



Licence EEA
3^{ème} Année

Module : Circuits et Composants Analogique CCA

Année Universitaire : 2008/2009

Introduction à l’outil de Simulation de circuits Electronique PSpice version 9.1

M. Réda YAHIAOUI
Mme Isabelle LAJOIE

Sommaire :

| | |
|---|----|
| 1 Introduction - Installation du logiciel : | 4 |
| 2 Principe de fonctionnement : | 5 |
| 3 Créer un schéma : | 6 |
| 4 Simuler et observer les résultats : | 10 |
| 5 Remarques et compléments utiles : | 13 |
| 6 Générer un Circuit Imprimé : | 14 |
| TP Pspice 1 | 17 |
| TP Pspice 2 | 19 |
| TP Pspice 3 | 21 |

Le module Circuits et Composants Analogique de la Licence EEA 3^{ème} année comprend 30 heures de travaux Pratiques réparties en 10 séances de 3 heures.

Les quatre premières séances sont consacrées à la présentation, et la prise en main du logiciel Pspice version 9.1 à travers un exemple et trois TP d'application. Les six autres séances à des TP d'analogique sur maquette LabDec.

Chaque étudiant devra faire un compte rendu précis de son travail qu'il faudra remettre à l'enseignant. La rédaction des comptes rendus de travaux pratiques fera apparaître chaque fois toutes les étapes de l'analyse des problèmes posés en utilisant les outils descriptifs appropriés : schémas, commentaires,

La présentation sera **claire**, **aérée** et **soignée**. Dès que celle-ci s'impose, une conclusion clairement exposée fera la synthèse des différentes étapes du travail demandé.

La **non remise** du compte rendu, l'absence non justifiée à une séance entraînera l'attribution d'une **note nulle**.

Coordonnées des intervenants de TP CCA

| |
|--|
| M. Réda YAHIAOUI : Bureau 243B Email : reda.yahiaoui@univ-fcomte.fr Tel UFR-ST : 03 81 66 57 53 Tel FEMTO-ST : 03 81 85 39 36 |
| Mme Isabelle LAJOIE : Bureau 241B Email : ilajoie@femto-st.fr Tel UFR-ST : 03 81 66 63 57 Tel FEMTO-ST : 03 81 85 39 93 |

1 Introduction - Installation du logiciel :

Ce chapitre permet de démarrer sur PSpice (Simulation Program with Integrated Circuits Emphasis), de comprendre et d'utiliser les fonctionnalités essentielles de ce logiciel. Il constitue un résumé de ce qu'il faut savoir. En effet, le logiciel est (trop) complet et il serait illusoire de vouloir en faire une description détaillée en quelques pages. Malgré sa relative complexité, la prise en main des fonctions reste relativement simple.

1.1 Configuration matérielle :

- PC ou compatible Pentium (200 Mhz minimum) avec 64 Mo minimum pour Windows NT et Windows 2000.
- Plates-formes: Windows NT 4.0 (avec service Pack 4 ou plus), Windows 95, Windows 98, Windows 2000.
- Orcad Capture CIS requière Microsoft Data Access Pack (ODBC) 3.5 ou plus récent, DCOM95 (nécessaire pour Windows 95) et Microsoft Internet Explorer 4.0 ou plus récent. Ces modules vous seront proposés lors de l'installation si cela est nécessaire.

☞ Pour les étudiants disposant d'ordinateurs moins performants, on peut utiliser des versions plus anciennes fonctionnant quasiment de la même façon (mais ne permettant pas la création de circuits imprimés).

1.2 Procédure d'installation : (valable quelle que soit la version)

☞ Source sur Internet :

Télécharger le fichier à partir de l'adresse suivante :

<http://www.electronics-lab.com/downloads/schematic/013/>

Lancez **SETUP.EXE**

☞ Source sur CD-ROM :

- Lancez le programme **SETUP.EXE** sur la racine du CD-ROM ou sélectionnez simplement l'icône « Install Evaluation Software » sous Windows (il apparaîtra automatiquement après insertion du CD-ROM).
- **Lors de l'installation n'oubliez pas de cocher sur l'option « Schematics » lors de l'étape montrée dans la figure ci-dessous :**

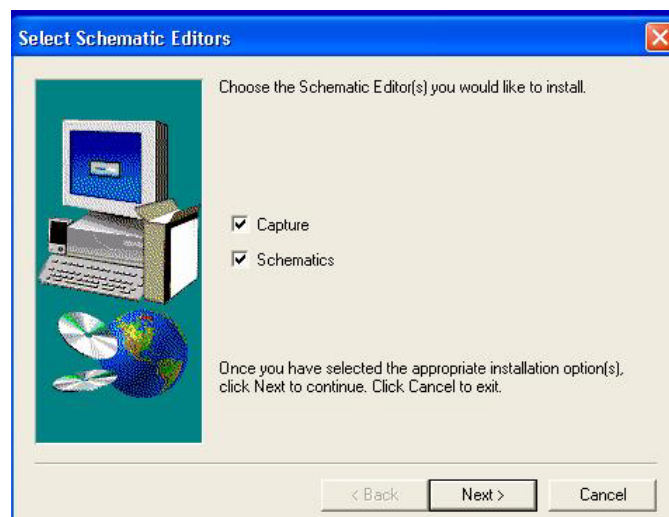


Fig.1 : Options d'installation de Pspice 9.1

1.3 Limitations :

La version utilisée est une version de démonstration, ses principales limitations sont les suivantes :

- Bibliothèque de composants limitée,
- Saisie de schémas limitée à une page A4 et 50 composants,
- Simulation limitée à 64 nœuds, 10 transistors ou 65 fonctions logiques,
- Conception de circuits imprimés limitée à 30 composants, 50 pistes et 2 couches (double face),
- Optimisation limitée à un seul paramètre,

Quoi qu’il en soit, ces limitations devraient rarement être atteintes pour tout ce qui va être fait durant les TPs de licence EEA.

2 Principe de fonctionnement :

PSpice est un logiciel complet dont on peut observer la plate-forme ci-dessous :

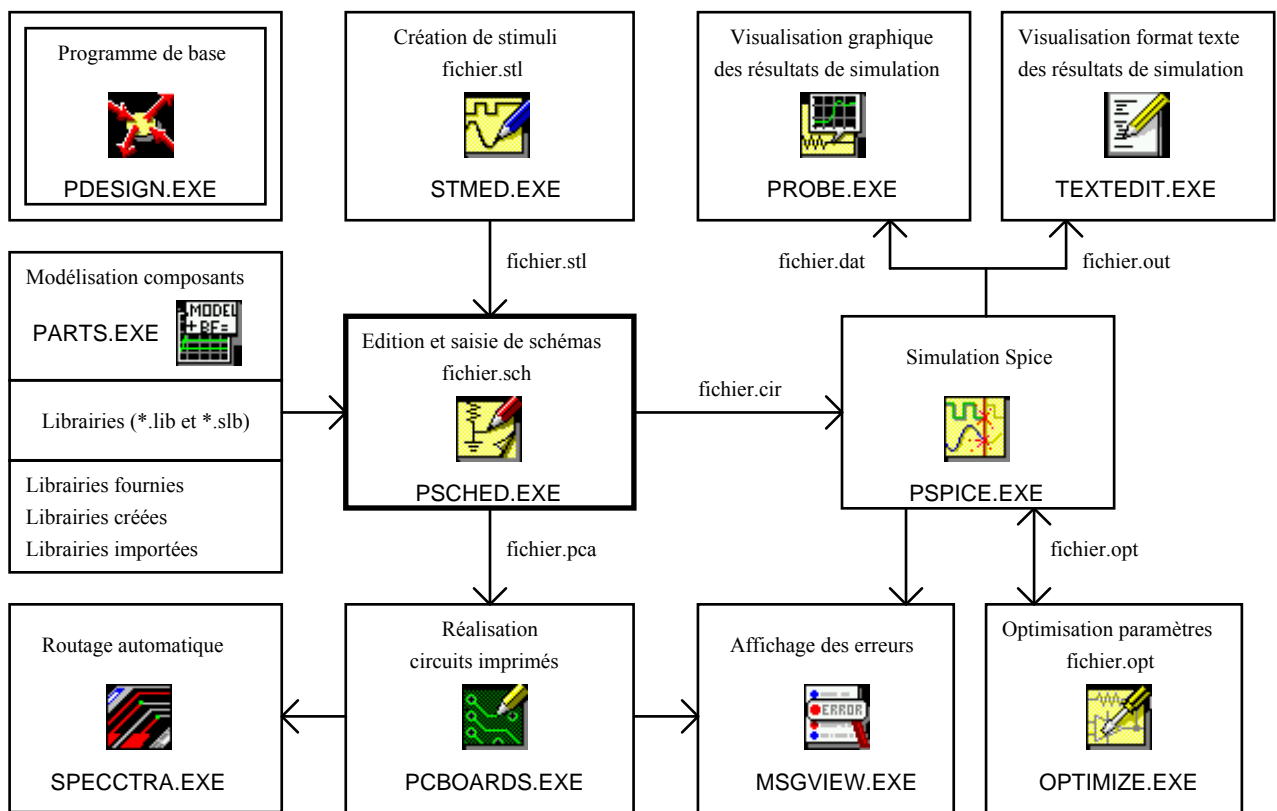


Fig.2 : Plate forme générale du logiciel PSpice

- Saisie de schéma :

Pour créer un circuit sous PSpice, on dessine tout d’abord le schéma électrique à l’aide de l’éditeur graphique PSCHED.EXE en utilisant des composants disponibles dans des bibliothèques ou en les créant (éventuellement en utilisant PARTS.EXE qui permet de créer des modèles de composants à partir des abaques constructeur, la version de démonstration étant limitée à la caractérisation des diodes).

