

**Gilles Brocard**

# **LTspice XVII**

## **Manuel de référence**

**Commandes et applications**

**DUNOD**

Toutes les marques citées dans cet ouvrage sont des marques déposées  
par leurs propriétaires respectifs

Conception graphique de la couverture : Pierre-André Gualino  
Illustrations de couverture : BigTunaOnline / Shutterstock.com

Le pictogramme qui figure ci-contre mérite une explication. Son objet est d'alerter le lecteur sur la menace que représente pour l'avenir de l'écrit, particulièrement dans le domaine de l'édition technique et universitaire, le développement massif du photocopillage.

Le Code de la propriété intellectuelle du 1<sup>er</sup> juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique s'est généralisée dans les établissements

d'enseignement supérieur, provoquant une baisse brutale des achats de livres et de revues, au point que la possibilité même pour

les auteurs de créer des œuvres nouvelles et de les faire éditer correctement est aujourd'hui menacée.

Nous rappelons donc que toute reproduction, partielle ou totale, de la présente publication est interdite sans autorisation de l'auteur, de son éditeur ou du Centre français d'exploitation du droit de copie (CFC, 20, rue des Grands-Augustins, 75006 Paris).



© Dunod, 2020

11 rue Paul Bert, 92240 Malakoff

[www.dunod.com](http://www.dunod.com)

ISBN 978-2-10-076893-6

Le Code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5, 2<sup>o</sup> et 3<sup>o</sup> a), d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite » (art. L. 122-4).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

# Table des matières

<b>Pourquoi un troisième ouvrage consacré à LTspice XVII ?</b>	<b>XIII</b>
<b>Préface</b>	<b>XVII</b>
<b>1 Les nouveautés de LTspice XVII</b>	<b>1</b>
1.1 Les principaux apports de la nouvelle version LTspice XVII	2
1.2 Création et attribution de nouveaux répertoires	3
1.3 LTspice XVII, dialogue amélioré avec Windows	5
1.4 Nouveautés concernant l'interface graphique de saisie	7
1.5 Accès plus facile au diagramme de l'œil	11
1.6 Machine d'état arbitraire	11
1.7 Petit retour sur LTspice IV	14
<b>2 Fonctionnement et premier exemple</b>	<b>15</b>
2.1 Première utilisation de LTspice XVII	15
2.2 Fonctionnement de LTspice XVII	16
2.3 Les menus de la phase de démarrage de LTspice XVII	16
2.4 Un exemple détaillé étape par étape	20
<b>3 Éditeur graphique de schémas</b>	<b>47</b>
3.1 Les commandes de LTspice XVII	47
3.2 L'éditeur graphique de schémas	47
3.3 Bases de données de composants	63
3.4 Saisie d'un nouveau schéma	65
3.5 Rappel des règles d'utilisation de l'éditeur de schémas	75
<b>4 Syntaxe et éditeur de composants</b>	<b>77</b>
4.1 Règles syntaxiques générales sous LTspice XVII	77
4.2 Éditeurs de valeurs de composants	80

4.3 Procédures pour accéder aux modèles usuels ou complexes des composants	83
4.4 Attribution des lignes de l'éditeur d'attributs	89
4.5 Affichage des attributs d'un composant utilisant deux modèles	90
<b>5 Éditeur de symboles et liens hiérarchiques</b>	<b>95</b>
5.1 Menu éditeur de symboles	95
5.2 Première étape, dessinez le corps du symbole	98
5.3 Deuxième étape, ajouter des bornes de raccordement	98
5.4 Troisième étape, ajouter ou modifier des attributs	99
5.5 Les appels possibles à partir d'un symbole	101
5.6 Attributs visibles attachés au symbole	102
5.7 Génération automatique de symboles à partir d'une partie de schéma	103
5.8 Génération automatique de symboles à partir d'une Netlist	103
5.9 Liens hiérarchiques avec LTspice XVII	107
5.10 Règles d'utilisation de la hiérarchie	108
5.11 Règles à respecter pour la construction hiérarchique	109
5.12 Les commandes du menu Hierarchy	111
5.13 Exemple : déroulement d'une construction hiérarchique simple à deux niveaux	111
5.14 Exportation du répertoire Hierarchy	113
5.15 Interactivité entre le niveau bas et le niveau haut	114
<b>6 Éditeur de Netlists</b>	<b>115</b>
6.1 Origine historique des Netlists	115
6.2 La Netlist, un passage obligé	115
6.3 Structure, syntaxe et conventions des Netlists	116
6.4 Exemple de Netlist	116
6.5 Menus de l'éditeur de Netlists	117
6.6 La rédaction d'une Netlist	118
6.7 Syntaxe du fichier Netlist .cir, .net ou .sp	119
6.8 Comment ouvrir l'éditeur de Netlists à partir d'un schéma	120
6.9 Exécution d'une Netlist	122

6.10	Exportation d'une Netlist correspondant à un schéma	123
6.11	Commandes systèmes utilisées dans les Netlists	124
<b>7</b>	<b>Éditeur graphique et sortie numérique</b>	<b>125</b>
7.1	Affichage des résultats du calcul de la simulation	125
7.2	Comment sélectionner des points de mesures sur votre schéma ?	125
7.3	Comment afficher une mesure sur l'oscilloscope virtuel ?	126
7.4	Utilisation des menus	129
7.5	Choisir les mesures à afficher	136
7.6	Ajouter une trace ou un écran supplémentaire	137
7.7	Fonctions Zoom	139
7.8	Opérations mathématiques dans l'oscilloscope virtuel	139
7.9	Demander le calcul d'une expression mathématique	140
7.10	Fonctions définies par l'utilisateur	141
7.11	Modifications des échelles des axes	142
7.12	Utilisation de l'oscilloscope virtuel en mode X-Y	144
7.13	Le menu contextuel et les échelles	145
7.14	Autres paramétrages des échelles	146
7.15	Gestion de l'oscilloscope virtuel multitraces	148
7.16	Informations concernant les traces de l'oscilloscope virtuel	149
7.17	Autres aménagements des traces de l'oscilloscope virtuel	150
7.18	Contrôles des couleurs de l'oscilloscope virtuel	151
7.19	Deux curseurs de mesures	152
7.20	Affichage des coordonnées dans le bandeau bas	155
7.21	Enregistrer la configuration de l'oscilloscope virtuel	156
7.22	Accélération du chargement des fichiers	156
7.23	Mémoire RAM et espace d'adressage	158
7.24	Présentation du fichier tampon SPICE Error Log	158
7.25	La commande .four	158
7.26	La combinaison des commandes .step et .meas	160
7.27	Réglage d'un asservissement	168
7.28	Bien choisir la résolution de la boucle .step	173

