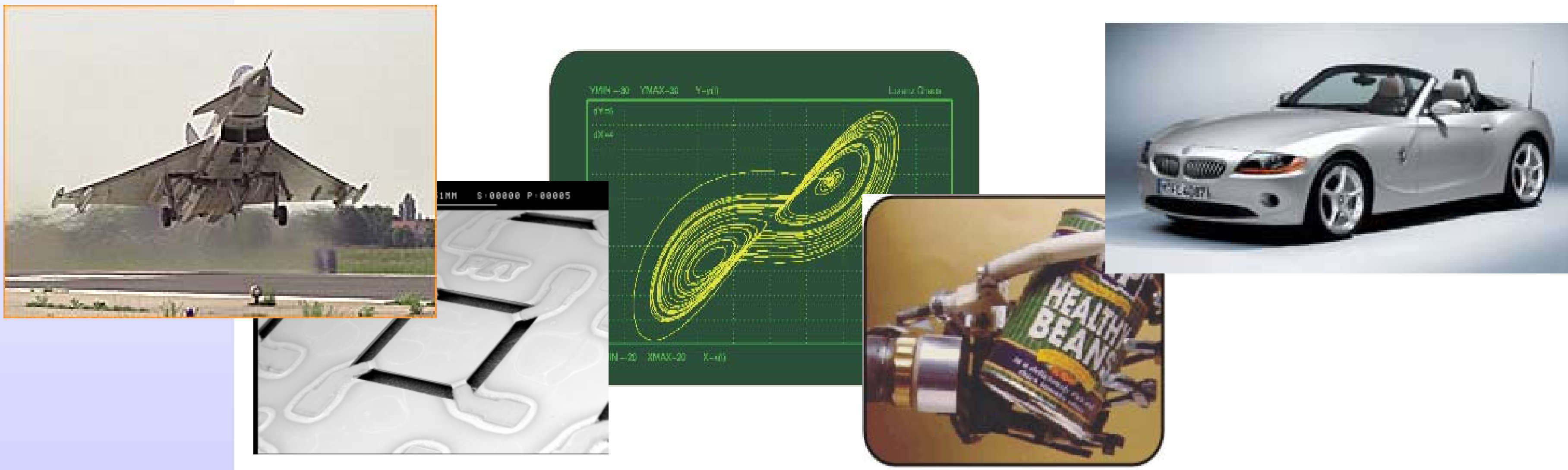
Standard VHDL 1076.1.1 Packages for Multiple Energy Domain Support



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

Domain	Generic	Electrical	Mechanical Rectilinear	Mechanical Rotary	Thermal	Hydraulic
Through Variable	y(t)	Current Amps	Force N	Torque NM	Heat Flow Joules/sec	Fluid Flow M^3/sec
Across Variable	x(t)	Voltage Volts	Velocity M/sec	Ang. Velocity Rad/sec	Temperature K	Pressure N/M^2
Dissipator	$x(t) = k_1 y(t)$	V = I R	f = kV	$T = k\omega$	T = Qr	$P = k_1 U$
Delay	$K_2 dy(t)/dt = x(t)$	V = L dI/dt	V = kdf/dt	$\omega = kdT/dt$	$Q = k_1\ dT/dt$	$U= k\ _2dP/$
Accumulator	$K_3 dx(t)/dt = y(t)$	I = C dV/dt	f = m dV/dt	$T = J\ d\omega/dt$	$T = k\ _2dQ/dt$	$P= k_3\ dU/$
Generators	y(t) /, x(t) known	← *Constant through or across sources* →				
Power	x(t) * y(t)	V * I	F * V	$T * \omega$	Q * T	P * U

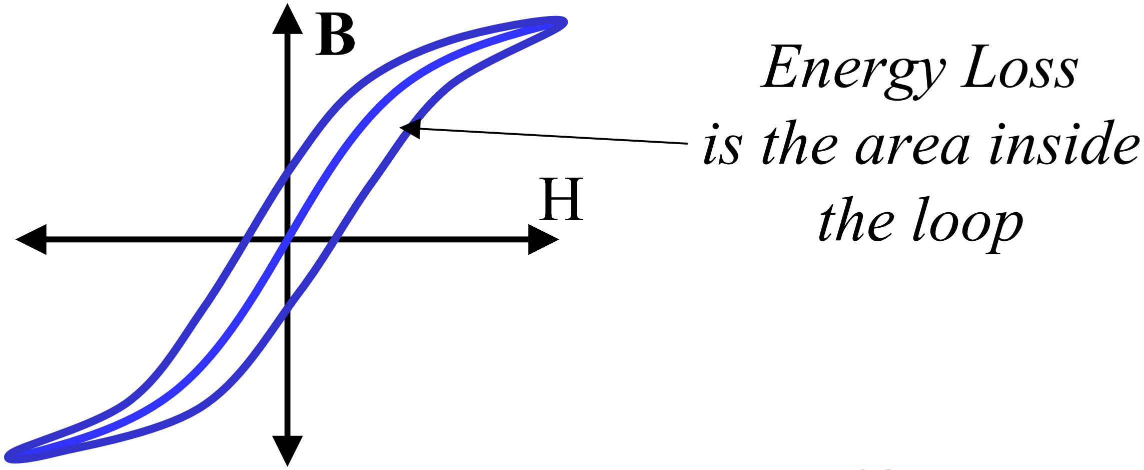
Magnetics are slightly different

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

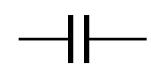
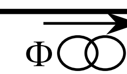
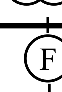
- e.g. in the electrical domain, VI = Power
- To obtain the power in the magnetic domain requires

$$E = \int H \cdot dB$$

$$P = \frac{dE}{dt}$$



Summary of Magnetic Technology

	Definition	Equation	Device
Through Variable	Flux Tesla	Φ	
Across Variable	MMF Amperes	mmf	
Dissipator	Loss	mmf = L * d Φ/dt	
Delay	Permeance	mmf = Φ * R	
Accumulator	not meaningful physically	Φ = C dmmf/dt	
Through Generator	Flux source	Φ = const	
Across Generator	MMF Source	mmf = const	

