

Attributs de symboles pour gEDA/gaf

Ales V. Hvezda

1er septembre 2004*

*Ce document est écrit sous GFDL (<http://www.gnu.org/copyleft/fdl.html>)
par Ales (ahvezda@geda.seul.org). Il a été traduit par Iznogood de
<http://iznogood-factory.org>.

Contents

1	Survol	4
2	Qu'est ce qu'un attribut?	4
3	Attributs pour symboles	4
3.1	device	4
3.2	graphical	4
3.3	description	5
3.4	author	5
3.5	comment	5
3.6	pinseq	5
3.7	pinnumber	6
3.8	pintype	6
3.9	pinlabel	7
3.10	numslots	7
3.11	slotdef	7
3.12	footprint	8
3.13	documentation	8
4	Attributs de Schémas seuls	8
4.1	netname	8
4.2	source	9
5	Attributs de Symboles et de Schémas	9
5.1	refdes	9
5.2	slot	9
5.3	net	9
5.4	value	10
6	Attributs Obsolètes	10
6.1	uref	10
6.2	name	10
6.3	label	10
6.4	pin#	10

6.5 slot#	10
6.6 type	10
6.7 email	11
7 Historique de la révision du Document	11

1 Survol

Ce document décrit tous les attributs utilisés dans gEDA/gaf (GPL'd Electronic Design Automation / Gschem And Friends). Ce document est divisé en plusieurs sections : ce survol, les attributs pour symboles, les attributs pour schémas, les attributs qui peuvent apparaître dans les symboles et dans les schémas et les attributs qui sont obsolètes ou dépréciés.

Dans ce document, les noms d'attribut sont en **gras** et les exemples sont dans la police `typewriter`.

2 Qu'est ce qu'un attribut?

Les attributs dans le système gEDA/gaf ne sont rien de plus que des objets texte qui ont la forme : **name**=valeur. Les noms peuvent tout représenter tant qu'il n'y pas le signe d'égalité. Les valeurs peuvent être aussi longues que désirées tant qu'il y a quelque chose (par opposition à rien)? **nom**= (sans partie de valeur) n'est pas un attribut valide. Il ne peut pas non plus y avoir d'espaces immédiatement avant ou après le signe égal.

Les attributs peuvent être attachés à une partie du symbole. Si l'attribut transporte des informations spécifique à un objet, alors l'attribut doit être attaché directement à l'objet, sinon l'attribut doit avoir un comportement libre ou flottant. Les attributs au comportement libre existent simplement dans le fichier symbole comme un objet texte qui prend la forme **name**=valeur.

3 Attributs pour symboles

3.1 device

device= est le nom de composant d'un symbole et est nécessaire pour gnetlist.

device= doit être placé quelque part dans le symbole et est rendu invisible. C'est un attribut au comportement libre ou flottant. Si l'objet est graphique alors **device**= doit être positionné à none (**device**=none) et attache un attribut **graphical**= (3.2). Ne perturbez pas cet attribut avec un simple label de texte qui est le nom du composant. Ne mettez pas d'espaces dans le nom du composant ; quelques programmes qui n'aiment pas les espaces dans les spécifications des composants. Le nom de composant est généralement en majuscule.

Exemples : `device=7400` `device=CONNECTOR_10` `device=NPN_TRANSISTOR`

3.2 graphical

Les symboles qui n'ont pas de signification électrique ou de circuit ont besoin d'un attribut **graphical**=1. Les symboles comme les boîtes de titres sont pure-

ment graphiques. Tout symbole qui possède un **graphical=1** est ignoré par gnetlist.

graphical=1 doit exister quelque part dans le symbole et être rendu invisible. C'est un attribut au comportement libre ou flottant. N'oubliez pas d'initialiser **device=none** (3.1).

Exemple : `graphical=1`

3.3 description

L'attribut **description** fournit une simple ligne de description de ce qu'est sensé représenter un symbole.

Exemple : `description=4 portes NAND avec 2 entrées`

3.4 author

L'attribut **author** identifie le nom de l'auteur de ce symbole et son adresse courriel. Cet attribut est optionnel mais il est sympa de connaître l'auteur d'un symbole. Il sert aussi de point de contact dans le cas o il y aurait des questions sur l'efficacité du symbole. Cet attribut possède une forme libre et peut aussi inclure le nom des personnes qui ont modifié le symbole de même que plusieurs adresses de courriel. Il est sûrement aussi une bonne idée d'altérer l'adresse de courriel pour éviter le spam.

Exemple : `author=Ales Hvezda ahvezdaATgeda.seul.org`

3.5 comment

L'attribut **comment** peut tout contenir. Cet attribut peut contenir toute information additionnelle qui peut ne pas être contenue dans un autre attribut. Il peut y avoir plusieurs instances de cet attribut.

