convenient.

From DR18, DR62, DR73, DR77.

DO46 Only static dimensional analysis is retained, that is all dimensional verifications are to be performed at compile time only.

8. Simulation mechanisms

DO47 As in VHDL, simulation mechanisms are part of the norm, but simulation algorithms ought not to be specified. However, mixed-mode analog-digital simulation mechanisms should be defined precisely in order to ensure a reasonable degree of portability for the models written in VHDL-A.

DO48 Both electrical simulation of digital circuits (with simplified techniques like Iterated Timing Analysis or Waveform Relaxation) and precise simulation of analog circuits (with accurate SPICE-like techniques) should be possible.

From DR74.

8.1. Time management

DO49 Obviously, the analog simulation should follow a different time management than the digital simulator. However, the simulation of a mixed model requires that events on digital signals and evaluations of analog quantities be ordered along a single time axis.

From DR2, DR67, DR68, DR176.

DO50 The definition of time synchronisation between analog and digital parts should be complete and precise enough to ensure a reasonable degree of portability. In addition, it should be transparent to the user (no explicit use of “last active postponed process”).

From DR68, DR69.

8.2. Initialisation

DO51 The initialisation of a mixed model must include DC analysis for the analog parts (in addition to the normal procedure for the digital parts) ; this may involve iterations.

From DR47, DR179.

8.3. Simulation control

DO52 As an analog simulator usually implements some form of iterative methods to compute the state of the circuit, some control over the iterations is needed.

Possible controls may be:
• To specify the accuracy at which the electrical quantities are computed.
• To select specific simulation techniques or algorithms on a block-by-block basis, a block being an analog black-box for example.
• To specify the format of the analog output waveforms, possibly with some post-processing (for example, to compute the power dissipation).

Only the communication mechanism to pass these data to the simulator should be defined, not the way the simulator should handle them.

From DR16.

9. Open questions

9.1. Relations

This style of description has been illustrated in requirements DR12 and DR14, but only on simple examples (elements with only two pins, or structural equivalent circuit for the bipolar transistor). It is necessary to explore the limits of this style:
• is it possible to describe an element having three or more pins using only equations?
• what if several relations are declared within one architecture (does this describe a "structure" ; will the relations interact through nodes)?
• use of "if-then-else" within relations?
• is it necessary to explicitly declare a new variable with each new equation?

9.2. Procedural models

This style of description has been described, with very few details, in requirements DR4, DR70 and DR75. It is used in several existing simulators (FAS, MAST, also M). It is very natural if the equation-set is built using the "modified nodal method" (each node corresponds to one variable: the node voltage, and one equation : KCL). It may include such features as access to the ‘old' value of an analog quantity, access to the value of the time-step, ... It is necessary to explore also the limits of this style.

9.3. Modification to the digital simulation kernel

It will be necessary to introduce at least some modifications to the digital part of VHDL-92 to allow for well-defined mixed-mode descriptions and simulations. These modifications will emerge from the A-to-D interaction mechanisms, and from the problem of time synchronisation. The following could be sufficient:

1) The wait statement is extended to allow processes to be sensitive to analog quantities; examples:

   wait on N.V;   wait on PIN1.I;   wait on COMP1.PIN2.V;

   Notes:
   • the visibility rules will have to be refined
   • a thresholding function is required, associated with the analog quantity to which a process is sensitive.

   This sensitivity takes effect as explained below.

2) The simulation cycle is modified:

   2.1) When the digital "kernel" reaches the point where all processes have been run and have suspended (including "postponed" processes), instead of advancing time (to the "next digital date", which is strictly greater than the current date), the digital kernel should be capable of halting and giving control back to some "general kernel", which will take care of analog simulation.

   2.2) The digital kernel is capable of being re-started by the "general kernel" with the following order : advance time to the "next digital date" and perform normally at that date (in this case there is no real modification of the simulation cycle).

   2.3) The digital kernel is also capable of being re-started by the "general kernel" with the following order : advance time to date T0 (with T0 strictly smaller than the "next digital date") and at that time, accept "unforeseen events". These events come from the analog part, and as far as the digital kernel is concerned, they simply imply resuming the execution of one or several processes currently suspended on a "wait on N.V" statement.

   With these modifications, and with a few assumptions on the analog simulator, time synchronisation can be achieved.

9.4. Assumptions on the analog simulator

When started, the analog simulator never produces "zero-delay-events" (in other words, its time-step is never zero); this seems a reasonable assumption to place on any analog simulator, and it might be crucial when specifying time synchronisation.