<b>8 Les commandes</b>	<b>177</b>
8.1 Définition d'une commande	177
8.2 .options paramètres modifiant l'exécution d'une simulation	184
8.3 .ic – fixer les conditions initiales pour une simulation temporelle	188
8.4 .savebias – enregistrer un point de fonctionnement DC	190
8.5 .loadbias – charger un point de fonctionnement DC	191
8.6 .net – calcul des paramètres d'un réseau avec une simulation AC	191
8.7 .nodeset – conditions initiales pour l'analyse DC	192
<b>9 Les six simulations principales</b>	<b>193</b>
9.1 Présentation des six simulations principales	193
9.2 Critères de choix concernant les simulations	197
9.3 .op – simulation d'un point de polarisation continu	200
9.4 .dc – simulation continue avec balayage (une à trois sources)	202
9.5 .tf – simulation de la fonction de transfert (gain, impédance d'entrée et de sortie)	203
9.6 .ac – simulation d'un signal AC autour d'un point de polarisation	204
9.7 .noise – simulation de bruit	207
9.8 .tran – simulation temporelle (non linéaire)	209
9.9 Paramétrages de la simulation temporelle .tran	211
<b>10 Trois analyses dédiées</b>	<b>223</b>
10.1 Présentation de ces trois analyses	223
10.2 .temp – simulation d'un balayage de température	223
10.3 .four – éditer les harmoniques sous forme numérique	226
10.4 Comment fonctionne l'analyse FFT	227
<b>11 Les composants passifs et actifs</b>	<b>241</b>
11.1 Préambules à l'utilisation des paramètres d'un modèle de composant	241
11.2 Les composants passifs	241
11.3 Le modèle générique de la résistance R	242
11.4 Le modèle générique du condensateur C	244
11.5 Le modèle générique de l'inducteur L	246
11.6 Les composants actifs	246

11.7 Le modèle générique de la diode D	247
11.8 Le modèle générique du transistor bipolaire Q	251
11.9 Le modèle générique du transistor JFET J	260
11.10 Le modèle générique du transistor MOSFET M	262
11.11 Le modèle générique du transistor MOSFET à double diffusion verticale	266
11.12 Le modèle générique du transistor MESFET Z	269
11.13 Le modèle générique du transistor IGBT	271
11.14 Les modèles de transistor unijonction, phototransistor, thyristor et TRIAC	272
11.15 Coefficient multiplicateur de composant en parallèle ou en série	273
<b>12 Les selfs, les transformateurs et les inductions mutuelles</b>	<b>277</b>
12.1 Le modèle générique de l'inducteur L (sans saturation)	277
12.2 Self et inducteur à air sans circuit magnétique saturable	277
12.3 Le sous-modèle générique d'inducteur L	280
12.4 Le modèle CHAN d'inducteur L	282
12.5 Le modèle de transformateur	286
12.6 Transformateur avec saturation et hystérésis du matériau magnétique	293
<b>13 Importer un composant avec Internet ou le créer</b>	<b>297</b>
13.1 Les composants modélisés compatibles SPICE	297
13.2 Les différents sites Internet	299
13.3 Le site NXP	299
13.4 Le site de téléchargement de l'IUT de Cachan	306
13.5 Le site de téléchargement de Texas Instruments	307
13.6 Les sites de téléchargement	309
13.7 Quelles différences entre modèle et sous-circuit ?	314
13.8 Comment modifier un modèle ?	319
13.9 Comment ajouter un modèle à LTspice ?	320
13.10 Comment ajouter un sous-circuit ?	326
13.11 Création d'un sous-circuit étape par étape	337

13.12 Utilisation des modèles et des sous-circuits	344
13.13 Commande .lib et .inc	344
<b>14 Les commandes .func .meas .param .step et accolade</b>	<b>345</b>
14.1 Les commandes	345
14.2 La commande .func	346
14.3 La commande .meas	349
14.4 Les commandes accolades .param .ako .step	362
14.5 Les accolades	362
14.6 La commande .param et la fonction param	365
14.7 La commande .step	367
14.8 Questions fréquentes concernant la commande .step	374
<b>15 Les commandes ako: .four .wave .model .subckt .include .lib .ic .save</b>	<b>389</b>
15.1 La fonction ako:	389
15.2 Commande .four	396
15.3 Commande .wave	396
15.4 Commande .model	398
15.5 Commande .subckt	400
15.6 Commandes .include et .lib	400
15.7 Commande .ic	401
15.8 Commande .save	402
<b>16 La commande .options et le panneau de contrôle</b>	<b>405</b>
16.1 Commande .options	405
16.2 Les dix onglets du panneau de contrôle	413
16.3 Raccourcis clavier	431
16.4 Onglet Choix de couleur : Color Preferences	433
16.5 Ouverture d'un éditeur	435
<b>17 Éditeur des sources de tension et de courant</b>	<b>437</b>
17.1 Deux types de sources et deux éditeurs	437
17.2 Deux grands types de sources, dépendantes ou indépendantes	438