- Simulation :

Après la saisie du schéma, on peut choisir un type de simulation puis l’effectuer. Le logiciel appelle alors automatiquement PSPICE.EXE qui réalise la simulation numérique proprement dite. Les résultats sont fournis soit sous forme de texte (affichage avec l’éditeur TEXTEDIT.EXE), soit sous forme graphique (visualisation avec PROBE.EXE). On peut aussi, à l’aide de OPTIMIZE.EXE, lancer automatiquement des simulations en vue de l’optimisation d’un paramètre (trouver la bonne résistance pour avoir un gain donné par exemple). Quoi qu’il arrive, l’utilisation est « transparente » pour l’utilisateur : Le logiciel s’occupe tout seul de lancer les différents programmes.

↗ A l'origine, il n'existait pas d'éditeur de schémas, le circuit était décrit dans un fichier texte (*.CIR) comme dans un programme par une suite de lignes (Composant, connexions, type, valeur), les commandes de simulation étaient obtenues à l'aide d'instructions. Sous DOS, on lançait alors le fichier (*.CIR) sur PSpice, qui fournissait les résultats dans des fichiers de type *.OUT (résultats texte) et *.DAT (résultats graphiques) que l'on devait ensuite manipuler « Manuellement ». Ces fichiers existent toujours mais il n'est plus nécessaire de les utiliser directement. Quand on lance une simulation sur PSpice 9.1, un fichier (*.CIR) est automatiquement créé à partir du schéma (*.SCH), la simulation est lancée automatiquement et sort toujours les résultats dans des fichiers (*.OUT et *.DAT) qui sont eux-mêmes automatiquement appelés. Ceci explique en partie à quoi correspondent les nombreux fichiers que PSpice crée avec chaque schéma.

- Création du circuit imprimé

Quand le circuit possède les caractéristiques désirées, on peut l'exporter sous PCBOARD.EXE qui est un programme de création de circuits imprimés. Il est possible, soit de router manuellement, soit d'utiliser le routeur automatique SPECCTRA. Une fois le circuit terminé, on peut sortir les résultats sous diverses formes en vue de la réalisation (dessin du typon, liste des composants, schéma d'implantation, plan de perçage, fichier Gerber, ...)

3 Créer un schéma :

↗ Comme dans le cas de la plupart des logiciels de conception assistée par ordinateur, il est plus aisé de traiter un exemple concret quand on veut apprendre à utiliser un tel logiciel. Il s'agit donc de décrire l'étude d'un schéma type. Ceci est la méthode efficace et rapide pour prendre en main PSpice. Cette initiation devrait permettre à un débutant de mieux maîtriser les bases de PSpice en moins d'une demi-journée. Ces bases pourront être enrichies par la suite à des fins professionnelles. A cet effet, tout ce qui apparaîtra par la suite dans un cadre en pointillés correspond aux actions à effectuer.

Dans PSpice, le « Centre nerveux » est l'éditeur de schémas Schematics (PSCHED.EXE). Il s'agit d'abord de le lancer. Nous allons dès lors créer un petit circuit passe-bande à Ampli-Op (ce circuit est différent de ceux des TP 1, 2 et 3 que vous allez faire) :

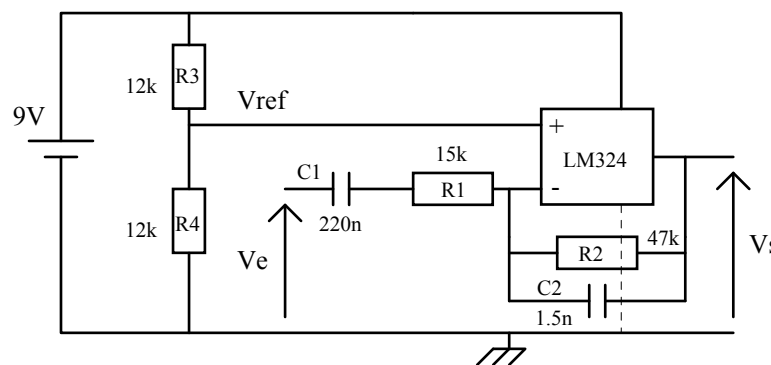


Fig.3 : Circuit électronique à simuler

3.1 Appel des composants :

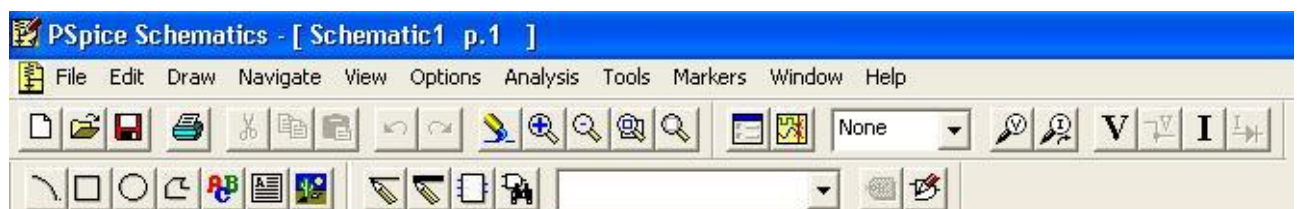


Fig.4: Principaux Menus et icônes e la version PSpice 9.1

Comme sur la plupart des logiciels sous Windows, les fenêtres de PSpice se présentent avec une barre supérieure et des menus déroulants. Un ensemble d'icônes (paramétrables) sont des raccourcis pour les commandes les plus courantes. Un nouveau document est ouvert par défaut, il faut tout d'abord lui donner un nom, pour cela faire :

File ⇒ Save as en donnant un nom choisi.

On peut dès lors commencer le dessin, l'appel des composants se fait par **Draw ⇒ Get New Part** qui affiche une fenêtre (Fig.5) permettant la sélection (on peut aussi utiliser les icônes correspondantes si elles sont paramétrées). Pour rechercher un composant, on peut :

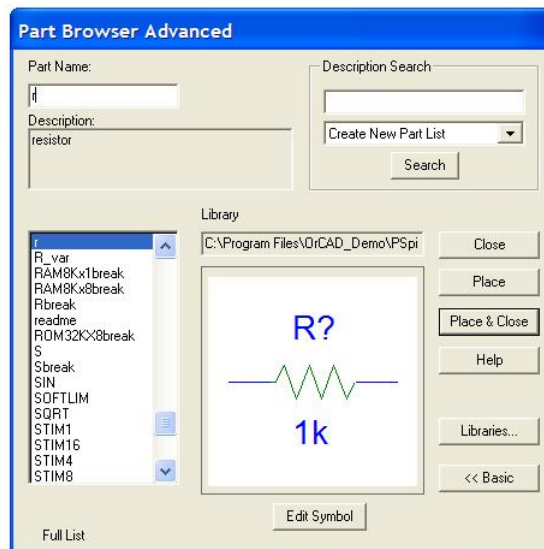


Fig.5 : Fenêtre pour la sélection des composants

- ① Se « Promener » dans les librairies (Browser),
- ② Taper le nom du composant si on le connaît avec éventuellement les caractères * et ? : **Part Name = 7*0?** affichera tous les composants dont le nom commence par 7 et qui possèdent un 0 en avant-dernière position,
- ③ Effectuer une recherche descriptive : **Advanced >> ⇒ Description Search = *invert*** listera tous les composants qui possèdent la chaîne « invert » (inverseur) dans leur description. Pour avoir de nouveau accès à l'ensemble des composants des librairies, remettre ensuite **Description Search = *** et cliquer sur **Search**.

Une fois le composant désiré sélectionné, validez **Place**, **Place&Close** ou appuyez sur la touche **Entrée**. Il suffit de le placer où on désire à l'aide de la souris : *Le bouton gauche* permet de placer autant de composants de ce type que l'on veut, *le bouton droit* permet de sortir du placement. Pour tourner le composant, utilisez **Edit ⇒ Rotate** ou le raccourci **CTRL+R**, pour faire un effet miroir utilisez **Edit ⇒ Flip** ou **CTRL+F**.

*⇒ Pour un débutant, il faut juste savoir que la simulation est possible car chaque composant fait appel à un modèle dont le nom apparaît lorsque l'on double-clique dessus. Ces modèles sont relativement complets et réalistes, ce qui permet des simulations fiables. Il faut savoir qu'il est possible de créer ou modifier un modèle associé à un composant (En général en utilisant les données **des Data Book constructeurs**). On peut toutefois se contenter d'utiliser les modèles fournis, ou d'ajouter les modèles que distribuent gracieusement la plupart des constructeurs (par exemple sur Internet).*

3.2 Dénominations des principaux éléments simulables :

⇒ Dans un souci de simplification, on se limite à l'essentiel.