Fundamental Constants

Constant	Unit	Name	Default
Electron charge	C	PHYS_Q	602_176_462e-19
permittivity of vacuum	F/m	PHYS_EPS0	8.854_187_817e-12
permeability of vacuum	H/m	PHYS_MU0	4.0e-7 * MATH_PI
Boltzmann's constant	J/K	PHYS_K	1.380_6503e-23
Acceleration due to gravity	ms^{-2}	PHYS_GRAVITY	9.806_65
Conversion between Kelvin and degree Celsius	-	PHYS_CTOK	273.15
Velocity of light in a vacuum	m/s	PHYS_C	299_792_458.0
Planck's constant	-	PHYS_H	6.626_068_76e-34
Planck's constant divided by 2pi	-	PHYS_H_OVER_2_PI	PHYS_H/MATH_2_PI

Material Constants

Constant	Unit	Name
Relative permittivity of silicon	-	PHYS_EPS_SI
Relative permittivity of silicon dioxide	-	PHYS_EPS_SIO2
Young's Modulus for silicon	Pa	PHYS_E_SI
Young's Modulus for silicon dioxide	Pa	PHYS_E_SIO2
Young's Modulus for polysilicon	Pa	PHYS_E_POLY
Poisson's Ratio for silicon	100 orientation	PHYS_NU_POLY
Density of Polysilicon	Kg/m^3	PHYS_RHO_POLY
Density of Silicon-Dioxide	Kg/m^3	PHYS_RHO_SIO2
Ambient Temperature	K	AMBIENT_TEMPERATURE
Ambient Pressure	Pa	AMBIENT_PRESSURE
Ambient Luminance		AMBIENT_LUMINANCE

Domain natures

Technology	Through	Across
Electrical	current	voltage
Magnetic	flux	mmf
Thermal	temperature	heat flow
Translational	velocity	force
Rotational	angular velocity	torque
Fluidic	flow rate	pressure
Radiant	luminous flux	luminous intensity

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";
```

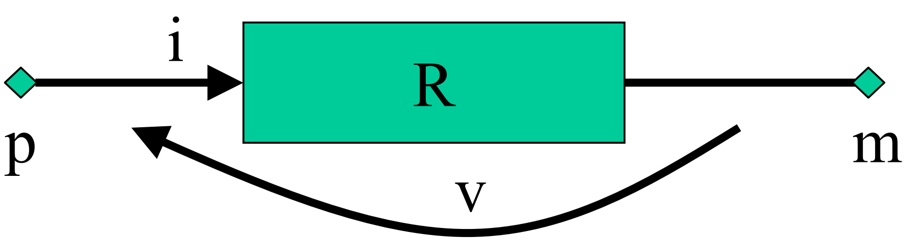
Simple Example: resistor

```
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;
```

```
SUBTYPE voltage is REAL;
SUBTYPE current is REAL;
NATURE electrical IS
        voltage ACROSS
        current THROUGH
        ground REFERENCE;
```



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)

E-ballot Process (November 2003-January 2004)

Examples: Mixed-domain modeling

Electro-Magnetic Model

- Winding Model
- Linear Core

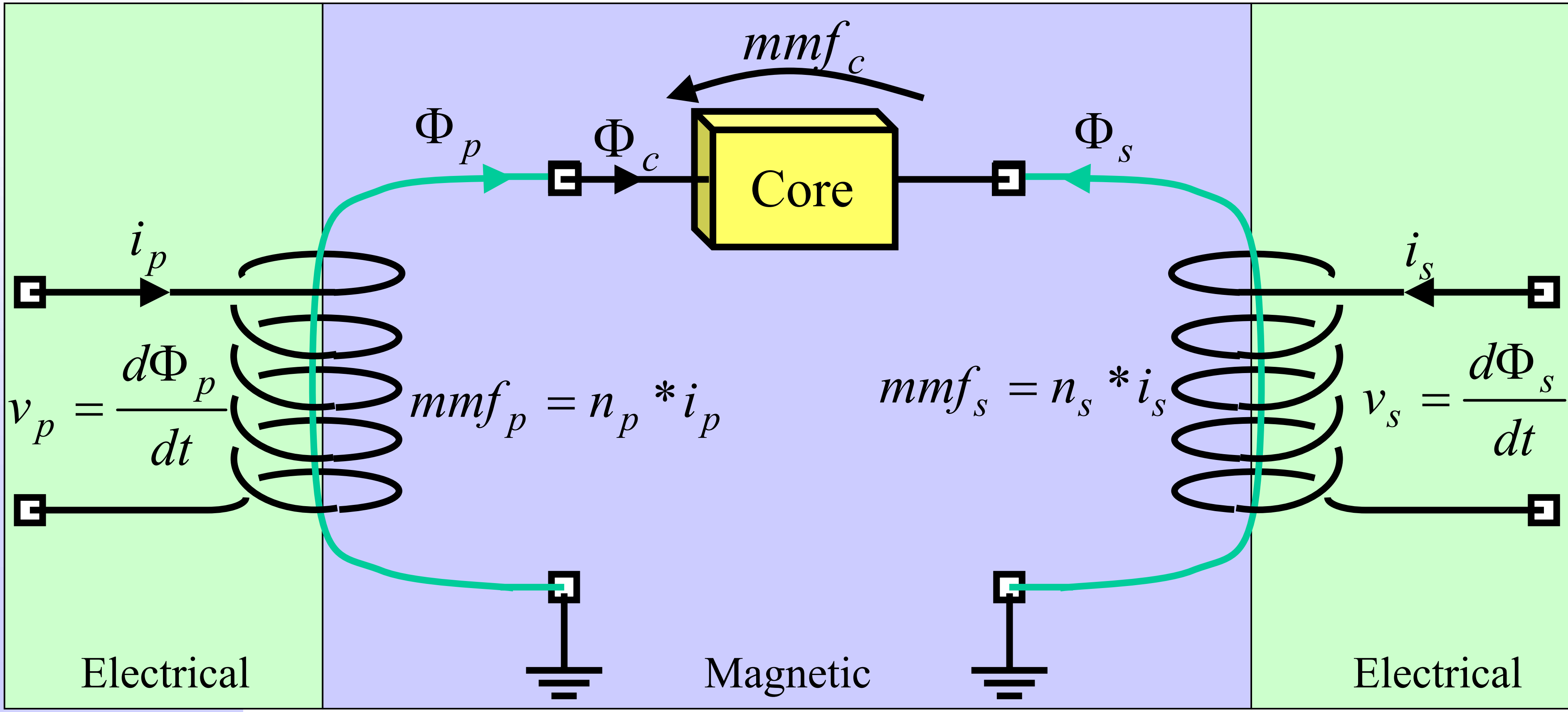
Non-linear magnetic model

- Non-linear core model – Jiles-Atherton

Thermal extensions:

- Electrical-Magnetic-Thermal model

Example Transformer Structure



Winding Model in VHDL-AMS

Ampere's Law

$$NI = \oint_l H \cdot dl$$

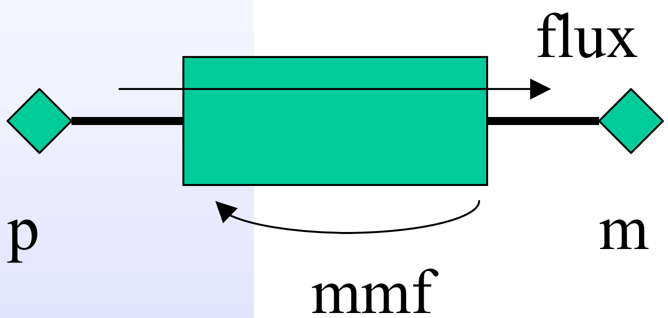
Faraday's Law

$$V = -N \frac{d\Phi}{dt}$$