17.3 Toute simulation nécessite la présence d'une source indépendante	438
17.4 Comment placer une source dans un schéma	440
17.5 V – source de tension indépendante (STNC)	442
17.6 I – source de courant indépendante (SCNC)	444
17.7 Load – charge active indépendante (SCCV)	447
17.8 L'éditeur de sources indépendantes	447
17.9 Sources dépendantes	473
17.10 E – source de tension dépendante en tension (STCT)	473
17.11 F – source de courant commandée en courant (SCCC)	477
17.12 G – source de courant commandée en tension (SCCT)	480
17.13 H – source de tension commandée en courant (STCC)	482
17.14 Source B de tension ou de courant	483
17.15 B – source de tension arbitraire, non linéaire (STC)	483
17.16 B – sources de courant arbitraire, non linéaire (SCC)	485
17.17 L'éditeur d'attributs pour les sources dépendantes	486
<b>18 La logique et les fonctions annexes</b>	<b>489</b>
18.1 Caractéristiques communes aux circuits logiques	489
18.2 Les portes logiques standards	490
18.3 Les portes logiques avec triggers de Schmitt	490
18.4 Les bascules logiques	491
18.5 PhaseDet : comparateur de phase avec sortie en courant	492
18.6 SampleHold : échantillonneur bloqueur	495
18.7 Modulate et Modulate2 : modulateur de fréquence et d'amplitude	496
18.8 Enregistrer une polarisation DC avec .savebias	499
18.9 Changer un point de fonctionnement DC avec .loadbias	500
18.10 Le diagramme de l'œil, la fonction baudrate	501
18.11 Paramétrer l'axe des abscisses .ac list	506
18.12 Le cryptage	510
18.13 Les options pouvant être ajoutées lors du lancement de LTspice	512
18.14 Mots réservés	513
18.15 Précision des calculs avec LTspice	514

18.16	Tableau d'attribution du modèle générique de composants	515
18.17	Interrupteurs commandés S et W	516
18.18	S – interrupteur commandé par une tension (deux modèles)	516
18.19	W – interrupteur commandé par un courant (un modèle)	519
18.20	O – ligne de transmission avec perte (un modèle)	523
18.21	T – ligne de transmission sans perte (un modèle)	524
18.22	U – ligne de transmission RC (un modèle)	524
<b>19</b>	<b>Monte-Carlo et Worst Case</b>	<b>527</b>
19.1	Présentation de Monte-Carlo	527
19.2	Présentation de Worst Case	538
<b>20</b>	<b>SOAtherm, modèle thermique pour MOSFET avec dissipateur</b>	<b>573</b>
20.1	SOAtherm pour LTspice XVII	573
20.2	Comment utiliser le modèle SOAtherm avec LTspice XVII ?	574
20.3	Les modèles SOAtherm	574
20.4	Comment faire fonctionner le modèle SOAtherm avec LTspice XVII	575
20.5	Le modèle SOAtherm fonctionne-t-il avec tous les MOSFET de LTspice XVII ?	575
20.6	Que se passe-t-il si on appelle un MOS non compatible avec SOAtherm ?	576
20.7	Comment reconnaître un MOS compatible SOAtherm ?	576
20.8	Comment accède-t-on aux bornes ?	578
20.9	Comment utiliser le deuxième modèle SOAtherm-HeatSink ?	584
20.10	Quelle est la réalité du flux thermique ?	588
20.11	Comment fonctionnent les modèles SOAtherm ?	589
20.12	Influence de la constante de temps ?	590
20.13	Caractéristiques du dissipateur ?	592
20.14	Comment régler l'équilibrage thermique de MOS en parallèle ?	594
20.15	Conclusion générale	599

<b>21 Quelques exemples</b>	<b>601</b>
21.1 Différence entre simulations LTspice XVII et réalité	601
21.2 La simulation du bruit .noise	602
21.3 Une fonction PWL pour créer un générateur de signaux arbitraires en tension ou en courant	605
21.4 L'oscilloscope virtuel : changer le paramétrage des axes des traces de l'oscilloscope	606
21.5 La stabilité des amplificateurs opérationnels avec une simulation AC (ou comment jongler avec les pôles et les zéros)	608
21.6 Comment améliorer la vitesse d'exécution de LTspice XVII	615
21.7 Un gain de temps à la mise au point, la détection synchrone	623
<b>Annexe</b>	<b>631</b>
<b>Index</b>	<b>635</b>



# Pourquoi un troisième ouvrage consacré à LTspice XVII ?

LTspice IV a vu le jour fin novembre 2008, avec le succès que l'on connaît, puis en 2016 Mike Engelhardt achève LTspice XVII, il a été mis en ligne par Linear Technology en septembre 2016.

Le premier livre, *Le simulateur LTspice IV, manuel, méthodes et applications*, suivi de *LTspice IV, nouvelles commandes*, ont été écrits et dédiés à la version IV. Le moment de faire le point sur toutes les nouveautés était arrivé et la demande des utilisateurs s'est fait sentir de plus en plus pressante. Ce troisième tome rassemble donc tout ce qu'il faut savoir pour bien utiliser LTspice XVII et surtout pour découvrir les nouvelles commandes dont certaines apportent un réel confort d'utilisation et un accroissement de la puissance (SOAtherm par exemple).

Durant de nombreuses années, les mises à jour se sont succédé au rythme d'une par mois, mais le nombre d'ajouts, dont certains importants, se sont stabilisés début 2019. Durant la même période, un événement a également influé le cours de la vie du logiciel, en 2017 Linear Technologie a été absorbée par **Analog Devices**. C'est sur le site d'**Analog Devices** qu'aujourd'hui vous devrez vous rendre pour télécharger la dernière version de LTspice XVII. Soyons honnête, LTspice XVII n'a pas souffert de cette opération, bien au contraire, la qualité du site, et le nombre d'éléments téléchargeables concernant LTspice, ainsi que le nombre de modèles de composants incorporés à LTspice XVII ont très notablement augmenté et le principe de la gratuité du logiciel a fort heureusement été conservé par Analog Devices.

