

# Tutorial Cadence Virtuoso®

## (Les premiers pas)

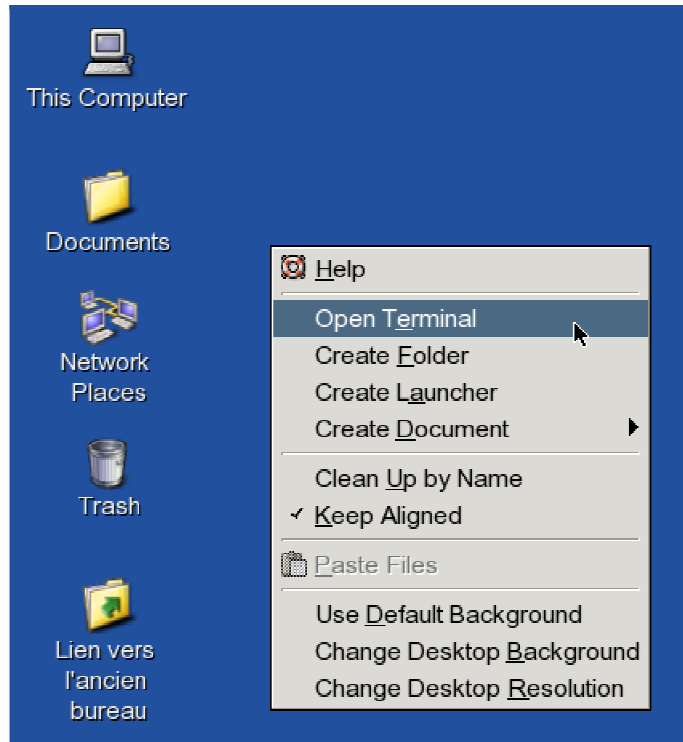
- Cadence Virtuoso® IC6.1.500.3
- Design Kit AustriaMicroSystems (AMS) HIT-Kit 4.00
- Process : c35b4c3 (0.35µm CMOS 4 métaux)

### Table des matières

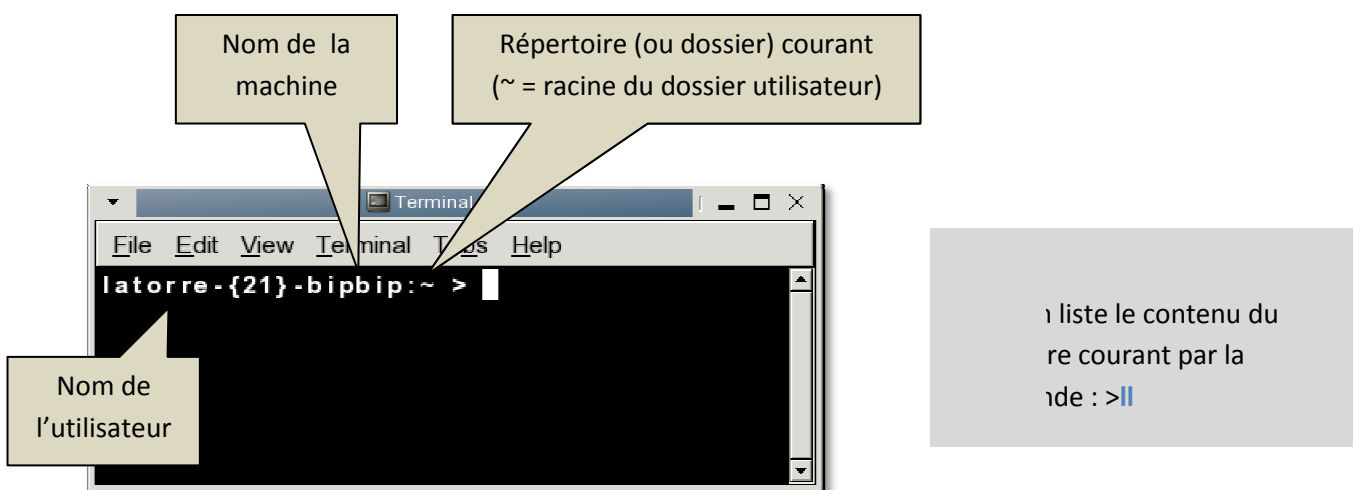
Login et ouverture d'un terminal .....	2
1. Préparation d'un répertoire de travail pour Cadence.....	3
2. Premier démarrage de Cadence Virtuoso .....	4
3. Aperçu du Library Manager.....	6
4. Création d'une librairie de travail.....	10
5. Edition de schéma .....	11
Placement des composants (ou <i>instances</i> ) .....	12
Interconnexions.....	14
6. Simulation.....	16
6.1. Simulation d'un point de polarisation simple .....	21
6.2. Examen des résultats.....	27
6.3. Simulation du point de polarisation avec une variable (sweep) .....	31
6.4. Analyse paramétrique du point de polarisation avec une variable (sweep).....	34

## Login et ouverture d'un terminal

Faire : **clic-droit** sur le bureau, puis **Open Terminal**.



Une fenêtre **Terminal** apparaît, avec une ligne de texte qui affiche des informations et vous invite à saisir une commande (invite de commande). Il est important de se familiariser avec informations disponibles sur cette invite de commande :



## 1. Préparation d'un répertoire de travail pour Cadence

Depuis le terminal ouvert, créer un répertoire de travail sous la racine de votre compte utilisateur (par exemple « TP\_CIA »). Si le répertoire existe déjà, cette étape doit être sautée.

```
~>mkdir TP_CIA
```

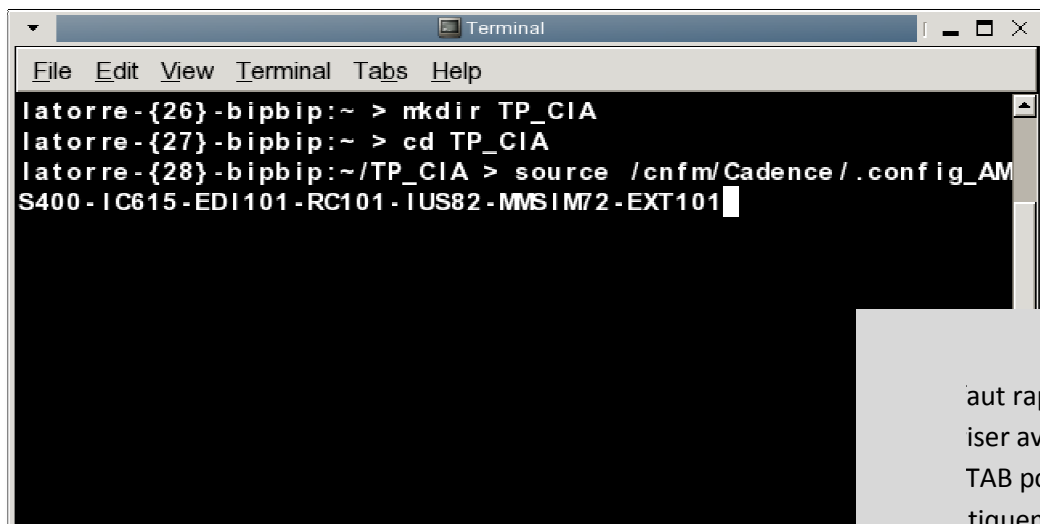
Aller dans le répertoire de travail (et vérifier que le répertoire courant sur l'invite de commande à bien changé :

```
~>cd TP_CIA
```

veau

Pour pouvoir lancer l'environnement **Cadence**, il faut définir des variables d'environnement, et des chemins (path) vers les fichiers exécutables. Les variables d'environnement permettent (entre autre) de configurer les outils Cadence pour la technologie CMOS ciblée. Dans le cas des TP CIA, nous allons utiliser la technologie CMOS 0,35µm du fondeur *AustriaMicrosystems (AMS)*. Les nombreuses lignes de commandes nécessaires à la configuration de **Cadence** pour cette technologie sont regroupées dans un fichier script que nous pouvons lire et exécuter en une seule commande :

```
~/TP_CIA> source /cnfm/Cadence/.config_AMS400-IC615-EDI101-RC101-IUS82-MMSIM72-EXT101
```



peut rapidement se  
utiliser avec l'usage de la  
TAB pour compléter  
tiquement (et  
ement !) la saisie de  
nde.

Le script affiche dans le terminal un rappel des différentes commandes valides pour tous les outils **Cadence** disponibles. On peut consulter cette notice en utilisant l'ascenseur vertical.

## 2. Premier démarrage de Cadence Virtuoso

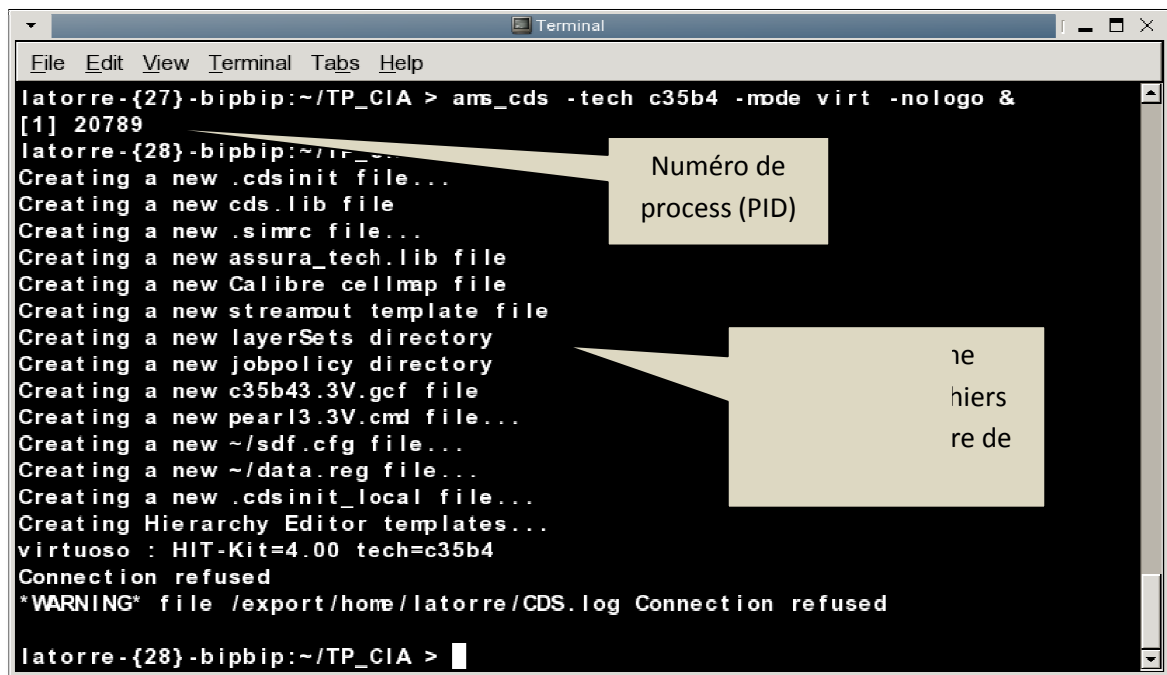
La suite d'outils **Cadence** est constituée d'une multitude de modules permettant d'aborder différents aspects de la conception de circuits microélectronique. **Virtuoso** est un module qui offre tout les outils nécessaires à la conception de circuits « full-custom », depuis la saisie de schéma jusqu'au dessin des masques de fabrication. Le flot de conception de circuits numériques est supporté par d'autres modules.

Le lancement de **Cadence Virtuoso** configuré pour la technologie **AMS c35b4** (0.35µm CMOS, 4 métaux) s'effectue à l'aide de la commande :

```
~/TP_CIA > ams_cds -tech c35b4 -mode virt -nologo &
```

Soyez attentif aux messages qui apparaissent dans le terminal, ils peuvent indiquer d'éventuels problèmes qui empêchent le lancement correct de **Cadence** (Il peut aussi y avoir des avertissements sans importance...)

Lors du premier lancement, **Cadence** prépare le répertoire courant avec une structure de fichiers dont on peut suivre la création depuis le terminal.



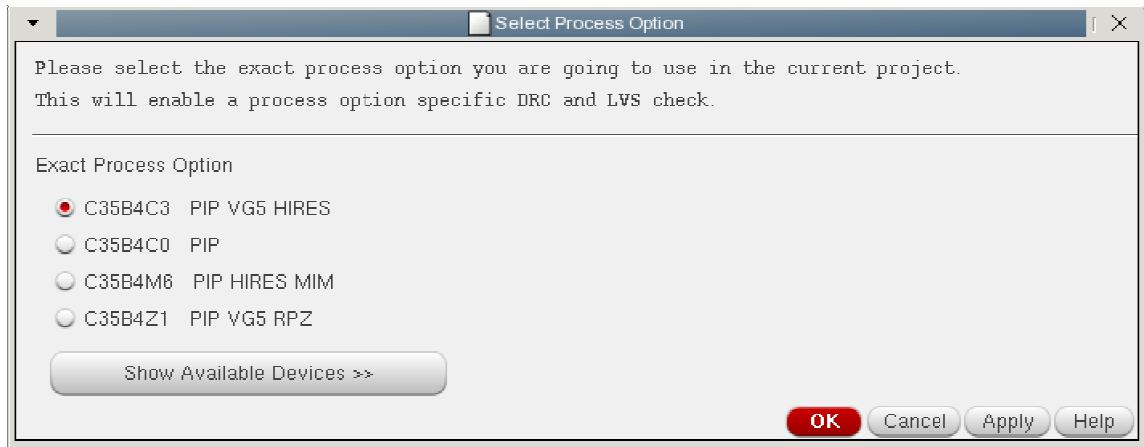
```
latorre-{27}-bipbip:~/TP_CIA > ams_cds -tech c35b4 -mode virt -nologo &
[1] 20789
latorre-{28}-bipbip:~/TP_CIA >
Creating a new .cdsinit file...
Creating a new cds.lib file
Creating a new .simrc file...
Creating a new assura_tech.lib file
Creating a new Calibre cellmap file
Creating a new streamout template file
Creating a new layerSets directory
Creating a new jobpolicy directory
Creating a new c35b43.3V.gcf file
Creating a new pearl3.3V.cmd file...
Creating a new ~/sdf.cfg file...
Creating a new ~/data.reg file...
Creating a new .cdsinit_local file...
Creating Hierarchy Editor templates...
virtuoso : HIT-Kit=4.00 tech=c35b4
Connection refused
*WARNING* file /export/home/latorre/CDS.log Connection refused
latorre-{28}-bipbip:~/TP_CIA >
```

Numéro de process (PID)

ne  
hiers  
re de

Quelques secondes plus tard (si tout s'est bien passé), l'environnement **Cadence** est lancé se qui se traduit à l'écran par l'apparition de 3 fenêtres :

- Une fenêtre **What's new?**
- Une fenêtre **Virtuoso® 6.1.5**
- Une fenêtre **Select Process Option** qu'il faut renseigner avant que toute autre action soit possible :



Sélectionner le procédé de fabrication **C35B4C3** puis faire **OK**.