Les composants analogiques :

| Eléments | Dénomination PSpice |
|--------------------------------------|--|
| Eléments passifs | R, L, C, K (circuit magnétique) ou T (ligne de transmission) |
| Diode xxxx | Dxxxx |
| Transistor Bipolaire xxxx | Qxxxx |
| Transistor MOSFET xxxx | Mxxxx |
| Transistor JFET xxxx | Jxxxx |
| Transistor IGBT xxxx | Zxxxx |
| Composant défini par un sous-circuit | Xxxxx |

Les composants numériques :

La version de démonstration 9.1 possède un assez large panel de composants numériques, essentiellement de la famille TTL et même un exemple de composant logique programmable (PLD ou PAL). L'appel des fonctions des divers composants se fait par leur nom (74xx) et ils sont affichés selon la norme américaine (*si besoin était, on pourrait redéfinir leur dessin avec la norme européenne*). PSpice étant essentiellement utilisé pour la simulation analogique, on ne s'étendra pas sur le sujet.

Les sources :

| Sources | Dénomination PSpice |
|--|---------------------|
| Potentiel 0 | AGND |
| Source continue | VDC, IDC |
| Source continue avec composante alternative | VSIN, ISIN |
| Signal carré de rapport cyclique réglable | VPULSE, IPULSE |
| Source définie par segments (Vi, ti) ou (Ii, ti) | VPWL, IPWL |
| Source de tension contrôlée en tension | Exxx |
| Source de courant contrôlée en courant | Fxxx |
| Source de courant contrôlée en tension | Gxxx |
| Source de tension contrôlée en courant | Hxxx |

3.3 Connexions et paramètres des composants :

Les fils de connexion :

Les fils de connexion se tracent simplement avec la souris à l'aide de **Draw** ⇒ **Wire** ou à l'aide de la combinaison **CTRL+W**.

Donner leurs valeurs aux composants :

Une fois le circuit créé (ou lors de sa création), il faut entrer les valeurs et les paramètres. Pour le faire, c'est tout simple, **il suffit de double-cliquer sur l'élément considéré**. PSpice connaît les facteurs :

$$f = 10^{-15}, p = 10^{-12}, n = 10^{-9}, u = 10^{-6}, m = 10^{-3}, k = 10^{+3}, meg = 10^{+6}, g = 10^{+9}.$$

Il n'est pas nécessaire de mettre les unités (ce sont les unités standard par défaut), mais il ne faut pas faire de confusion, PSpice **confondant majuscules et minuscules** (⚠ M = m = 10⁻³). On donne ainsi les valeurs aux résistances et capacités. En double-cliquant sur un fil, on peut nommer les signaux (Vs pour une sortie par exemple). Pour les sources, PSpice demande un certain nombre de paramètres qui sont peut-être moins clairs et qui seront détaillés dans ce qui suit.

↪ En cas d'erreur et pour effacer un ou plusieurs fils ou composants, il suffit de les mettre en surbrillance en cliquant dessus avec la souris (ils apparaissent alors en rouge) et d'appuyer sur la touche Suppr (ou de faire **Edit** ⇒ **Delete**). Pour déplacer un composant, il suffit de se placer dessus avec la souris, maintenir cliqué et de le déplacer.

Dessinez le circuit en utilisant comme Ampli-Op le LM324, prenez une source VDC pour l'alimentation et une source VSIN pour le signal d'entrée. Placez ensuite les fils de connexion. Au cours du dessin ou une fois le dessin achevé, donnez leurs valeurs aux composants et nommez les signaux Ve et Vs.

3.4 Les sources et leurs paramètres :

Avant de faire une description des sources, il faut tout d'abord connaître les notions liées à la simulation. Les sources peuvent posséder deux paramètres différents (l'un, l'autre ou les deux). Le premier est lié à la simulation fréquentielle, l'autre à la simulation temporelle,

Dans l'analyse fréquentielle (AC Sweep en Anglais), le simulateur utilise le circuit AC équivalent et fait varier la fréquence de la source d'excitation petits signaux. A cet effet, PSpice réalise un calcul linéaire dans le domaine fréquentiel. Ce type d'analyse *ne présage donc pas des phénomènes de non-linéarité et de saturation et de certains composants*.

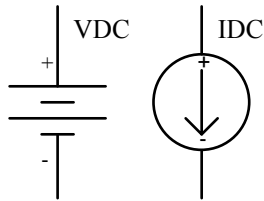
Dans l'analyse transitoire ou temporelle (*Transient* en Anglais), PSpice effectue une simulation en fonction du temps, c'est la seule simulation qui permet de prendre en considération les phénomènes non-linéaires et donc d'observer la forme réelle des signaux. A cet effet PSpice utilise les paramètres temporels des sources.

⇒ Le potentiel 0 :

Après simulation, PSpice permet d'afficher n'importe quel potentiel du circuit. A cet effet, il faut nécessairement un potentiel de référence (composant AGND = potentiel 0). ⚠ **PSpice refusera donc toute simulation s'il n'y a pas au moins un potentiel AGND dans le circuit. De même qu'il indiquera une erreur s'il y a des potentiels non référencés (erreur floating node).**

N'oubliez pas le potentiel 0

⇒ Les sources continues :



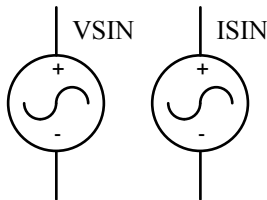
DC = Valeur continue
SIMULATIONONLY = Signifie que la source sera ignorée lors de la conception CI

Donnez la valeur 9V à la source continue en double-cliquant dessus, puis sur « DC= » et en entrant le paramètre dans la zone d'édition.

⇒ *Avez les anciennes versions de PSpice, on utilisait une source VSRC avec paramètre AC = 0.*

⇒ Les sources alternatives :

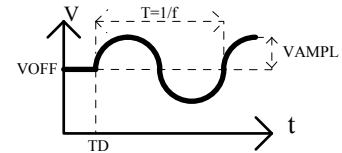
- Pour l'analyse fréquentielle (AC Sweep) PSpice utilise :



DC = Valeur de la composante continue
AC = Amplitude de la composante alternative

- Pour l'analyse transitoire ou temporelle (Transient) PSpice utilise :

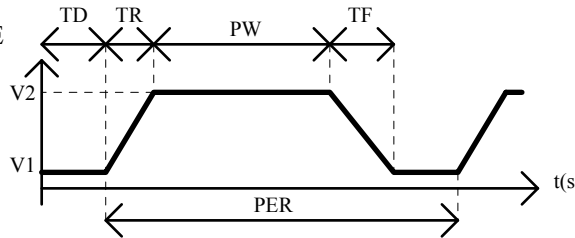
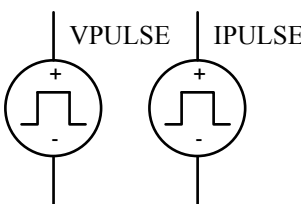
VOFF = Valeur de la composante continue
VAMPL = Amplitude de la composante alternative
FREQ = Fréquence en Hz
TD = Délai d'apparition du signal (0 par défaut)
DF = Coefficient d'amortissement (0 par défaut)
PHASE = Phase (0 par défaut)



⇒ *Ces deux sources conviennent pour tout type de simulation. On peut aussi utiliser les sources VAC et IAC, **mais elles ne permettent pas l'analyse temporelle (Transient)**. Les sources VSRC et ISRC sont du même type que VSIN et ISIN mais elles utilisent pour l'analyse transitoire une fonction du temps choisie par l'utilisateur (paramètre TRAN).*

Donnez alors les caractéristiques de la source VSIN : 10mV d'amplitude, 1KHz, sans composante continue, délai d'apparition du signal nul. Utilisez ces valeurs aussi bien pour les analyses fréquentielle (DC, AC) que transitoire (VOFF, VAMPL, FREQ).

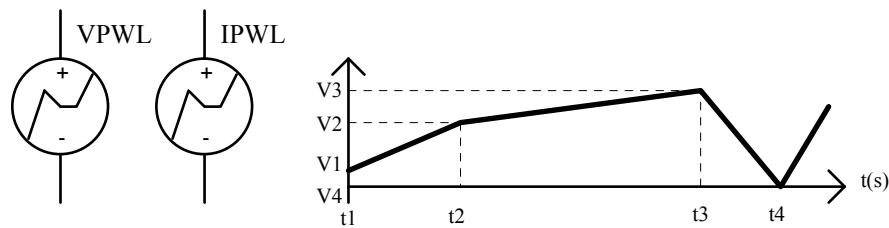
⇒ Les sources de signal « carré » :



TD : Delay Time
TR : Rise Time (>0!)
PW : Pulse Width
TF : Fall Time (>0!)
PER : Period

⇒ *Ce type de source est évidemment utilisé en analyse transitoire. Si on effectue une analyse fréquentielle, la source devient sinusoïdale et utilise les paramètres AC et DC. On doit aussi noter que TF et TR ne doivent pas être nuls (ce qui correspond à la réalité). Pour avoir des signaux « carrés », il faut utiliser de très faibles valeurs.*

⇒ Les sources définies par segments :



↷ Comme les sources de signal carré, ce type de source est évidemment utilisé en analyse transitoire. Si on effectue une analyse fréquentielle, la source devient sinusoïdale et utilise les paramètres AC et DC.

↷ Pour ceux qui voudraient aller plus loin, sachez qu'il existe d'autres types de sources, notamment VEXP qui est une source exponentielle et VSFFM une source modulée en fréquence.