Plusieurs commandes importantes ont été ajoutées ou réaménagées pour faciliter l'utilisation de LTspice XVII (**.machine** ou **SOAtherm**), d'autres ont été nettement améliorées (les éditeurs de **.meas** et **.step** en particulier) et des applications toujours plus puissantes sont apparues, s'appuyant sur des méthodes performantes (**Monte Carlo** et **Worst Case**). Le blog dédié à LTspice depuis 2011 traite une soixantaine de sujets ayant trait à LTspice, à l'aide de tutoriels PDF, de vidéos et d'exemples téléchargeables.

Depuis quelques années, de nombreuses personnes, en charge de la conception de schémas électroniques, ont adopté LTspice XVII pour eux-mêmes ou pour leurs équipes de designers. Ce choix leur a apporté un gain de temps important, a réduit

leur budget de création de schéma et surtout de mise au point. Cela a également amélioré la qualité des circuits produits et a augmenté l'efficacité globale de production.

De nombreuses questions, souvent les mêmes, ont été posées à l'auteur/formateur au cours des sessions de formation en entreprise. Ce troisième tome apporte des réponses documentées et détaillées aux questions les plus fréquentes sur toutes les nouveautés de LTspice XVII. Près de sept cents illustrations et cinq cents exemples, tous téléchargeables sur le site de Dunod, viennent enrichir ces différents sujets.

Toutes ces nouveautés intégrant LTspice XVII font l'objet de chapitres dédiés qui constituent un ouvrage complémentaire des deux tomes précédents. Ce livre est organisé en vingt et un chapitres regroupant les sujets par thème. Par contre, certains sujets importants sont présentés de plusieurs manières pour mieux répondre aux nombreuses questions qu'ils ont suscitées.

Le **chapitre 1** présente succinctement les ajouts et suppressions de la dernière version de LTspice XVII.

Le **chapitre 2** présente la procédure de création d'un montage à l'aide de l'éditeur graphique de schémas et le lancement des commandes de base.

Le **chapitre 3** présente en détail toutes les commandes de l'éditeur graphique de schémas.

Le **chapitre 4** développe l'utilisation de la syntaxe propre à LTspice et présente l'éditeur de composants.

Le **chapitre 5** traite de l'édition des symboles et des liens hiérarchiques dont la maîtrise fait gagner beaucoup de temps lors de la création de schémas.

Le **chapitre 6** traite de l'éditeur de Netlist utile à ceux qui souhaitent maîtriser les fondements de LTspice XVII.

Le **chapitre 7** présente l'affichage des mesures avec l'oscilloscope virtuel dont la maîtrise ouvre la voie à des applications extrêmement puissantes (réglage des asservissement, circuit HF, etc.). Il présente l'affichage graphique de deuxième niveau, réalisé à partir des données contenues dans le fichier tampon **Spice Error Log**.

Le **chapitres 8, 9 et 10** traitent des sujets essentiels qui sont les neuf simulations et leurs paramétrages.

Les **chapitres 11 et 12** présentent les différentes familles de composants passifs et actifs fournis au moment de l'installation de LTspice XVII.

Le **chapitre 13** traite de la création et de l'importation de modèles et de sous-circuits à partir des sites Internet des fabricants de composants électroniques.

Les **chapitres 14 et 15** détaillent l'utilisation et le paramétrage des commandes **.func**, **.meas**, accolades, **.param**, **ako:**, **.step**, **.four**, **.wave**, **.model**, **.subckt**, **.inc**, **.lib**, **.ic** et **.save**.

Le **chapitre 16** présente la commande **.options** et son complément, le **panneau de contrôle**.

Le **chapitre 17** présente les différentes sortes de sources E, F, G et H pour la tension et le courant ainsi que les sources Bv et Bi.

## Pourquoi un troisième ouvrage consacré à LTspice XVII ?

Le **chapitre 18** rassemble toutes les fonctions annexes, l'affichage du diagramme de l'œil et les mots réservés.

Le **chapitre 19** détaille l'utilisation pratique de fonction Monte-Carlo et Worst Case pour traiter un schéma avec un nombre élevé de composants.

Le **chapitre 20** montre comment exploiter le modèle **SOAtherm** avec LTspice XVII et donne plusieurs exemples concrets d'utilisation.

Le **chapitre 21** termine avec huit exemples présentant des facettes souvent ignorées de LTspice XVII.

Certaines illustrations ont été empruntées aux ouvrages précédents, car elles ne comportent aucune différence entre leur version pour LTspice IV ou pour LTspice XVII.

Nous espérons que ce troisième volume permettra aux ingénieurs et aux techniciens de trouver des réponses précises à toutes leurs questions concernant LTspice XVII dans sa toute dernière version (août 2019).

Au-delà, l'auteur reste disponible et répondra à toutes vos questions, il peut être joint à son adresse mail **Brocard.gilles.b26@gmail.com** ou au cours des sessions de formation qu'il effectue en entreprise.



# Préface

LTspice IV est un outil global de simulation orientée **électronique analogique**. Il regroupe des qualités rarement rassemblées dans un même logiciel.

**1) Des éditeurs ergonomiques** dédiés à chaque domaine d'activité : création de schémas, de mesures (oscilloscope numérique), d'analyse spectrale (FFT), de génération de symboles, de valeurs et attributs de composant, de Netlist, de couleurs, etc.

**2) Des bases de données** comprenant des dizaines de milliers de modèles de composants et de sous-circuits paramétrables pour s'adapter à vos besoins particuliers. De nombreuses fonctions complexes sont aussi proposées, elles permettent un gain de temps appréciable.

**3) Des commandes puissantes** qui permettent de réaliser de vrais programmes de simulation entièrement adaptés à chaque problème.

**4) Des outils de paramétrage** et de personnalisation très complets. Ils permettent d'ajuster les fonctionnalités du logiciel à vos besoins spécifiques.

**5) Des outils de création de modèles et de sous-circuit** pour étendre facilement les bases de données de composants existantes ou les adapter à vos souhaits.

