

Chapitre VI: Description des modèles SPICE de différents éléments des circuits électroniques

I. Introduction

L'appel au noyau SPICE est complètement transparent pour l'utilisateur : il suffit d'entrer le circuit sous forme graphique puis de lancer la simulation. Pour effectuer les simulations, SPICE a besoin de connaître la manière dont se comportent les composants utilisés. Il utilisera donc des modèles des composants qu'il doit simuler. Ces modèles sont décrits sous formes de fichiers écrits en langage SPICE.

II. Les types de fichiers des modèles SPICE

II.1. Les formats des fichiers :

Le modèle SPICE d'un composant électronique est décrit par un fichier au format texte. Ce fichier texte est éditabile avec n'importe quel éditeur de texte : Notepad, Wordpad, ou même Edit pour DOS,...

Il existe de nombreux modèles SPICE à télécharger sur les sites des fabricants de composants électroniques. Ces modèles sont souvent enregistrés avec l'une des extensions suivantes : .CKT, .MOD, ou .MDL. Par exemple: le modèle SPICE de l'amplificateur opérationnel *UA741* est décrit dans un fichier texte nomé *UA741.CKT*.

On peut également trouver des fichiers intégrant plusieurs modèles des composants portants l'extension **.LIB**. L'avantage principal de ces fichiers multi-modèles est de limiter le nombre de fichiers. Cependant, l'inconvénient majeur est qu'ils sont plus longs à traiter lors de la simulation (temps de simulation énorme).

II.2. Le fichier .MODEL:

Un fichier .MODEL permet de personnaliser les caractéristiques d'une primitive SPICE paramétrable (comme : diode, transistor...). La façon de la déclaration de ce type de fichier est illustrée sur la figure IV.1. Ce fichier permet de définir un nouveau composant *10TQ045*, c'est une diode puisque la primitive de base utilisée est D.

```

nom du modèle créé
Commentaires (description du modèle)
Déclaration du modèle
Redéfinition des paramètres
primitive à redéfinir
à la ligne...
*Copyright (c) Symmetry Design Systems*
* Model generated on Sep 17, 96
* MODEL FORMAT: SPICE3
.MODEL 10tq045 D (IS=9.81804e-08 RS=0.00739225
+N=0.993075 EG=0.6 XTI=0.5 BV=45 IBV=0.0001
+CJO=1.42434e-09 VJ=1.5 M=0.479175 FC=0.5 TT=0
+KTF=0 AF=1 )

```

Figure IV.1 : Déclaration du fichier .Model.

Dans un modèle décrit par un fichier. Model :

- On ne peut pas changer que des valeurs de paramètres prédéfinis (ex : tension de seuil d'une diode, coefficient de température d'une résistance,...).
- On peut définir autant de modèles que l'on veut. Il suffira d'affecter à chaque modèle un nom différent comme: *IN4001*, *IN4002*,...
- On peut introduire des commandes .MODEL à l'intérieur même d'un macro-circuit .SUBCKT.

II.3. Le fichier .SUBCKT

Un fichier .SUBCKT permet de créer un macro-circuit. Ce dernier représente un circuit composé par l'association de plusieurs primitives SPICE. Pour toutes les versions de SPICE, un fichier .SUBCKT est constitué comme ceci (figure IV.2):

```

nom du modèle créé
Commentaires (description du modèle)
Déclaration du modèle
Description du modèle (langage SPICE)
Fin du modèle
connexions logicielles du modèle
*self réelle newport
*brochage
*
.SUBCKT L18R154 1 2
L0 1 4 1.50E-04
V1 4 3
B1 1 4 I=I(V1)^3 * -1.24E-05/(3*1.50E-04)
RDC 3 2 0.145
CP 1 2 5.42E-12
RP 1 2 2.10E+05
.ENDS L18R154

```

Figure IV.2 : Déclaration du fichier .SUBCKT.

Dans un fichier de type .SUBCKT :

- Un commentaire s'écrit à la suite du caractère * (il est optionnel). Ce commentaire est souvent utilisé pour définir une équivalence entre la connexion logicielle du modèle (numéro des nœuds de connexion) et le brochage du composant réel (figure IV.2).
- Le composant est décrit sous la forme d'une Netlist à l'aide du langage SPICE : Toutes les primitives utilisées sont connectées ensemble avec des nœuds. Chaque nœud étant affecté d'un numéro, l'ensemble structuré des connexions constituera la Netlist.
- La commande .ENDS permet d'informer SPICE que la description du modèle est terminée.