The analog simulator can not tell its "next analog date" (date of the next time- step to be computed by the analog simulator). On the other hand, when computing a time-step, it can store old values (values of the previous time step); thus it can be told either to "accept" a time-step (forget old values) or to "reject" a time-step and to compute (possibly by interpolation) a shorter one. This is a soft form of backtracking.

With these assumptions, and with the above modifications on the digital simulator, time synchronisation can be achieved.

9.5. Miscellaneous questions

• Limits of the LRM: Will the LRM describe the structure of the mixed simulator, the time synchronisation mechanism, the method for building the equation-set, the method for solving it?

• Error tolerance (DR6) and portability (DR69); N.I returns 0 or epsilon?

• 64 bits? (left to the implementation?)

• Statement for "connecting" two nodes (DR177)?

• "node-set" to help DC.

Appendix A: Detailed analysis of design requirements

DR1 Mark Brown	**Continuous time.**
Continuous time:	Supported. Analog quantities vary continuously in time but the simulator computes them only at discrete time-points.
Analog simulation algorithms are proprietary:	Supported. The analog part of the VHDL-A LRM will be less detailed than the digital part.
Kirchhoff's laws:	Supported. Kirchhoff's laws form the basis of the semantics for analog simulation.

DR2 Mark Brown	**Conversion between continuous and discrete time.**
	Supported.

DR3 Mark Brown	**Network of interconnected elements.**
No direction:	Supported.
A node connects two or more terminals:	Supported.
VHDL 92 component statements are good for analog structures:	Supported.
N-Port theory:	Not completely understood, please provide a tutorial!
A branch between two terminals:	Branch current not unique !?
Modal and domain conversions:	??

DR4 Mark Brown	**Port behavior.**
KCL and KVL:	Supported.
Branch current seen as function of other analog quantities:	Supported.
Modelling by relationship:	Supported. Assuming “relationship” is the same as “equations”.

DR5 Mark Brown	**Operators.**
Threshold function for A-to-D:	Supported.
Threshold function for D-to-A:	Supported. To be refined.
Time derivative:	Supported.
Transformation between time and frequency, partial derivative:	??

DR6 Mark Brown	**Error tolerance.**
	Not understood.

DR7 Mark Brown	**Miscellaneous requirements.**
Math package for transcendental operations (and floating-point exponent):	Supported.

Complex numbers (rectangular or polar):	Supported. But it is second-priority for transient analysis of mixed circuits.
Physical units:	To be clarified.

DR12 C. Le Faou et al **New key-words and electrical objects.**

Modeling by equation:	Supported. Adding an equation to the equation set should be complemented by adding an independent variable to the set of variables...
Key-words introducing new objects:	Supported. Essentially a re-phrasing of Kirchhoff's laws.
New type "electrical":	Syntactic issues ("type electrical is record…"). It is perhaps too early to propose detailed syntactic forms.
Time derivative:	Supported.

DR13 C. Le Faou et al **Input parameter for entities.**

Supported.
Except using parameters in the digital domain: this needs further study.

DR14 C. Le Faou et al **Package and user library for electrical modelling.**

It is too early to propose detailed syntax ; the examples are all very simple ones, and tend to suggest that this method is limited to elements having only two pins, possibly parameters, and no internal states. To be studied further.

DR16 D. Rouquier **New key-words and underlying semantics for electrical modelling.**

Not considered (too long, too syntactical ; the main ideas have been rephrased in subsequent requirements by the same author).

DR17 A.S. Gilman **Analog modelling.**

	Not completely understood.
Micro-wave circuits:	Not supported.
Use of "concurrent" (global?) variables to model analog parts:	Not supported.
Analog part operates in parallel with the digital one:	Supported.
Simulation algorithm is implementation-defined:	Supported.
Upward compatibility:	Supported.
'DOT for time derivative:	Interesting alternative to d-dt()…

DR18 A.S. Gilman **Physical quantities.**

	Not completely understood.
Dimensional analysis in the language:	Supported.

Floating-point values for physical quantities:	Supported.

DR19 **A.S. Gilman**	**Baseline character set.**
	Not considered (already considered by VHDL-92).

