Proposal to Extend Frequency Domain Analysis in VHDL-AMS

Joachim Haase
Fraunhofer-Institut Integrierte Schaltungen
Branch Lab Design Automation (EAS) Dresden

Ewald Hessel and Heinz-Theo Mammen
Hella KGaA Hueck & Co Lippstadt
VHDL-AMS Approach for Time and Frequency Domain Analysis

**VHDL-AMS Code**

- Nonlinear equations for QUIESCENT_DOMAIN
- DAE’s for TIME_DOMAIN
  \[ F(x, x', p, t) = 0 \]

- Complex equations for FREQUENCY_DOMAIN
  \[
  \left. \frac{\partial F}{\partial x} \right|_{x=0} \cdot X(j\omega) + \left. \frac{\partial F}{\partial x'} \right|_{x' = 0} \cdot j\omega \cdot X(j\omega) = RS(j\omega)
  \]

Approximation may be a difficult step
Purpose of the Proposal

Motivation

- Whereas the translation from time to frequency domain models is easy to handle, it is much more difficult to derive consistent time models from frequency domain descriptions in the general case. Nevertheless, there exist several problems where only a frequency domain characteristic is given.

- A typical case is the description of the terminal behavior by frequency-dependent parameters – given for instance by a Touchstone file. Translation into a linear network may be difficult.

Objective

- However, it makes sense to extend the possibilities of behavioral modeling languages to provide models that are only used for frequency analysis and avoid the translation from frequency to time domain models.

- We show how, with little VHDL-AMS standard extensions, this can be done.
Inconsistencies of Descriptions

Counter-argument
Models for TIME_DOMAIN and FREQUENCY_DOMAIN can be inconsistent.

However, this is already with the current VHDL-AMS LRM possible

```vhdl
library IEEE;
use IEEE.ELECTRICAL_SYSTEMS.all;
entity L_EXAMPLE is
generic (V0 : REAL := 1.0;
         LT : REAL := 1.0;
         LF : REAL := 10.0);
port (terminal P, N : ELECTRICAL);
end entity BENCH_AC;

architecture A0 of L_EXAMPLE is
quantity V across I through P to N;
begin
if DOMAIN = QUIESCENT_DOMAIN use
  V == V0;
elsif DOMAIN = TIME_DOMAIN use
  V == LT * I´DOT;
else
  V == LF * I´DOT;
end use;
end architecture A0;
```
Core of Proposal

The domain signal is set to FREQUENCY_DOMAIN. The function FREQUENCY delivers the current simulation frequency. The value of the time delivered by the function NOW is 0. The system for the frequency analysis at \( \tilde{\omega} = 2\pi \cdot f \) is given by the linearized form

\[
\frac{\partial \tilde{F}}{\partial x}(\text{arg}) \cdot X(j\tilde{\omega}) + \frac{\partial \tilde{F}}{\partial x'}(\text{arg}) \cdot j\tilde{\omega} \cdot X(j\tilde{\omega}) = RS(j\tilde{\omega})
\]

at \( (\text{arg}) = (x_0, x_0', 0, f) \). The linearization is carried out around the QUIESCENT_DOMAIN operating point as described in the current VHDL-AMS LRM.

All other parts of the standard are not affected.
Fundamental Example

\[ X_1(j\omega) = Y(j\omega) \cdot X_2(j\omega) = (Y \cdot \text{RE}(j\omega) + j \cdot Y \cdot \text{IM}(j\omega)) \cdot X_2(j\omega) \]

equivalent to ( \(\omega \neq 0\) )

\[ X_1(j\omega) = Y \cdot \text{RE}(j\omega) \cdot X_2(j\omega) + \frac{Y \cdot \text{IM}(j\omega)}{\omega} \cdot j\omega \cdot X_2(j\omega) \]

-- simultaneous statement

if \(\text{DOMAIN} = \text{FREQUENCY\_DOMAIN}\) use

\[ X_1 \equiv Y \cdot \text{RE}(\text{FREQUENCY}) \times X_2 \]
\[ + \frac{Y \cdot \text{IM}(\text{FREQUENCY})}{(\text{MATH\_2\_PI} \times \text{FREQUENCY})} \times X_2' \times \text{DOT} \]

end use;
Details – Quiescent Domain Analysis

In order to be able to apply the given procedure, the VHDL-AMS standard should allow for the use of the real-valued function FREQUENCY in simultaneous statements. The evaluation of the statements can be done in the following way where the DAE similar to (1) has to be evaluated

\[ \tilde{F}(x, x', p, t, f') = 0 \]  

(11)

with frequency \( f \).

1) Operating point analysis: The domain signal is set to QUIESCENT_DOMAIN. The value of time \( t \) delivered by the function NOW is 0. The value of frequency \( f \) delivered by the function FREQUENCY is 0. The following system extended by initial conditions has to be solved

\[ \tilde{F}(x_0, x_0', p, 0, 0) = 0 \]  

(12)
2) Time domain analysis: The domain signal is set to TIME_DOMAIN. The function NOW delivers the current simulation time. The value of frequency f delivered by the function FREQUENCY is 0. The following DAE system has to be solved

\[ \tilde{F}(x, x', p, t, 0) = 0 \]  \hspace{1cm} (13)
3) **Frequency domain analysis:** The domain signal is set to FREQUENCYDOMAIN. The function FREQUENCY delivers the current simulation frequency \( \tilde{f} \). The value of the time delivered by the function NOW is 0. The system for the frequency analysis at \( \tilde{\omega} = 2\pi \cdot \tilde{f} \) is given by the linearized form

\[
\frac{\partial \tilde{F}}{\partial x}(\text{arg}) \cdot X(j\tilde{\omega}) + \frac{\partial \tilde{F}}{\partial x'}(\text{arg}) \cdot j\tilde{\omega} \cdot X(j\tilde{\omega}) = RS(j\tilde{\omega}) \quad (14)
\]

at \((\text{arg}) = (x_0, x'_0, 0, \tilde{f})\). The linearization is carried out around the \( x_0, x'_0 \) QUIESCENT_DOMAIN operating point as described in the sections 7.6 and 12.6.5.4 of [5].
Typical Application – EMC Analysis