II.4. Utilisation des modèles SPICE :

L'objectif d'un modèle est évidemment d'être utilisé lors d'une simulation. La procédure générale pour utiliser un modèle est structurée comme suit :

- Placement des connexions électriques, puis dessin du symbole correspondant,
- création du composant grâce à la commande appropriée,
- copie du fichier-modèle dans le répertoire des modèles propre au logiciel,
- affectation d'un modèle à ce nouveau composant. En règle générale il faut éditer les propriétés du composant, spécifier le nom et le chemin d'accès du modèle, et enfin préciser le lien entre broches physiques (choisies lors de l'édition du symbole) et broches logicielles (définies dans la ligne .SUBCKT).

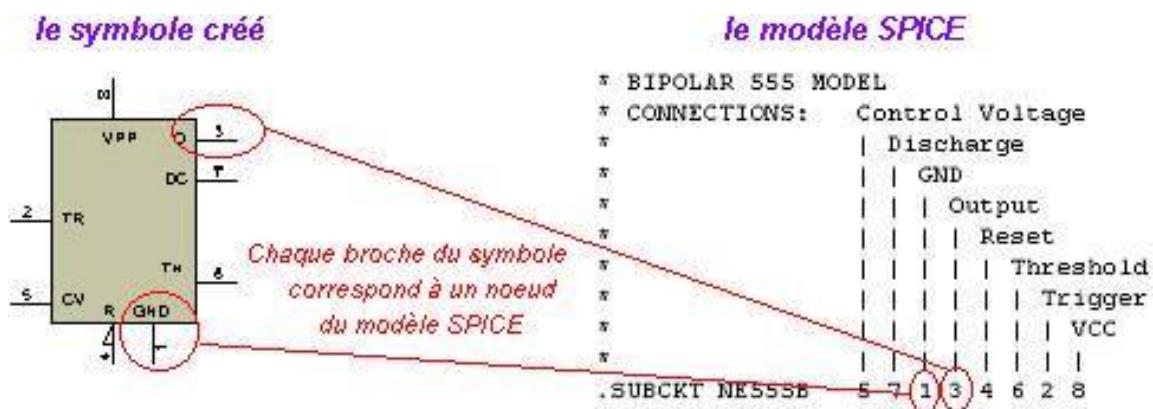


Figure IV.3 : Lien entre broches physiques et broches logicielles.

III. Syntaxe du langage SPICE

III.1. Définitions:

- Le langage SPICE est une suite de lignes de code SPICE, chacune de ces lignes place ou définit un composant.

- Pour fixer la valeur d'une primitive ou d'un paramètre, on peut utiliser indifféremment la notation réelle (ex: -5.54, .8, 1176,...), exponentielle (ex : 12E-09, 10.1E+02,...) ou enfin la notation symbolique : on écrit le coefficient multiplicateur à la suite du nombre (ex : 32.4P, 17.2MEG, .25M,...). Les coefficients disponibles dans SPICE sont :

F: femto, P: pico, N: nano, U: micro, M: milli, K: kilo, MEG: méga, G: giga, T : téra.

- Le noeud 0 est réservé par SPICE : c'est la masse . Il peut cependant être utilisé pour relier des composants à la masse du circuit.

III.2. Placement des primitives simples "de bas niveau"

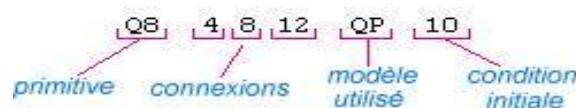
La constitution d'une ligne de code SPICE plaçant une primitive simple (non paramétrable) est décrite comme suit :



La ligne de cet exemple permet de brancher une self nommée L_0 entre les nœuds 1 et 4, cette self a pour valeur $150 \mu H$.

III.3. Placement des primitives paramétrables "de haut niveau"

Une ligne de code SPICE plaçant une primitive paramétrable est décrite comme suit :



Dans cet exemple, on placera un transistor nommé Q8 entre les noeuds 4, 8 et 12. Ce transistor sera un modèle QP (défini ailleurs grâce à une ligne .MODEL) qui aura pour valeur initiale 10.

Remarque : les primitives R (résistance) et D (diode) peuvent être définies sans nom du modèle, dans ce cas le logiciel SPICE utilisera le modèle par défaut (presque idéal).

ex : D4 3 4 ou R6 10 11 4,7K.

IV. Modélisation des primitives avec SPICE

IV.1 Définition:

La primitive est l'élément de base qui sert à la construction des circuits et des modèles électriques. Chaque primitive correspond donc à un composant du circuit. Il existe deux types de primitives :

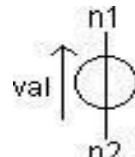
- les primitives simples : ce sont les plus basiques, on peut simplement leur affecter une valeur. Il est impossible de redéfinir les paramètres internes. *Par exemple : self, condensateur, générateurs purs ou commandés,...*

- les primitives paramétrables : on peut redéfinir leurs paramètres grâce à la commande .MODEL. *Par exemple : résistance, transistor, diode, MOS,...*

IV.2 Modélisation des sources :

a. Source de tension :

Vn n1 n2 type val



Avec :

n1 : est le nœud correspond au pôle positif de la source de tension.

n2 : est le nœud correspond au pôle négatif de la source de tension.

