Systèmes Hétérogènes
TP 1 : initiation à la modélisation sous VHDL-AMS

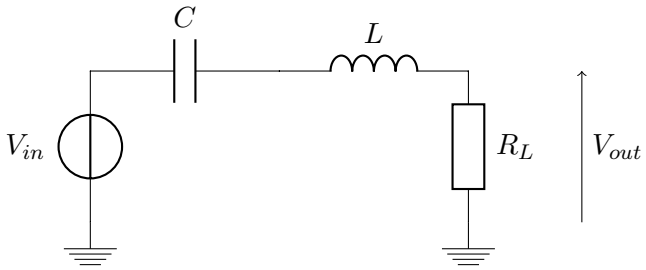
Ce TP a pour objectifs :

- la familiarisation avec la syntaxe du VHDL-AMS,
- la prise en main de l'environnement de simulation *Mentor*.

Vous rendrez à la fin du TP un compte rendu par personne sous forme numérique, dans lequel vous incluerez les codes permettant de répondre aux questions et vos réponses. Les comptes rendus doivent être transmis par couriel à l'adresse *florian.kolbl(at)ensea.fr* (vous remplacerez le (at) par un caractère arobase).

1 Première simulation : Circuit analogique du second ordre

On cherche à modéliser le circuit bouchon suivant :



1.1 Création de l'environnement de travail

1. La première étape consiste à créer un espace et une bibliothèque de travail, pour cela :
   (a) créer un dossier (*mkdir*) dans lequel vous tarvaillerez, puis accéder à ce dossier,
   (b) créer une librairie de travail, en utilisant la commande : **valib ma_librairie**,
   Rmq : le nom n'est pas nécessairement celui de du dossier que vous venez de créer,
   (c) désigner cette même librairie comme librairie de travail, à laquelle seront rattachés les modèles issus de la compilation, vous utiliserez pour cela la commande : **vasetlib ma_librairie**.

A ce stade, vous pouvez commencer à écrire les fichiers vhdl pour décrire le circuit. Le ligociel que vous venez d'utiliser a été développée par *Mentor* et fait partie de la suite *ADMS*. Cette suite contient les outils pour simuler et visualiser les résultats. Elle n'est pas limitée au VHDL-AMS, et dispose également de solveurs de type *Spice* et *Verilog*. Cette suite est très utilisée dans la conception de circuits intégrés outre-atlantique, alors que *Cadence* est d'avantage utilisé en Europe. Les Flots de conception sont rigoureusement identiques sur les deux outils, de même pour le code VHDL-AMS, qui peut être compilé avec n'importe quelle plateforme adaptée. Les commandes du logiciel sont données en annexe.

1.2 Création des sous-modèles pour le schéma

2. On commence la modélisation par le schéma de la source de tension. Le code vous est donné ci-dessous, vous l'analyserez dans cette question :

```
1  library IEEE;
2  use IEEE.math_real.all;
3  use IEEE.electrical_systems.all;
4
5  entity vsource is
6    generic(
7      a
8      ac_mag, ac_phase : real :=0.0;
9    );
10   port(
11     terminal p, m : electrical
12   );
13 end entity vsource;
14
15 architecture bce of vsource is
16   quantity vsrc through p to m;
17   quantity ac_spec : real spectrum ac_mag, ac_phase;
18 begin
19   if (DOMAIN = QUIESCENT_DOMAIN or DOMAIN = TIME_DOMAIN) use
20     vsrc == src_ampl*sin(MATH_2_PI * src_freq * NOW);
21   else
22     vsrc == ac_spec;
23 end architecture bce;
```

   (a) Quelle librairies sont chargées sur les lignes 1 à 3 ? Que permettent-elles de faire ?

   (b) Quels sont les numéros des lignes de code du fichier qui sont concernées par ces libraries ?

   (c) Que réalise le test conditionnel en ligne 19 ? Expliquez brièvement le fonctionnement du script.

