Standard VHDL 1076.1.1 Packages for Multiple Energy Domain Support

Peter R. Wilson & Andrew D. Brown
School of Electronics and Computer Science
University of Southampton, UK

H. Alan Mantooth
Department of Electrical Engineering
University of Arkansas, USA
Outline

Overview
• Requirements
• Multiple Domain Modeling
• Packages Overview
• Simple Example

Current Status
• Draft Standard
• IEEE Process

Examples
• Multiple Domains
• Electrical-Magnetic-Thermal
• Electro-Mechanical
Requirements: Model Authors

Interoperability

- Models must be interoperable with models written by any author
- Terminal ports and terminals must have an agreed set of natures

Infrastructure

- Mathematical Functions
  - IEEE Std 1076.2-1996
- Physical Constants
  - Fundamental Constants
    - e.g. Speed of light, Boltzmann’s constant, electronic charge
  - Material Properties
    - e.g. Permittivity of Silicon
Requirements: Model Users

Model Exchange
- Interoperability
- Models must have an internationally accepted set of units
  - This has implications for connections and parameters

Simulation Accuracy
- Tolerance codes to support the VHDL-AMS tolerance model

Conveniences
- Unit Symbols and names
  - E.g. (V) & Voltage
- Scale factors
  - Kilo, mega, giga etc
Requirements: Tool vendors

**Interoperability**
- Agreed upon natures

**Simulation Accuracy**
- Support for VHDL-AMS tolerance model

**Ability to write numerically robust models**
- Selection of units of quantities such that unknowns have roughly the same scale
BMAS: Comment at Panel Session

One comment at the panel session (paraphrased):
“Oh, we can do all this multi-domain modeling in the electrical domain – we don’t need those other types”

Some comments on that:

• **Units**
  - Example: it may be that the output is a thermal pin, but is it C or K?
  - Example: Imperial or SI – Hubble telescope!

• **Black Box**
  - What if the model is black box – can you assume the units are consistent without knowledge of the model content? I suggest not.
  - Standard units and types mean that models can be written portably whether or not they model detail is available or not.

• **Tolerances**
  - The multiple domain approach allows different tolerance mechanisms to be defined for different technologies – potentially important for stiff systems (hydraulics?)
BMAS: Comment at Panel Session

Some more comments...

• Checking
  • Standardised units and connections means that limited unit checking can take place within models and systems

• Interoperability
  • Enables models to delivered to the user community that can easily be used in a variety of tools
  • Ensures consistency of units and symbols across vendors

• While not a requirement of the VHDL-AMS standard package definitions, it is helpful for the integration of other model languages (e.g. Verilog-A) in a system simulation to have a well defined standard set of interfaces.
## Multiple Domain Requirements

<table>
<thead>
<tr>
<th>Domain</th>
<th>Generic</th>
<th>Electrical</th>
<th>Mechanical Rectilinear</th>
<th>Mechanical Rotary</th>
<th>Thermal</th>
<th>Hydraulic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Across Variable</td>
<td>x(t)</td>
<td>Voltage Volts</td>
<td>Velocity M/sec</td>
<td>Ang. Velocity Rad/sec</td>
<td>Temperature K</td>
<td>Pressure N/M^2</td>
</tr>
<tr>
<td>Dissipator</td>
<td>x(t) = k_1 y(t)</td>
<td>V = I R</td>
<td>f = kV</td>
<td>T = k_\omega</td>
<td>T = Qr</td>
<td>P = k_1 U</td>
</tr>
<tr>
<td>Delay</td>
<td>K_2 dy(t)/dt = x(t)</td>
<td>V = L dl/dt</td>
<td>V = kdf/dt</td>
<td>\omega = kdT/dt</td>
<td>Q = k_1 dT/dt</td>
<td>U = k_2 dP/dt</td>
</tr>
<tr>
<td>Accumulator</td>
<td>K_3 dx(t)/dt = y(t)</td>
<td>I = C dV/dt</td>
<td>f = m dV/dt</td>
<td>T = J d\omega/dt</td>
<td>T = k_2 dQ/dt</td>
<td>P = k_3 dU/dt</td>
</tr>
<tr>
<td>Generators</td>
<td>y(t) /, x(t) known</td>
<td>Constant through or across sources</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td>x(t) * y(t)</td>
<td>V * I</td>
<td>F * V</td>
<td>T * \omega</td>
<td>Q * T</td>
<td>P * U</td>
</tr>
</tbody>
</table>
Magnetics are slightly different