**6) Le meilleur noyau SPICE disponible** regroupant de nombreuses qualités le démarquant de ses concurrents : *extrême rapidité*, *robustesse à toute épreuve* et *grande fiabilité* de traitement même pour des schémas complexes. Il exploite également les processeurs jusqu'à huit cœurs. Ce noyau est une avancée notable du noyau Berkeley original.

**7) Six simulations** paramétrables et adaptables répondant à l'ensemble des besoins des électroniciens les plus exigeants. La plupart sont dédiées aux circuits linéaires, mais la simulation **Transient** sait traiter les circuits non linéaires.

**8) Il existe, via Internet, une communauté d'utilisateurs de LTspice XVII** comprenant plusieurs millions d'utilisateurs, principalement professionnels.

**9) Plusieurs centaines de milliers de modèles de composants** sont immédiatement accessibles sur Internet. La majorité des fabricants proposent un modèle SPICE de chaque composant de leur catalogue.



# Les nouveautés de LTspice XVII

La nouvelle version du logiciel LTspice est disponible depuis la fin 2016. Elle est téléchargeable gratuitement sur le site de la société Analog Devices qui a racheté Linear Technology début 2017. Voici l'écran de démarrage de LTspice XVII, tel qu'il se présente au premier lancement du logiciel (figure 1.1).

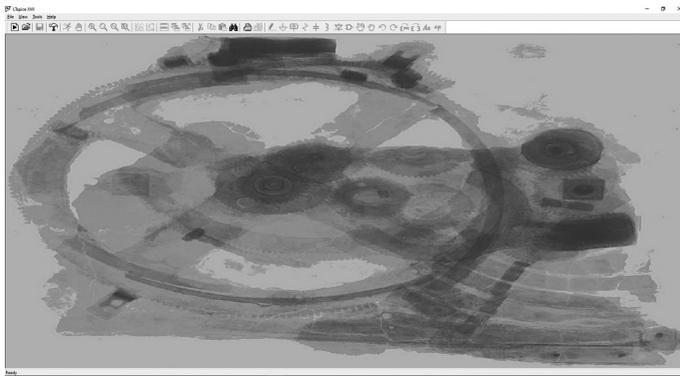


Figure 1.1

Cette nouvelle version est compatible avec le système d'exploitation Windows 10 qui est aujourd'hui la référence. Les nouveautés sont nombreuses comme vous pourrez vous en rendre compte au cours de la lecture des 20 chapitres de cet ouvrage. Dans sa catégorie, LTspice XVII est un logiciel hors norme, il est le leader incontesté de la simulation analogique, de plus il est depuis plus de 30 ans entièrement gratuit et régulièrement mis à jour et amélioré.

Son créateur Mike Engelhardt nous a apporté avec la nouvelle version XVII un confort amélioré avec l'utilisation de Unicode, une plus grande interactivité avec Windows 10, de nombreuses améliorations pour toutes les commandes et une puissance de calcul des simulations encore améliorée. Nous dédions cet ouvrage à Mike Engelhardt pour cette nouvelle version et encore bravo à lui pour le travail d'amélioration constante depuis près de 30 ans.

Comme nous venons de l'évoquer, LTspice XVII comporte de nombreuses améliorations par rapport à LTspice IV. Bien que ces améliorations touchent tous les secteurs, plusieurs chapitres présentent un nombre de nouveautés plus important.

- ▶ Le **chapitre 3** vous présentera les nouveautés touchant l'éditeur graphique, ce sont par exemple l'amélioration de la saisie de la plupart des composants, la gestion des commentaires avec les caractères accentués et mille autres améliorations petites ou grandes qui facilitent la création des schémas.
- ▶ Le **chapitre 7** aborde les nouveautés de l'affichage des mesures et notamment celles de l'oscilloscope virtuel et l'affichage de la FFT.
- ▶ Les **chapitres 11 et 12** vous montreront les bases de composants nettement augmentées par rapport à LTspice IV ainsi que l'ajout de nouveaux modèles comme l'IGBT.
- ▶ Les **chapitres 14 et 15** vous dévoileront les commandes les plus importantes. Elles ont maintenant chacune un éditeur dédié qui évite de devoir mémoriser leur syntaxe.
- ▶ Le **chapitre 16** vous montrera que certains paramétrages de la commande **.Options** ont disparu et que d'autres ont été ajoutés, l'un des plus importants est **SOAccounting**, c'est la déclaration du **modèle SOAtherm**. Le panneau de contrôle comporte également de nombreuses modifications, retraits et ajouts. A été ajouté un nouvel onglet qui permet de personnaliser la gestion des répertoires et notamment la gestion de vos propres créations.
- ▶ Le **chapitre 20** détaille le **modèle SOAtherm** qui permet l'étude thermique des MOSFET dans les schémas où des transitoires de puissance peuvent générer des pics de température destructifs au niveau de la puce.

## 1.1 Les principaux apports de la nouvelle version LTspice XVII

1) L'introduction d'un ensemble de cinq commandes permettant l'utilisation de LTspice avec **un langage déclaratif appelé machine d'état aléatoire**. C'est un ensemble de commandes totalement nouvelles qui ouvre une nouvelle voie vers les développements de l'électronique analogique les plus récents.

2) LTspice XVII présente **une interactivité complète avec Windows**. Par conséquent, sont devenus possibles le détachement des fenêtres de LTspice, le couper/coller, l'utilisation des caractères Unicode, l'affichage des icônes, etc. (paragraphe 1.3).

3) Les fenêtres de saisie de **l'interface graphique** utilisateur de LTspice ont été complètement **repensées et grandement améliorées**. Cela évite de devoir apprendre les règles syntaxiques contraignantes de LTspice, notamment lors de la saisie des paramètres de la plupart des composants (paragraphe 1.4).