On devait alors observer (mis à part les sonde) quelque chose qui ressemble à :

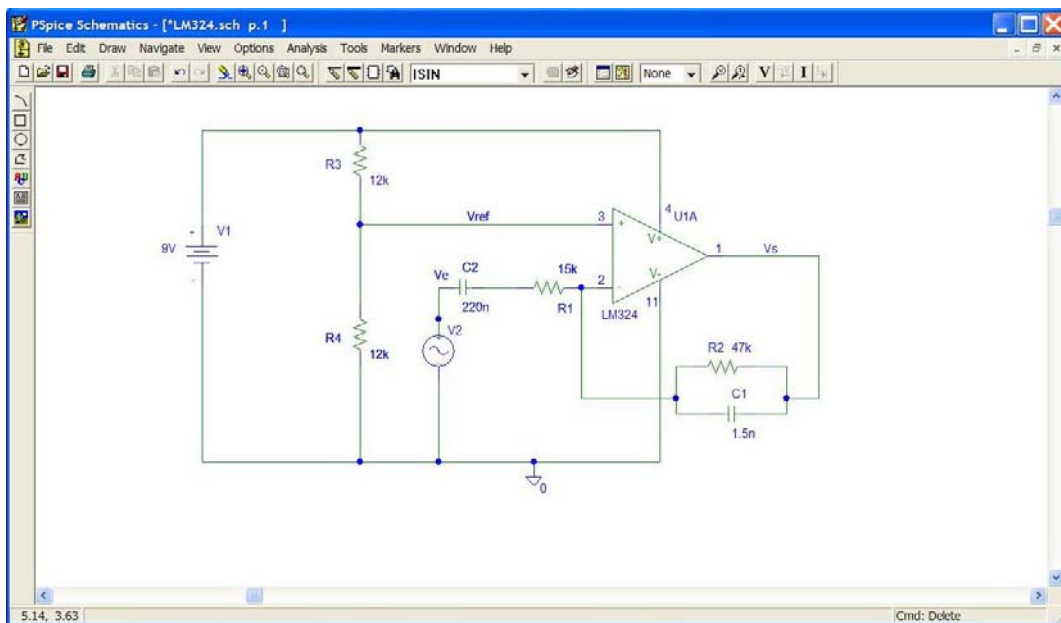


Fig.6 : Saisie sur PSpice du Circuit Electronique à simuler

4 Simuler et observer les résultats :

Il existe de nombreux types de simulations, on les configure à l'aide de **Analysis** ⇒ **Setup** :

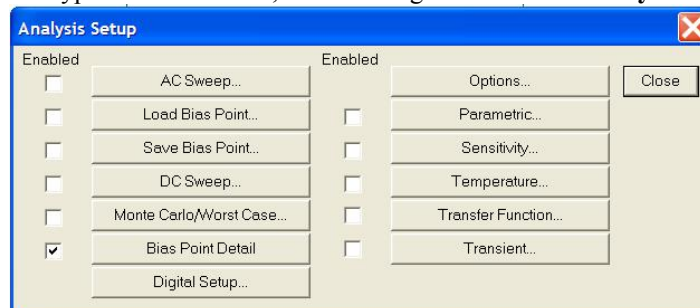


Fig.7 : Menu Analysis Setup

Pour la plupart des applications, on peut dire que l'on utilise essentiellement les 4 simulations :

- ① Bias Point Detail : Analyse du point de fonctionnement,
- ② DC Sweep : Analyse en continu,
- ③ AC Sweep : Analyse fréquentielle petit signaux,
- ④ Transient : Analyse temporelle (transitoire).

Les simulations se valident en cochant les cases (☒) des analyses désirées, chacune d'elles se configure en cliquant sur leur nom. Après paramétrage, on les lance à l'aide de **Analysis** ⇒ **Simulate**.

4.1 Simulation du point de fonctionnement :

☒ **Bias Point Detail** : Effectue une analyse statique en n'utilisant que le paramètre DC des sources (sans tenir compte des autres paramètres). Après cette simulation, on peut placer des sondes Viewpoint (**Voltmètre**) et Iprobe (**Ampèremètre**) qui permettent d'afficher les tensions et courants continus en divers points du circuit. La commande **Analysis** ⇒ **Examine Output** permet d'examiner le fichier texte des résultats. Ce fichier peut sembler un peu obscur au débutant, il faut savoir qu'il contient toutes les informations utiles du point de fonctionnement avec le potentiel de tous les nœuds du circuit, le courant débité par les sources, ...

Placez une sonde Viewpoint (**Draw** ⇒ **Get new part** ⇒ **Viewpoint**) au niveau de Vref et une sonde Iprobe (**Draw** ⇒ **Get new part** ⇒ **Iprobe**) en série sur R₄ (effacez le fil avant), vérifiez que **Bias Point Detail** est validé (☒) dans le menu (**Analysis** ⇒ **Setup**) et effectuez la simulation. On remarque que les points de fonctionnement sont dès lors affichés sur le schéma en unités standard (Volt et Ampère). A l'aide de **Analysis** ⇒ **Examine Output** observez ensuite le fichier des résultats.

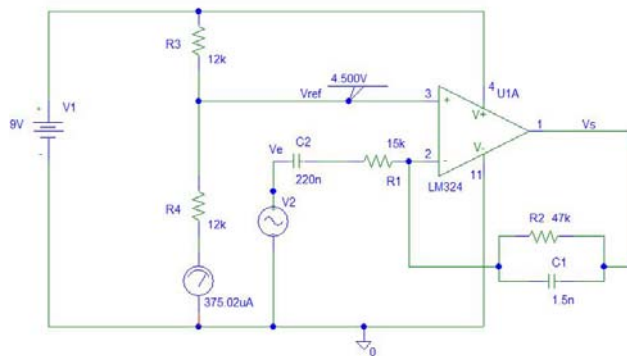
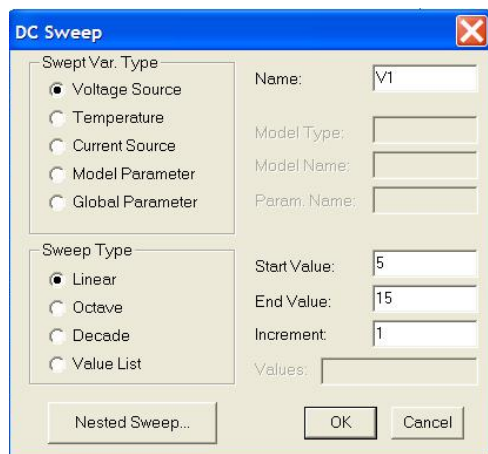


Fig.8 : Circuit Electronique à simuler avec les sondes « Viewpoint & Iprobe »

4.2 Analyse en continu :

☒ **DC Sweep** : On étudie les variations du point de fonctionnement du système en faisant varier les valeurs d'une ou plusieurs sources continues. Cette analyse équivaut à effectuer plusieurs simulations du point de fonctionnement en faisant varier un ou plusieurs paramètres. Ces derniers peuvent être la température, un paramètre du modèle d'un des éléments du circuit, une variable globale et la tension ou le courant d'une source. Après la simulation, le programme PROBE de visualisation graphique est automatiquement lancé.



Validez DC Sweep et demandez une variation linéaire de V1 de 5 à 15 V avec un incrément de 1V, lancez ensuite la simulation. Dès que PSpice a terminé, il lance PROBE avec V_V1 en abscisse. Son utilisation est assez ergonomique et ne devrait pas poser de problème. On désire par exemple afficher Vref ou Vs en fonction de la tension d'alimentation. Pour cela faites **Trace** ⇒ **Add** et sélectionnez V(Vref) (ou tapez-le manuellement), faire de même avec V(Vs). Sur cet exemple simple on remarque évidemment que les polarisations d'entrée et de sortie restent à (V1)/2 (à cause du pont R2-R3).

La fenêtre de Probe devrait alors apparaître à peu près comme ceci :

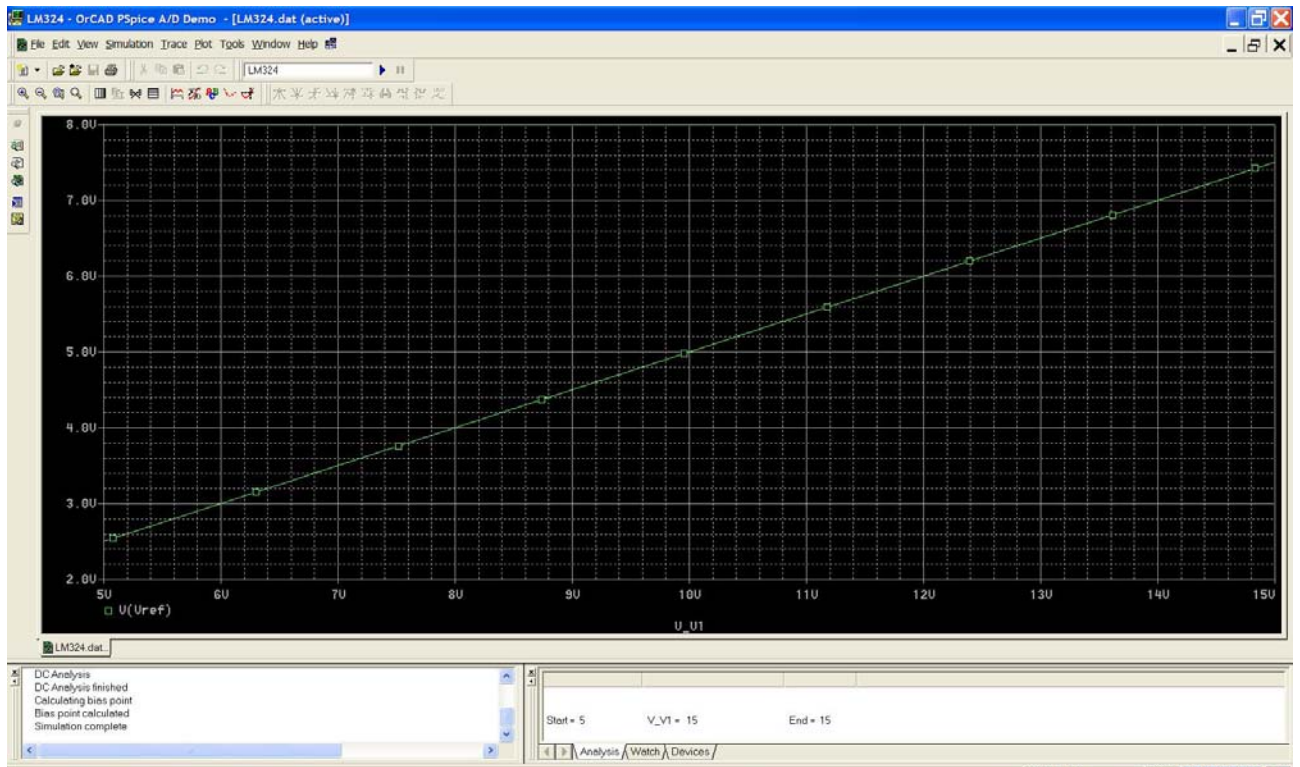
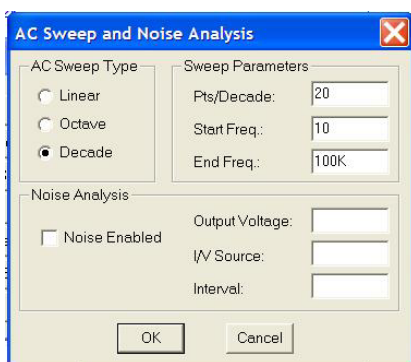