Unlike the Other technologies, Magnetics use the B,H or Flux,MMF Energy planes

- e.g. in the electrical domain, VI = Power
- To obtain the power in the magnetic domain requires

\[ E = \int H \cdot dB \]

\[ P = \frac{dE}{dt} \]

Energy Loss is the area inside the loop
# Summary of Magnetic Technology

<table>
<thead>
<tr>
<th></th>
<th>Definition</th>
<th>Equation</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Through Variable</td>
<td>Flux</td>
<td>( \Phi )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tesla</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Across Variable</td>
<td>MMF</td>
<td>( \text{mmf} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Amperes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dissipator</td>
<td>Loss</td>
<td>( \text{mmf} = L \cdot \frac{d\Phi}{dt} )</td>
<td></td>
</tr>
<tr>
<td>Delay</td>
<td>Permeance</td>
<td>( \text{mmf} = \Phi \cdot R )</td>
<td></td>
</tr>
<tr>
<td>Accumulator</td>
<td>not meaningful physically</td>
<td>( \Phi = C \frac{d\text{mmf}}{dt} )</td>
<td></td>
</tr>
<tr>
<td>Through Generator</td>
<td>Flux source</td>
<td>( \Phi = \text{const} )</td>
<td></td>
</tr>
<tr>
<td>Across Generator</td>
<td>MMF Source</td>
<td>( \text{mmf} = \text{const} )</td>
<td></td>
</tr>
<tr>
<td>Constant</td>
<td>Unit</td>
<td>Name</td>
<td>Default</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>------------</td>
<td>-------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>Electron charge</td>
<td>C</td>
<td>PHYS_Q</td>
<td>602_176_462e-19</td>
</tr>
<tr>
<td>permittivity of vacuum</td>
<td>F/m</td>
<td>PHYS_EPS0</td>
<td>8.854_187_817e-12</td>
</tr>
<tr>
<td>permeability of vacuum</td>
<td>H/m</td>
<td>PHYS_MU0</td>
<td>4.0e-7 * MATH_PI</td>
</tr>
<tr>
<td>Boltzmann's constant</td>
<td>J/K</td>
<td>PHYS_K</td>
<td>1.380_6503e-23</td>
</tr>
<tr>
<td>Acceleration due to gravity</td>
<td>ms⁻²</td>
<td>PHYS_GRAVITY</td>
<td>9.806_65</td>
</tr>
<tr>
<td>Conversion between Kelvin and degree Celsius</td>
<td>-</td>
<td>PHYS_CTOK</td>
<td>273.15</td>
</tr>
<tr>
<td>Velocity of light in a vacuum</td>
<td>m/s</td>
<td>PHYS_C</td>
<td>299_792_458.0</td>
</tr>
<tr>
<td>Planck's constant</td>
<td>-</td>
<td>PHYS_H</td>
<td>6.626_068_76e-34</td>
</tr>
<tr>
<td>Planck's constant divided by 2π</td>
<td>-</td>
<td>PHYS_H_OVER_2_PI</td>
<td>PHYS_H/MATH_2_PI</td>
</tr>
</tbody>
</table>
### Material Constants

<table>
<thead>
<tr>
<th>Constant</th>
<th>Unit</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative permittivity of silicon</td>
<td>-</td>
<td>PHYS_EPS_SI</td>
</tr>
<tr>
<td>Relative permittivity of silicon dioxide</td>
<td>-</td>
<td>PHYS_EPS_SIO2</td>
</tr>
<tr>
<td>Young's Modulus for silicon</td>
<td>Pa</td>
<td>PHYS_E_SI</td>
</tr>
<tr>
<td>Young's Modulus for silicon dioxide</td>
<td>Pa</td>
<td>PHYS_E_SIO2</td>
</tr>
<tr>
<td>Young's Modulus for polysilicon</td>
<td>Pa</td>
<td>PHYS_E_POLY</td>
</tr>
<tr>
<td>Poisson's Ratio for silicon</td>
<td>100 orientation</td>
<td>PHYS_NU_POLY</td>
</tr>
<tr>
<td>Density of Polysilicon</td>
<td>Kg/m³</td>
<td>PHYS_RHO_POLY</td>
</tr>
<tr>
<td>Density of Silicon-Dioxide</td>
<td>Kg/m³</td>
<td>PHYS_RHO_SIO2</td>
</tr>
<tr>
<td>Ambient Temperature</td>
<td>K</td>
<td>AMBIENT_TEMPERATURE</td>
</tr>
<tr>
<td>Ambient Pressure</td>
<td>Pa</td>
<td>AMBIENT_PRESSURE</td>
</tr>
<tr>
<td>Ambient Luminance</td>
<td></td>
<td>AMBIENT_LUMINANCE</td>
</tr>
</tbody>
</table>
## Domain natures