Une nouvelle fenêtre **What's New in IC6.1.500 Overview** apparaît. Fermer cette fenêtre en faisant :

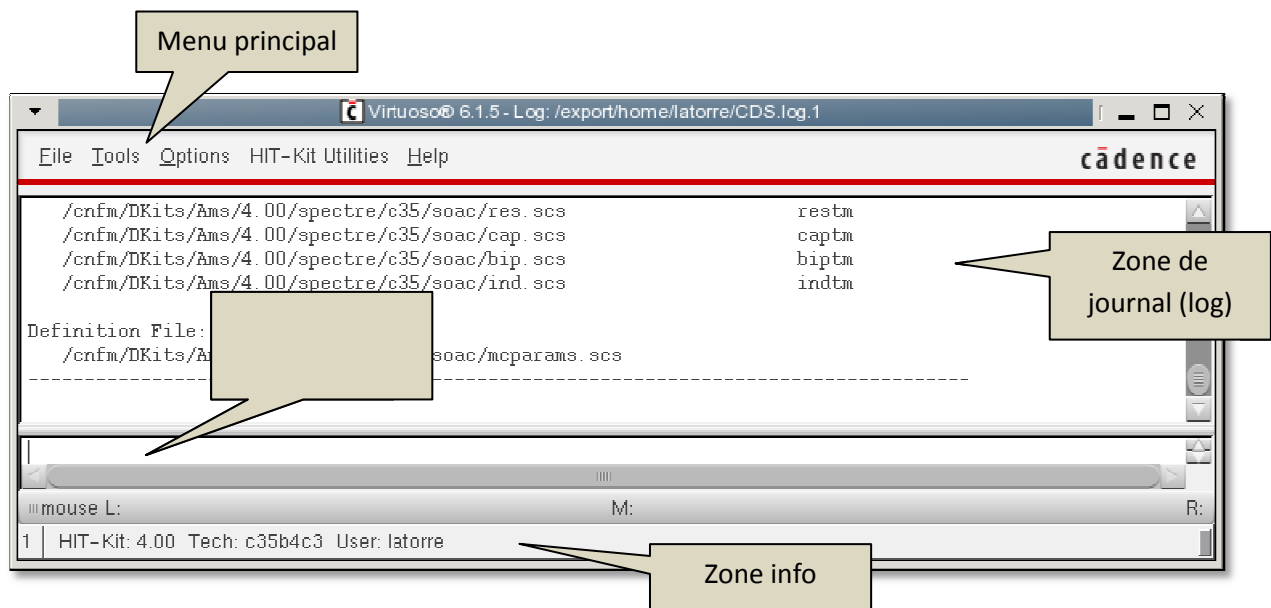
**File > Close and Do Not Show Again**

Fermer également la fenêtre **What's new?** :

**File > Off at Startup**

**File > Close**

La fenêtre restante (**Virtuoso® 6.1.5**) est la fenêtre principale de l'environnement **Virtuoso**. Son menu permet l'accès à tous les outils et options. Elle affiche un journal (log) qui permet de suivre l'exécution des différentes commandes et d'identifier l'origine des problèmes lors que ceux-ci se présentent. Il faut en permanence garder cette fenêtre ouverte en bonne place sur le bureau et constamment porter attention aux messages pouvant apparaître dans la zone de journal. Pour les utilisateurs avancés, il est également possible de saisir directement des commandes en ligne depuis un champ dédié à cet usage.

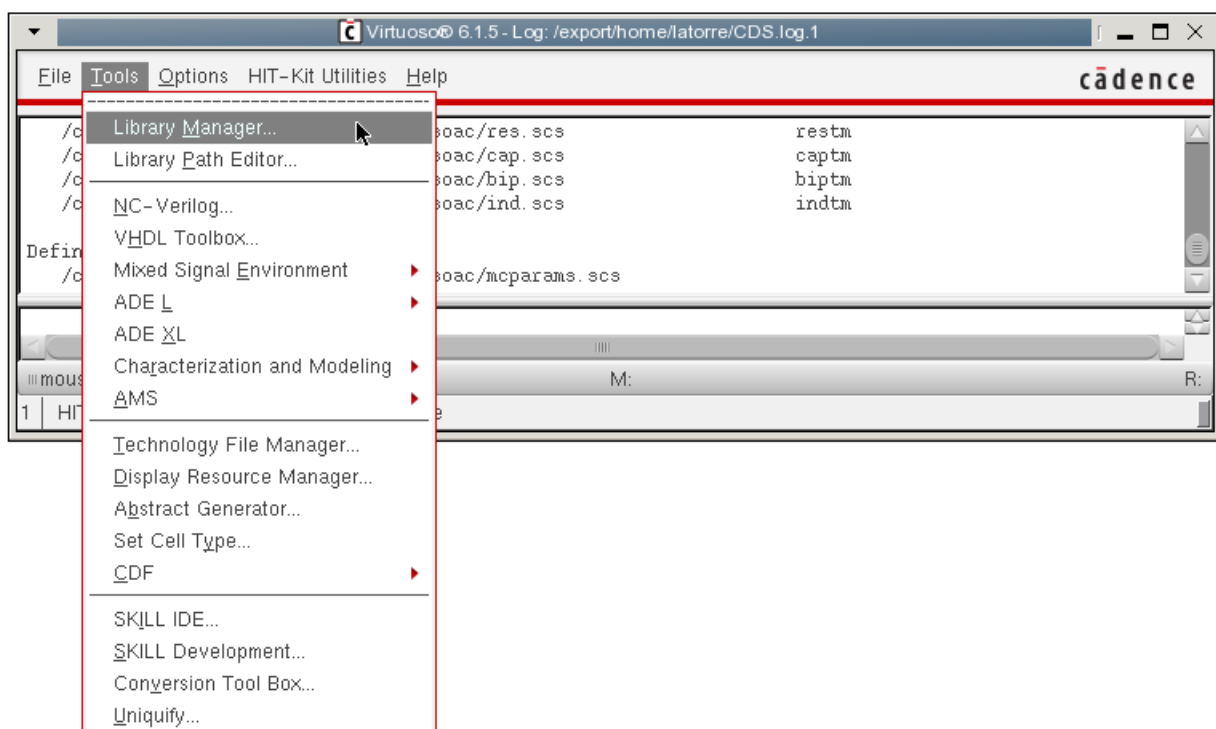


Utiliser l'ascenseur vertical pour prendre connaissance des messages inscrits au journal, relatifs au lancement de **Virtuoso**. L'ensemble des fichiers de configuration et bibliothèques permettant de travailler sur une technologie cible dans **Cadence** est appelé **Design Kit (DK)**. Ici, nous utilisons le Design Kit du fondeur **AMS** baptisé **HIT-Kit v.4.00**.

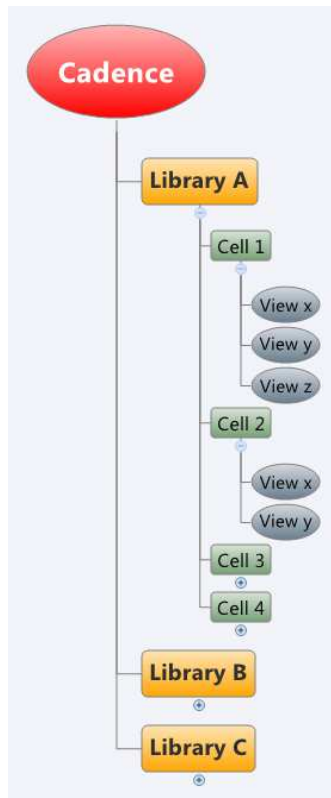
### 3. Aperçu du Library Manager

Depuis le menu principal faire :

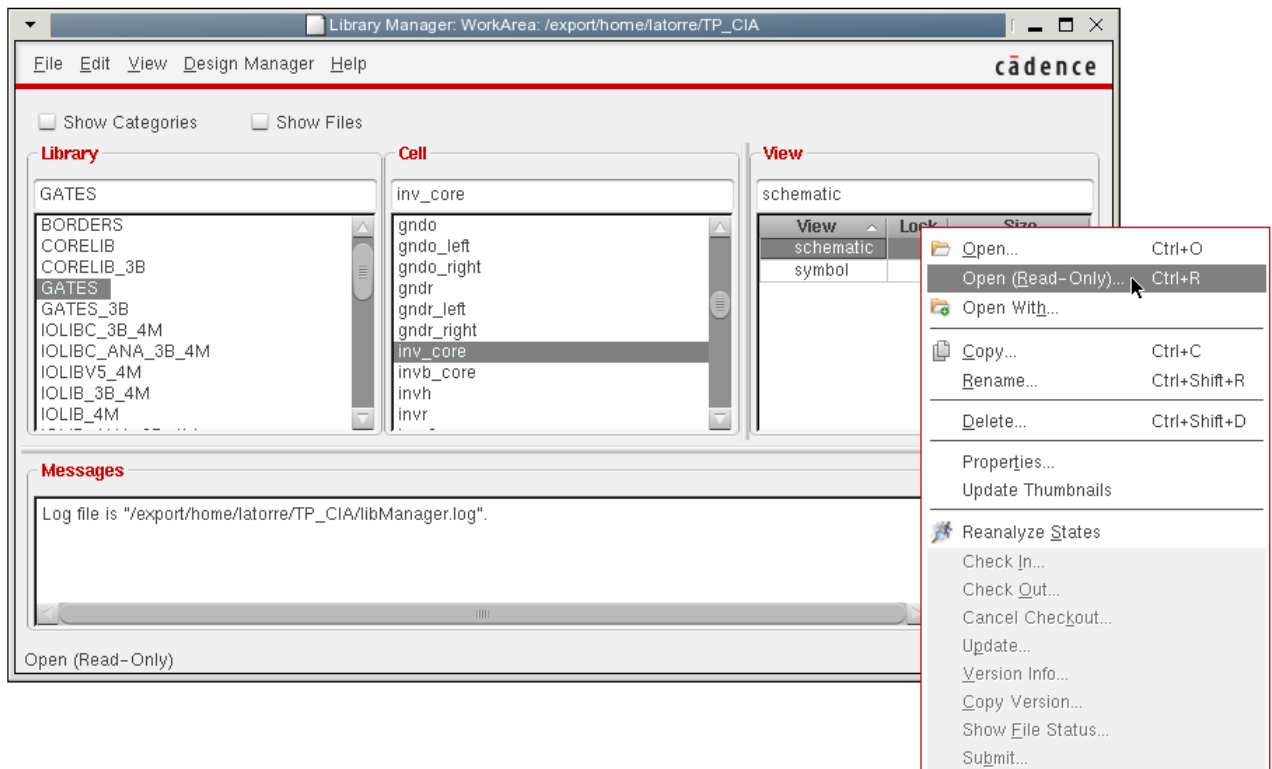
**Tools > Library Manager**




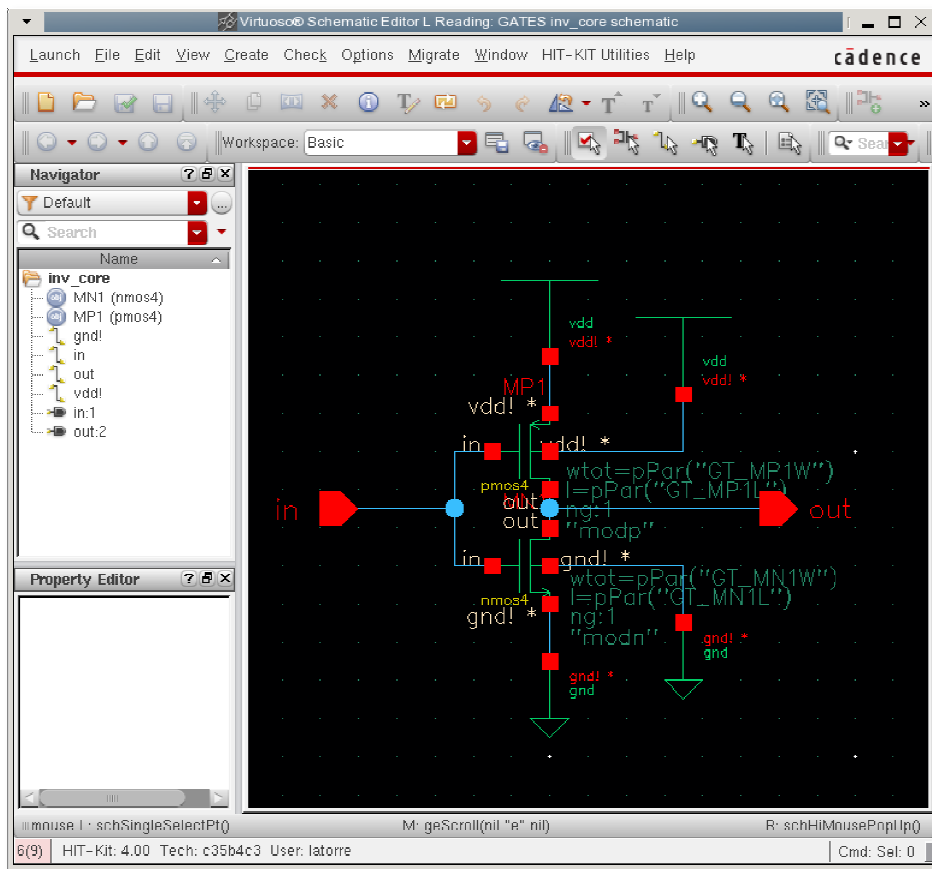
Le **Library Manager** s'ouvre. C'est un navigateur à partir duquel la plupart des opérations sur les bibliothèques, cellules et vues sont possibles. Les relations hiérarchiques entre bibliothèques, cellules et vues sont illustrées sur la figure suivante :



A titre d'exemple, ouvrir (en lecture seule) la vue *schematic* de la cellule *inv\_core* de la librairie *GATES*. Pour cela sélectionner dans le *Library Manager* à tour de rôle la librairie, puis la cellule, puis effectuer un **clik-droit** sur la vue *schematic* et faire **Open (Read Only)** :



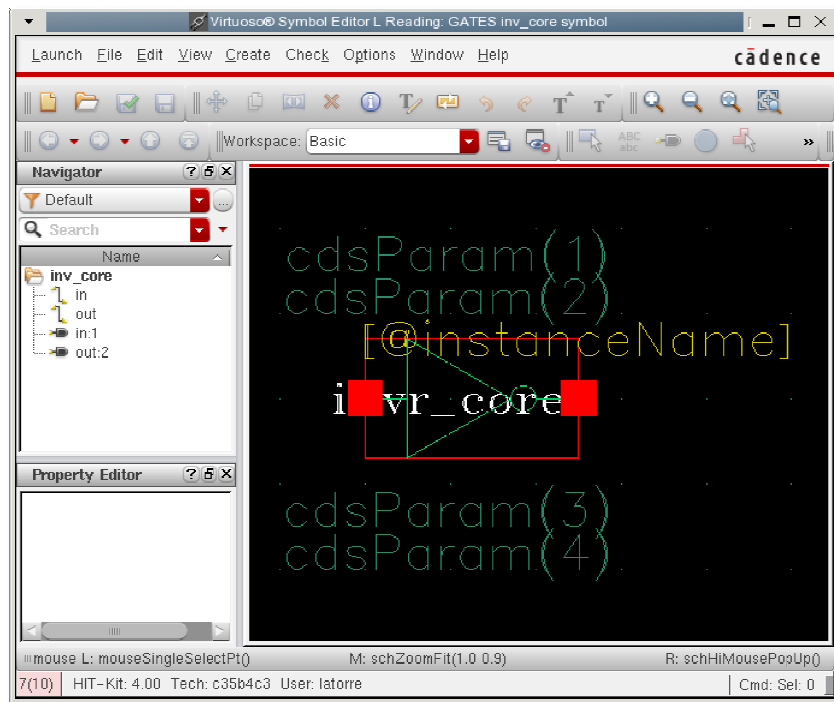
L'éditeur de schémas s'ouvre, et en effectuant quelques zooms (  ), on reconnaît le schéma d'un inverseur logique CMOS :



Fermer l'éditeur de schéma : **File > Close**

Ouvrir de la même façon la vue **symbol** de cette même cellule en lecture seule.