Fig.8 : Fenêtre de Probe avec V_{Ref} en fonction de V_1

Si l'on veut examiner des points précis, on peut utiliser le curseur à l'aide de **Trace** ⇒ **Cursor** ⇒ **Display**, ce qui permet de déplacer deux curseurs sur une ou plusieurs courbes. A cet effet, il faut tout d'abord sélectionner la courbe sur la quelle on veut déplacer le curseur (en cliquant sur sa légende en bas à gauche). On peut ensuite déplacer les curseurs en utilisant les boutons gauche et droits de la souris ou les flèches du clavier (avec ou sans Shift).

4.3 Analyse fréquentielle :

AC Sweep : Analyse fréquentielle petit signaux. Le simulateur remplace le schéma par son équivalent petits signaux lorsqu'il est polarisé avec les paramètres DC des sources. Il permet de réaliser une analyse fréquentielle (réponse en fréquence, diagramme de Bode, ...). ⚡ La simulation s'effectue en complexes en utilisant seulement les paramètres AC et DC des sources et ne permet donc pas de déterminer si par exemple le circuit n'est pas saturé. En cas de doute, on effectuera au préalable une analyse transitoire.



Dévalidez DC Sweep, validez AC Sweep et demandez une analyse par décade de 10Hz à 100KHz avec 20pts/décade, lancez alors la simulation. Sous PROBE, tracez d'abord la transmittance en tapant $V(Vs)/V(Ve)$. Effacez-la (**Trace** ⇒ **Delete All**) et demandez la courbe de gain à l'aide de : $DB(V(Vs)/V(Ve))$ [La macro $DB()$ vaut $20 \cdot \log_{10}()$]. A l'aide des deux curseurs il est alors aisé de repérer les fréquences de coupure et d'en déduire la bande passante à -3dB (la fenêtre des curseurs affiche les deux coordonnées et leurs différences). Il existe aussi des macro calculant automatiquement les paramètres : en faisant **Trace** ⇒ **Eval Goal Function** ⇒ **Bandwidth(V(vs)/V(ve),3)** on évalue la bande passante à -3dB. On dispose ainsi de toute une série de macro fort utiles (temps de montée, temps de réponse, ...)

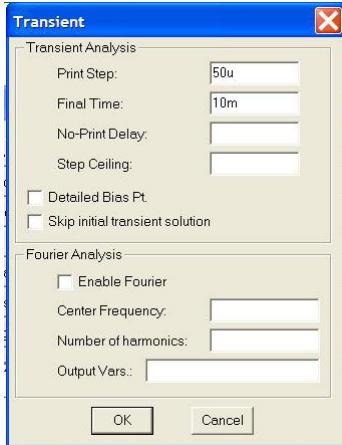
↪ La liste des Macro de calcul de fonctions (DB, valeur absolue, ...) apparaît dans la fenêtre Add Traces de Probe, la liste des macro permettant de calculer les paramètres (temps de montée, de réponse, ...) est accessible dans le menu **Trace** ⇒ **Goal Functions**. On peut assez facilement éditer, modifier ou créer une macro.

↪ Les courbes peuvent être imprimées ou sauvegardées à l'aide respectivement de **File** ⇒ **Print** et de **Window** ⇒ **Display Control**. On peut aussi insérer les graphes (ou les schémas) dans un autre document (Un document Word par exemple) à l'aide de **Window** ⇒ **Copy to Clipboard** puis en faisant **Edition** ⇒ **Coller** (sous Word). Cependant l'image est enregistrée en point par point, ce qui occupe beaucoup de place et il faut modifier la couleur du fond (noire par défaut, pour la modifier rechercher le mot Colors dans l'aide de Probe).

Si on veut le diagramme de Bode complet, on peut aussi tracer la phase simplement sur un autre graphe, pour cela faire **Plot** ⇒ **Add Plot** puis demander de tracer $P(V(Vs))$ [éventuellement $-P(V(Ve))$ mais elle est nulle ici].

4.4 Analyse temporelle :

Transient : Permet une analyse temporelle (transitoire) du fonctionnement du circuit, c'est la seule analyse valable pour observer un signal de sortie (voir s'il n'y a pas saturation,...). PSpice stockera les résultats avec un pas temporel **Print Step**, et pour $t \in [\text{No-Print Delay}, \text{Final Time}]$. Le paramètre **Step Ceiling** correspond au pas maximal de calcul du simulateur (si les courbes sont trop segmentées, il faut lui imposer une faible valeur).



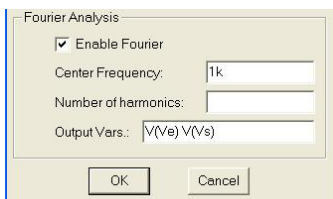
Commencez par changer le signal sinusoïdal d'entrée en signal carré, à cet effet, remplacez la source VSIN par une source VPULSE ($V1=0$, $V2=100m$, $TD=0$, $TR=1n$, $TF=1n$, $PW=0.5m$, $PER=1m$). Dévalidez alors AC Sweep, validez Transient sur [0,10ms] avec un pas d'écriture de $50\mu s$. Lancez la simulation, demandez à visualiser $V(Vs)$. On observe bien la réponse d'un second ordre. Sur un autre graphe (**Plot** ⇒ **Add plot**) à cause du point de fonctionnement, affichez $V(Ve)$. On observe donc bien en parallèle les signaux d'entrée et de sortie. Mieux : demandez la transformée de Fourier (à l'aide de **Trace** ⇒ **Fourier** ou de l'icône correspondante). On observe alors la transformée rapide de Fourier (FFT) qui montre clairement que le circuit diminue les harmoniques de rang élevé !

5 Remarques et compléments utiles :

Développement en série de Fourier :

Par affichage sous Probe, on peut effectuer la transformée de Fourier d'un signal. Il existe aussi une fonction qui permet de réaliser le développement en série de Fourier (DSF). Pour cela, il faut retourner dans le menu de l'analyse transitoire, valider Fourier (Enable Fourier), donner la fréquence du signal et nommer le signal dont on veut le DSF (par défaut le nombre d'harmoniques est 10). Après simulation, sous Schematics faire **Analysis** ⇒ **Examine Output**. En fin de fichier on trouve les résultats du DSF avec les différents harmoniques, leur composante ainsi que leur phase et le taux de distorsion harmonique.

Le taux de distorsion harmonique (THD) est assez précis, le DSF est par contre peu précis pour les faibles valeurs.

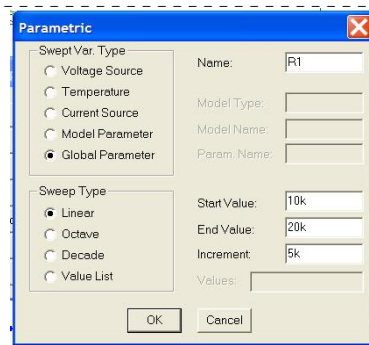


En conservant en entrée le signal carré, validez le DSF, donnez la fréquence 1KHz ($=1/(1ms)$) et demandez l'analyse pour les signaux $V(Ve)$ et $V(Vs)$. Simulez et observez les résultats : On remarque clairement que le filtre diminue le taux de distorsion.

On remarquera que notre passe-bande a bien diminué l'amplitude des harmoniques de rang élevé, ce qui a diminué le THD.