4) **La vitesse a été notablement améliorée** lors du calcul de certaines simulations. C'est d'autant plus étonnant, que la vitesse était l'une des caractéristiques saillantes qui démarquaient déjà LTspice de ses concurrents.

## 1.2 Création et attribution de nouveaux répertoires

5) **LTspice XVII prend en charge UNICODE.** Ainsi les caractères accentués, cédille, etc. sont pris en charge (en fait tous les caractères saisissables avec le clavier AZERTY), c'est un changement particulièrement agréable pour la rédaction des commentaires et le choix des noms des composants (paragraphe 1.3).

6) L'ensemble des menus et **la gestion de la saisie sont réorganisés**, ainsi la présentation des menus, sous-menus et fenêtres contextuelles ont été réaménagées à l'occasion de cette nouvelle version. L'utilisateur régulier de **LTspice XVII** appréciera les efforts d'ergonomie apportés à toutes les phases de l'utilisation du logiciel notamment pour les menus contextuels c'est-à-dire ceux que l'on appelle d'un clic droit lorsqu'on est, par exemple, dans un schéma ou dans l'oscillateur virtuel.

7) **LTspice XVII prend maintenant en charge les versions 32 et 64 bits de Windows**, ce qui n'était pas le cas avec LTspice IV. **LTspice XVII** est compatible avec les dernières versions de Windows : Windows 10 et **Microsoft intends** (paragraphe 1.3).

8) **LTspice XVII a subi un lifting au niveau de son noyau**, cela a toujours été une priorité pour Mike Engelhardt, créateur et programmeur du logiciel LTspice.

9) Vous trouverez maintenant le répertoire LTC contenant l'ensemble des éléments de LTspice XVII dans **C:/Programmes** et plus dans **C:/Programmes(x86)**. La liste des modifications faites dans LTspiceXVII est accessible facilement, lorsque vous faites **Sync Release** ou que vous répondez positivement à l'invite **Update**. Pour l'afficher, allez dans le **menu Help** et lancez la commande **Show change Log**.

10) La version XVII peut utiliser **plusieurs périphériques d'affichage physiques**, en effet la gestion multi moniteur est maintenant intégrée (paragraphe 1.3).

11) D'autres ajouts sont apparus comme le **modèle d'IGBT**, la **récupération logicielle de diodes** et la **machine à états arbitraire** que nous détaillons plus loin.

12) LTspice XVII permet d'ajouter un chemin d'accès aux répertoires des symboles utilisateurs (fichiers .asy) et aux répertoires de bibliothèques ajoutées (fichiers .sub et .lib) (paragraphe 1.2).

## 1.2 Création et attribution de nouveaux répertoires

### Symbole LTspice

Pour créer ces répertoires, vous devez :

a) utiliser l'explorateur de Windows pour créer les répertoires souhaités. Vous ouvrez le répertoire et vous cliquez sur le bandeau d'explorer où vous récupérez le chemin du répertoire créé (figure 1.2).

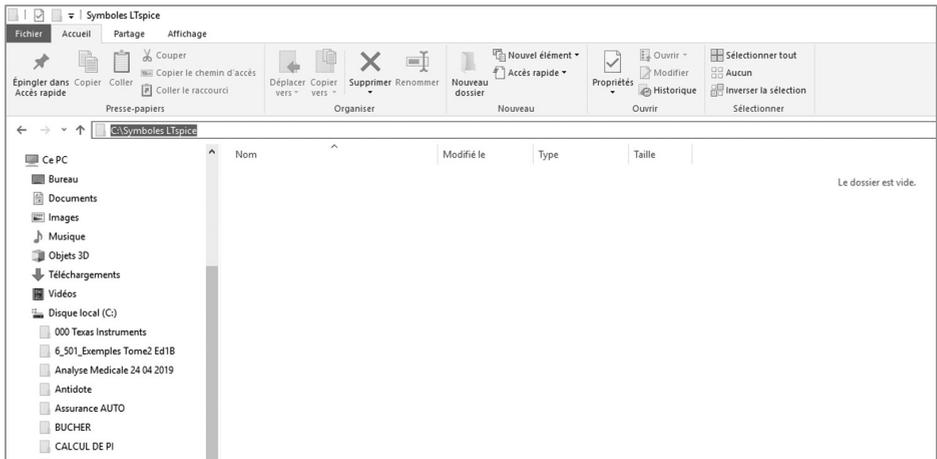


Figure 1.2

b) Ouvrir LTspice XVII et aller dans le menu suivant :

- Tools/Control Panel/Sym. & Lib. Search Paths (figure 1.3).



Figure 1.3

Vous déposez le chemin prélevé par explorer avec la commande couper/coller (figure 1.4).

### 1.3 LTspice XVII, dialogue amélioré avec Windows

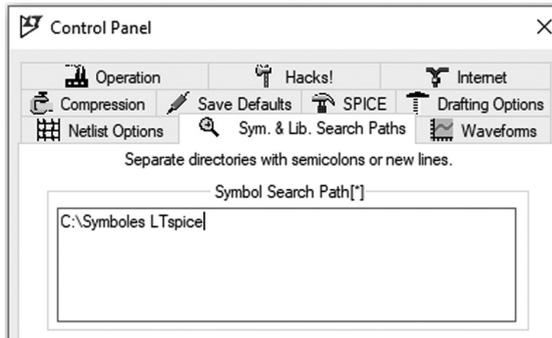


Figure 1.4

Vous sortez sans oublier de valider, à partir de ce moment vous pouvez utiliser ce répertoire pour exporter ou importer vos symboles comme vous pouvez le voir sur la figure 1.5. La procédure est rigoureusement la même pour la création de répertoires de bibliothèque.