<table>
<thead>
<tr>
<th>Technology</th>
<th>Through</th>
<th>Across</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical</td>
<td>current</td>
<td>voltage</td>
</tr>
<tr>
<td>Magnetic</td>
<td>flux</td>
<td>mmf</td>
</tr>
<tr>
<td>Thermal</td>
<td>temperature</td>
<td>heat flow</td>
</tr>
<tr>
<td>Translational</td>
<td>velocity</td>
<td>force</td>
</tr>
<tr>
<td>Rotational</td>
<td>angular velocity</td>
<td>torque</td>
</tr>
<tr>
<td>Fluidic</td>
<td>flow rate</td>
<td>pressure</td>
</tr>
<tr>
<td>Radiant</td>
<td>luminous flux</td>
<td>luminous intensity</td>
</tr>
</tbody>
</table>
Units and Symbols: e.g. electrical

-- attribute declarations
attribute UNIT of VOLTAGE : subtype is "Volt";
attribute UNIT of CURRENT : subtype is "Ampere";
attribute UNIT of CHARGE : subtype is "Coulomb";
attribute UNIT of RESISTANCE : subtype is "Ohm";
attribute UNIT of CAPACITANCE : subtype is "Farad";
attribute UNIT of MMF : subtype is "Ampere";
attribute UNIT of FLUX : subtype is "Weber";
attribute UNIT of INDUCTANCE : subtype is "Henry";
attribute UNIT of FLUX_DENSITY : subtype is "Tesla";
attribute UNIT of FIELD_STRENGTH : subtype is "Amperes per meter";
attribute SYMBOL of VOLTAGE : subtype is "V";
attribute SYMBOL of CURRENT : subtype is "A";
attribute SYMBOL of CHARGE : subtype is "C";
attribute SYMBOL of RESISTANCE : subtype is "Ohm";
attribute SYMBOL of CAPACITANCE : subtype is "F";
attribute SYMBOL of MMF : subtype is "A";
attribute SYMBOL of FLUX : subtype is "W";
attribute SYMBOL of INDUCTANCE : subtype is "H";
attribute SYMBOL of FLUX_DENSITY : subtype is "T";
attribute SYMBOL of FIELD_STRENGTH : subtype is "A/m";
Tolerances: e.g. electrical

-- subtype declarations
subtype VOLTAGE     is REAL tolerance "DEFAULT_VOLTAGE";
subtype CURRENT     is REAL tolerance "DEFAULT_CURRENT";
subtype CHARGE      is REAL tolerance "DEFAULT_CHARGE";
subtype RESISTANCE  is REAL tolerance "DEFAULT_RESISTANCE";
subtype CAPACITANCE is REAL tolerance "DEFAULT_CAPACITANCE";
subtype MMF         is REAL tolerance "DEFAULT_MMF";
subtype FLUX        is REAL tolerance "DEFAULT_FLUX";
subtype INDUCTANCE  is REAL tolerance "DEFAULT_INDUCTANCE";
Simple Example: resistor

USE work.electrical_system.ALL;

ENTITY r IS
  GENERIC ( rnom : real );
  PORT (TERMINAL p, m: electrical);
END ENTITY r;

ARCHITECTURE simple OF r IS
  QUANTITY v ACROSS i THROUGH p TO m;
BEGIN
  i == v / rnom;
END ARCHITECTURE simple;
IEEE Standard 1076.1.1: Background