Exemple : `comment=Ceci est un commentaire dans un symbole`

3.6 pinseq

Cet attribut est utilisé pour pour donner à chaque broche un nombre unique ou une séquence. Toutes les broches doivent avoir un attribut **pinseq=#** attaché à l'objet briche. Cet attribut doit être caché. Cet attribut est utilisé extensivement par gschem et gnetlist.

gnetlist sortira les broches dans l'ordre de croissance de leur numérotation. La séquence débute à 1 et doit augmenter sans sauter aucun nombre. Cet attribut n'est pas le numéro de broche (i.e. le numéro de la broche du composant,

in	Input
out	Output
io	Input/Output
oc	Open collector
oe	Open emitter
pas	Passive
tp	Totem pole
tri	Tristate (high impedance)
clk	Clock
pwr	Power/Ground

Table 1: **pintype** values

comme GND est 7 sur un TTL). Pour les numéros de broches, voyez l'attribut **pinnumber** (3.7).

Exemples : `pinseq=1 pinseq=2 pinseq=3`

Cet attribut remplace l'attribut obsolète `pin#=#`.

3.7 pinnumber

Cet attribut est le numéro de broche (i.e. comme GND est 7 sur un 74 TTL). Toutes les broches doivent avoir un attribut **pinnumber=#** attaché à l'objet broche.

Vous pouvez avoir des nombres ou des lettres pour la valeur. Cet attribut doit être visible avec la valeur seule de visible. Vous avez aussi besoin d'un attribut **pinseq** (3.6).

Exemples : `pinnumber=1 pinnumber=13 pinnumber=A0`

Cet attribut remplace l'attribut obsolète `pin#=#`.

3.8 pintype

Chaque broche doit avoir un attribut **pintype=valeur** qui lui est rattaché et doit être caché. La table 1 montre les valeurs valides pour cet attribut.

Cet attribut n'est pas utilisé extensivement dans la bibliothèque de symboles mais il sera utilisé pour le DRC et le netlisting.

Exemples : `pintype=clk pintype=in pintype=pas`

3.9 pinlabel

Cet attribut labelise un objet broche. Cet attribut est utilisé en premier par gnetlist pour supporter les conceptions hiérarchiques.

Cet attribut doit être attaché à la broche et rester visible. Veuillez rendre, SVP, cet attribut vert (au lieu de la couleur jaune par défaut).

Exemples : `pinlabel=A0 pinlabel=DATA1 pinlabel=CLK`

3.10 numslots

Si un composant possède des slots multiples dans un boîtier physique (tel qu'un 7400 (NAND) qui possède 4 NANDs par boîtier) alors vous avez besoin d'un attribut `numslots=#`. Le # est le nombre de slots qui sont dans le composant physique. `numslots=#` doit exister quelque part dans le symbole et doit être rendu invisible. C'est un attribut au comportement libre ou flottant. Si le symbole n'a pas besoin du slotting, alors mettez `numslots=0` dans le fichier symbole.

Exemple : `numslots=4`

3.11 slotdef

Si un composant possède des slots multiples dans un boîtier physique alors vous devez attacher un `slotdef=slotnumber :#, #, #...` pour chaque composant dans le boîtier physique.

Le slotnumber correspond au numéro de slot. La colonne après le numéro de slot est nécessaire. Par exemple, si un composant possède 4 slots alors il y aura des attributs `slotdef=1 :...`, `slotdef=2 :...`, `slotdef=3 :...`, et `slotdef=4 :...` quelquepart dans le symbole et doit être invisible. C'est un attribut au comportement libre ou flottant.

Le # possède une correspondance une-à-une pour l'attribut `pinseq` et spécifie quel `pinnumber=#` est utilisé pendant l'affichage (gschem) ou le netlisting (gnetlist).

Il est aussi recommandé que tous les symboles qui ont des slots aient un attribut `slot=1` (5.2) attaché de la même manière que l'attribut `device=` (3.1).

Voyez 7400-1.sym pour avoir un exemple concret.

Exemples : `slotdef=1 :1,2,3 slotdef=2 :4,5,6 slotdef=3 :7,8,9`

Cet attribut remplace l'attribut obsolète `slot#=#`.

3.12 footprint

footprint=package_name doit exister quelque part dans le symbole et être rendu invisible. Cet attribut est utilisé par gnetlist et est pour le boîtier de PCB.

Attachez cet attribut juste comme l'attribut **device**= (3.1). C'est un attribut au comportement libre ou flottant.

package_name est l'empreinte pcb ou le type de boîtier tel DIP14 ou DIP40. Bien que cet attribut soit dépendant du boîtier pcb, les conventions gEDA/gaf existent pour rendre cet attribut aussi portable que possible, facilitant la collaboration et le partage entre les utilisateurs. Voyez les *Conventions de nommage des Empreintes* dans le *Document de Création de Symboles de gEDA/gaf*.

Si le symbole ne possède pas d'empreinte alors la valeur de **footprint**= doit être mise à none. Si l'empreinte doit être surchargée dans un schéma alors la valeur de **footprint**= doit être mise à none. Si l'empreinte n'est pas connue alors la valeur de **footprint**= doit être mise à unknown.

3.13 documentation

documentation=documentation_locator peut exister quelque part dans le symbole et doit être rendu invisible. Cet attribut est utilisé par gschemdoc pour trouver la documentation adaptée au symbole ou plutôt le composant ou le boîtier associé au symbole.