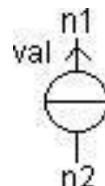
type : fixe le type de la tension : continu (type=DC) ou sinusoïdal (type=AC).

val : représente la tension, exprimée en V.

Exemple : *V2 3 5 DC 10* définit un générateur de tension V2 entre les nœuds 3 et 5, la tension est continue et vaut 10 V.

b. Source de courant :

In n1 n2 type val



Avec :

n1 : est le nœud correspond au pôle positif de la source de courant.

n2 : est le nœud correspond au pôle négatif de la source de courant.

type : fixe le type du courant : continu (type=DC) ou sinusoïdal (type=AC).

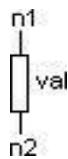
val : représente le courant, exprimé en A.

Exemple : *I4 13 23 DC 24E-03* place un générateur de courant I4 entre les nœuds 13 et 23, dont le courant est continu et vaut 24 mA.

IV.3 Modélisation des composants électriques passifs:

a. Résistance :

Rn n1 n2 val (modele)



Avec :

n1 et *n2* : sont les pattes de la résistance.

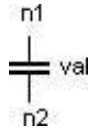
val : est la valeur de la résistance (en Ohms).

modele : donne le nom du modèle à utiliser pour la simulation.

Exemple : *R12 3 2 4.7K* positionne une résistance R_{12} de 4,7 K entre les nœuds 3 et 2.

b. Condensateur :

Cn n1 n2 val(v0)



Avec :

n1 et *n2* : sont les pattes du condensateur.

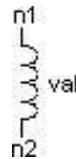
val : est la valeur de la capacité (en Farad).

v0 : est la tension initiale aux bornes du condensateur en V (facultatif).

Exemple : *C1 13 14 22N* place un condensateur de 22 nF entre les nœuds 13 et 14.

c. Inductance :

Ln n1 n2 val (i0)



Avec :

n1 et *n2* : sont les pattes de la self.

val : est la valeur de la self (en Henry).

i0 : est le courant initial dans la self en A (facultatif).

Exemple : *L10 4 7 2.2U* place une inductance de 2.2 μ H entre les nœuds 4 et 7.

IV.4 Modélisation des composants électroniques actifs:

a. Diodes :

Dn n1 n2 (modele) (OFF)



Avec

n1 : est l'anode A,

n2 : est la cathode K,

modele : donne le nom du modèle à utiliser pour la simulation,

OFF : est une condition initiale facultative permettant de bloquer au départ la diode.

Si on a redéfini une diode avec .MODEL, il y'a un modèle prédéfini par défaut :

.MODEL nom D (IC=? TEMP=? AREA=? IS=? RS=? N=? TT=? CJO=? VJ=? M=? EG=? XTI=? KF=? AF=? FC=? BV=? IBV=? TNOM=?)

nom : est le nom du modèle créé,

IS : est le courant de saturation,

RS : est la résistance passante,

CJO : est la capacité de jonction,

VJ : est le potentiel de jonction,

BV : est la tension inverse,

IVB : est le courant à la tension inverse,

TEMP : est la température de simulation (27°C par défaut),

TNOM : est la température de mesure des paramètres.

Exemple :

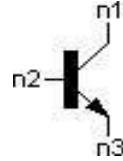
D2 3 2 IN4003

.MODEL IN4003 D (RS=.04 CJO=55E-12 IS=1.38E-09....)

Cela place une diode *IN4003*, anode au nœud 3 et cathode au nœud 2. La ligne .MODEL décrit la diode *IN4003* en question.

b. Transistors bipolaires:

Qn n1 n2 n3 modele (OFF)



Avec:

n1 : est le collecteur C,

n2 : est la base B,

n3 : est l'émetteur E,

modele : donne le nom du modèle de transistor à utiliser pour la simulation, défini avec .MODEL.

OFF : est une condition initiale facultative permettant de bloquer au départ le transistor.

.MODEL nom <NPN ou PNP>(ICVBE=? ICVCE=? AREA=? TEMP=? IS=? BF=? BR=? IKF=? IKR=? NF=? NR=? ISE=? ISC=? NE=? NC=? RE=? RC=? RB=? RBM=? IRB=? VAF=? VAR=? VJE=? VJC=? VJS=? MJC=? MJE=? MJS=? CJC=? CJE=? CJS=? TF=? TR=? XTF=? VTF=? ITF=? PTF=? XCJC=? XTB=? EG=? XTI=? FC=? KF=? AF=? TNOM=?)