Faire varier un paramètre :

Si l'on veut étudier le comportement d'un circuit avec un composant dont la valeur varie, il faut le déclarer en paramètre. A cet effet, au lieu de lui donner une valeur, donnez un nom de variable arbitraire entre accolades, par exemple {nom_var}, insérez ensuite le « composant spécial » « PARAM » et déclarez après avoir double-cliqué dessus $NAME1=nom_var$ et la valeur par défaut *valeur*). Pour réaliser une simulation en fonction de ce paramètre, faites **Analysis** ⇒ **Setup** ⇒ **Parametric** et déclarez la variable nom_var en paramètre global (Global Parameter) et choisissez les variations désirées. Choisissez enfin le type général de simulation.



Remplacez la source carrée par la source sinusoïdale initiale, déclarez R1 en paramètre (valeur {V_R1} par exemple), insérez le composant PARAM et donnez la valeur par défaut (NAME1=V_R1, VALUE1=15k). Dans le menu de simulation (Setup) dévalidez l'analyse transitoire, validez l'analyse fréquentielle et demandez une analyse paramétrique linéaire sur le paramètre global V_R1 de 10k à 20k par pas de 5k. On peut alors afficher sous Probe les différentes réponse pour les différentes valeurs de R1. En double-cliquant sur le symbole (en bas à gauche) de chaque courbe, on peut entre autres avoir la valeur du paramètre pour la courbe.

Utilisation de conditions initiales - Cas particulier des oscillateurs :

Pour l'étude de certains oscillateurs, PSpice a un défaut qui nécessite une astuce. En effet, avant toute analyse, il effectue une simulation du point de fonctionnement stable et il débute l'analyse en supposant le circuit dans ce point de fonctionnement. ⚡ **A cause de ce principe, les oscillateurs peuvent ne jamais démarrer.** Cette remarque est particulièrement vraie pour l'astable à 2 transistors et 2 condensateurs : dans la réalité les tensions de bruit font qu'une des capacités est initialement à un potentiel différent de l'autre or PSpice les placera au même potentiel au démarrage, ce qui empêchera les oscillations de débiter. Pour remédier à cet inconvénient, il suffit de déclarer qu'une des capacités est initialement chargée en double-cliquant sur la capacité concernée et en déclarant la tension initiale dans « IC= »(Initial Condition). On doit ensuite demander une analyse transitoire en tenant compte des conditions initiales (valider Skip initial transient solution dans le Setup de Transient). La simulation est alors correcte.

📖 L'aide (Help) de PSpice est assez complète. A moins d'avoir oublié toute notion de la langue de Shakespeare, je conseille vivement à tous ceux qui veulent progresser de la consulter en cas de problème.

6 Générer un Circuit Imprimé :

Une fois le circuit créé, on peut réaliser le circuit imprimé correspondant à l'aide de l'interface PCBoard.

📖 Comme cette fonction est relativement complexe, je vais ici me borner à donner une « recette de cuisine » permettant de réaliser un circuit imprimé simple face à partir du circuit exemple.

6.1 Exporter le schéma du circuit sous PCBoard :

Epurier le schéma :

Il faut tout d'abord savoir que PSpice ne placera pas sur le circuit imprimé les composants propres à la simulation (caractéristique SIMULATIONONLY). Ainsi, si l'on veut des pastilles pour pouvoir câbler les alimentations, les entrées et les sorties, il faudra l'indiquer. Avec la version de démonstration (qui possède peu de symboles de connecteurs), il faudra donc placer sur le schéma des éléments « Jump » (1 pin d'un connecteur 9 pin) partout où cela est nécessaire. Il faut ensuite nommer ces pin, ⚡ PSpice n'aime pas certains noms, en commençant le nom par P il n'y a plus de problème. ⚡ D'autre part les sondes de type IPROBE doivent être enlevées, car lors de la création de CI, elles sont bizarrement remplacées par des circuits ouverts.

Association d'empreinte physique : (Packaging)

Il faut ensuite associer à chaque composant du schéma une empreinte physique, c'est à dire indiquer à quoi correspond son empreinte sur un circuit imprimé (le type de boîtier de chaque composant). On peut le faire « manuellement » en sélectionnant les attributs de chaque composant et en indiquant le type de boîtier dans la ligne « PKGTYPE= », [Ex : PKGTYPE=DIP14 (Dual Inline Package 14, circuit 14 pattes standard)]. Comme les composants des bibliothèques ont des empreintes par défaut, on se contentera généralement de faire une association automatique à l'aide de **Tools ⇒ Package ⇒ OK**,

Génération de la Netlist :

Il s'agit ensuite de créer la Netlist, c'est à dire le fichier qu'utilisera l'interface de création de circuits imprimés. Elle contient la liste des composants à souder avec leurs connexions. Pour cela, il suffit de faire **Tools ⇒ Create layout netlist ⇒ OK**. Le logiciel effectue un DRC (entrées en l'air, courts-circuits,...) et indique les erreurs et avertissements (notamment s'il y a des composants de type SIMULATIONONLY : ne pas en tenir compte)

Lancer PCBoard :

Maintenant que nous avons toutes les informations utiles, on peut lancer l'interface PCBoard à l'aide de **Tools** ⇒ **Run PCBoard** ⇒ **Yes**. Si tout se passe correctement, PCBoard devrait se lancer et afficher les composants en « Tas ».

 Nous allons implanter notre passe-bande sur circuit imprimé simple face afin de réaliser un exemple de maquette pédagogique. Pour cela, il faut d'abord placer des composants « Jump » sur l'alimentation positive, la masse, l'entrée et la sortie et de les nommer correctement (Palim, Pgnnd, Pin, Pout). Enlevez aussi la sonde Iprobe et remplacez-la par un fil. Sauvegardez le circuit et lancez **Tools** ⇒ **Package**, lancez **Tools** ⇒ **Create layout netlist** puis lancez PCBoard (**Tools** ⇒ **Run PCBoard** ⇒ **Yes**). S'il y a des erreurs, les corriger et recommencer.

6.2 Placer les composants :

Comme nous sommes en France, nous allons demander des mesures en millimètres et non en pouces (inch). Pour cela faire **Tools** ⇒ **Options** et cocher l'option (mm).

Il s'agit dès lors de placer les composants sur le circuit imprimé. Pour faciliter le routage, on veillera évidemment à les placer de sorte à minimiser les longueurs des pistes et les croisements de fils. Tout se fait à l'aide de la souris. Pour sélectionner texte, composant ou groupe de composant, cliquez dessus ou définissez un rectangle avec la souris. On peut ensuite le déplacer, le tourner (CTRL+R), faire des miroirs (CTRL+M).

☞ ⚙️ *Faire un effet miroir avec un circuit intégré équivaut à le placer sur la face soudure ! D'autre part pour pouvoir déplacer certains composants il faut parfois d'abord déplacer le texte qui est au-dessus, comme par exemple pour les pastilles « Jump ». Pour régler d'éventuels problèmes de sélection, on peut aussi utiliser le filtre de sélection **Configure** ⇒ **Selection Filter** en ne cochant que ce que l'on veut pouvoir déplacer.*

On place ensuite les composants comme on le désire pour tout le circuit. Les connexions qui ne sont pas encore routées apparaissent sous forme de fils élastiques (appelés RATs en anglais). Il se peut que lors du déplacement des composants, on embrouille cette toile d'araignée de fils. Pour minimiser la longueur de ces connexions, je conseille de faire souvent **Tools** ⇒ **Optimize Rats** qui minimise la longueur des connexions pour le nouveau placement. Ou mieux, on peut demander l'optimisation automatique des connexions à chaque déplacement de composant en cochant l'option Optimize Rats dans **Tools** ⇒ **Options**.

☞ *Pour une maquette pédagogique on veillera généralement à placer les entrées à droite, les sorties à gauche et les alimentations en haut du circuit. Après routage, on remplacera aussi les pastilles de connexion Jump par des plus grosses si on veut mettre des fiches banane 4mm (double-cliquer sur les pastilles concernées et changer le paramètre VIA_PADSTACK).*

 Réalisez le placement des composants pour notre maquette

6.3 Routage automatique :

⚙️ Le routage automatique réclame énormément de mémoire, il faut en outre fermer toutes les fenêtres excepté PCBoard et Design Manager (allez sur toutes les fenêtres ouvertes inutiles à l'aide de ALT+TAB, puis fermez-les).

Il faut tout d'abord définir les limites du circuit imprimé à l'aide de **Draw** ⇒ **Board Signal Keepin**, c'est à dire la zone hors de laquelle les pistes ne doivent pas se trouver.

Ensuite, le routeur est initialement prévu pour dessiner des circuits imprimés double-face (ou multicouches), il faut donc utiliser une astuce car on ne peut lui demander de router sur une seule face. On peut par contre lui demander de router sur deux faces en lui interdisant de passer par une des faces (appréciez la nuance !). Pour cela se placer sur face composants (Component) ou soudure (Solder) à l'aide de l'icône correspondante, puis faire **Draw** ⇒ **Keepout** et dessiner une figure englobant le circuit : on interdit ainsi au routeur de passer sur une des deux couches.

Avant de demander le routage automatique, il faut définir ses paramètres. Pour cela faire **Tools** ⇒ **CCT:Setup** et choisir la largeur de piste par défaut (Width) et l'espace minimal entre deux pistes (Clearance), par exemple 1,27 et 0,654 pour avoir des pistes assez larges; validez (OK). A l'aide de **Tools** ⇒ **CCT:Net Rules**, on peut ensuite définir des paramètres supplémentaires (par exemple si certaines pistes doivent être plus larges que les autres) dans ce menu on contrôle totalement les paramètres du routage, il existe de nombreuses options qui sont relativement complexes (je n'en ai pas testé la moitié) mais néanmoins fort utiles. Je conseille par défaut de valider aussi **DFM** ⇒ **Miter Bends** ⇒ **Set Default Miters** qui effectue un nettoyage après routage et minimise les longueurs des connexions en autorisant les pistes à 45°.