```
USE work.electrical_systems.ALL;
ENTITY wind IS
        GENERIC (r,n : real :=0.0);
        PORT (  TERMINAL ep,em : electrical;
                TERMINAL 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

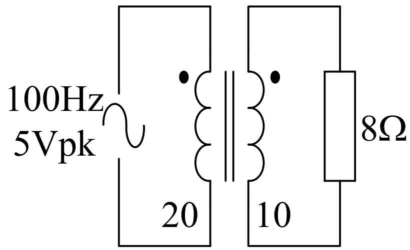


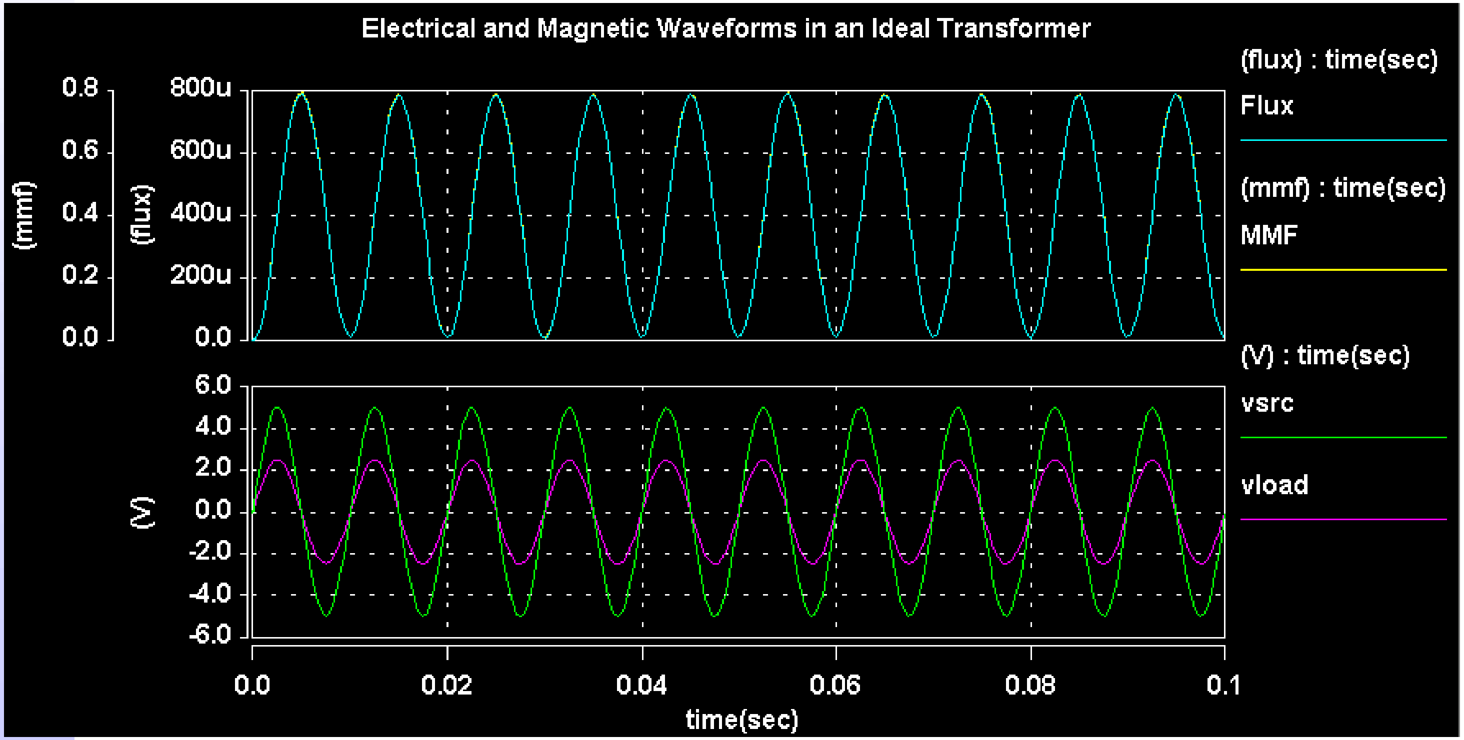
MMF = Reluctance * Flux
Flux = Permeance * MMF
Permeance = 1/Reluctance
= μ0*μr*area/len

```
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 == (mu0 * ur * area / len ) * mmf;
END ARCHITECTURE simple;
```

Transformer Simulation





Simulation carried out with VeriasHDL - Avant!