   (d) Compilez le fichier tel-quel, en utilisant la la commande **vacom mon_modele.vhd**.

3. Réalisez et compilez :

   (a) un fichier *.vhd* générique qui modélise une résistance idéale,

   (b) un fichier *.vhd* générique qui modélise une inductance idéale,

   (c) un fichier *.vhd* générique qui modélise une capacité idéale, et qui permet de paramétrer la condition initiale en tension de la capacité.

4. créer le fichier '*filtre_bouchon.vhd*' qui assure la connection des composants, on utilisera les valeurs suivantes :

   - $R_L = 50\Omega$, $L = 1\mu H$, $C = 1\mu F$,
   - en temporel, une fréquence de $10MHz$ et une amplitude de 1 volt,

- en fréquentiel une fréquence qui permet de simplifier la formule du gain pour n'avoir qu'a visualiser la sortie.

attention : lors de l'écriture, vous pouvez avoir une entity mais vide, les entrée sorties ne sont pas gérées par le simulateur. Tout point de mesure doit être déclaré comme une quantité.

simulation et validation du modèle

5. Exprimer la fonction de transfert

$$H(\jmath\omega) = \frac{v_{out}(\jmath\omega)}{v_{in}(\jmath\omega)}$$

en fonction de R, L et C.

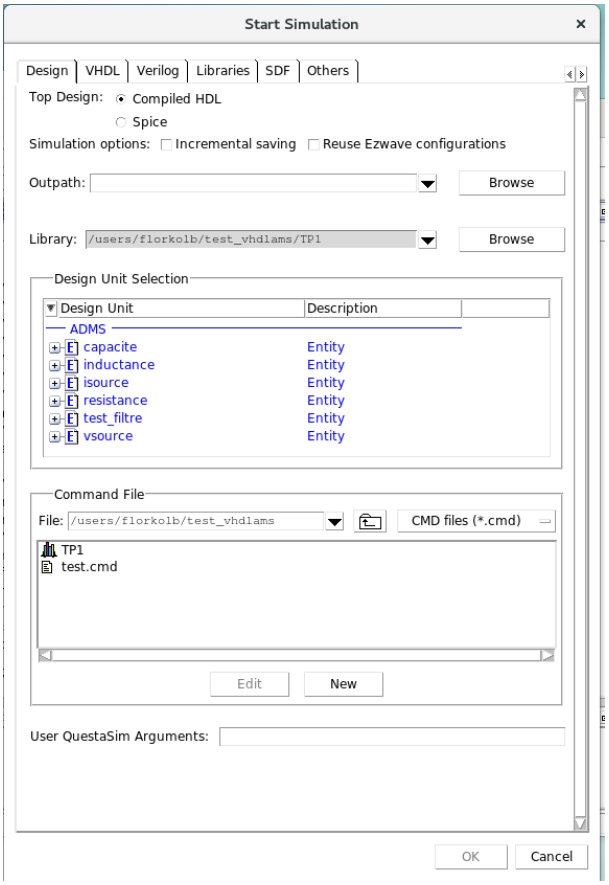
6. En utilisant *Python* (un exemple de script simple est donné en annexe), tracez le diagramme de Bode du circuit bouchon.

une simulation sous *ADMS* se déroule en deux étapes :