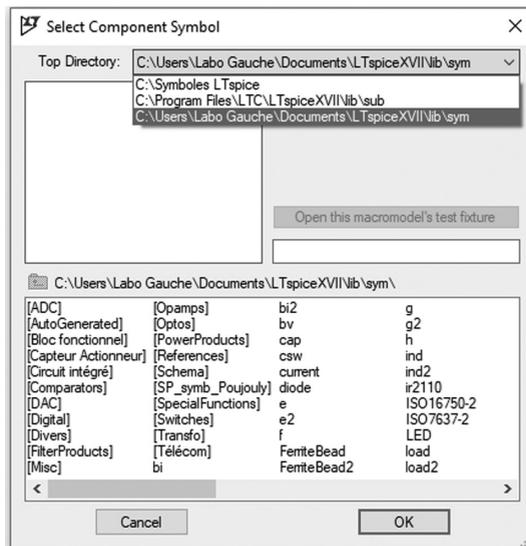


Figure 1.5

## 1.3 LTspice XVII, dialogue amélioré avec Windows

De nombreuses améliorations sont apportées avec la version LTspice XVII vis-à-vis des échanges avec Windows 10. Il serait fastidieux de les énumérer toutes, aussi allons-nous présenter les plus marquantes. Elles sont beaucoup plus nombreuses, vous pourrez les découvrir au gré de votre utilisation quotidienne.

À partir de LTspice XVII, vous avez une icône illustrant le schéma appelé par cette icône à la place d'une icône muette comme cela était le cas avec LTspice IV. Cela est très pratique et permet de faciliter les recherches de schémas (figure 1.6).

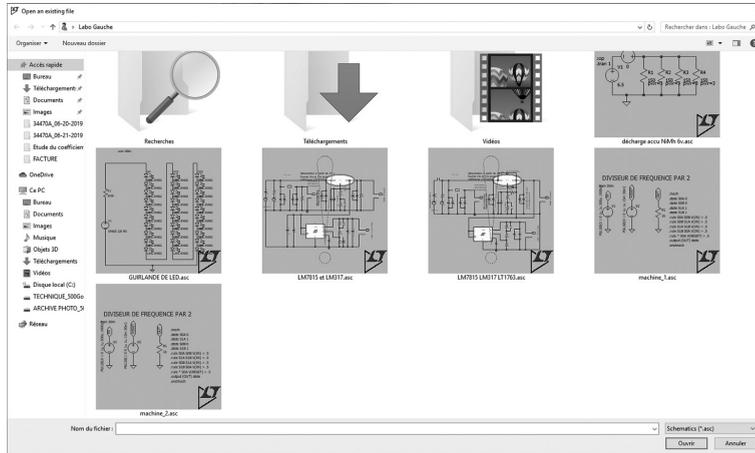


Figure 1.6

De plus certaines informations apparaissent sur l'icône, telles que l'absence de composant et autres erreurs (figure 1.7).

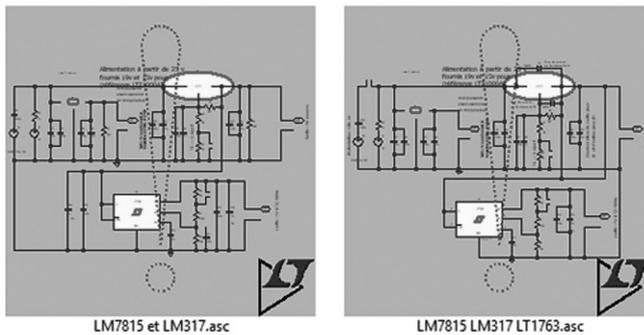


Figure 1.7

Vous pouvez également détacher les fenêtres et les placer où vous voulez sur le bureau de Windows, indépendamment de LTspice. La commande correspondante **Float Window** est atteinte en cliquant droit sur la partie (schéma ou oscillogramme) que vous souhaitez rendre autonome (figure 1.8).

Ensuite, la partie détachée est sur le bureau et vous pouvez la manipuler librement comme n'importe quelle autre fenêtre avec Windows (figure 1.9).

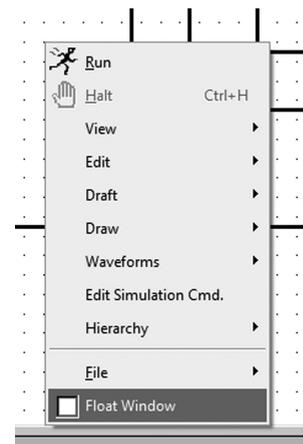


Figure 1.8

## 1.4 Nouveautés concernant l'interface graphique de saisie

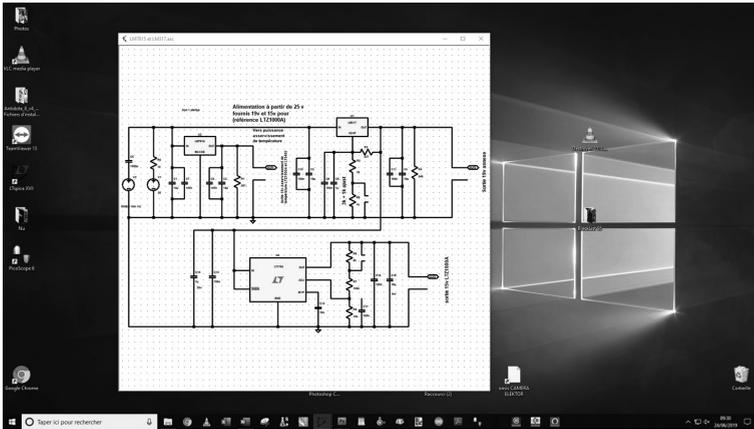


Figure 1.9

La troisième amélioration est la possibilité d'utiliser les caractères propres à chaque pays dans les commentaires. Cela est dû au fait que LTspice XVII est maintenant compatible avec les caractères Unicode.

Avec LTspice IV, lorsque vous utilisiez un caractère accentué, il n'était tout simplement pas pris en compte et donc pas affiché. Aujourd'hui avec LTspice XVII, tous les caractères sont affichés sans restriction (figure 1.10).