Non-linear Hysteresis: Jiles Atherton

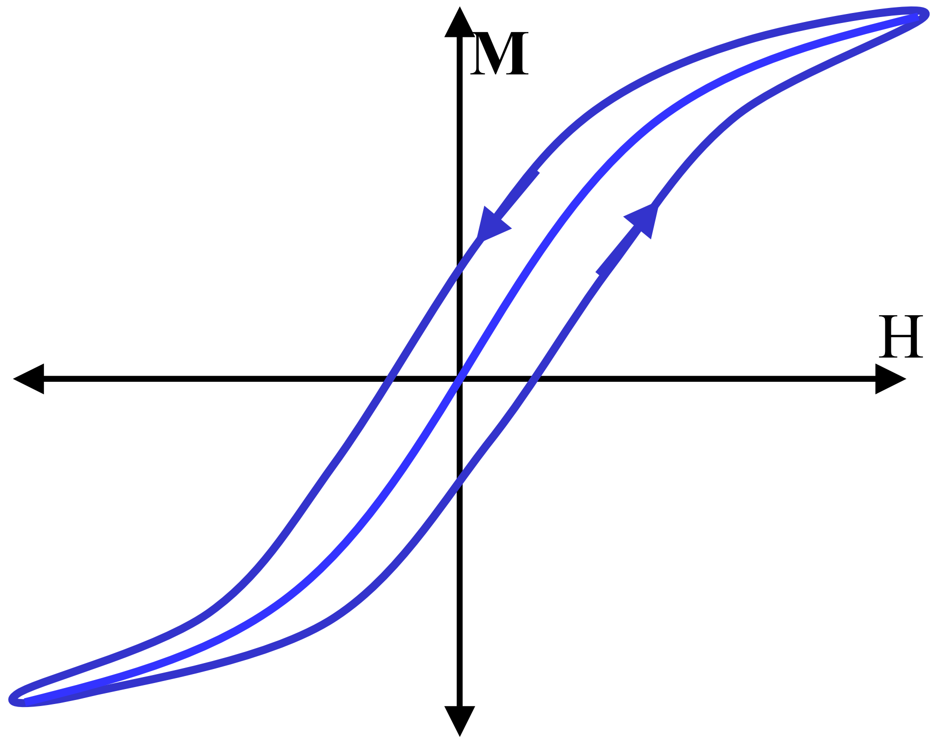
Differential Equation approach to modeling ferromagnetic hysteresis

Reversible Magnetization

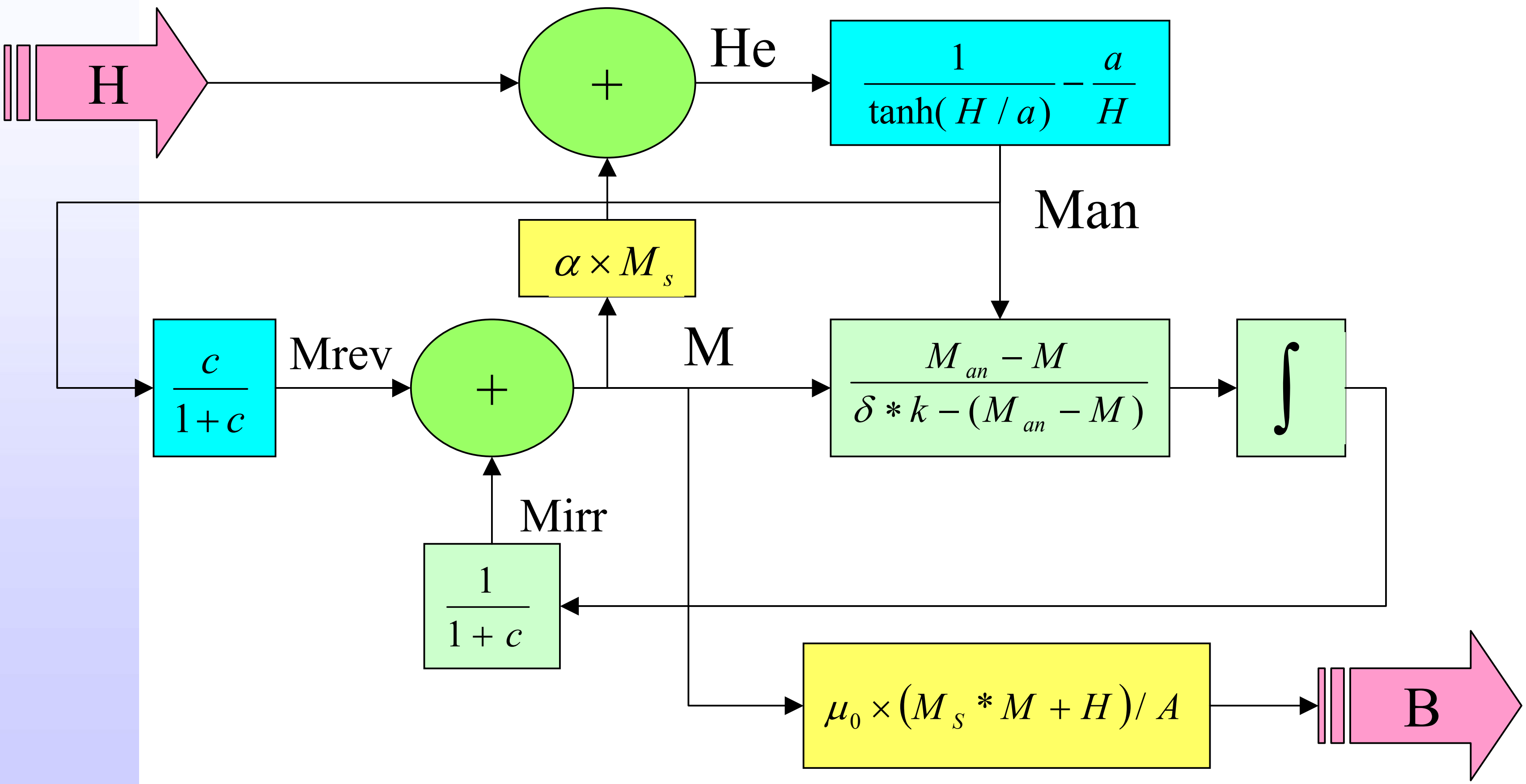
$$M_{an} = \frac{1}{\tanh(H/a)} - \frac{a}{H}$$