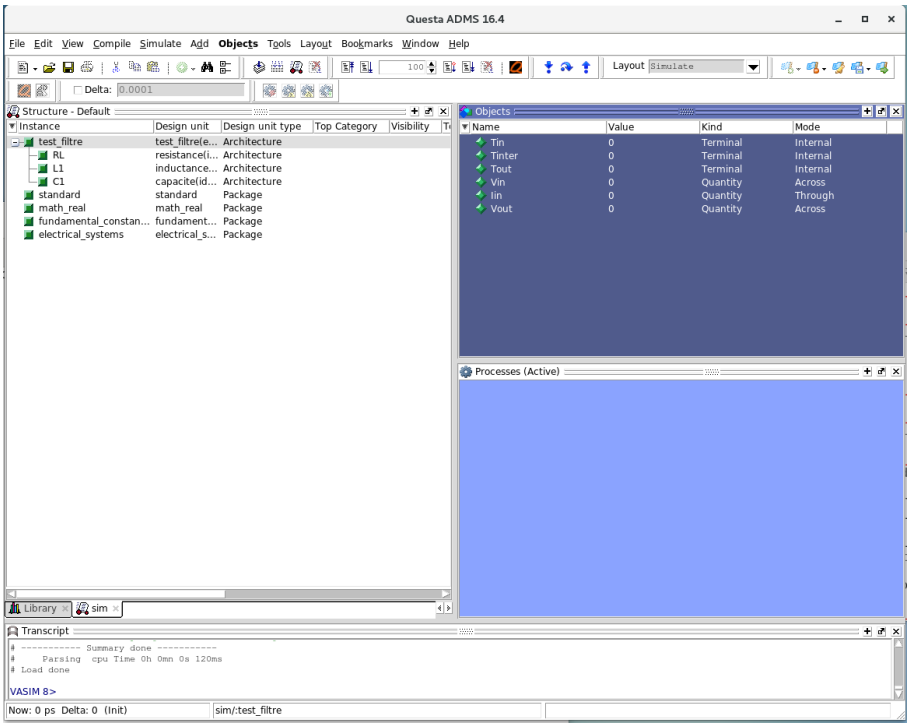
- la rédaction d'un fichier de commande *.cmd* qui contient les instructions de simulation. Ces instructions sont identiques aux commandes spice. A titre d'exemple, le code suivant permet de réaliser une simulation temporelle (*transiant*) :

```
1 * TEST TEMPOREL DU FILTRE BOUCHON
2 .option voltage_loop_severity = warning
3 .tran 1u 5m 0u 0.5u
4 .end
```

- le lancement de l'environnement graphique de simulation, grâce à la commande **vasim**. Une Fenêtre apparait alors comme illustré dessous.



Vous pouvez alors sélectionner le fichier *.vhd* à simuler ainsi que le fichier *.cmd* correspondant à la simulation demandée. Une fois les fichiers sélectionnés, lorsque que vous cliquez sur OK, une autre fenêtre s'ouvre automatiquement : il s'agit de la fenetre permettant d'accéder aux résultats de simulation, comme illustré ci dessous :



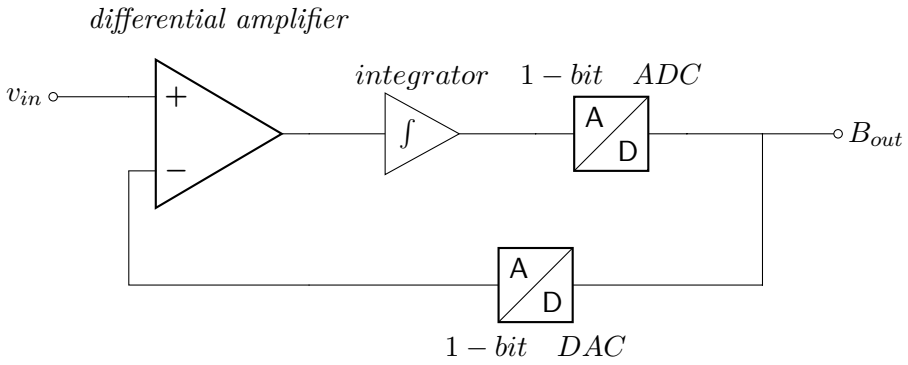
Dans cette fenêtre sont listée les quantités et terminaux déclarés dans le entités et architectures. L'affichage est possible en sélectionnant la grandeur à affichier, puis un cliq droit et '*add wave*'. Le lancement de la simulation est possible en lançant la commande **runn -all** dans l'invite de commande ou en cliquant sur le bouton suivant dans la bare des tâches :

7. lancer la simulation temporelle donnée dans l'exemple de fichier *.cmd*.
   (a) Expliquez le fonctionnement du fichier de commande en étudiant la réponse donnée par le logiciel.
   (b) Que vallent l'amplitude et la phase de la tension de sortie ? Est-ce cohérent avec les résultats obtenus sous *Matlab* ?