Exemple:

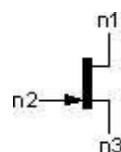
Q3 1 3 2 2N2222

.MODEL 2N2222 NPN (IS=3.108E-15 XTI=3 EG=1.11 ...)

Cela place *Q3*, un transistor *NPN* 2N2222 défini par le .MODEL. La base est le nœud 3, le collecteur est le nœud 1 et l'émetteur est le 2.

c. Transistors à effet de champ

Jn n1 n2 n3 modele (OFF)



Avec:

n1 : est le drain D,

n2 : est la grille G,

n3 : est la source S,

modele : indique le nom du modèle à utiliser pour la simulation, redéfini avec .MODEL.

OFF : est une condition initiale facultative : permet de bloquer au départ le transistor.

.MODEL nom <NJF ou PJF>(AREA=? TEMP=? VTO=? BETA=? LAMBDA=? IS=? RD=? RS=? CGS=? CGD=? PB=? FC=? B=? KF=? AF=? TNOM=?)

nom : est le nom du modèle créé,

VTO : est la tension de seuil,

BETA : est le paramètre de transconductance,

LAMBDA : est le paramètre de modulation de largeur du canal,

IS : est le courant de saturation de grille,

RD : est la résistance de drain,

RS : est la résistance de source,

CGS : est la capacité de G-S,

CGD : est la capacité de G-D,

TEMP : est la température de simulation (27°C par défaut),

TNOM : est la température de mesure des paramètres,

Exemple :

J2 1 3 2 J105

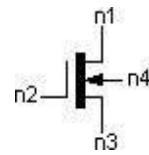
.MODEL J105 NJF (VTO=-7.25 BETA=5M LAMBDA=.035 RD=4.20E-01 ...)

Cela positionne *J2*, un transistor *JFET* canal N type *J105* défini par le .MODEL. La grille est au nœud 3, le drain est le nœud 1 et la source est le nœud 2.

d. Transistors MOS-FET :

Mn n1 n2 n3 n4 modele (OFF)

Avec:



n1 : est le drain D,

n2 : est la grille G,

n3 : est la source S,

n4 : est le substrat,

modele : donne le nom du modèle à utiliser pour la simulation, défini avec .MODEL.

OFF : est une condition initiale facultative : permet de bloquer au départ le transistor.

➤ Les *MOSFET* sont un cas particulier de SPICE : il existe plusieurs types de modèles différents (dits LEVELs).

SPICE 3F5 peut en effet simuler 7 niveaux de modèles (MOS1, MOS2, MOS3, BSIM1, BSIM2, MOS6 et BSIM3).

➤ Les modèles MOSx comportent les paramètres suivants :

.MODEL nom <NMOS ou PMOS> (LEVEL=? L=? W=? ADE=? AS=? PD=? PS=? NRD=? NRS=? OFF=? ICVDS=? ICVGS=? ICVBS=? TEMP=? VTO=? KP=? GAMMA=? PHI=? LAMBDA=? IS=? RD=? RS=? CBD=? CBS=? PB=? CGSO=? CDSO=? CGBO=? KF=? AF=? RSH=? CJ=? MJ=? CJSW=? MJSW=? JS=? TOX=? LD=? UO=? UCRIT=? UEXP=? VMAX=? NEFF=? FC=? NSUB=? NSS=? NFS=? TPG=? XJ=? XD=? ALPHA=? ETA=? DELTA=? THETA=? KAPPA=? TNOM=?)

Avec :

LEVEL=x : correspond au niveau de modèle utilisé (sur PSPICE les niveaux 1 à 3 sont acceptés)

Les paramètres débutant par *C*: décrivent les capacités parasites,

TOX : correspond à l'épaisseur de l'oxyde de grille,

VTO correspond à la tension de seuil,

Af et *Kf* correspondent aux paramètres pour l'analyse de bruit.

Exemple :

M4 2 5 6 6 NMOD1

.MODEL NMOD1 NMOS(LEVEL=1 VTO=2.474 RS=1.68 RD=0.0 ...)

Cela positionne *M4*, un *MOSFET* canal *N* de type *NMOD1* défini par .MODEL. La grille est le nœud 5, le drain est le nœud 2 et le substrat est connecté avec la source au nœud 6.

IV. Conclusion

Dans ce chapitre, nous avons présenté les différents types de fichiers des modèles SPICE ainsi que la syntaxe SPICE pour le placement d'une primitive que se soit simple ou paramétrable. Egalement, la modélisation de toutes les primitives (sources électriques, composants électriques passifs et composants électroniques actifs) est décrite en détail dans ce chapitre.