Fermer l'éditeur de symbole : **File > Close**

On peut distinguer 3 catégories de bibliothèques :

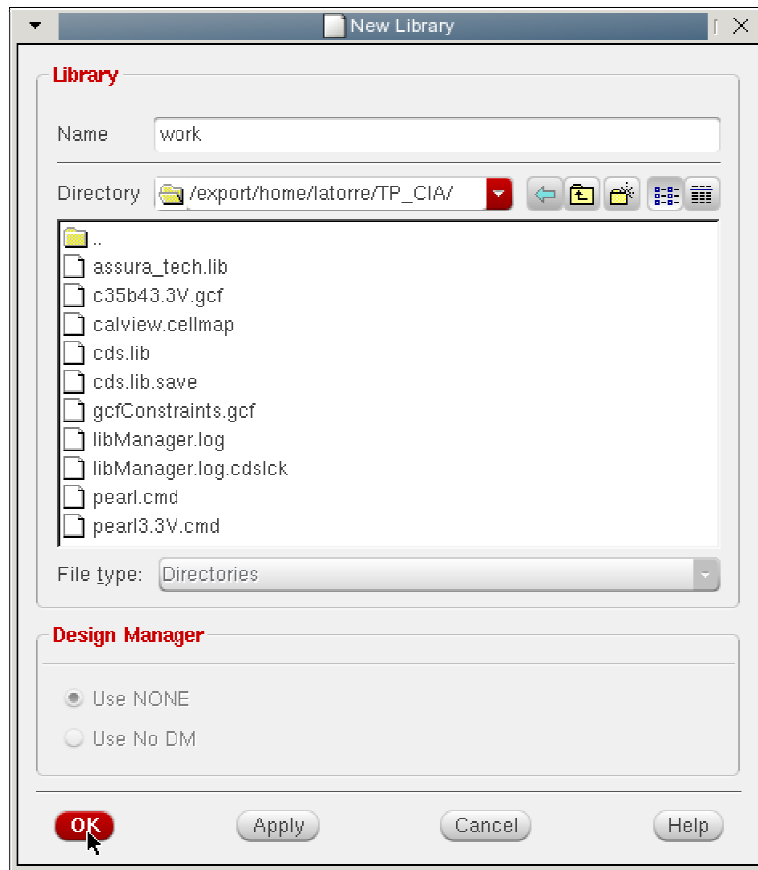
- Les bibliothèques « Fondateur » : ce sont les bibliothèques qui font partie du Design Kit parmi lesquelles on trouve par exemple :
  - o **PRIMLIB** qui contiennent les composants élémentaires (transistors, diodes, résistances, capacités) de la technologie cible (c35b4)
  - o **GATES** qui contient les portes logiques élémentaires
  - o **CORELIB** qui contient toutes les cellules logiques standards utilisées en synthèse automatique de circuits numériques
  - o **A\_CELLS** qui contient des cellules standard analogiques : AOP, Comparateurs, Portes de transmission, DAC, ADC...
  - o **TECH\_C35B4** qui regroupe des informations d'ordre technologique.
- Les bibliothèques « Cadence » : ces bibliothèques proposent des cellules indépendantes de la technologie cible, comme par exemple :
  - o **analogLib** qui rassemble des composants génériques (transistors et composants passifs) ainsi que diverses sources de tensions et courants indispensables à la simulation de circuits.
  - o **ahdLib** et **functional** qui proposent des blocs fonctionnels permettant la modélisation de systèmes de façon comparable à ce qu'il est possible de faire avec Matlab Simulink®.
- Les bibliothèques « utilisateur » : ce sont les bibliothèques de travail, à l'intérieur desquelles on organise ses propres développements de cellules.

## 4. Création d'une librairie de travail

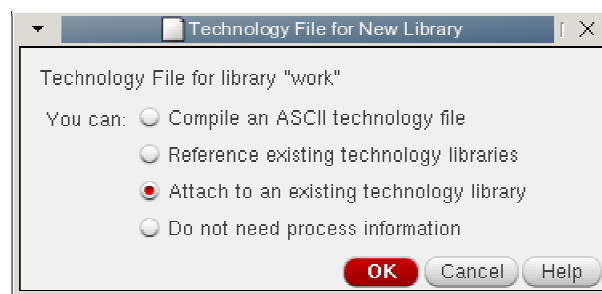
Depuis le **Library Manager**, faire :

**File > New > Library**

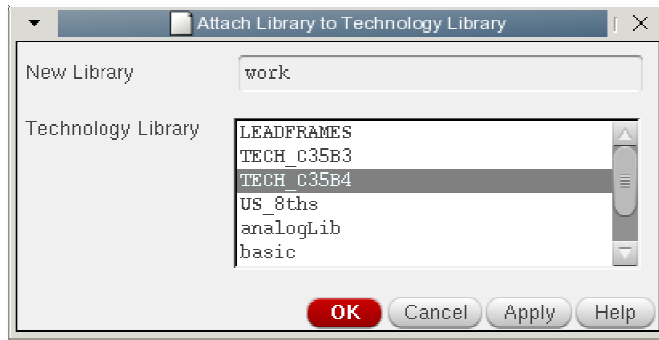
Dans le champ **Name** de la fenêtre **New Library** saisir le nom de la librairie à créer (par exemple **work**), puis faire **OK**.



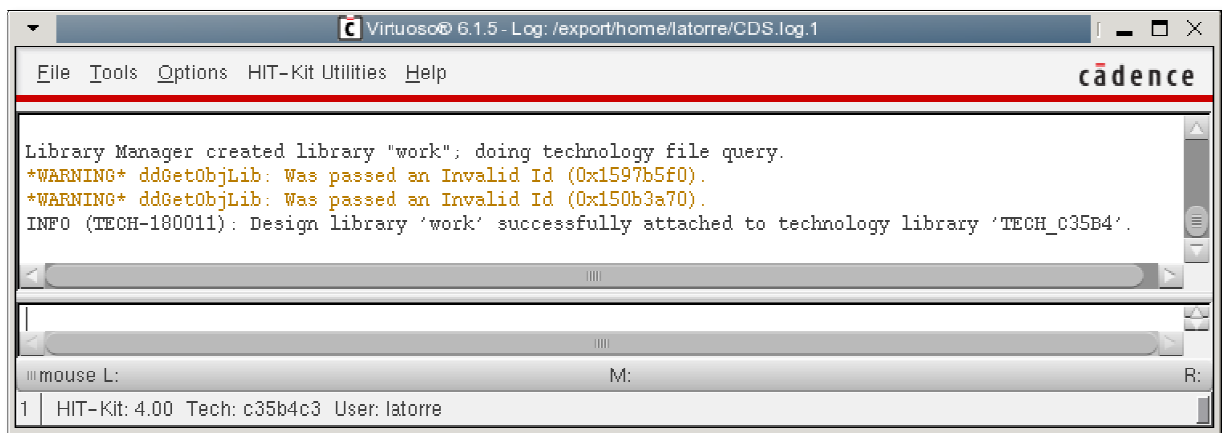
**Cadence** demande alors de quelle façon il peut associer une technologie cible à cette nouvelle librairie. Laisser l'option par défaut **Attach to an existing technology library** et faire **OK**.



La fenêtre suivante demande à quelle librairie technologique il faut attacher la librairie en cours de création. Sélectionner **TECH\_C35B4** et faire **OK**.



Contrôler les messages relatifs à la création de la nouvelle librairie dans le journal (fenêtre **Virtuoso® 6.1.5**) :



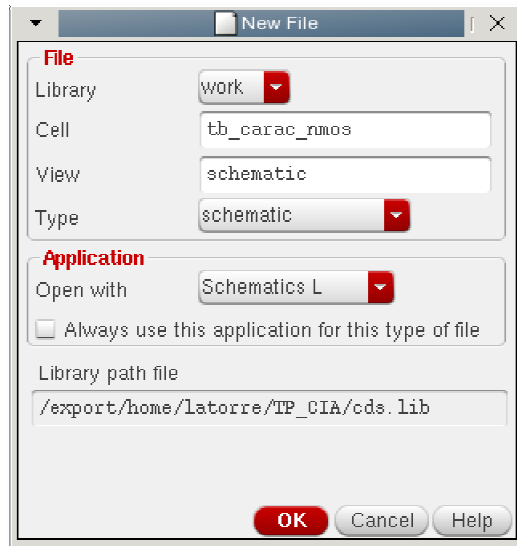
Vérifier que la librairie **work** apparaît maintenant dans le **Library Manager**.

## 5. Edition de schéma

Depuis le **Library Manager**, sélectionner la librairie **work**, puis faire :

**File > New > Cell View**

Dans la fenêtre **New File**, saisir un nom pour la cellule à créer. Nous allons réaliser un schéma de simulation (*test bench*) permettant de caractériser les transistors NMOS de la technologie **AMS**. On peut ainsi nommer la cellule **tb\_carac\_nmos**. Vérifier que le nom de la vue (**View**) est **schematic**, et que le type de vue (**Type**) est également **schematic**. L'application par défaut pour éditer une vue schématique est **Schematic L**. Faire **OK**.



L'éditeur de schéma (fenêtre **Schematic Editor**) s'ouvre. L'édition de schéma est un processus simple qui consiste à :

- Placer des composants sur le schéma et les paramétrer ;
- tirer des fils entre les terminaux de ces composants ;
- éventuellement placer des broches d'E/S, si le schéma représente lui-même un composant qui doit être intégré dans un autre schéma de niveau hiérarchique supérieur. Dans l'exemple qui suit, nous allons réaliser un schéma de simulation (*test bench*) sans broches d'E/S.

### Placement des composants (ou instances)

Depuis la fenêtre **Schematic Editor** faire :

**Create > Instance** (ou , ou la touche **I**)

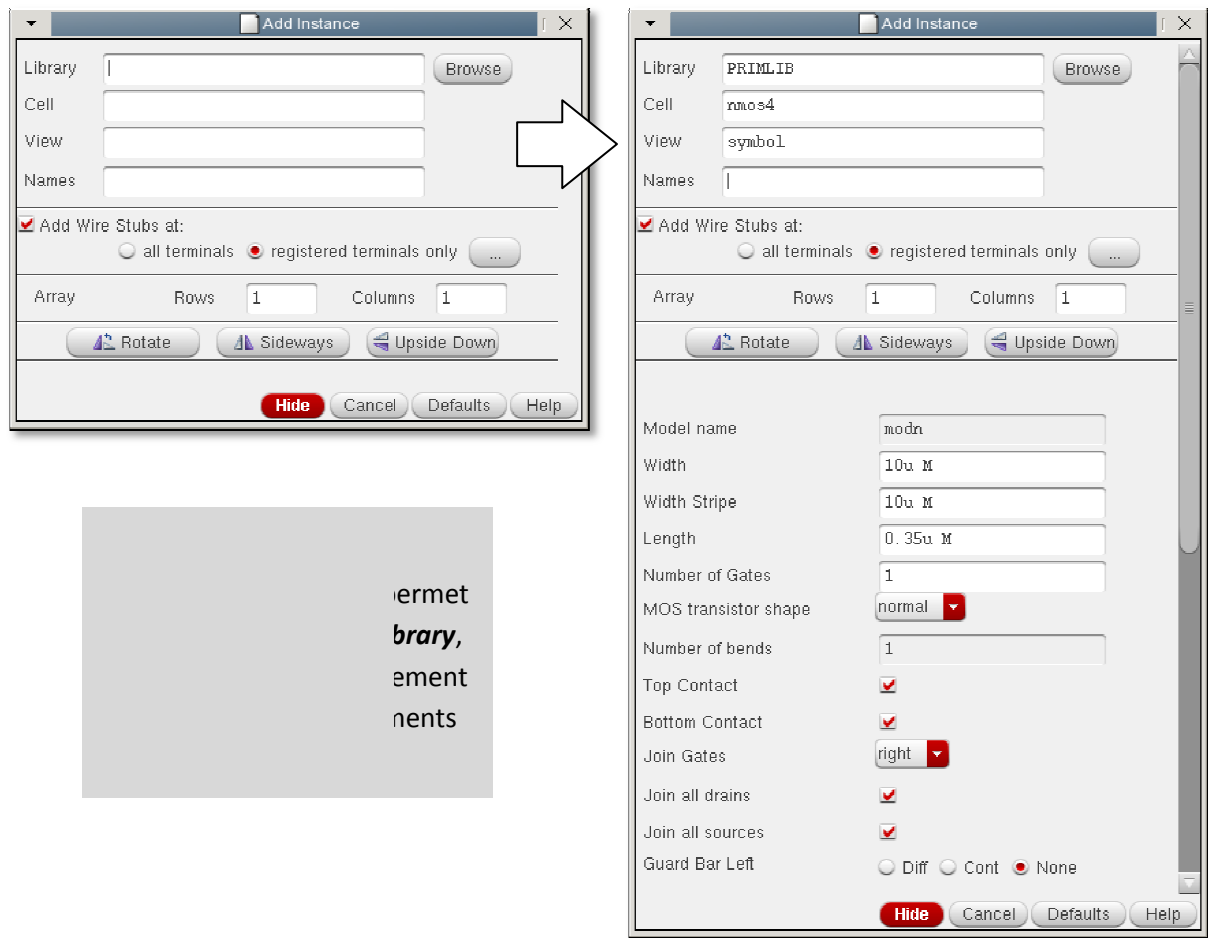
La fenêtre **Add Instance** s'ouvre. Nous allons placer un transistor NMOS. Remplir les champs :

**Library** : PRIMLIB

**Cell** : nmos4

**View** : symbol

A ce stade, l'apparence de la fenêtre change et affiche les paramètres associés à la cellule en cours d'instanciation (nmos4). Il est possible ici de saisir ici les dimensions du transistor (W, L) ainsi que des informations topologiques. Laisser les paramètres et les dimensions par défaut (L=0.35µm, W=10µm).

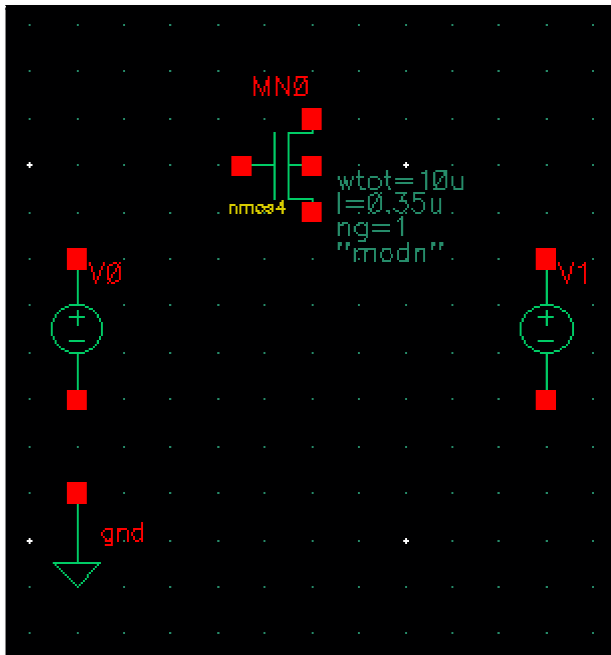


Sans fermer la fenêtre **Add Instance**, faire glisser la souris sur la fenêtre **Schematic Editor**. Un symbole de transistor est attaché au curseur de la souris. Choisir un emplacement dans la fenêtre d'édition (arbitrairement) et cliquer une fois pour poser un transistor. L'éditeur vous invite à poursuivre la pose de transistors. Appuyer sur **ESC** pour terminer la commande.