8. On cherche maintenant à afficher le diagramme de Bode du circuit bouchon par la simuatlion.
   (a) En vous aidant du fichier d'aide de *Mentor* donné avec le TP, créer un nouveau ficher de commande réalisant une simulation AC.
   Rmq : pour la simulation AC, l'affichage des grandeurs souhaité est géré directement dans le fichier de commande et non dans l'interface graphique, cette dernière ne servant que pour l'affichage final.
   (b) Les résultats sont-ils cohérents avec ceux obtenus sous python ?
   (c) Calculer à partir de vos résultats le facteur de qualité pour les valeurs de composants utilisées.

2 Simulation d'un système mixte

On souhaite étudier un système mixte, on prendra pour objectif de simuler un modulateur $\Delta\Sigma$ qui est un bloc de base des convertisseurs Analogiques/Numérique du même nom. Ce bloc a le schéma interne suivant :



Ces blocs sont décrit dans deux notes techniques :

- un tutorial de Maxim sur les ADC $\Sigma\Delta$,
- une note technique de Texas Instrument sur le même sujet.

9. Créer, compiler, tester et valider des modèles comportementaux VHL-AMS pour chacun des blocs du modulateur. Vous pouvez vous aider des notes techniques citée si dessus pour identifier le contenu du modèle. Vous testerez chaque bloc indépendament en créant une simulation ou un test-bench.

10. On souhaite moduler un signal sinusoïdal compris entre 0 et $3.3V$ de fréquence $1kHz$, soit une pleine échelle en tension d'un circuit en technologie CMOS $0.35\mu m$. Determinez les paramètres idéaux pour chaque sous-circuits, réalisez une simulation système et expliquez brièvement le fonctionnement du circuit.

Annexes

Exemple de script Python pour le tracé de Bode

```python
from scipy import signal
import matplotlib.pyplot as plt

s1 = signal.lti([1], [1, 1])
w, mag, phase = signal.bode(s1)

plt.figure()
plt.semilogx(w, mag)    # Bode magnitude plot
plt.figure()
plt.semilogx(w, phase)  # Bode phase plot
plt.show()
```

Commandes terminal de la suite ADMS

ADVance MS Quick Reference Guide

Mentor Graphics web site:
www.mentor.com

Mentor Graphics Support:
North America
Phone: 1-800-547-4303
www.mentor/supportnet
Worldwide
www.mentor.com/supportnet/support_offices.html

Installation / Environment / Licensing

Documentation
Getting Started with ADVance MS: *adms_gs.pdf*
ADVance MS User's Manual: *adms_ur.pdf*
VHDL-AMS Quick Reference: *vhdlams_qr.pdf*

Environment Variables

LM_LICENSE_FILE	**Required**	Pathname of *anacad.license*
ADMSIM	Optional	Pathname of *adms.ini* file

PATH Environment Variable
Add /<install_dir>/$anacad/bin to $PATH

Starting the License Server
Copy *anacad.license* to /<install_dir>/$anacad/mgls/
Run /<install_dir>/$anacad/com/setup_mgls

Invoking ADVance MS
Run /<install_dir>/$anacad/bin/vasim

Key Simulation Control Commands (.cmd file)