Irreversible Magnetization

$$\frac{dM_{irr}}{dH} = \frac{M_{an} - M}{\delta * k - (M_{an} - M)}$$



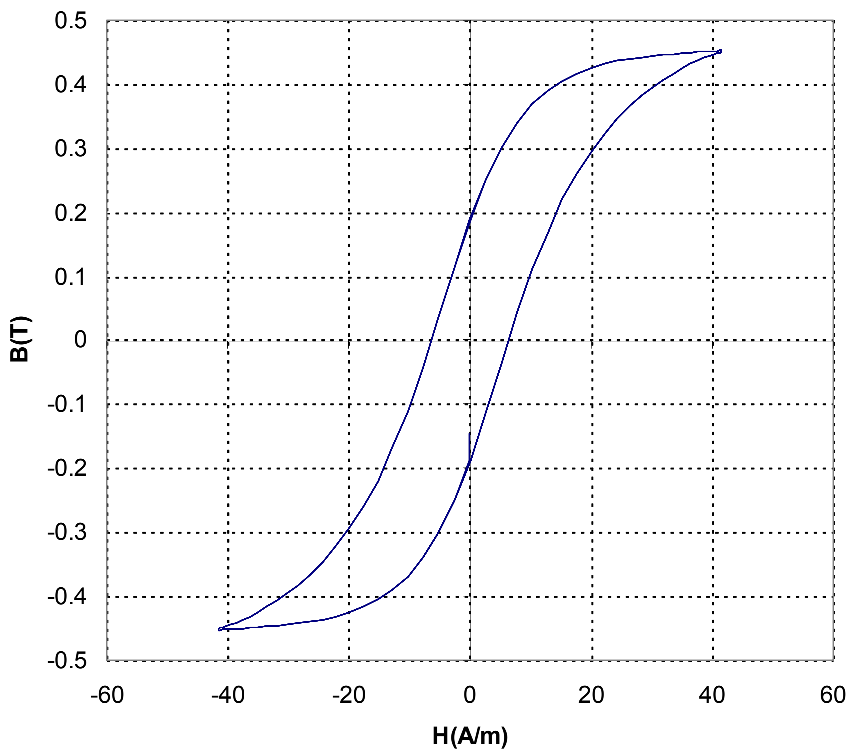
Jiles-Atherton Model Structure



Jiles-Atherton model in VHDL-AMS