System under test
described by frequency characteristics

noise source with
given spectral characteristic

spectrum of load voltage
has to be determined
Tool Support

Supported in an undocumented way by some tools
Further Examples

**S11 – Parameter- Description (1)**

\[ B(j\omega) = S11(j\omega) \cdot A(j\omega) \]

with

\[ A(j\omega) = \frac{1}{2\sqrt{R_0}} \cdot (V(j\omega) + R_0 \cdot I(j\omega)) \]

\[ B(j\omega) = \frac{1}{2\sqrt{R_0}} \cdot (V(j\omega) - R_0 \cdot I(j\omega)) \]
Further Examples

S11 – Parameter- Description (2)

library IEEE, FUNDAMENTALS VDA;
use IEEE.ELECTRICAL_SYSTEMS.all;
use IEEE.MATH_REAL.all;
use FUNDAMENTALS_VDA.TLU_VDA.all;

entity Z_S11_AC is

  generic ( 
    F      : REAL_VECTOR;
    S11_RE : REAL_VECTOR;
    S11_IM : REAL_VECTOR;
    R0     : REAL := 50.0
  );

  port ( 
    terminal N1 : ELECTRICAL;
    terminal N2 : ELECTRICAL
  );

end entity Z_S11_AC;

architecture A0_AC of Z_S11_AC is

  quantity V across I through N1 to N2;
  quantity AS, BS : REAL; -- A, B * 2*SQRT(2)

begin

  assert DOMAIN /= TIME_DOMAIN
    report
    "ERROR: Time domain model is not implemented."
    severity ERROR;

  AS := V + R0*I;
  BS := V - R0*I;

  if FREQUENCY = 0.0 use

    BS := LOOKUP_1D (FREQUENCY, F, S11_RE) * AS;

  else

    BS := LOOKUP_1D (FREQUENCY, F, S11_RE) * AS
          + LOOKUP_1D (FREQUENCY, F, S11_IM)
            /MATH_2_PI/FREQUENCY * AS’*DOT;

  end use;

end architecture A0_AC;
Further Examples

Lossy Line (1)

\[
\begin{bmatrix}
V_2(j\omega) \\
-I_2(j\omega)
\end{bmatrix} =
\begin{bmatrix}
A_{11}(j\omega) & A_{12}(j\omega) \\
A_{21}(j\omega) & A_{22}(j\omega)
\end{bmatrix}
\begin{bmatrix}
V_1(j\omega) \\
I_1(j\omega)
\end{bmatrix}
\]

\[A_{12}(j\omega) = ZW(j\omega) \cdot \sinh(\gamma(j\omega) \cdot LEN)\]
Further Examples

Lossy Line (2)

library IEEE;
use IEEE.ELECTRICAL_SYSTEMS.all;
use IEEE.MATH_REAL.all;
use IEEE.MATH_COMPLEX.all;

entity LINE_AC is

  generic (  
    L  : REAL;
    C  : REAL;
    LEN : REAL := 1.0;
    R  : REAL := 0.0;
    G  : REAL := 0.0);

  port (  
    terminal N1 : ELECTRICAL;
    terminal N2 : ELECTRICAL;
    terminal N3 : ELECTRICAL;
    terminal N4 : ELECTRICAL);

begin
  assert C > 0.0 or G > 0.0;
  assert L > 0.0 or R > 0.0;
end entity LINE_AC;

architecture A0_AC of LINE_AC is

  function DETERMINE_A12 (LEN, L, C, R, G, F : REAL) return COMPLEX is
    variable ARGUMENT : COMPLEX;
    variable RESULT : COMPLEX;
  begin
    if not (G = 0.0 and F = 0.0) then
      ARGUMENT := DETERMINE_GAMMA_TIMES_LEN
                  (LEN, L, C, R, G, F);
      RESULT := COSH(ARGUMENT);
    else
      RESULT.RE := 1.0;
      RESULT.IM := 0.0;
    end if;

    if F = 0.0 then
      RESULT.IM := 0.0;
    else
      RESULT.IM := RESULT.IM/MATH_0_PI/F;
    end if;
  return RESULT;
end function DETERMINE_A12;

...
Further Examples

Lossy Line (3)

quantity V1 across I1 through N1 to N2;
quantity V2 across I2 through N3 to N4;

begin

V2 ==
  DETERMINE_A11(LEN,L,C,R,G,FREQUENCY) .RE * V1
  + DETERMINE_A11(LEN,L,C,R,G,FREQUENCY) .IM * V1' DOT
  + DETERMINE_A12(LEN,L,C,R,G,FREQUENCY) .RE * I1
  + DETERMINE_A12(LEN,L,C,R,G,FREQUENCY) .IM * I1' DOT;

I2 ==
  - DETERMINE_A21(LEN,L,C,R,G,FREQUENCY) .RE * V1
  - DETERMINE_A21(LEN,L,C,R,G,FREQUENCY) .IM * V1' DOT
  - DETERMINE_A22(LEN,L,C,R,G,FREQUENCY) .RE * I1
  - DETERMINE_A22(LEN,L,C,R,G,FREQUENCY) .IM * I1' DOT;

assert DOMAIN /= TIME_DOMAIN
report
"ERROR: Time domain model is not implemented."
severity ERROR;

end architecture A0_AC;