Renouveler l'opération et poser dans le schéma :

- Deux sources de tension continue (librairie : **analogLib**, cellule : **vdc**, vue : **symbol**)
- La masse (librairie : **analogLib**, cellule : **gnd**, vue : **symbol**)

Arranger la disposition des symboles (**clik & drag**) de façon à peu près comparable à ceci :




) permet de  
 atiquement


**Interconnexions**

Les connexions entre les composants sont représentées par des fils (**wire**) dont les extrémités peuvent être attachées :

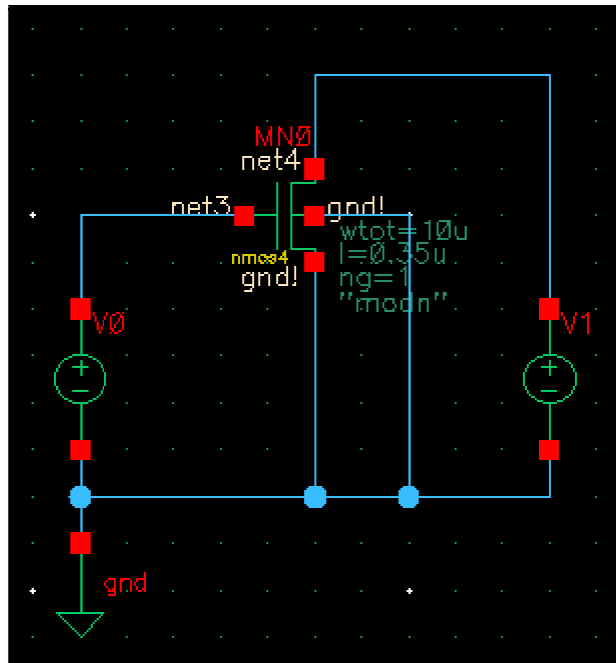
- soit à un terminal de composant (carré rouge)
- soit à une broche d'E/S
- soit à un autre fil


Il existe plusieurs façons d'invoquer l'outil de tracé des interconnexions :

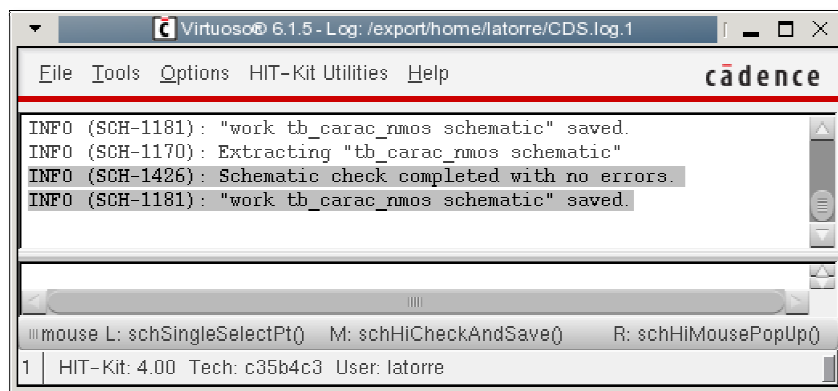
- par un simple **clik & drag** du point de départ (obligatoirement un terminal) vers un point d'arrivée quelconque.
- par le menu **Create > Wire (narrow)** qui vous invite alors à cliquer sur un point de départ (quelconque), puis sur un point d'arrivée (quelconque).
- par le raccourci  qui est équivalent à l'invocation par le menu, ou encore par un appui sur la touche **W**.

Un fil existant peut être sélectionné, supprimé (touche **Suppr** ou ), ou déplacé par simple **clik & drag**.

Réaliser le routage du schéma de la façon suivante :



A ce stade, le schéma est terminé, et nous pouvons demander à l'outil d'en vérifier la cohérence (vérifier par exemple, qu'il ne reste pas de terminal non connecté) en faisant **File > Check and Save** (ou ). Vérifier alors dans le journal le résultat de l'étape de vérification, qui ne doit pas indiquer d'erreurs.



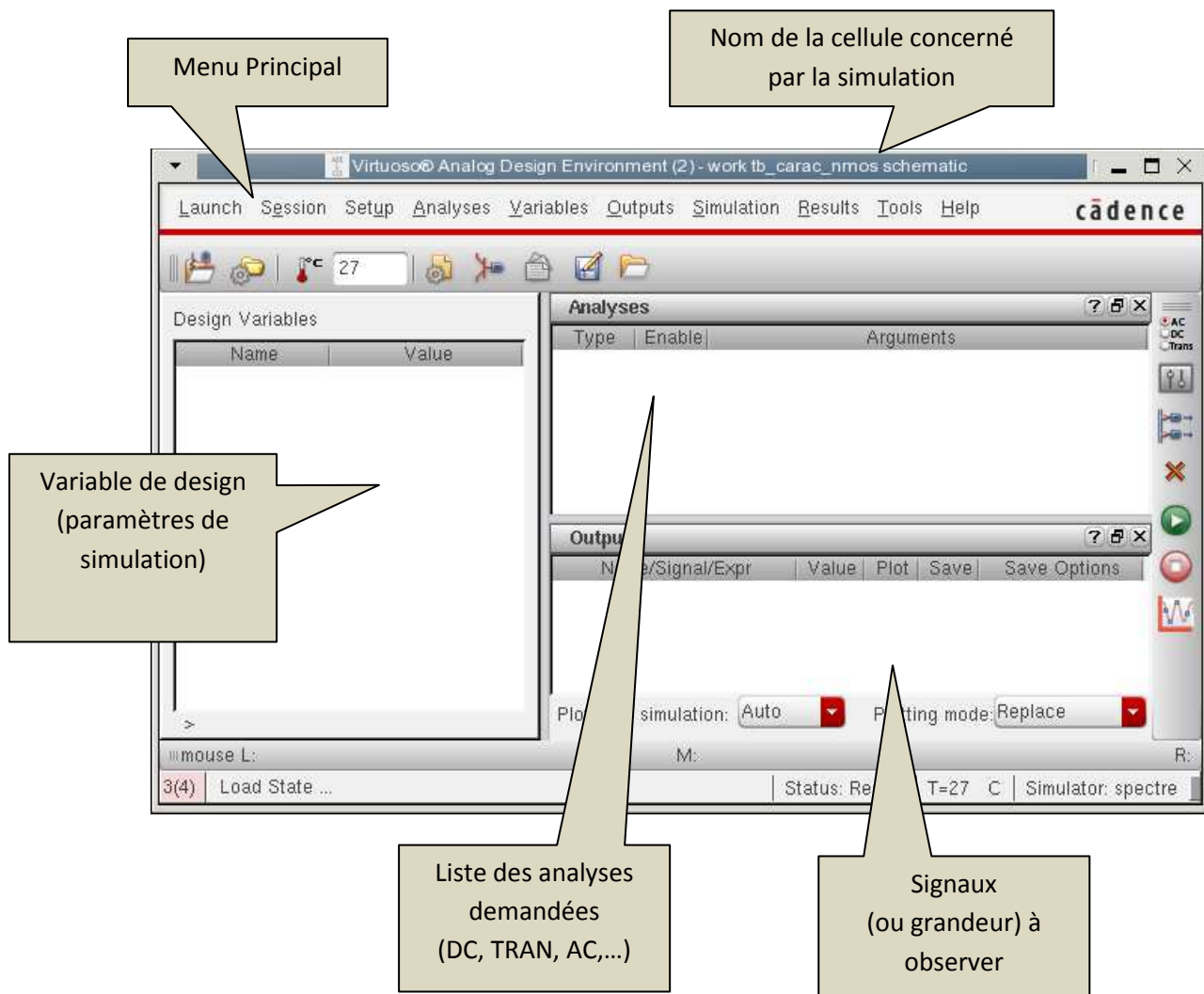
## 6. Simulation

Depuis la fenêtre **Schematic Editor**, avec la cellule **tb\_carac\_nmos** ouverte, faire :

**Launch > ADE L**

La fenêtre **Analog Design Environment** s'ouvre. Il s'agit de la fenêtre de contrôle du simulateur. La fenêtre est divisée en 3 zones :

- La zone **Design Variables**, qui liste les variables utilisées pour paramétrer les composants du schéma
- La zone **Analyses**, qui liste les différentes analyses (au sens SPICE) programmées, comme par exemple un point de polarisation (.OP ou .DC), une analyse transitoire (.TRAN), une analyse en fréquence (.AC), etc...
- La zone **Outputs** qui liste les signaux ou grandeurs scalaires à observer

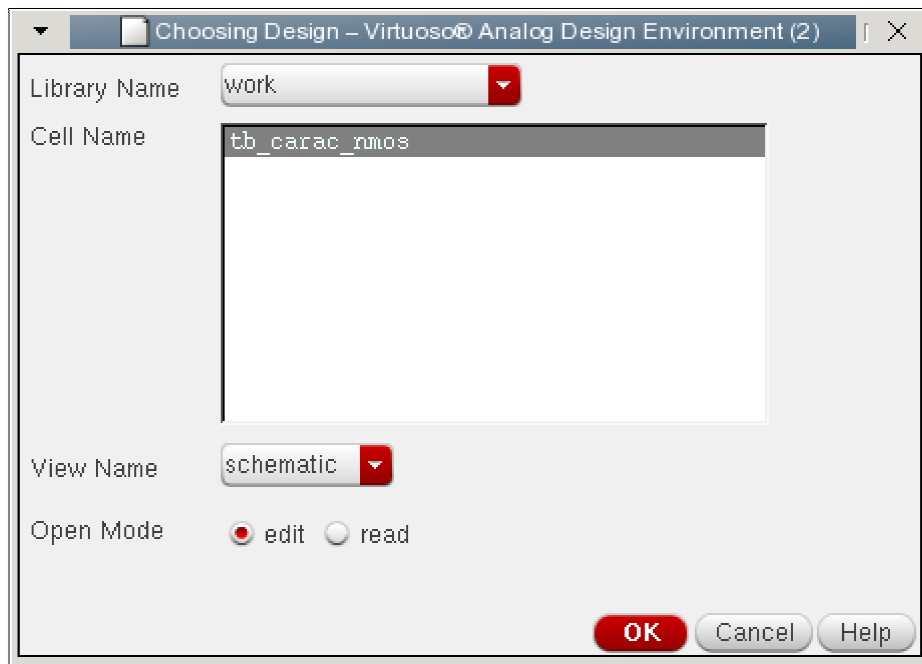




Avant toute simulation, il faut vérifier que l'environnement de simulation est correctement paramétré : Quel est le schéma associé à la simulation, quel simulateur utilise-t-on, et avec quels modèles de composants...

Depuis le menu principal de la fenêtre **Analog Design Environment** faire :

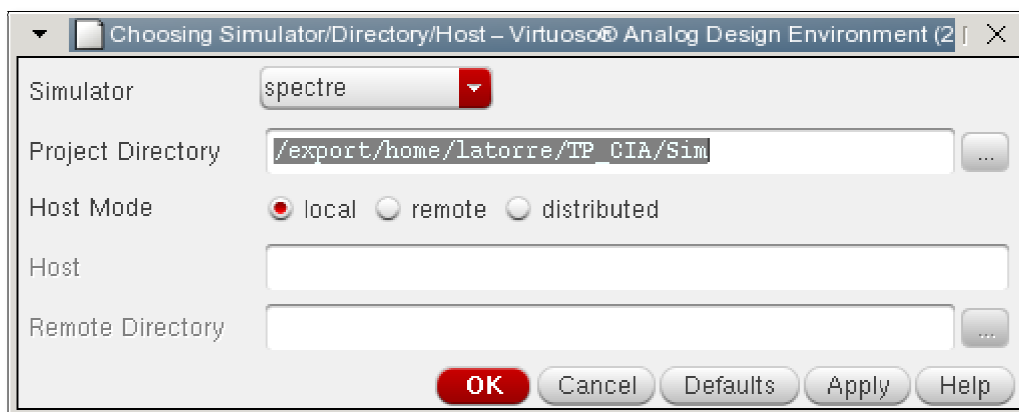
**Setup > Design** (ou )



Dans la fenêtre **Choosing Design**, vérifier que le schéma associé à l'environnement de simulation est bien celui souhaité (**tb\_carac\_nmos**), puis faire **OK**.

Depuis le menu principal de la fenêtre **Analog Design Environment** faire :

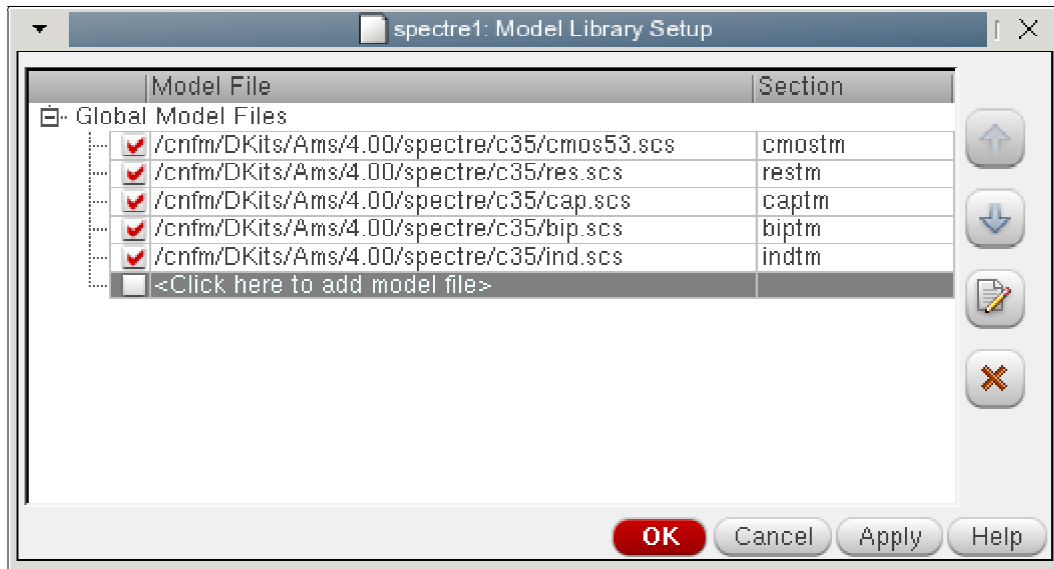
**Setup > Simulator/Directory/Host** (ou )



Dans la fenêtre **Choosing Simulator/Directory/Host**, vérifier que le simulateur est bien **spectre**, et que le répertoire de simulation est bien sous votre répertoire de projet Cadence (~/TP\_CIA). Faire **OK**.