Attachez cet attribut juste comme l'attribut **device**= (3.1). C'est un attribut au comportement libre ou flottant.

documentation_locator est soit le nom de base du nom de fichier pour la documentation ou c'est l'URL internet complète (Uniform Resource Locator). Si c'est le nom de fichier, une tentative sera faite pour le chercher dans le répertoire partagé gEDA local nommé documentation.

Exemple de nom de fichier : `documentation=sn74ls00.pdf`

Exemple d'URL : `documentation=http://www-s.ti.com/sc/ds/sn74ls00.pdf`

4 Attributs de Schémas seuls

4.1 netname

Cet attribut doit être attaché à un objet réseau pour donner un nom. Les multiples noms de réseau pour connecter les segments de liaisons est découragé. Tous les réseaux qui ont la même valeur sont considérés comme connectés électriquement. Cet attribut n'est pas valide à l'intérieur d'un symbole (car vous ne pouvez avoir de liaisons à l'intérieur de symboles).

Exemples : `netname=DATA0_H netname=CLK_L`

4.2 source

L'attribut **source=** est utilisé pour spécifier qu'un symbole possède un schéma sous-jacent. Cet attribut est attaché directement à un composant.

Cet attribut devrait être seulement attaché au composant instancié dans le schéma. Attachez l'attribut à un composant et spécifiez le nom de fichier (pas le chemin) du schéma sous-jacent (comme `block.sch`) pour la valeur. Le schéma spécifié doit être dans le chemin de la bibliothèque source. Cet attribut peut être attaché de multiples fois avec différentes valeurs, ce qui signifie basiquement qu'il existe des schémas sous-jacents multiples.

Exemples : `source=underlying.sch source=memory.sch`

5 Attributs de Symboles et de Schémas

5.1 refdes

Cet attribut est utilisé pour spécifier le désigneur de références d'un composant instancié particulier. Il doit être sur TOUS les composants qui ont une signification électrique. Cet attribut peut être aussi à l'intérieur d'un symbole (il sera promu, i.e. attaché à l'extérieur d'un symbole, s'il est visible) pour fournir une valeur `refdes` par défaut (tel que `U?`).

Exemples : `refdes=U1 refdes=R10 refdes=CONN1`

5.2 slot

Cet attribut est utilisé pour spécifier le slot d'un composant qui en possède. Il doit être attaché à un composant instancié. Cet attribut peut aussi être à l'intérieur d'un symbole (il sera promu, i.e. attaché à l'extérieur d'un symbole, s'il est visible) pour fournir un slot par défaut.

5.3 net

L'attribut **net=** est utilisé pour créer power/ground et des liaisons arbitraires. Veuillez aller voir, SVP, le Mini-HOWTO `netattrib` pour plus d'informations. Lorsqu'un attribut est à l'intérieur d'un symbole, il est utilisé pour créer des liaisons. Lorsque cet attribut est attaché à un composant instancié (dans un schéma) alors **net=** peut aussi être utilisé pour créer de nouvelles liaisons et peut être utilisé pour surcharger les liaisons existantes.

5.4 value

Utilisé principalement pour le netlister de spice afin de spécifier la valeur des éléments divers. Aucune traduction n'est faite dessus et elle est placée telle quelle dans la netlist.

Exemples : `value=1K` `value=10V`

6 Attributs Obsolètes

6.1 uref

L'attribut **uref=** est obsolète et ne peut pas être utilisé. Il a été utilisé pour fournir la même information que **refdes** (5.1).

6.2 name

L'attribut **name=** ne doit pas être attaché ou apparaître dans le symbole. C'est considéré comme ambigu. **name=** n'a jamais été utilisé par gEDA/gaf.

6.3 label

L'attribut **label=** est obsolète et ne peut être utilisé. Il a été utilisé pour donner aux liaisons des noms/labels et pour les labels des broches. L'attribut de remplacement pour ceci a été le **netname** (4.1) et **pinlabel** (3.9) respectivement.

6.4 pin#

L'attribut **pin#=#** est obsolète et ne peut pas être utilisé. Il a été utilisé pour fournir une séquence et une information de nombre pour les broches. Les attributs de remplacement pour ceci sont **pinseq** (3.6) et **pinnumber** (3.7).

6.5 slot#

L'attribut **slot#=#** est obsolète et ne peut pas être utilisé. Il a été utilisé pour fournir une information sur les slots des composants. L'attribut de remplacement est **slotdef** (3.11).

6.6 type

L'attribut **type=** est obsolète et ne doit pas être utilisé. Il a été utilisé pour fournir des informations de type sur les broches. L'attribut de remplacement pour ceci est **pintype** (3.8).

6.7 email

L'information de cet attribut a été mélangé avec **author** (3.4).

7 Historique de la révision du Document

July 14th, 2002	Created attributes.tex from attributes.txt.
July 14th, 2002	Updated doc to be in sync with post-20020527.
August 25th, 2002	Added obsolete type= attribute.
September 14, 2002	Added description= attribute. Minor fixes
October 7, 2002	Added doc= attribute ; Egil Kvaleberg.
February 11, 2002	Added reference to footprint conventions.
February 23, 2002	Added author=, email=, and comment= attributes.