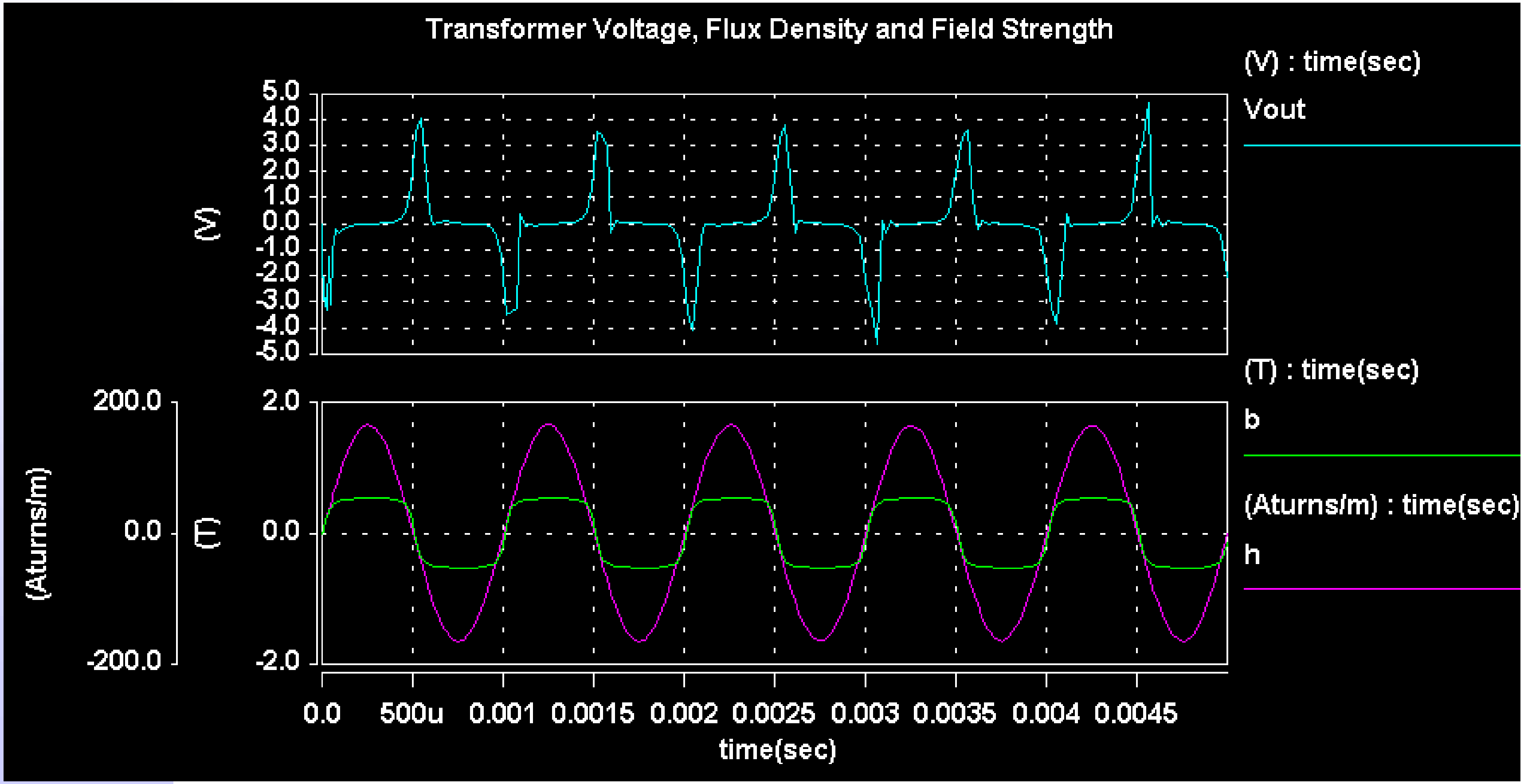
BH Curve

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



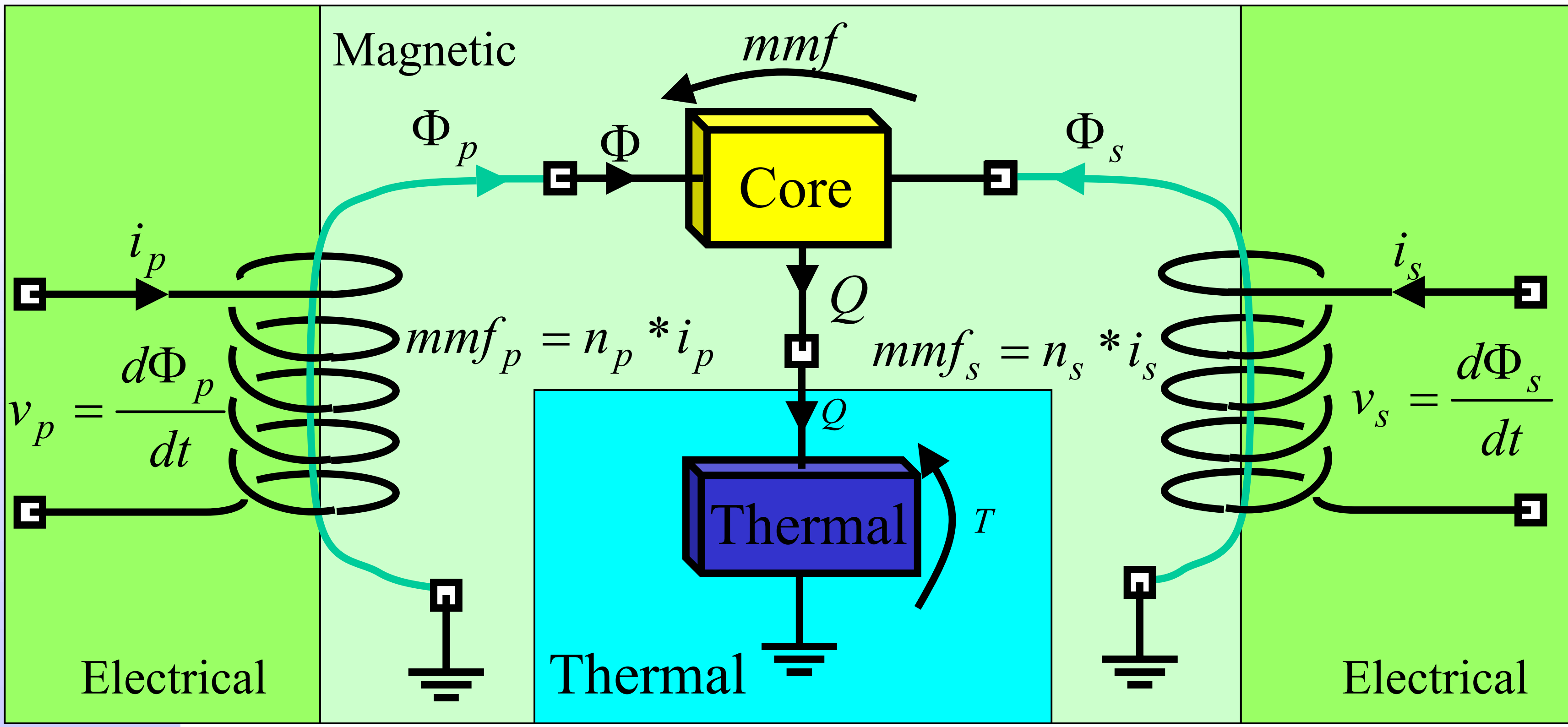
Results obtained using Mentor Graphics' Design Station