DR46 **M. Altmae**	**Correspondence between VHDL objects and alternative representations in other netlist formats.**
	Not completely understood.
A-to-D and D-to-A interactions:	Seems to suggest A-to-D and D-to-A interactions by direct connection of (VHDL) signals to (SPICE) nodes ; this seems the only way during the interim phase (using the "foreign interface" before VHDL-A is standardised) but it will not be the case in VHDL-A.
Programs to adapt SPICE libraries to VHDL-A:	Supported.

DR47 **M. Altmae**	**Modify the VHDL initialization to include DC analysis for analog parts.**
	Supported. But we should find a way to avoid iterations…

DR62 **D. Rodriguez et al**	**Dimensional equations.**
Dimensional analysis in the language:	Supported.
Detailed syntax:	Too early.

DR63 **D. Borrionne et al**	**New key-words and electrical objects. Modification and review of CLF+1 (= DR12).**
Connecting several PINs to a NODE does not imply the same semantics as connecting several PORTs to a SIGNAL:	Supported.
Proposed syntax:	Too early.

DR64 **D. Borrionne et al**	**Access to the current value crossing an electrical component.**
Need to access branch currents:	Supported.
Proposed syntax:	Too early.

DR65 **D. Rouquier et al**	**At least two labels.**
	Not considered (not technical; in fact, as there will be two different standards, it will be easy to check whether a given simulator is conformant to 1076 or to 1076.1).
Static criterion about circuit type:	The criterion to tell whether a circuit is digital, analog, or mixed will be known only after elaboration.

DR66 **D. Rouquier et al**	**Re-use digital constructs.**
	Supported. In fact, to be monitored during the detailed design of the language.

DR67 D. Rouquier et al — Definition of analog time.

Supported.

DR68 D. Rouquier — Time synchronisation.

Conversion between analog and digital time:	Supported.
Unique time axis:	Supported.
Proposed strategy:	To be studied in greater details ; such a model (of the mixed simulator) must be viewed as conceptual, otherwise we might enter domain of "over-specification".

DR69 D. Rouquier — Portability of analog models.

To be analysed by vendors of mixed simulators!

DR70 D. Rouquier — Use of existing methods.

General:	Supported.
Modified Nodal Method:	Over-specification?
Procedural models:	Supported. To be refined.
Extra equations (and associated extra variables):	Supported.
Old value, time-step:	Supported. See procedural models.
Time derivative:	Supported.

DR71 D. Rouquier et al — User control over digital to analog interactions.

General:	Supported.
Details:	Depend on the exact definition of analog behavioral modelling in VHDL-A. The problem of discontinuity may be solved by the "slope function" associated with any signal used (in read-only mode) in the analog part.

DR72 D. Rouquier et al — User control over analog to digital interactions.

Supported.

DR73 N. Whitaker — Dimensional analysis of physical types.

Supported.

DR74 D. Rouquier — Domain of application.

Supported.

DR75 D. Rouquier — Several modelling techniques.

Supported. The syntax of "analog procedural models" may be similar to that of digital processes, but the semantics will be completely different.

DR76 T. Rahkonen — **Consistency checks for excitations and results in different analog simulators.**

Not understood. This text seems to refer to a situation where different simulators are used in a very loosely coupled way, and the simulation of the whole circuit requires a lot a manual (and error-prone) operations, to feed the results of one simulator as stimuli to another simulator. It does not seem to envisage true mixed-mode simulation in VHDL-A.

DR77 A. Patterson — **Analog modelling requirements in VHDL.**

Supported. Perhaps phase-angle descriptions are second priority.

DR78 D. Rouquier — **Mixed entities required.**

Supported.

DR79 D. Rouquier — **Implementation issues.**

Supported.

DR176 C. Ussery — **Single timing semantics.**

Extensions to the timing model (due to analog):	Should be incorporated within VHDL 92.
Achieve a single simulation cycle mechanism:	Supported.

DR177 C. Ussery — **Pass-through of analog values.**

Supported.
Assuming that the mixed component in the associated example will have a mixed interface (at least two PINs, possibly some ports) and will contain, in addition to its ("purely digital") behavior, a "connect statement" (still to be invented) to indicate that the two nodes (coming from the source and going to the reader) are connected.

DR178 C. Ussery — **Fully intermixed designs.**

Supported.
"runtime statements" : to be clarified); closely linked to DR177.

DR179 K. Nolan — **Specification of initial analog conditions.**

Supported. Also user-supplied "hints" before DC are usefull (to help DC converge faster).

DR180 K. Nolan — **Specification of non-conserved analog systems.**

Not understood. Please provide tutorial!