Command Syntax	Description
.tran *tprint tstop [tstart]*	Activates transient analysis with print step and duration of analysis
.option NOASCII	Simulation does not generate ASCII output (.chi)
.plot dc/ac/tran V(*node*) I(*node*)	Plot simulation results for voltage/current at node
.plot FOUR ...	Plot Fast Fourier Transform results
.probe VTOP	Gives output showing all top level node voltages
.extract ...	Extract waveform characteristics
.ac dec|oct|lin *n fstart fstop*	AC analysis; start and stop frequencies can be set
.dc ...	DC analysis
.defmac *mac_name*(*arg*{, *arg*})	Macro definition with written arguments
.defwave	Waveform definition
.end	End Eldo netlist
.four label=*lname wname*	FFT select waveform
.ic v(*nn*)=*val* {v(*nn*)=*val*}	Initial transient analysis conditions
.include *fname*	Include a filename in an input netlist
.nodeset	DC analysis conditions
.noise *outv insrc nums* (X*subckt*)	Noise Analysis
.noisetran ...	Transient Noise Analysis
.op ...	DC Operating Point calculation
.option ...	Simulator configuration
.param *par=val | par={expr} | par="name"*	Global parameter declarations
.plotbus *bname*	Plotting of bus signals
.print ...	Printing of results
.setbus *bname pn*	Creates a bus *bname* with number of bits *pn*
.sigbus ...	Sets signal on a bus
.temp *ts*	Set circuit temperature
.use ...	Use previously simulated results

Key ADVance MS Commands (see User's Manual for more)

Command	Where Used **(Sh)ell** **(ADMS)**	Description For details on these commands refer to the ADVance MS User's Manual
vacom	Sh, ADMS	VHDL-AMS compiler
vadel	Sh, ADMS	Deletes a unit from a selected library
vadir	Sh, ADMS	Selectively lists the contents of a design library
valib	Sh, ADMS	Creates a design library
vasetlib	Sh	Sets the default working library
vaunlock	Sh	Restores the unlocked library **lib_name**
vasim	Sh, ADMS	VHDL-AMS and/or Verilog simulator
vamap	Sh, ADMS	Defines or displays library mappings
valog	Sh, ADMS	Verilog and Verilog-A compiler
import_adms	Sh, ADMS	Used to import from ADVance MS to ModelSim
import_ms	Sh, ADMS	Used to import digital portions from ModelSim to ADVance MS
add log	ADMS	Creates log files (*dou* and *cou*) for analysis with the "Wave" window
add log -delta	ADMS	**The si**mulator only has to solve analog data when it has sufficiently changed more than the given delta value
add wave	ADMS	Adds signals, quantities or terminals to the "Wave" window
add wave -delta	ADMS	**The si**mulator only has to solve analog data when it has sufficiently changed more than the given delta value
add list	ADMS	Will give signal output as an ASCII window
write list	ADMS	This will record the contents of the List window in an output file
batch_mode	ADMS	Typically used as a condition in an if statement. Returns 1 if ADVance MS is in batch mode and 0 if not
cd	Sh, ADMS	Changes directory
change	ADMS	Modifies the value of a VHDL-AMS variable or constant
examine	ADMS	Examines one or more VHDL-AMS item and displays its value in the "Transcripts" window
exit	ADMS	Exits the simulator and the ADVance MS application
find	ADMS	Displays full pathnames of matching VHDL-AMS items
force	ADMS	Forces interactive stimulus on VHDL-AMS nets
history	ADMS	Lists previous commands
ms	ADMS	Used with a simulation that uses ADVance MS and ModelSim together requires alternate sequential commands
noforce	ADMS	Removes the effect of any active **force** commands on selected VHDL-AMS items
probe	ADMS	Updates the mask on a net when running an AC analysis or running a *.do* file
pwd	Sh, ADMS	Displays current path
quit	ADMS	Exits the simulator and the ADVance MS application
restart	ADMS	Restarts the simulator
run	ADMS	Advances the simulation time
view	ADMS	Opens an ADVance MS window and brings it to the front