Une fois les paramètres définis, enregistrez et effectuez le routage automatique à l'aide de **Tools** ⇒ **Autoroute**. Pour les petits circuits, cela se déroule en général avec succès, sinon remplacez les composants problématiques et recommencez.

Routez automatiquement le circuit après avoir défini son contour et interdit de router sur la face composants, -----

Pour terminer, il ne reste qu'à sortir le résultat sur papier.

La liste des composants s'imprime aisément à l'aide de **File** ⇒ **Reports** ⇒ **Bill of Materials**.

Le Typon et le schéma d'implantation ne s'impriment pas à l'aide d'une simple requête, il faut d'abord créer une liste de ce qu'on veut imprimer (Job) en définissant pour chaque page la ou les couches qui doivent y être imprimées. Pour cela, faire **File** ⇒ **Job Setup** ⇒ **New**. Pour une sortie sur imprimante, il faut définir le centre de la feuille A4 en changeant les paramètres : **Plot Origin : x=-100mm, y=-150mm** ⇒ **OK**. On demande alors d'imprimer sur la page N°1 le contour et la face soudure : **Define Pages** ⇒ **New** et sélectionner **Board Signal Keepin** et **Solder**. Sur la page N°2 nous demanderons le schéma d'implantation : faire **New** (page) et sélectionner **Silktop** (contour des composants) et **Board Signal Keepin** ⇒ **OK** ⇒ **OK**. On peut dès lors imprimer **File** ⇒ **Print** ⇒ **New Job (Print)** ⇒ **Print the Job**. Il ne reste plus qu'à tirer le typon !

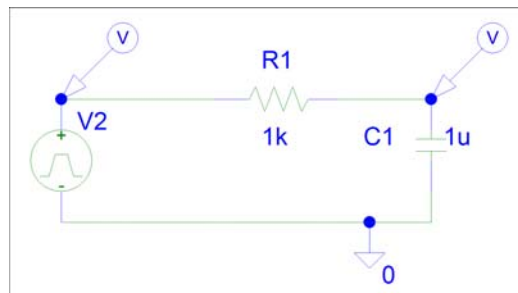
⚠ Le seul problème concerne les pastilles qui apparaissent non trouées sur le typon : Cela peut poser quelques problèmes au moment du perçage pour les circuits intégrés qui doivent normalement avoir des trous assez précisément placés.

TP Pspice 1

A- Circuit Passifs :

1- Filtre du 1^{er} ordre :

Le circuit à étudier est le circuit classique RC série :



On choisira $R = 1k\Omega$ et $C = 1\mu F$

Simulation de la réponse transitoire :

- a) $e(t)$ est un échelon de 3V.
 - Simuler l'évolution de la tension de sortie $s(t)$ en fonction du temps.
- b) $e(t)$ est de la forme $e(t) = E_{max} \sin(\omega t)$ avec $E_{max} = 2V$ et $\omega = 2513\text{rad/s}$
 - simuler la tension de sortie en fonction du temps.

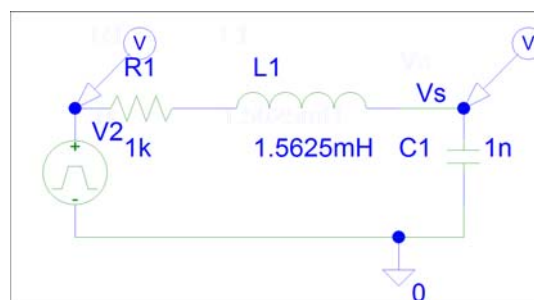
Simulation de la réponse harmonique :

On considère que $e(t)$ est un signal sinusoïdal permanent d'amplitude 1V.

- a) déterminer les courbes de Bode en module et en phase
- b) déterminer les lieux de Nyquist

2- Filtre du second ordre :

Le circuit à étudier est le circuit RLC suivant :



On choisira $R = 1k\Omega$, $L = 1.5625\text{mH}$ et $C = 1\text{nF}$

a- Simulation de la réponse transitoire :

$e(t)$ est un échelon de 3 V.

- Simuler l'évolution de la tension $s(t)$ en fonction du temps. Préciser le dépassement et la pseudo-période du signal de sortie

b- Simulation de la réponse harmonique :

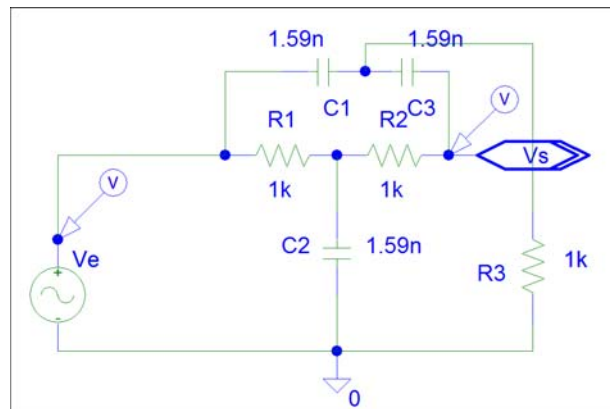
On considère que $e(t)$ est un signal sinusoïdal permanent d'amplitude 1V.

- Déterminer les courbes de Bode en module et en phase. Préciser la pulsation de résonance.

c- Comparer les résultats de a- et b- avec la théorie :

3- Filtre en T :

On considère le circuit suivant :

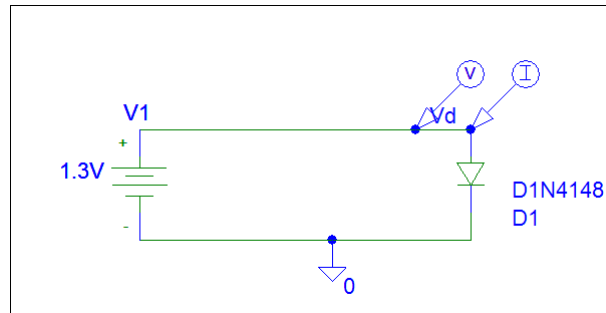


Donner la réponse harmonique de ce circuit pour $R = 1k\Omega$ et $C = 1.59nF$
 Tracer la réponse du filtre dans le plan de Bode.

TP Pspice 2

A- DIODE :

Il s'agit de tracer les caractéristiques directes et inverses de la diode pour différentes température. On considère le montage ci-dessous :



- 1) La diode est alimentée par une source de tension que l'on fera varier de 0 à 1.3V. Représenter la caractéristique directe pour trois températures différentes :

$$T = -55^{\circ}\text{C}, T = 25^{\circ}\text{C} \text{ et } T = 125^{\circ}\text{C}$$

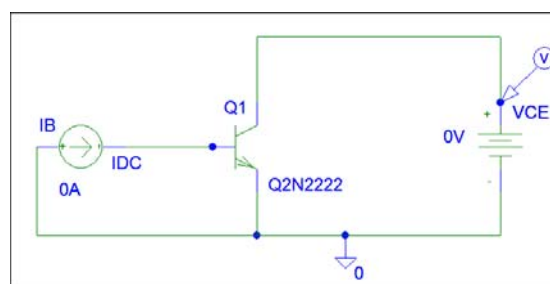
Commenter les courbes.

- 2) Pour les trois valeurs de température, observer le phénomène de claquage en traçant $I = f(V_{AK})$, $V_{AK} < 0$.

B- TRANSISTOR BIPOLAIRE :

a) Caractéristiques statiques :

- Réaliser la simulation de I_C en fonction de V_{CE} pour I_B constant. On choisira 5 valeurs de I_B comprises entre $100\mu\text{A}$ et 2mA . Le schéma permettant la simulation est le suivant :

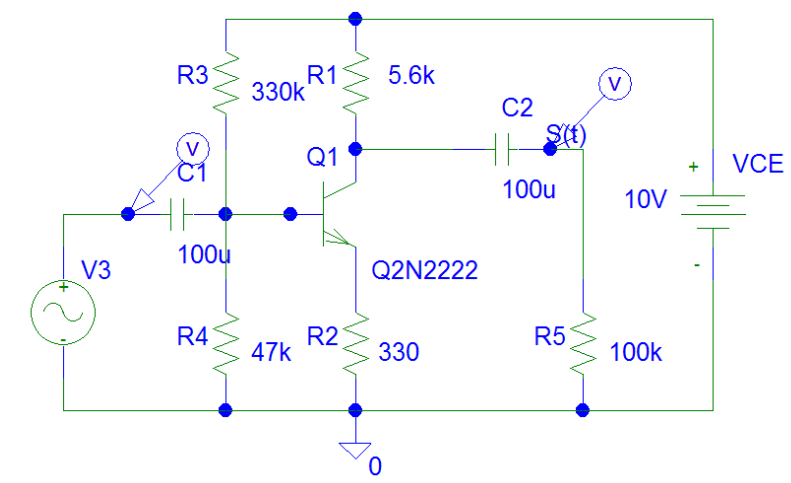


Déterminer sur les graphes les différentes zones de fonctionnement du transistor.