Depuis le menu principal de la fenêtre **Analog Design Environment** faire :

**Setup > Model Libraries** (ou )



Vérifier dans la fenêtre **Model Library Setup** qu'il existe des chemins vers les cartes modèles (fichier.scs) pour chaque type de composant :

- Transistors : **cmos53.scs**
- Résistances : **res.scs**
- Capacités : **cap.scs**
- Bipolaires (diodes et transistors) : **bip.scs**
- Inductances : **ind.scs**

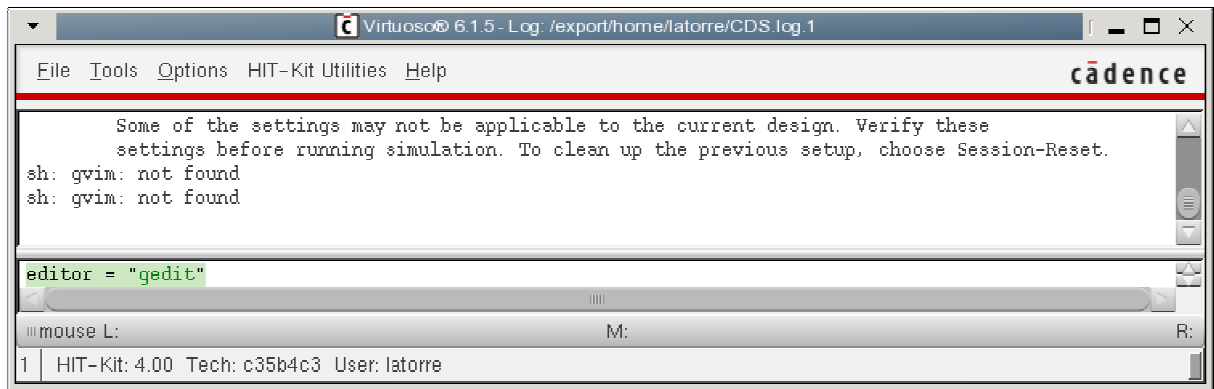
Depuis la fenêtre **Model Library Setup**, il est également possible d'ouvrir les cartes modèles des composants pour en consulter le contenu. Dans la liste, sélectionner la carte modèle des transistors (cmos53.scs), puis cliquer sur le bouton



Si rien ne se passe, consulter le journal dans la fenêtre **Virtuoso® 6.1.5**. Un message est probablement affiché pour signaler que l'éditeur de texte par défaut (**gvim**) n'a pas été trouvé sur la machine depuis laquelle nous exécutons Cadence.

Dans ce cas, nous devons indiquer à **Virtuoso** que nous souhaitons utiliser un autre éditeur de texte. Dans le champ dédié à la saisie de lignes de commande de la fenêtre **Virtuoso® 6.1.5** taper :

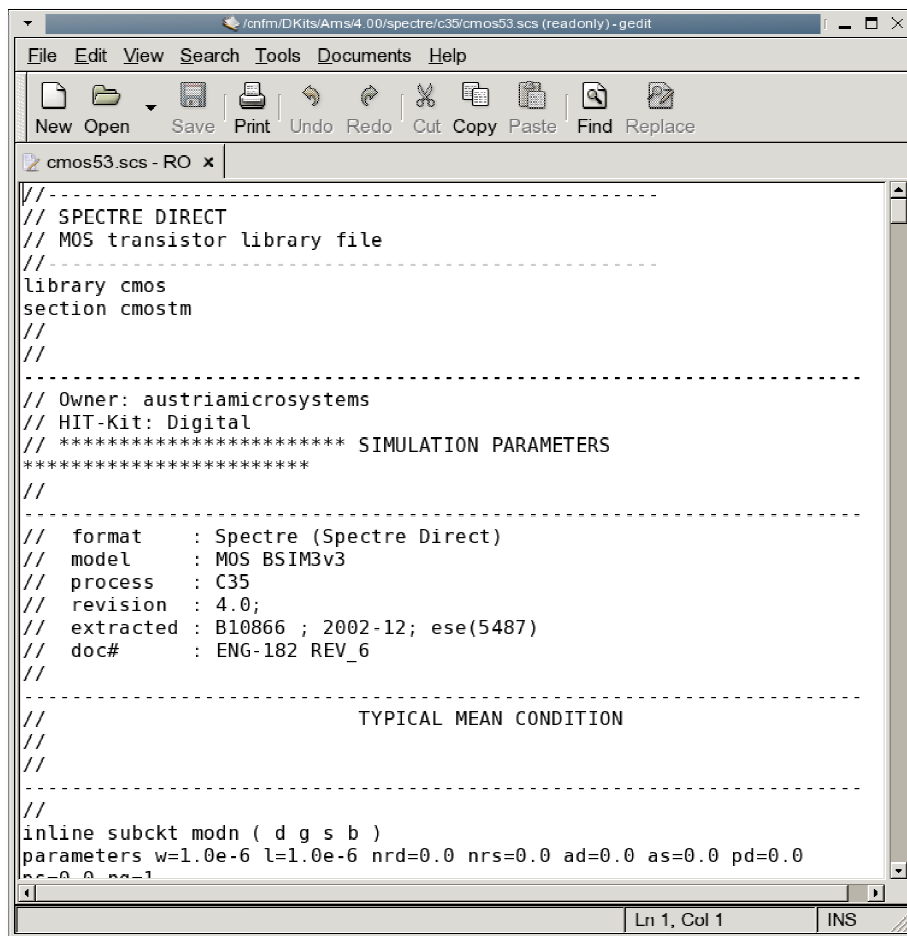
editor = "gedit" (puis Entrée)



Essayer à nouveau d'ouvrir la carte modèle des transistors depuis la fenêtre **Model Library Setup**. Après quelques secondes, l'éditeur **gedit** ouvre le fichier **cmos53.scs**.

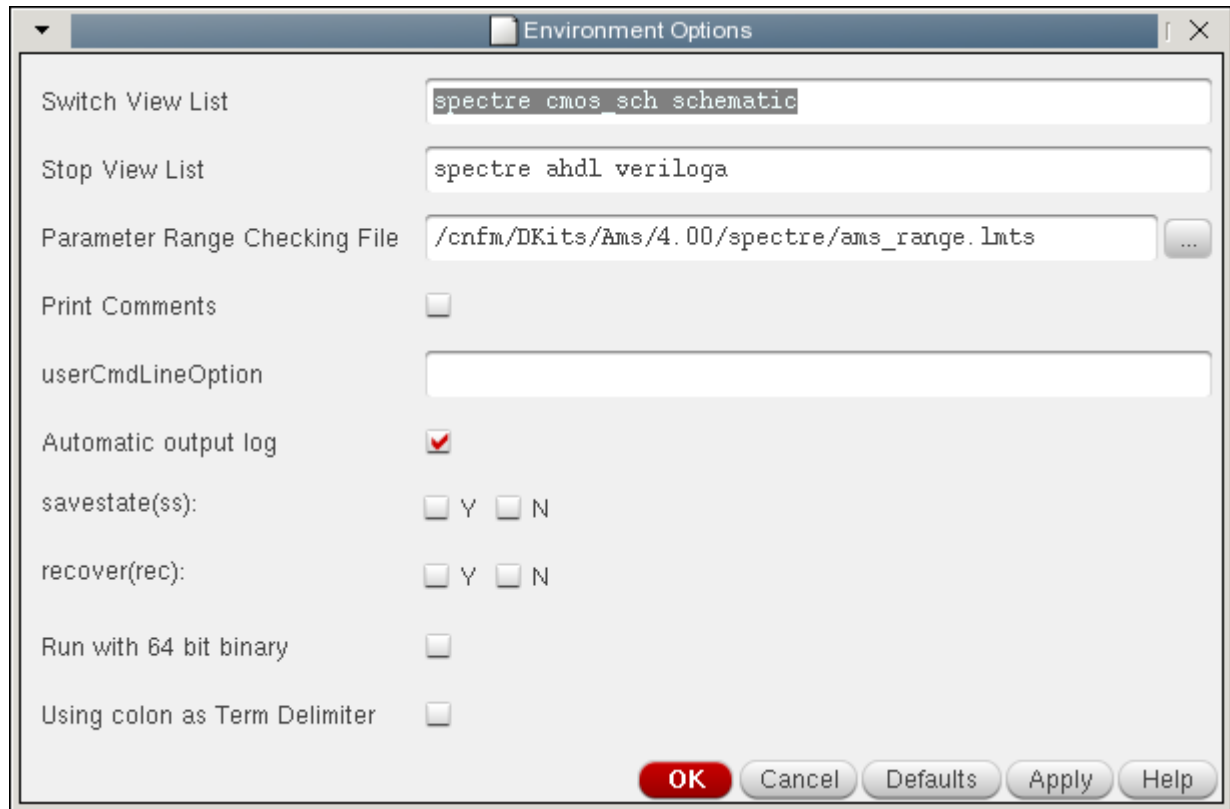
Utiliser des ascenseurs pour prendre connaissance de la structure de la carte modèle, puis quitter **gedit** en faisant **File > Quit**.

Fermer également la fenêtre **Model Library Setup** en faisant **OK**.



Depuis le menu principal de la fenêtre **Analog Design Environment** faire :

**Setup > Environment** (ou  )



Les deux listes **Switch View** et **Stop View** jouent un rôle de première plan au moment où le simulateur transforme un schéma (graphique) en une *netlist* (texte) au format **spectre** (similaire à **spice**). Pour faire court, l'outil explore la hiérarchie du schéma en « descendant » dans les vues de la **Switch View List**, et arrête l'exploration du schéma lorsqu'il trouve pour chaque composant une vue de la **Stop View List**.

Dans la grande majorité des cas qui vont nous intéresser :

- Si une vue **spectre** existe, l'outil de création de *netlist* s'arrête son exploration et considère qu'un modèle de simulation est directement associé à cette cellule.
- Sinon, si une vue **schematic** existe pour une cellule placée dans un schéma, l'outil de création de *netlist* explore cette vue.

A l'aide du **Library Manager**, vérifier qu'il existe des vues **spectre** pour tous les composants que nous avons placé dans le schéma **tb\_carac\_nmos**.

Fermer la fenêtre **Environment Options** en faisant **OK**.

## 6.1. Simulation d'un point de polarisation simple

Il s'agit ici de réaliser la simulation d'un simple point de polarisation sur le schéma [tb\\_carac\\_nmos](#).

A ce stade, le schéma n'est pas complet puisque nous n'avons pas paramétré les sources de tension **V0** ( $V_{GS}$ ) et **V1** ( $V_{DS}$ ).

Depuis la fenêtre d'édition de schéma, sélectionner la source de tension **V0** et saisir la valeur **1** (puis Entrée) dans le champ **DC voltage** du cadre **Property Editor**.


Si le cadre **Property Editor** n'apparaît pas dans la fenêtre, il faut l'invoquer en faisant depuis le menu principal de l'éditeur de schéma :

**Window > Assistants > Property Editor**

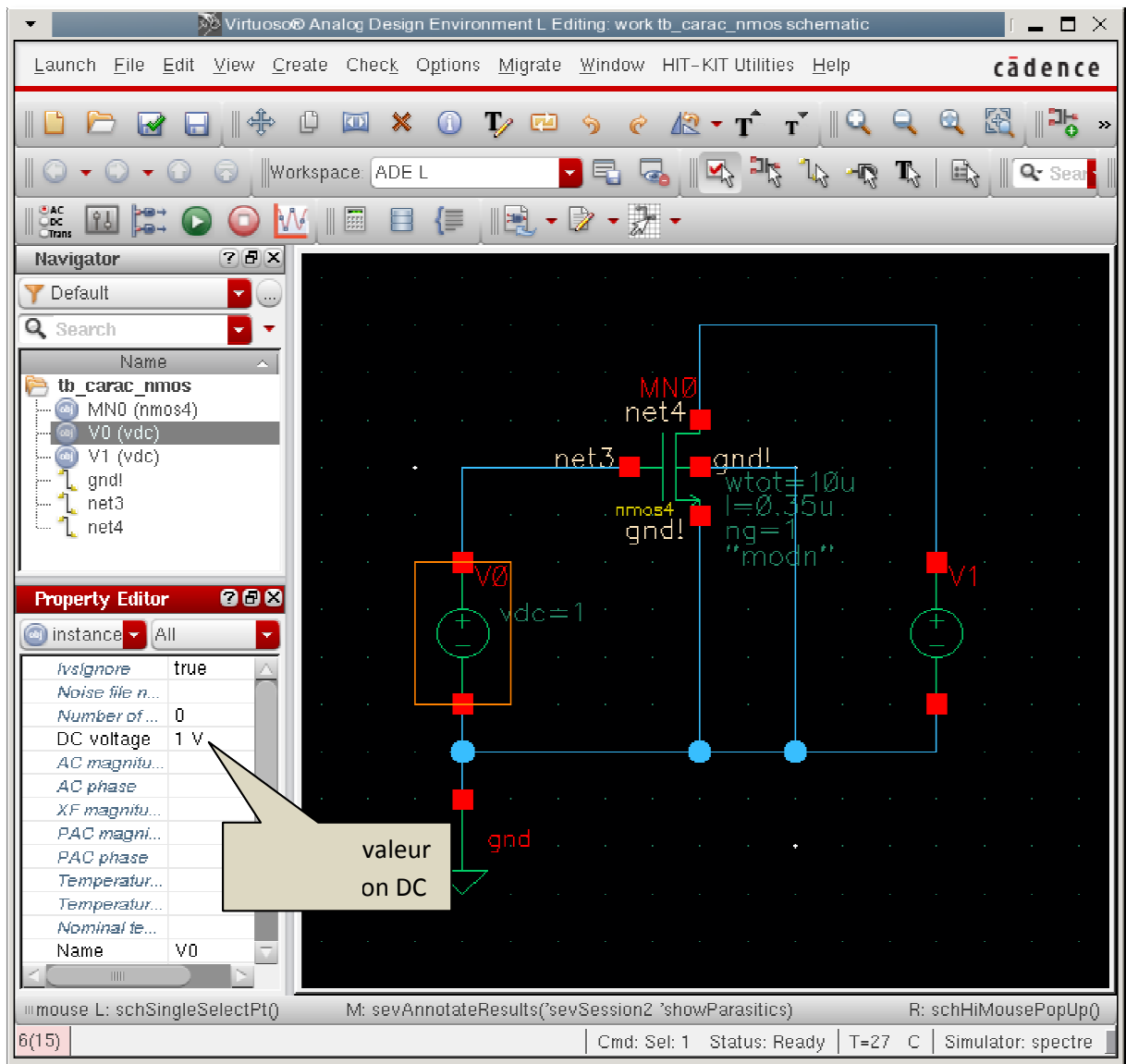
Alternativement, il est possible d'éditer les paramètres des composants en sélectionnant une instance (en cliquant dessus dans le schéma, ou dans le cadre **Navigator**), puis :

**Clic Droit > Properties**

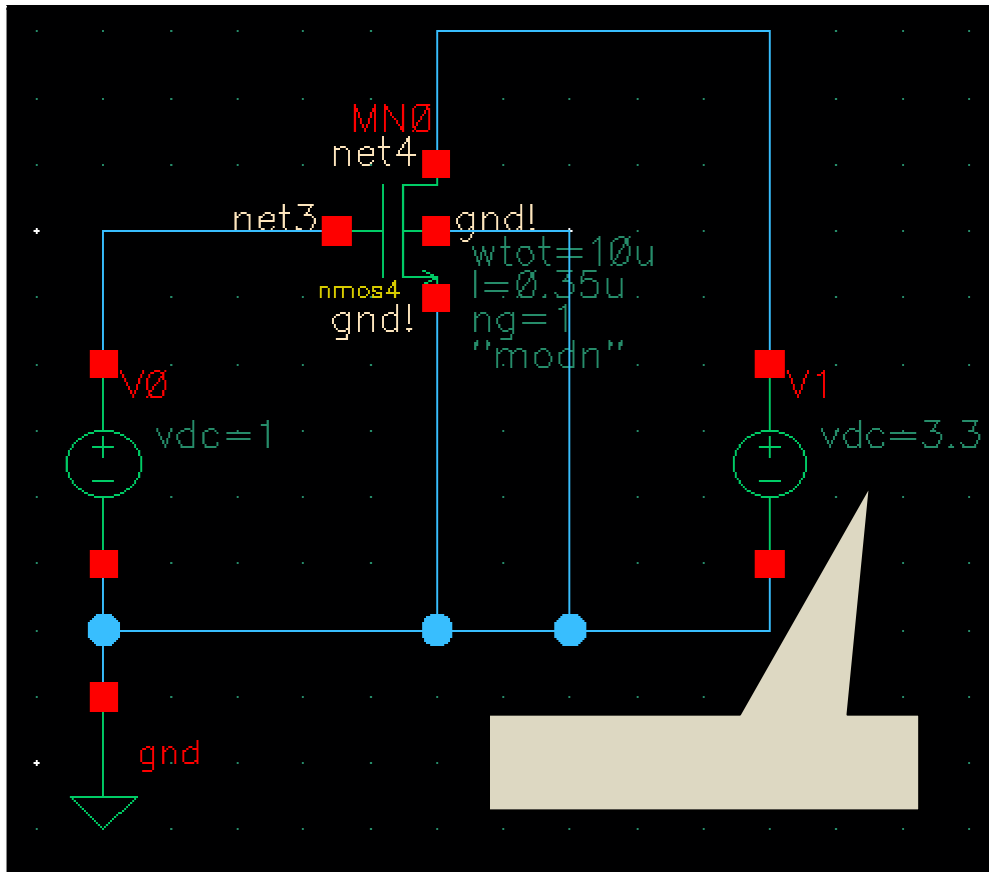
ou encore depuis le menu principal : **Edit > Properties > Objects**

ou encore à l'aide du bouton  , ou enfin par la touche de raccourci **Q**.