Non-linear Transformer: VHDL-AMS

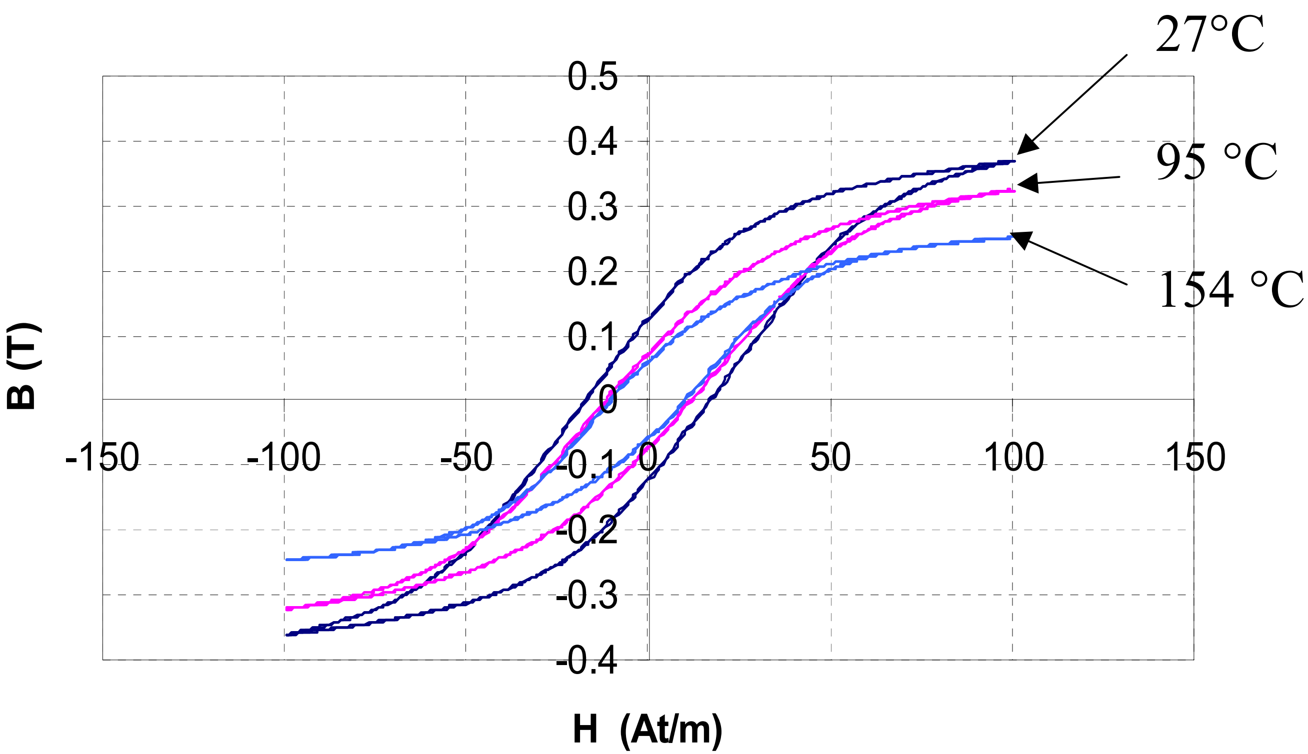


Results obtained using Avant!'s VeriasHDL

Extensions to include thermal effects

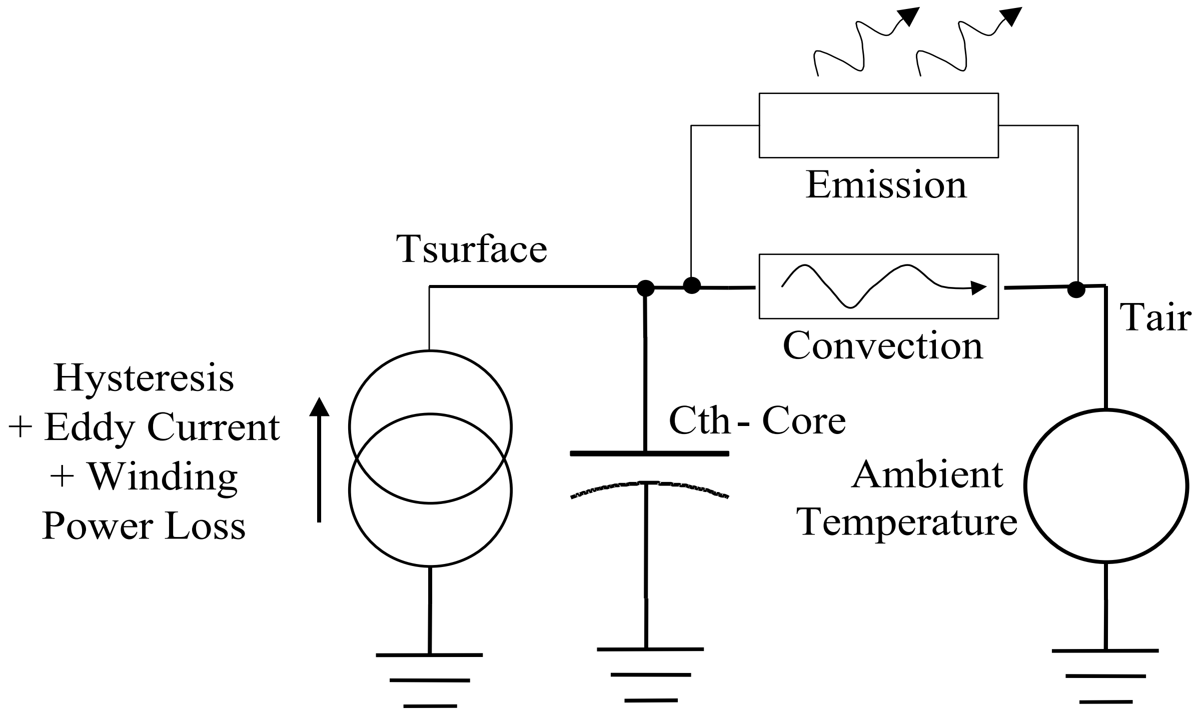


Variation with Temperature



Thermal model of the magnetic core

Assume uniform distribution of heating in the core material



Thermal Capacitance Model

```
use work.thermal_systems.all;

entity ctherm is
  generic (cth:real := 0.0);
  port (terminal th,tl : thermal);
end entity ctherm;

architecture simple of ctherm is
  quantity tc across heatfl through th to tl;
begin  -- simple architecture
  assert cth /= inf and cth /= undef
  report "cth specified incorrectly"
  severity error;

  heatfl == cth * tc'dot;
 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

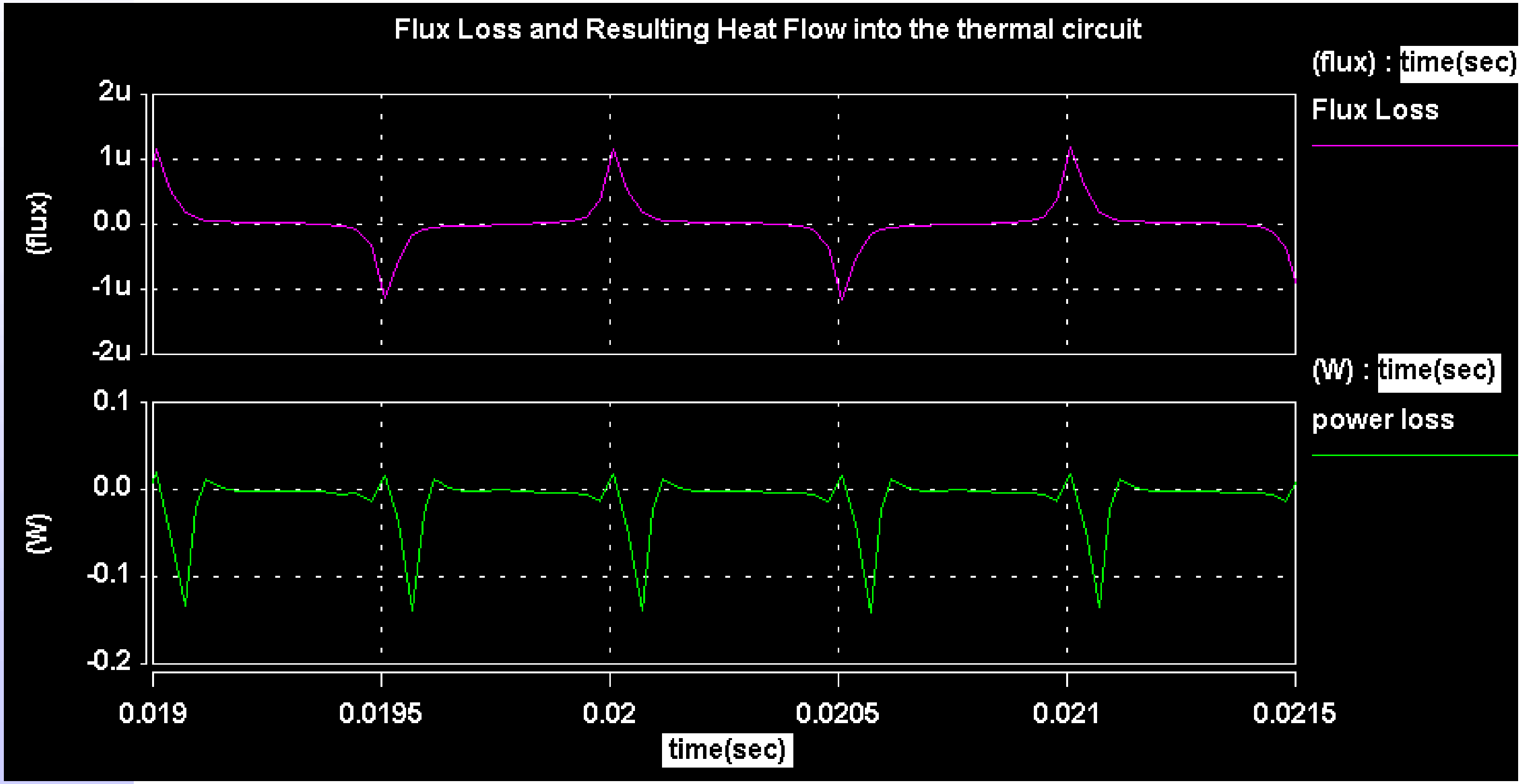
2: Use the irreversible proportion of the magnetization

- + 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

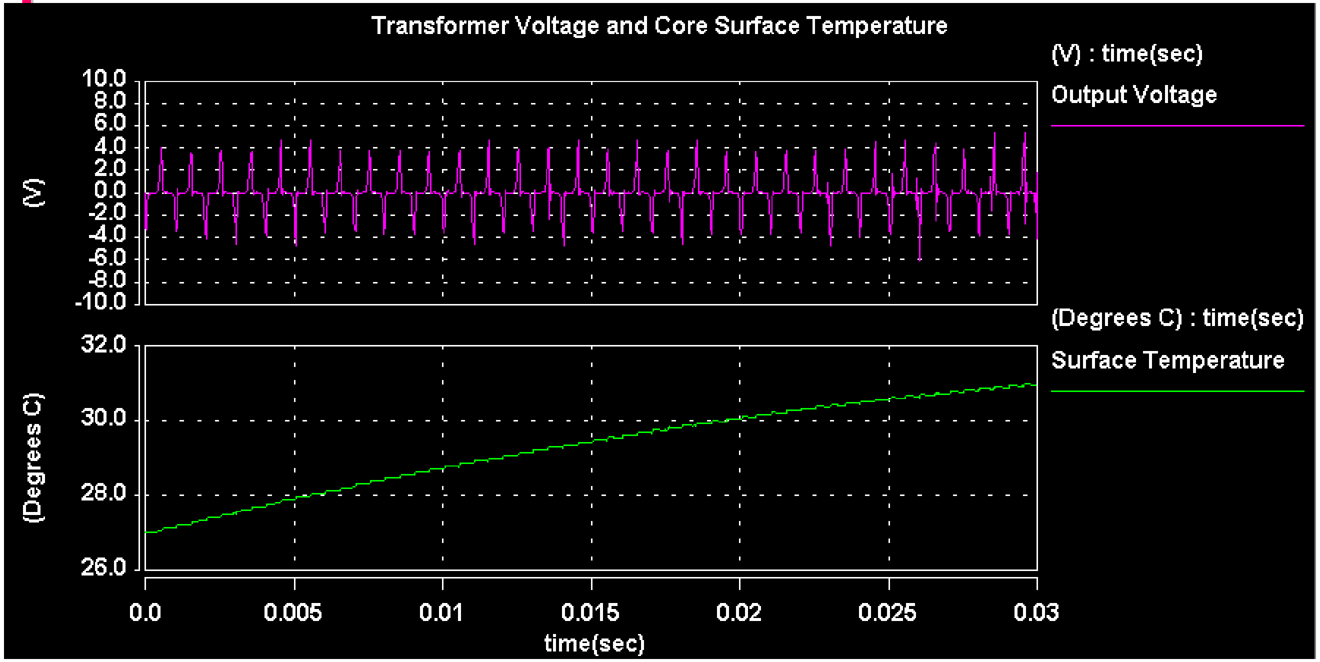
- Power is fed into the thermal circuit from the core loss directly
- Dynamic temperature changes in the core are modeled and used to modify the model parameters
- Dynamic thermal behaviour of the core material and surface is modeled
- Thermal effects will have a profound effect on the *electrical* behaviour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

Simulations including thermal model



Simulations including thermal model

Final Temperature rise accurate to within 0.3ºC



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

Electro-Mechanical

```
use work.electrical_systems.all;
use work.mechanical_systems.all;

entity dc_motor is
    generic (kt : real;
             j  : real;
             r  : real;
             ke : real;
             d  : real;
             l  : real);
    port (terminal p, m  : electrical;
          terminal rotor : rotational_v );
end entity dc_motor;

architecture behav of dc_motor is
    quantity w across t through rotor
         to rotational_v_ref;
    quantity v across i through p to m;
begin
  v == l*i'DOT + i*r + ke*w;
  t == i*kt - j*w'DOT - d*w;
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.