- On souhaite observer la variation du coefficient d'amplification β en fonction de la température pour trois valeurs de température précédentes.
- Tracer $\beta = f(I_C)$ pour $V_{CE} = 10\text{V}$ et comparer ses résultats avec les données constructeurs du 2N2222A.

b) Amplificateur :

On souhaite réaliser un amplificateur à transistor bipolaire. Il s'agit du montage émetteur commun à sortie sur collecteur :



Un peu de théorie :

- Déterminer théoriquement les valeurs des courants et tensions de polarisation.
- Le transistor étant modélisé par son quadripôle simplifié par linéarisation autour du point de fonctionnement (source de courant i_c et V_{BE}), faire le schéma en petit signaux. Calculer le gain, la fréquence de coupure et la bande passante du montage.

De la simulation pour vérifier.... :

- Réaliser les simulations correspondantes et comparer les résultats.
- On souhaite vérifier les résultats précédents avec pour entrée une sinusoïde de fréquence $f = 10 \text{ kHz}$.

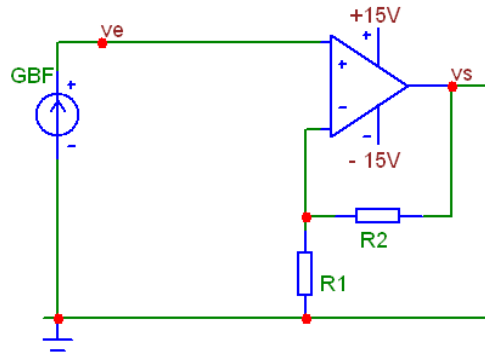
- 1) Simuler la sortie pour une amplitude d'entrée de 200mV.
- 2) Même question pour une amplitude d'entrée de 500mV.
- 3) Commenter les résultats.

TP Pspice 3

Modes de fonctionnement de l'amplificateur opérationnel

Montages à réaction négative

1. Amplificateur non inverseur



En considérant un modèle idéal d'amplificateur opérationnel, on trouve une amplification égale à : $T_0 = (R_1 + R_2) / R_1$; pour des valeurs raisonnables de T_0 , il n'y a pas de problème en basses fréquences.

On prendra par exemple :
 $R_1 = 10 \text{ k}\Omega$ et $R_2 = 100 \text{ k}\Omega$

- a) Amplification en basses fréquences.

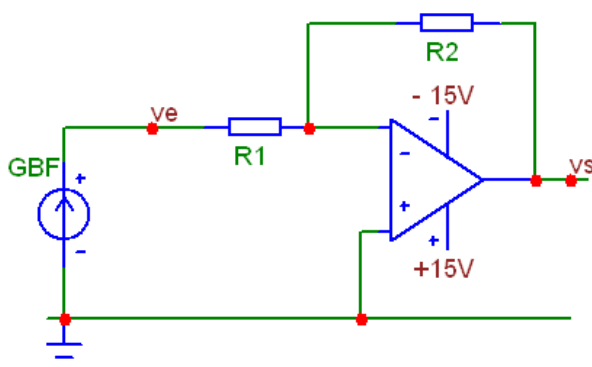
Avec V_e une tension sinusoïdale d'amplitude 1V (2Vpp) et de fréquence 1kHz, simuler le montage et relever en concordance de temps les tensions V_e et V_s ; en déduire la valeur de l'amplification en basses fréquences.

- b) Observation de la saturation.

Simuler votre montage en augmentant progressivement l'amplitude de V_e et visualiser le phénomène qui apparaît (saturation); pour un signal d'entrée d'amplitude 2V (4Vpp) relever en concordance de temps les tensions V_e et V_s . Conclure.

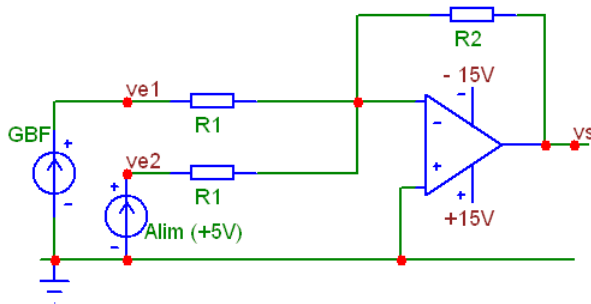
- c) Relever la caractéristique de transfert du montage $V_s = f(V_e)$.

2. Amplificateur inverseur.



En prenant un modèle idéal d'amplificateur Opérationnel, on trouve : $V_s / V_e = -(R_2 / R_1)$
 On prendra $R_1 = 10 \text{ k}\Omega$ et $R_2 = 1 \text{ M}\Omega$
 Simulez votre montage puis relevez la courbe de gain. Justifiez.

3. Sommateur.



$$v_s = -\frac{R_2}{R_1} \cdot (v_{e1} + v_{e2})$$

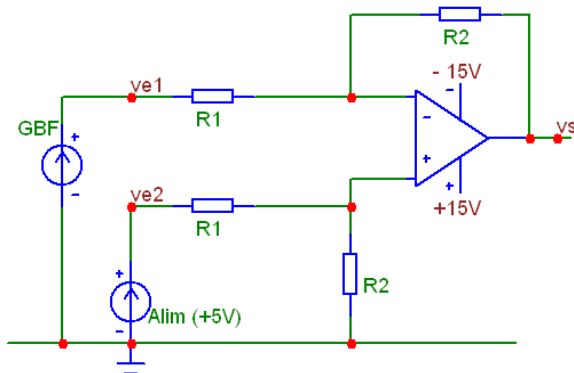
On prendra deux résistances égales $R_1 = 10k\Omega$ et une résistance $R_2 = 100k\Omega$

V_{e1} : sinus 1Vpp 1kHz

V_{e2} : tension continue 0,5V

Simulez votre montage et relevez V_{e1} V_{e2} et V_s
Expliquer

4. Soustracteur.



$$v_s = \frac{R_2}{R_1} \cdot (v_{e2} - v_{e1})$$

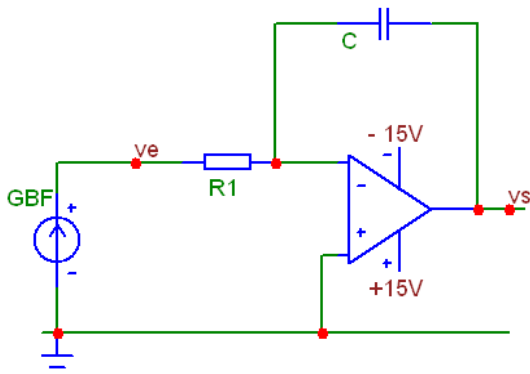
On prendra deux résistances égales $R_1 = 10k\Omega$ et deux résistances égales $R_2 = 100k\Omega$

V_{e1} : sinus 1Vpp 1kHz

V_{e2} : tension continue 0,5V

Simulez votre montage et relevez V_{e1} V_{e2} et V_s
Expliquer

5. Intégrateur.



La tension de sortie peut s'écrire :

$$v_s = -\frac{1}{R_1 \cdot C} \int_{t_0}^t v_e(t) \cdot dt$$

ce qui correspond à une fonction de transfert en régime sinusoïdal :

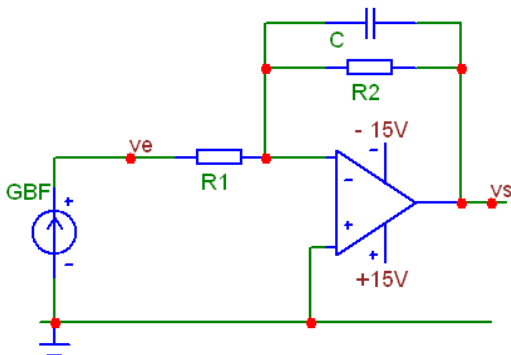
$$\underline{T} = \frac{-1}{jR_1 C \omega}$$

$$R_1 = 10 \text{ k}\Omega \quad C = 15 \text{ nF}$$

Ce montage ne peut pas fonctionner correctement s'il est utilisé seul, en effet la moindre tension continue présente en entrée (ou une tension de défaut) va entraîner l'amplificateur opérationnel en saturation. On peut aussi voir le problème à partir de la fonction de transfert : lorsque la fréquence tend vers 0 la fonction de transfert devient infiniment grande. (On peut aussi dire qu'en très basses fréquences l'impédance du condensateur devient très grande, ce qui revient donc à un montage en boucle ouverte)

Pour mettre le problème simulez-le avec une tension sinusoïdale $V_e \ll 1V_{pp} \text{ 1kHz}$ ». Expliquer.

Pour limiter l'amplification en basses fréquences on ajoute une résistance R_2 en parallèle avec le condensateur C : on obtient alors le montage « intégrateur corrigé » :



En très basses fréquences le montage se comporte comme un amplificateur inverseur,

$$\text{d'amplification } -\frac{R_2}{R_1}$$

On limitera l'amplification en basses fréquences à la valeur 100, ce qui revient à prendre $R_2 = 1M\Omega$

Le signal d'entrée V_e est maintenant un carré symétrique d'amplitude $E = 1 \text{ V}$ et de fréquence 1 kHz (supposé de valeur moyenne nulle).

Déterminer théoriquement le signal V_s . Simuler ensuite votre montage et comparer entre théorie et simulation.

En utilisant une structure inverseuse, réaliser un filtre de fonction de transfert :

$$\underline{T} = T_0 \cdot \frac{1}{1 + j \frac{f}{f_1}} \quad \text{avec } T_0 = -10 \quad \text{et } f_1 \text{ compris entre } 1,5 \text{ et } 2 \text{ kHz}$$

Simuler votre structure avec les valeurs que vous avez calculées et relever la courbe de gain. Conclure.