Une fenêtre **Edit Object Properties** s'ouvre alors, dans laquelle il est possible de saisir des valeurs pour les différents paramètres, et de les appliquer par le bouton **Apply** ou **OK**.



De la même façon, changer la valeur **DC voltage** de la source **V1** en 3.3V. Le schéma se présente maintenant comme ceci :



Depuis le menu principal de l'éditeur de schéma, faire :

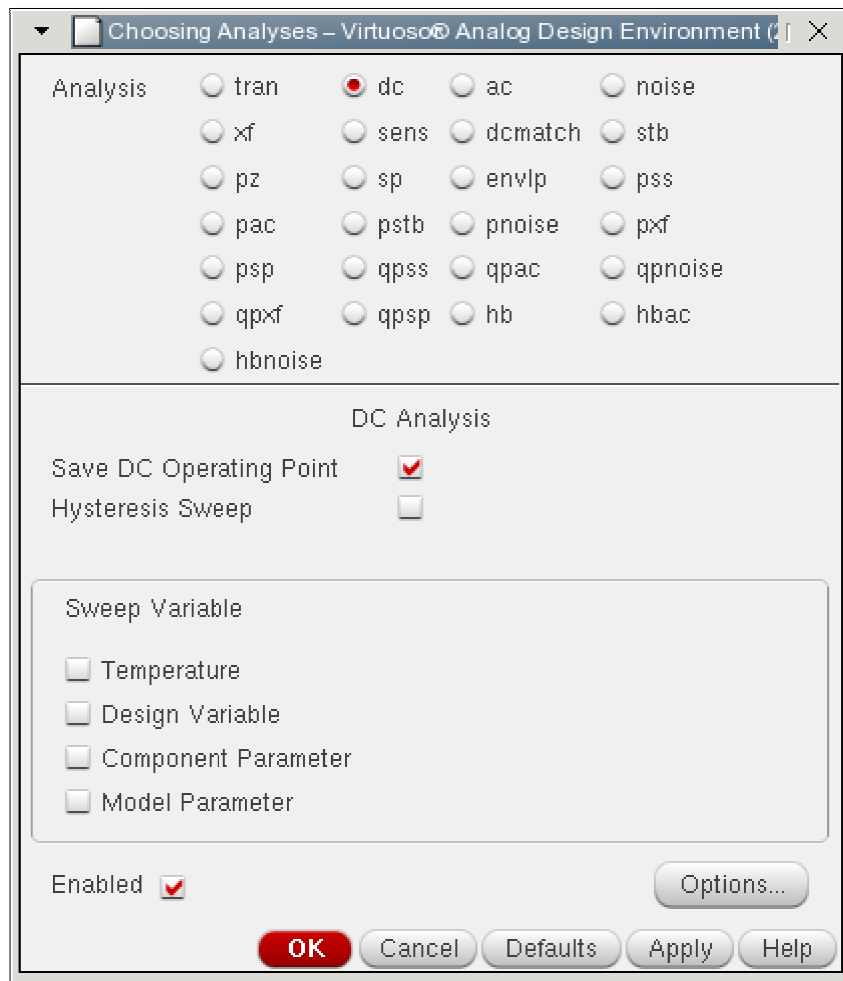
**File > Check and Save** (ou ). Contrôler l'absence d'erreur dans le journal.

Il s'agit maintenant de programmer l'analyse du point de fonctionnement. Revenir à la fenêtre **Analog Design Environment** et faire :

**Analyses > Choose** (ou )

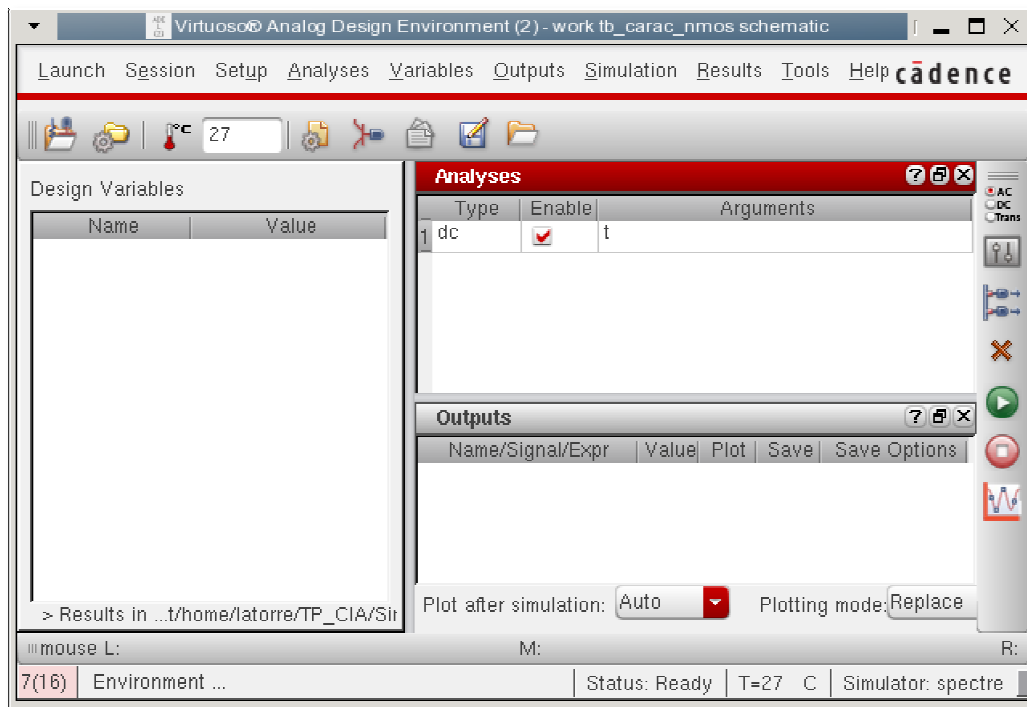
Dans la fenêtre **Choosing Analyses**, sélectionner une analyse **dc** et cocher l'option **Save DC Operating Point**. Cette option indique au simulateur que vous souhaitez enregistrer tous les résultats de simulation concernant le point de polarisation (tensions, courants, paramètres petit-signal des composants, ...)

Vérifier que l'option **Enabled** est cochée, et faire OK.



A ce stade, la fenêtre **Analog Design Environment** doit afficher ceci :





Lancer la simulation en faisant :

**Simulation > Netlist and Run** (ou )

Si la simulation ne démarre pas, consulter le journal principal qui doit certainement indiquer l'origine du problème.

Si tout ce passe bien, le netlist est construite, puis le simulateur est lancés. Ceci donne lieu à l'apparition d'une fenêtre de log. Consulter ce log et vérifier qu'il n'affiche pas de message d'erreur :

```

/export/home/latorra/TP_CIA/Sim/tb_carac_nmos/spectre/schematic/psf/spectre.out
File Help cadence

Time for EDE Visiting: CPU = 653 us, elapsed = 651.4 us.
Time accumulated: CPU = 351.677 ms, elapsed = 1.09255 s.
Peak resident memory used = 93 Mbytes.

Circuit inventory:
    nodes 2
    bsim3v3 1
    vsource 2

Time for parsing: CPU = 9.026 ms, elapsed = 38.5086 ms.
Time accumulated: CPU = 361.48 ms, elapsed = 1.18803 s.
Peak resident memory used = 97 Mbytes.

*****
DC Analysis 'dcOp'
*****
Important parameter values:
    reltol = 100e-06
    abstol(V) = 1 uV
    abstol(I) = 1 pA
    temp = 27 C
    tnom = 27 C
    tempeffects = all
    gmin = 1 pS
Convergence achieved in 2 iterations.
Total time required for dc analysis 'dcOp': CPU = 2.631 ms, elapsed = 5.0674 ms.
Time accumulated: CPU = 366.171 ms, elapsed = 1.2871 s.
Peak resident memory used = 97 Mbytes.

dcOpInfo: writing operating point information to rawfile.
designParamVals: writing netlist parameters to rawfile.
primitives: writing primitives to rawfile.
subckts: writing subcircuits to rawfile.

Aggregate audit (2:57:37 PM, Wed Sep 21, 2011):
Time used: CPU = 392 ms, elapsed = 1.51 s, util. = 25.9%.
Time spent in licensing: elapsed = 101 ms, percentage of total = 6.69%.
Peak memory used = 97 Mbytes.
Simulation started at: 2:57:35 PM, Wed Sep 21, 2011, ended at: 2:57:37 PM, Wed S
spectre completes with 0 errors, 0 warnings, and 1 notice.

La bonne
nouvelle est ici !
18
```

La simulation est terminée. Il reste à examiner les résultats. Avant toutefois d'afficher les résultats de simulation, faire :

**Simulation > Netlist > Display**

Cette commande ouvre une nouvelle fenêtre contenant la netlist construite en vue de la simulation. Cette netlist est au format **spectre**, qui présente de nombreuses similitudes avec le format **spice**.

```

// Generated for: spectre
// Generated on: Sep 21 14:57:33 2011
// Design library name: work
// Design cell name: tb_carac_nmos
// Design view name: schematic
simulator lang=spectre
global 0
include "/cnfm/DKits/Ams/4.00/spectre/c35/mcparams.scs"
include "/cnfm/DKits/Ams/4.00/spectre/c35/cmos53.scs" section=cmostm
include "/cnfm/DKits/Ams/4.00/spectre/c35/res.scs" section=restm
include "/cnfm/DKits/Ams/4.00/spectre/c35/cap.scs" section=captm
include "/cnfm/DKits/Ams/4.00/spectre/c35/bip.scs" section=biptm
include "/cnfm/DKits/Ams/4.00/spectre/c35/ind.scs" section=indtm

// Library name: work
// Cell name: tb_carac_nmos
// View name: schematic
MNO (net4 net3 0 0) modn w=10u l=0.35u as=1.1e-11 ad=1.1e-11 ps=12.2u \
    pd=12.2u nrd=0.06 nrs=0.06 ng=1
V1 (net4 0) vsource dc=3.3 type=dc
V0 (net3 0) vsource dc=1 type=dc
simulatorOptions options reltol=100e-6 vabstol=1e-6 iabstol=1e-12 temp=
    tnom=27 homotopy=all limit=delta scalem=1.0 scale=1.0 \
    compatible=spice2 gmin=1e-12 rforce=1 maxnotes=5 maxwarns=5 digits=
    cols=80 pivrel=1e-3 sensfile="../psf/sens.output" checklimitdest=ps
dcOp dc write="spectre.dc" maxiters=150 maxsteps=10000 annotate=status
dcOpInfo info what=oppoint where=rawfile
designParamVals info what=parameters where=rawfile
primitives info what=primitives where=rawfile
subckts info what=subckts where=rawfile
saveOptions options save=allpub

```

réation et l'affichage de la  
ivent être obtenus avant  
ilation soit lancée :

[> Netlist > Create](#)

problème de simulation,  
on reflexe de consulter la  
in de vérifier que la  
du schéma est conforme à  
ittendu.

Identifier dans la netlist les 3 composants du schéma ainsi que les paramètres associés, puis refermer la fenêtre.

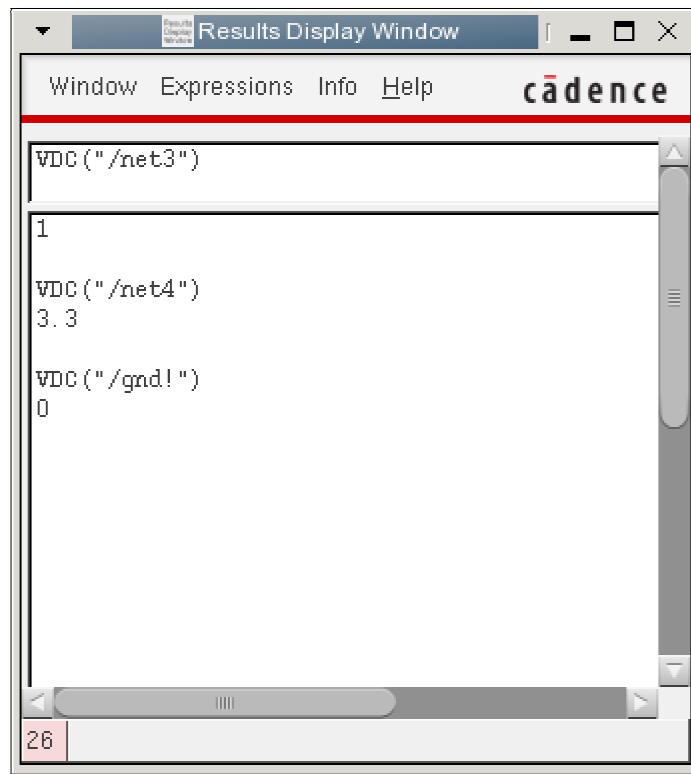
### 6.2. Examen des résultats

Une fois le calcul du point de fonctionnement terminé, aucun résultat ne s'affiche automatiquement car nous avons laissé le cadre **Output** de la fenêtre **Analog Design Environment** vide. Les résultats sont toutefois stockés en mémoire et il existe différents outils pour les consulter.

Depuis la fenêtre **Analog Design Environment** faire :

**Results > Print > DC Node Voltages**

Une fenêtre vide **Result Display Window** monte à l'écran et l'éditeur de schéma invite à sélectionner des nœuds (c.-à-d. des fils). Cliquer sur les différents fils du schéma et observer l'évolution de la fenêtre **Result Display Window**.



Fermer la fenêtre **Result Display Window** en faisant **Window > Close**.