1076.1.1 PAR requested in March 2002
PAR approved in May 2002
Initial ad hoc formation Summer 2001
Kickoff discussions FDL 2001 and BMAS 2001
WG meetings held at DAC 2002 (June), FDL 2002 (Sept.), DATE 2003 (March), FDL 2003 (Sep)
Completed compilation of VHDL packages from WG contributors
Have received significant review and feedback
IEEE Standard 1076.1.1: Current

First Draft of proposed standard has had IEEE editorial review

- response received 7th October 2003: very minor revisions

Initial Balloting Pool:

- Initial Balloting Pool List defined and email sent to proposed members
  - Please indicate as soon as possible your intention to ballot so we can finalise the balloting pool
  - If you did not get an email – you were not on the draft list

- If you would like to be in the balloting pool, contact Alan Mantooth (mantooth@engr.uark.edu) now – he’s here.

- Initial Balloting Pool has a (roughly) equal split between Industry, Academia and EDA
IEEE Standard 1076.1.1: Process

Paperwork in progress for an e-ballot

IEEE review editorial tasks arising:

- Modifications made to proposed standard
- Revised proposed standard will be posted on the 1076.1.1 web site
  - http://mixedsignal.eleg.uark.edu/stdpkgs.html
- Email will be sent to the VHDL-AMS reflector
- Final pre-ballot review period of 30 days will commence

Final Pre-ballot Review (October 2003)
Examples: Mixed-domain modeling

**Electro-Magnetic Model**
- Winding Model
- Linear Core

**Non-linear magnetic model**
- Non-linear core model – Jiles-Atherton

**Thermal extensions:**
- Electrical-Magnetic-Thermal model
Example Transformer Structure

\[ \Phi_p = \frac{d\Phi_p}{dt} \]

\[ mmf_p = n_p \cdot i_p \]

\[ mmf_s = n_s \cdot i_s \]

\[ \nu_s = \frac{d\Phi_s}{dt} \]
**Winding Model in VHDL-AMS**

**Ampere’s Law**

\[ NI = \oint H \cdot dl \]

**Faraday’s Law**

\[ V = -N \frac{d\Phi}{dt} \]

USE work.electrical_systems.ALL;
ENTITY wind IS
  GENERIC (r, n : real := 0.0);
  PORT ( ep, em : electrical;
        mp, mm : magnetic);
END ENTITY wind;

ARCHITECTURE simple OF wind IS
  QUANTITY mmf ACROSS f THROUGH mp TO mm;
  QUANTITY v ACROSS I THROUGH ep to em;
  QUANTITY vdrop voltage;
BEGIN
  vdrop == i*r;
  mmf == i*n;
  v == n*f'DOT + vdrop;
END ARCHITECTURE simple;
**Linear Core model in VHDL-AMS**

MMF = Reluctance * Flux
Flux = Permeance * MMF
Permeance = 1/Reluctance

\[ \mu_0 \cdot \mu_r \cdot \text{area}/\text{len} \]

**VHDL-AMS Code**

```vhdl
USE work.electrical_systems.ALL;
ENTITY core IS
  GENERIC ( ur, len, area : real);
  PORT (TERMINAL p, m : magnetic);
END ENTITY core;

ARCHITECTURE simple OF core IS
  QUANTITY
    mmf ACROSS flux THROUGH p TO m;
  BEGIN
    flux == (\(\mu_0 \cdot \mu_r \cdot \text{area} / \text{len}\)) \cdot mmf;
  END ARCHITECTURE simple;
```

Dr Peter R. Wilson – BMAS 2003
School of Electronics and Computer Science

University of Southampton, UK
Transformer Simulation

Simulation carried out with VeriasHDL - Avant!

Dr Peter R. Wilson – BMAS 2003
School of Electronics and Computer Science
University of Southampton, UK
Non-linear Hysteresis: Jiles Atherton

Differential Equation approach to modeling ferromagnetic hysteresis

**Reversible Magnetization**

\[
M_{an} = \frac{1}{\tanh\left(\frac{H}{a}\right)} - \frac{a}{H}
\]