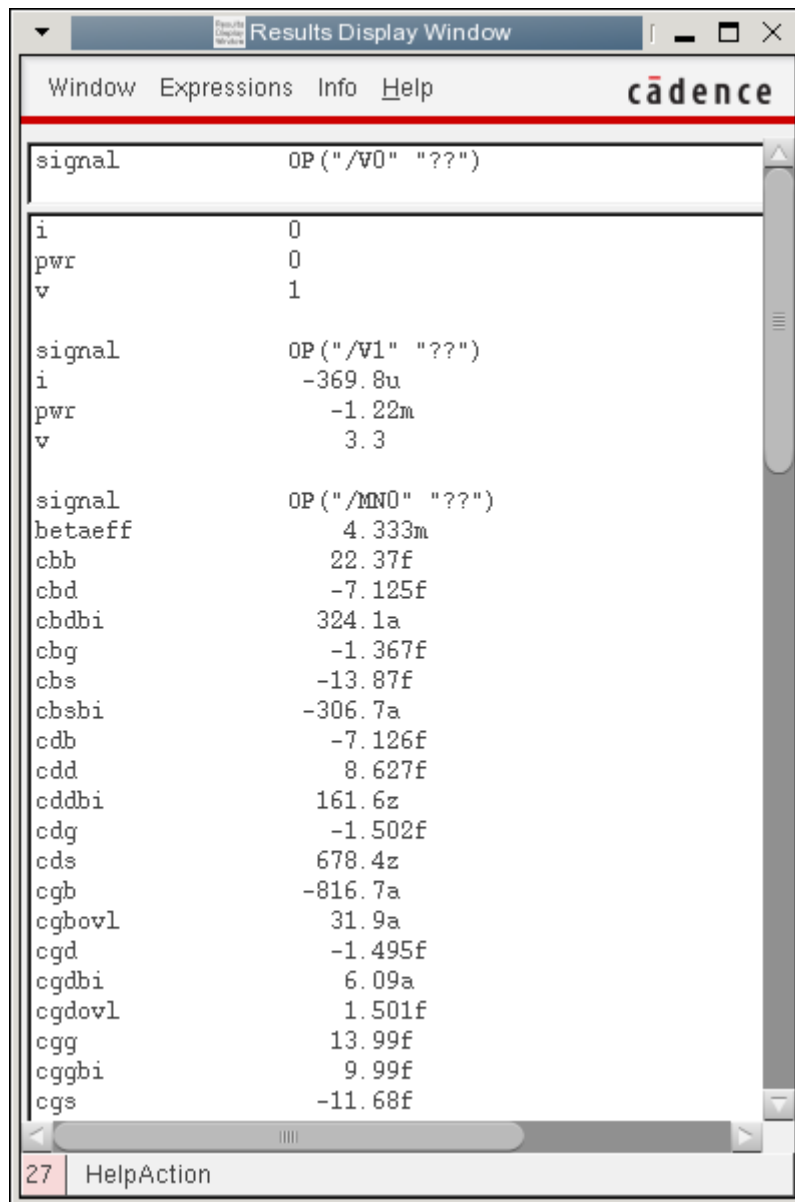
Depuis la fenêtre **Analog Design Environment** faire :

**Results > Print > DC Operating Points**

Une fenêtre vide **Result Display Window** monte à nouveau à l'écran et l'éditeur de schéma invite cette fois à sélectionner des instances (composants). Sélectionner sur le schéma, à tour de rôle, les deux sources de tension **VO** et **V1**, ainsi que le transistor **MNO** et observer l'évolution de la fenêtre **Result Display Window**.

Pour chaque source de tension, le calcul du point de polarisation permet de déterminer la tension, le courant et la puissance délivrés.

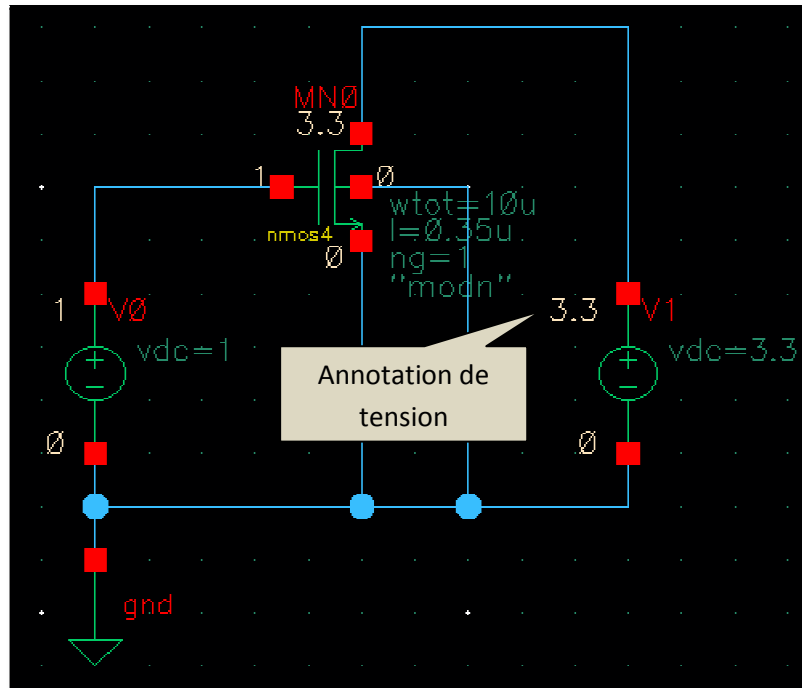
Pour un transistor, cette analyse fournit les coordonnées du point de polarisation (tensions, courants) ainsi que les valeurs calculées pour l'ensemble des paramètres du modèle « petit signal ». Utiliser l'ascenseur vertical pour consulter les paramètres disponibles, puis fermer la fenêtre **Result Display Window**.



Depuis la fenêtre **Analog Design Environment** faire maintenant :

**Results > Annotate > DC Node Voltages**

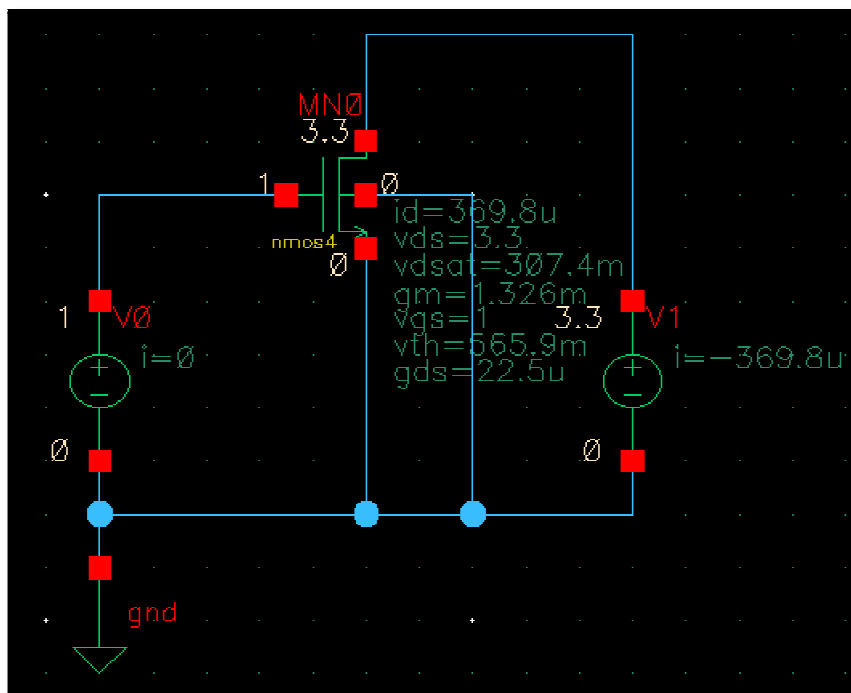
Examiner le schéma, les tensions sur chaque fil apparaissent maintenant à proximité des terminaux.



Depuis la fenêtre **Analog Design Environment** faire maintenant :

**Results > Annotate > DC Operating Point**

Examiner le schéma, les résultats de l'analyse sont affichés à proximité des instances. Pour ne pas charger le schéma toutefois, l'outil n'affiche pas tous les résultats disponibles tels que nous les avons observés auparavant. Il est possible de régler les paramètres que l'on souhaite voir via l'annotation de schéma, mais cela sort du cadre de ce tutorial.



### 6.3. Simulation du point de polarisation avec une variable (sweep)

Dans la simulation précédente, nous avons fixé les tensions  $V_{GS}$  et  $V_{DS}$  autour du transistor. Ainsi, le principal résultat de cette simulation est le calcul du courant  $I_D$ .

Supposons à présent que nous souhaitons, pour  $V_{GS}=1V$ , établir la caractéristique  $I_D=f(V_{DS})$ . Manuellement, nous pourrions éditer le paramètre **DC voltage** de la source **V1** (qui représente  $V_{DS}$ ), et lancer autant de fois le calcul du point de polarisation que nous souhaitons de points pour le tracé.

Heureusement, l'analyse **dc** dispose d'options permettant de faire varier un paramètre du schéma de façon à établir rapidement des caractéristiques statiques.

Depuis la fenêtre **Analog Design Environment** faire un **double-clic** sur la ligne correspondante à l'analyse **dc** déjà programmée (dans le cadre **Analyses**). La fenêtre **Choosing Analyses** s'ouvre avec les paramètres de l'analyse **dc** précédente. Ce qui nous intéresse maintenant se trouve dans le cadre **Sweep Variable**.

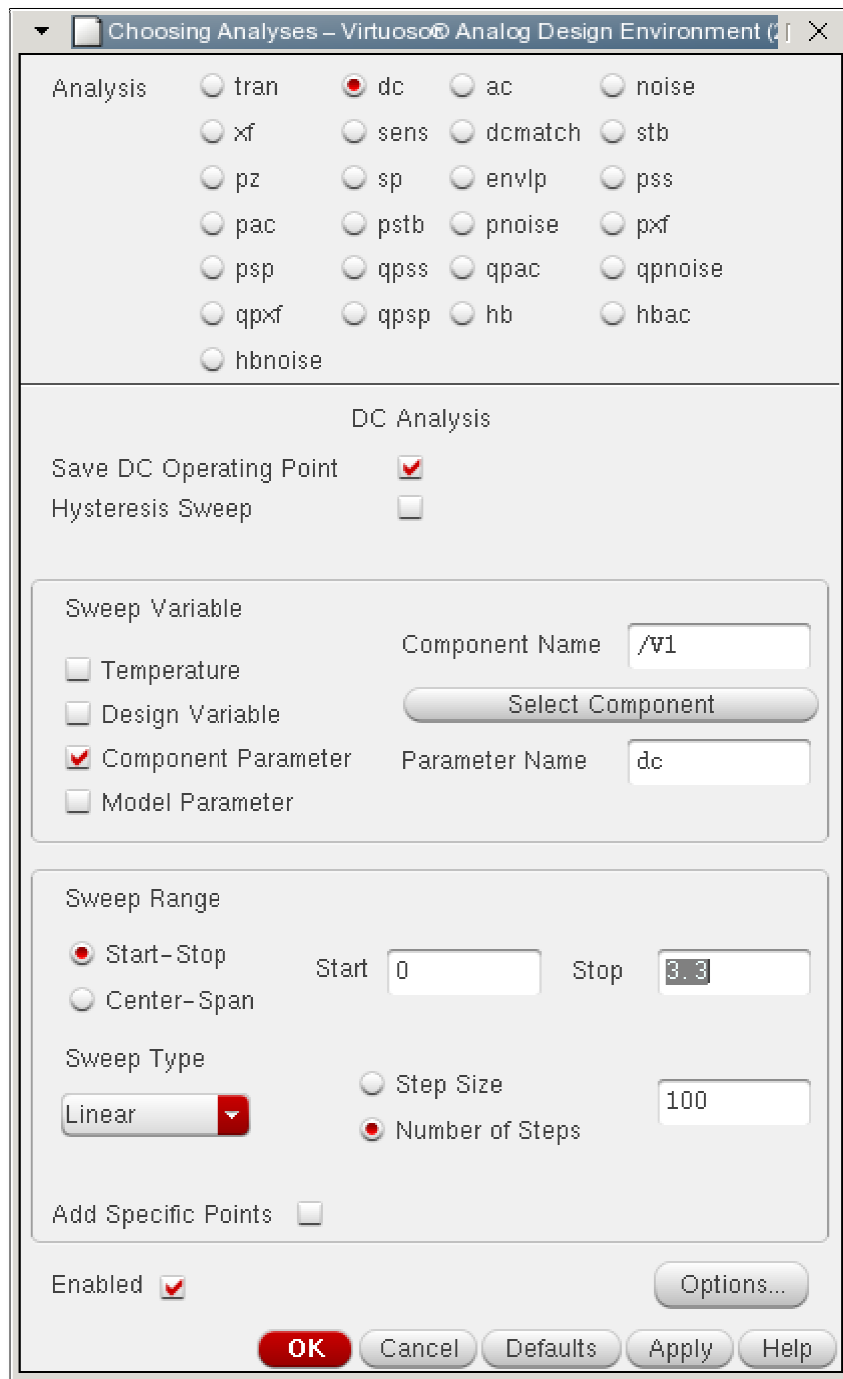
Pour réaliser une variation automatique de la tension  $V_{DS}$ , deux approches sont possibles :

- Nous pourrions éditer dans le schéma les propriétés de la source **V1** de façon à ce que son paramètre **DC voltage** ne soit plus défini par une valeur numérique, mais par un nom de variable. Dans ce cas, dans le cadre **Sweep Variable**, il nous faudrait choisir **Design Variable**.
- Une autre approche, plus directe est de sélectionner **Component Parameter**. C'est cette méthode que nous allons explorer.

Dans le cadre **Sweep Variable**, cocher **Component Parameter**. La fenêtre est alors réarrangée de façon à afficher les options relatives à ce choix. Cliquer sur le bouton **Select Component**. L'éditeur de schéma invite alors à sélectionner une instance. Cliquer sur la source de tension **V1**. Une fenêtre **Select Component Parameter** s'ouvre. Sélectionner la ligne **DC voltage** et faire **OK**. Observer que les champs **Component Name** et **Parameter Name** de la fenêtre de configuration de l'analyse **dc** se sont remplis automatiquement.

Nous souhaitons tracer la caractéristique pour  $V_{DS}$  variant entre 0V et 3.3V. Dans le cadre **Sweep Range**, saisir :

- **Start** : 0
- **Stop** : 3.3
- **Sweep Type** : **Linear**
- **Number of Steps** : 100



Vérifier que l'option **Enabled** est cochée et faire **OK**.

Nous allons maintenant indiquer au simulateur que nous voulons tracer l'évolution du courant  $I_D$ .

Depuis la fenêtre **Analog Design Environment** faire :

**Output > To Be Plotted > Select On Schematic**

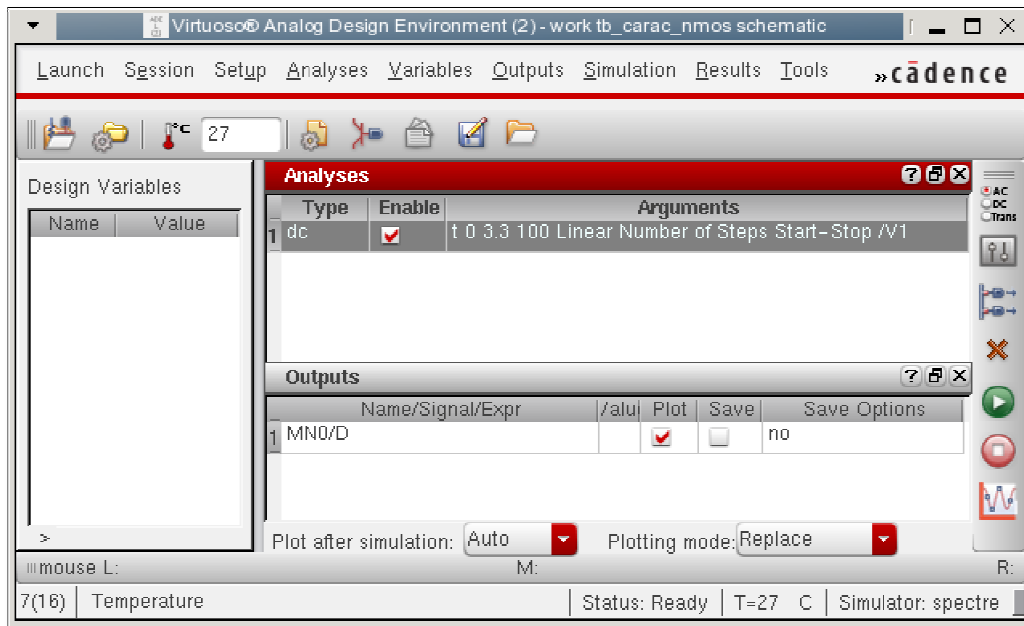
L'éditeur de schéma invite à sélectionner une grandeur à observer. Les fils représentent les tensions, et les terminaux des instances les courants. Cliquer sur le drain (carré rouge) du transistor. Un cercle



de couleur s'affiche alors autour de ce terminal et vous indique que la sélection est effective. Dans le même temps, une ligne **MNO/D** est apparue dans le cadre **Outputs** de la fenêtre **Analog Design Environment**.

On peut poursuivre le processus de sélection si l'on souhaite ajouter d'autres sorties de simulation (i.e. tracer d'autres caractéristiques). Dans notre exemple, terminer la commande en utilisant la touche **ESC** du clavier depuis l'éditeur de schéma.

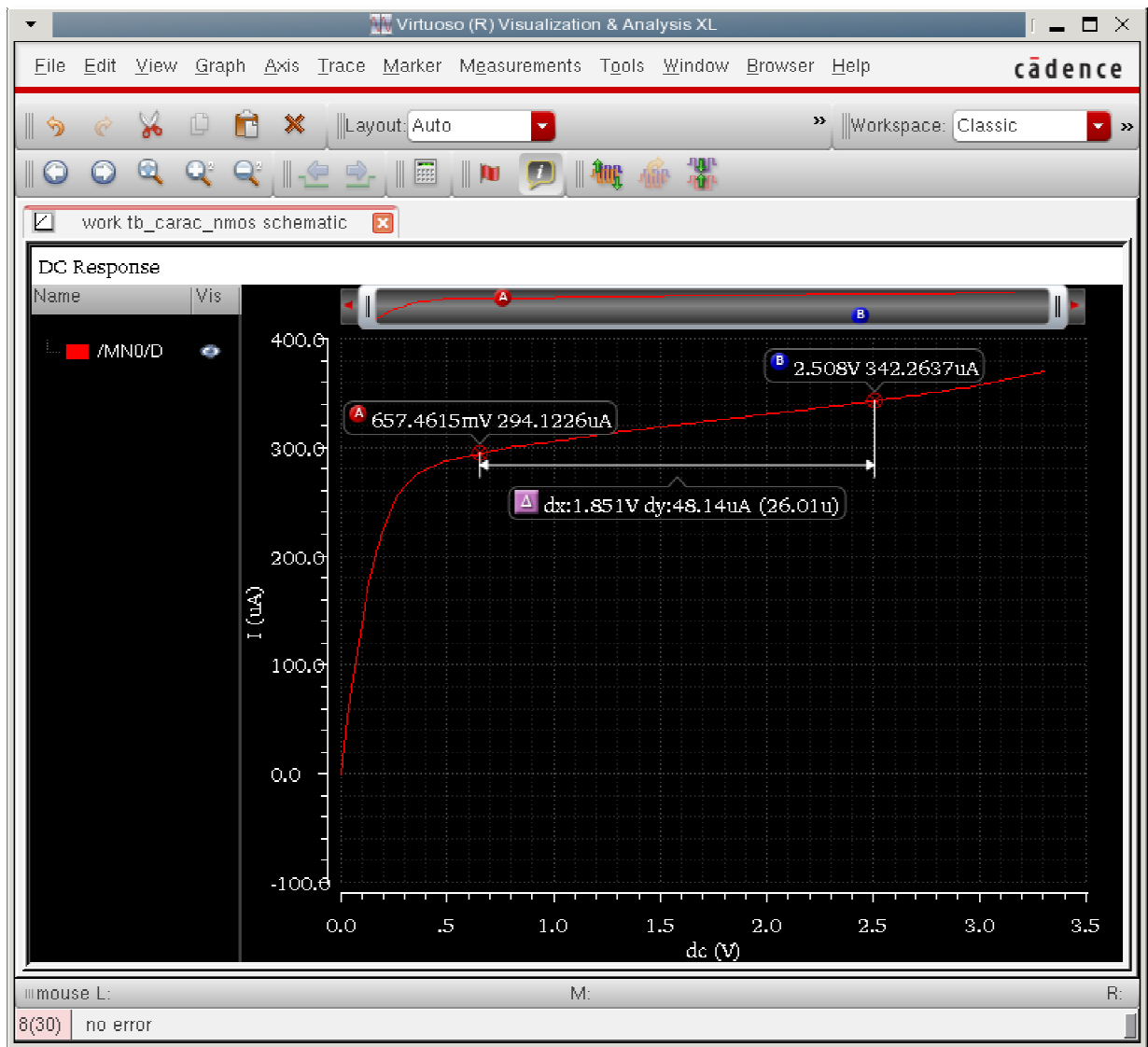
A ce stade, la fenêtre **Analog Design Environment** se présente ainsi :



Lancer la simulation en faisant :

**Simulation > Netlist and Run** (ou )

Observer le contenu du log de simulation. Si tout ce passe bien, quelques secondes plus tard, la fenêtre **Visualisation & Analysis XL** s'ouvre avec la caractéristique souhaitée :



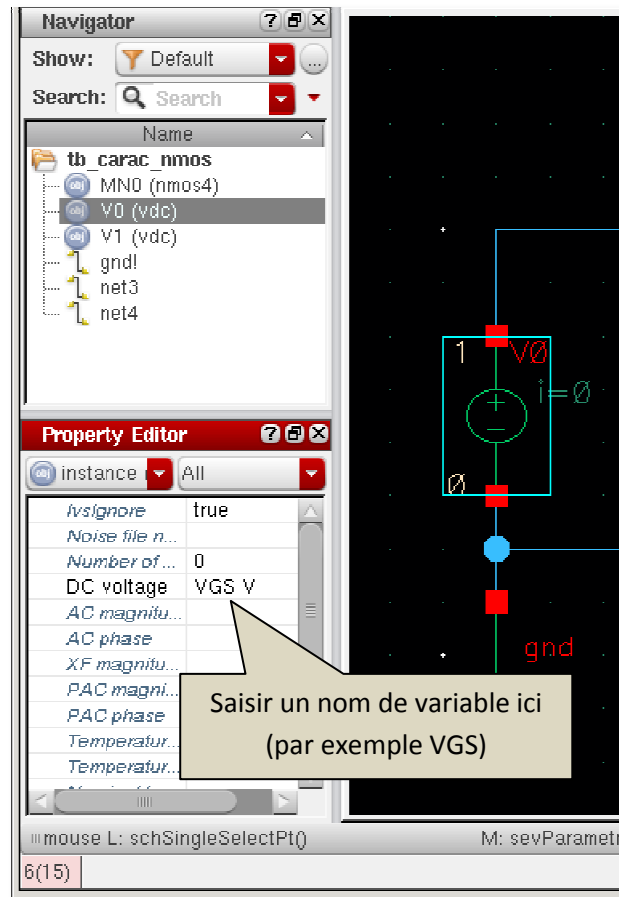
Dans la fenêtre **Visualisation & Analysis XL**, promener le curseur de la souris le long de la caractéristique pour obtenir des coordonnées ( $I_D$ ,  $V_{DS}$ ). Presser la touche **A** pour positionner un premier marqueur. Changer de position sur la courbe et presser la touche **B** pour positionner un second marqueur. Les informations **dx**, **dy** et la pente apparaissent aussitôt. Sélectionner un marqueur et utiliser la touche **Suppr** pour le supprimer.

#### 6.4. Analyse paramétrique du point de polarisation avec une variable (sweep)

Supposons maintenant que nous souhaitons tracer le réseau de caractéristiques  $I_D=f(V_{DS})$  pour différentes valeurs de la tension  $V_{GS}$ . Cela revient à répéter la simulation précédente plusieurs fois en faisant varier un paramètre de cette simulation qui n'est autre que le paramètre **DC voltage** de la source de tension **VO**.

Pour cela, nous avons besoin dans un premier temps de créer la variable  $V_{GS}$ .

Depuis l'éditeur de schéma, sélectionner la source de tension V0. Dans le cadre **Property Editor**, effacer la valeur '1 V' actuellement présente pour le paramètre DC voltage, et saisir à la place un nom qui peut être quelconque, mais qu'il vaut mieux choisir de façon pertinente. Prenons par exemple 'VGS'.



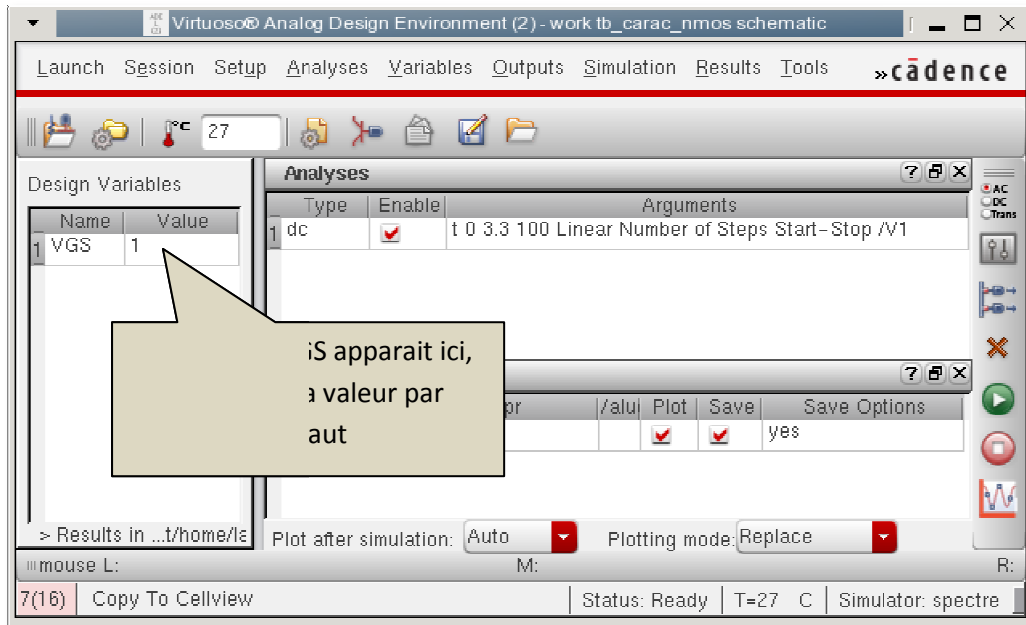
Faire :


**File > Check and Save** (ou )

Depuis la fenêtre **Analog Design Environment** faire :

**Variables > Copy From Cellview**

Le cadre **Design Variable** de la fenêtre **Analog Design Environment** affiche maintenant la variable **VGS**. Cliquer dans la colonne **Value** et saisir **1**. Ainsi configurée, la variable **VGS** vaut 1V. Cela veut dire qu'en pratique, nous n'avons pas modifié les conditions de simulation par rapport à la simulation précédente.



Lancer la simulation (  ) et vérifier que l'on obtient exactement la même caractéristique  $I_D=f(V_{GS})$  que précédemment. Noter la valeur du courant  $I_D$  autour de  $V_{DS} = 2V$  (environ  $330\mu A$ ) et fermer la fenêtre de visualisation de traces.

Modifier la valeur du paramètre **VGS** (par exemple 2V) et relancer la simulation. Mesurer le courant pour  $V_{DS} = 2V$  et conclure sur l'effet du paramètre **VGS**.

Nous souhaitons maintenant utiliser ce paramètre **VGS** pour tracer un réseau de caractéristiques  $I_D=f(V_{GS})$ . Depuis la fenêtre **Analog Design Environment**, faire :

#### Tools > Parametric Analysis

La fenêtre **Parametric Analysis** s'ouvre.

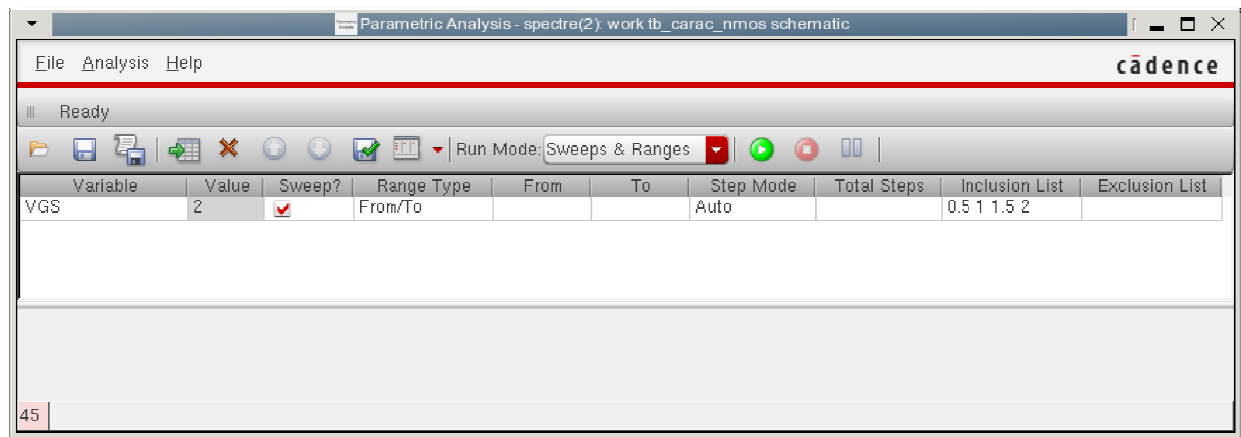
Cliquer dans la case **Add Variable**, puis dans la liste déroulante qui apparaît, sélectionner la variable **VGS**.

Il existe plusieurs méthodes pour spécifier les valeurs de **VGS** à inclure dans la simulation paramétrique :

- Automatique sur une plage de valeur (de type **From/To**) en spécifiant le pas, ou le nombre de points.
- Par des listes d'inclusion ou d'exclusion.

Nous allons pour l'exemple procéder avec une liste d'inclusion. Laisser les colonnes **From**, **To**, et **Total Steps** vides. Dans la colonne **Inclusion List**, taper :

0.5 1 1.5 2



Puis lancer l'analyse paramétrique en faisant :

**Analysis > Start Selected** (ou )

Une barre de progression apparaît dans la partie basse de la fenêtre ainsi qu'un journal de simulation. Après quelques secondes, l'analyse paramétrique se termine et la fenêtre de visualisation de tracés s'ouvre avec le réseau de caractéristiques attendu.