**Irreversible Magnetization**

\[
\frac{dM_{irr}}{dH} = \frac{M_{an} - M}{\delta \cdot k - (M_{an} - M)}
\]
Jiles-Atherton Model Structure

\[ H + \frac{1}{\tanh \left( \frac{H}{a} \right)} - \frac{a}{H} \]

\[ \alpha \times M_e \]

\[ \frac{c}{1+c} \]

\[ M_{\text{rev}} \]

\[ \frac{1}{1+c} \]

\[ \text{M} \]

\[ \int \frac{M_{an} - M}{\delta \cdot k - (M_{an} - M)} \]

\[ \mu_0 \times \left( M_s \times M + H \right) / A \]

\[ B \]
Jiles-Atherton model in VHDL-AMS

BH Curve

- B and H measured inside the Jiles Atherton Model of the core

Results obtained using Mentor Graphics’ Design Station

Peter R. Wilson
Non-linear Transformer: VHDL-AMS

Results obtained using Avant!'s VeriasHDL
Extensions to include thermal effects

\[ v_p = \frac{d\Phi_p}{dt} \]

\[ mmf_p = n_p * i_p \]

\[ \Phi_p \]

\[ \Phi \]

\[ mmf_s = n_s * i_s \]

\[ v_s = \frac{d\Phi_s}{dt} \]

\[ i_p \]

\[ i_s \]

Core

Thermal

Electrical

Thermal

Electrical
Variation with Temperature

-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5

H (At/m)

B (T)

27°C
95 °C
154 °C
Thermal model of the magnetic core

Assume uniform distribution of heating in the core material

Hysteresis + Eddy Current + Winding
Power Loss
Thermal Capacitance Model

1  use work.thermal_systems.all;
2
3 entity ctherm is
4    generic (cth:real := 0.0);
5    port (terminal th,tl : thermal);
6 end entity ctherm;
7
8 architecture simple of ctherm is
9    quantity tc across heatfl through th to tl;
10 begin -- simple architecture
11   assert cth /= inf and cth /= undef
12   report "cth specified incorrectly"
13   severity error;
14
15   heatfl == cth * tc'dot;
16 end architecture simple;
Calculating Power and Energy

We need to find the area inside the BH curve to get the energy lost and then the power dissipated.

1: Integrate H.db
- Valid only at the completion of each cycle
- + easy to calculate

2: Use the irreversible proportion of the magnetization
- + Calculate the Power actually lost directly
- + Valid at any instant of time
- - Difficult to be sure the value is really correct
Implications of dynamic model

Power is fed into the thermal circuit from the core loss directly.
Dynamic temperature changes in the core are modeled and used to modify the model parameters.
Dynamic thermal behaviour of the core material and surface is modeled.
Thermal effects will have a profound effect on the electrical behaviour in the circuit:
  - thermal demagnetization - Curie Temperature
Simulations including thermal model

Flux Loss and Resulting Heat Flow into the thermal circuit

(flux): time(sec)
Flux Loss

(W): time(sec)
power loss

Dr Peter R. Wilson – BMAS 2003
School of Electronics and Computer Science

University of Southampton, UK
Simulations including thermal model

Final Temperature rise accurate to within 0.3°C

Results obtained using Avant!’s VeriasHDL
Electro-mechanical: DC Motor

Model a simple dc motor using the standard motor equations

\[ V = L \frac{di}{dt} + iR + Ke\omega \]

\[ T = Kti - J \frac{d\omega}{dt} - D\omega \]

Notice the interaction between the electrical and rotational domains
Electrical Systems Design Group

Electro-Mechanical

```vhdl
1  use work.electrical_systems.all;
2  use work.mechanical_systems.all;
3
4  entity dc_motor is
5      generic (kt : real;
6               j  : real;
7               r  : real;
8               ke : real;
9               d  : real;
10              l  : real);
11     port (terminal p, m  : electrical;
12           terminal rotor : rotational_v);
13 end entity dc_motor;
14
15 architecture behav of dc_motor is
16     quantity w across t through rotor
17     to rotational_v_ref;
18     quantity v across i through p to m;
19 begin
20   v == l*i'DOT + i*r + ke*w;
21   t == i*kt - j*w'DOT - d*w;
22 end architecture behav;
```
Conclusions

These packages will provide a standard, interoperable and consistent framework for VHDL-AMS modeling from the author, user and vendor perspectives.

IEEE Std 1076.1.1 (proposed) is nearing the end of the standardization process
Extensive consultation with industry, academia and EDA vendors already taken place
Draft standard ready for comments
E-ballot process to be undertaken soon.