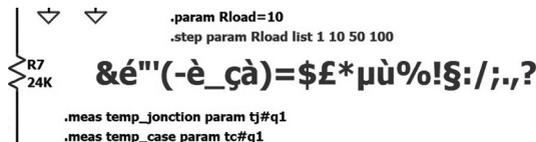


Figure 1.10

## 1.4 Nouveautés concernant l'interface graphique de saisie

LTspice XVII hérite du très bon éditeur graphique de schéma de LTspice IV, mais Mike Engelhardt a souhaité l'améliorer encore en le dotant de nouvelles fenêtres de saisie qui évitent le recours à une syntaxe contraignante pour l'utilisateur et les possibilités sont plus étendues.

Prenons comme exemple **steptemp.asc** dans les exemples **éducations**. Si vous cliquez droit sur la commande **.step temp -55 125 5** posée sur le schéma, vous ouvrirez la fenêtre suivante (figure 1.11).

Vous voyez immédiatement l'amélioration par rapport à la fenêtre de LTspice IV qui ne proposait qu'une petite fenêtre avec seulement la ligne du bas où la syntaxe

de l'écriture de la commande devait être connue. Aujourd'hui, cinq champs clairement identifiés vous permettent de modifier les valeurs choisies (figure 1.12).

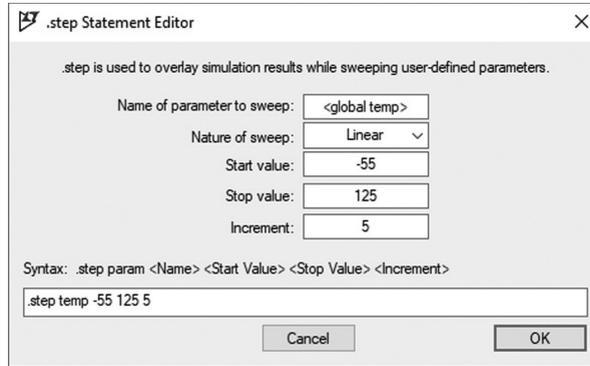


Figure 1.11

Comme vous pouvez le voir, la modification de **Stop Value** faite par l'utilisateur a modifié automatiquement sa valeur, au bon emplacement, dans la ligne de commande en respectant les règles de syntaxe, espacement, etc.

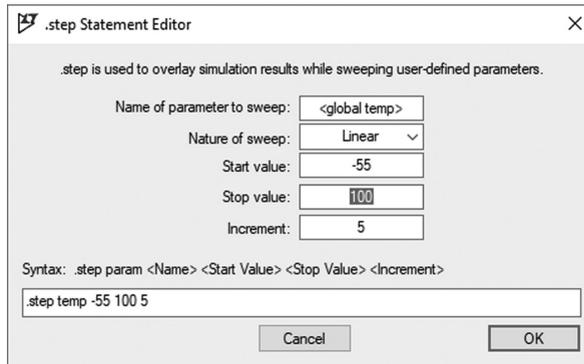


Figure 1.12

Vous pouvez sortir de cette fenêtre de deux manières différentes, soit par la touche **OK**, soit par **Cancel**. Si vous sortez par **Cancel**, vous ouvrez la fenêtre suivante (figure 1.13).

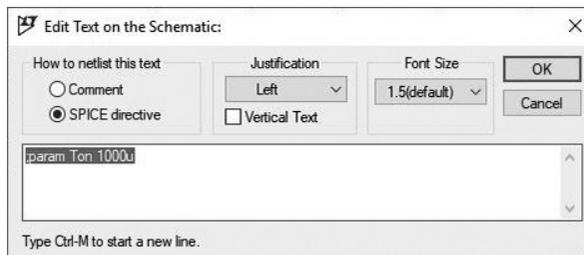


Figure 1.13

## 1.4 Nouveautés concernant l'interface graphique de saisie

Elle vous permet de valider la commande par la frappe de la touche **OK**, mais elle permet également de sélectionner **Comment** avant de valider par OK. Ceci est parfois ambigu, car la commande placée sur le schéma devient un commentaire et ne sera plus prise en compte lors du calcul de la simulation, pourtant elle apparaît précédée d'un point au début et en couleur noir, donc **à priori active** alors que ce n'est plus le cas !

**Attention**, cela peut être la cause d'erreur surtout si vous avez de nombreuses commandes accompagnant votre schéma. Le passage d'une commande en commentaire (donc inactif pour le simulateur) par l'action du remplacement du point par un point-virgule comme cela était la pratique courante avec LTspice IV reste utilisable, nous la conseillons, car son utilisation évite toute ambiguïté.

Si vous souhaitez changer de type d'énumération, par exemple remplacer **List**, pas de problème, LTspice XVII changera les champs et la syntaxe de la commande pour vous (figure 1.14).

Par contre, vous devrez indiquer au logiciel les nouvelles valeurs de la liste. Par exemple quatre nouvelles valeurs devront remplacer les valeurs existantes dans le champ commande du bas de la fenêtre.

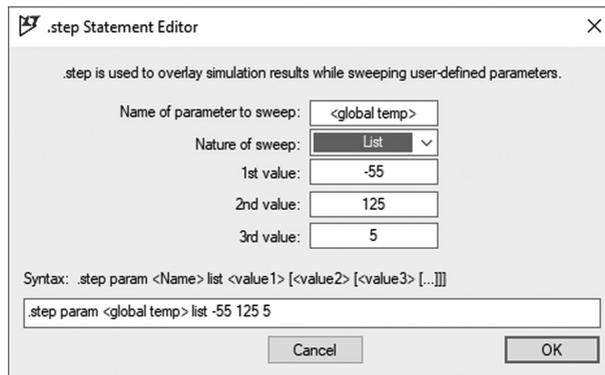


Figure 1.14

Prenons comme exemple les commandes suivantes (figure 1.15).

```
;param Ton 1000u  
.param R_LED = 8  
.step param Ton list 400u 800u 1200u 1600u
```

Figure 1.15

On peut, au moment de la mise au point, avoir besoin d'alterner entre la première ligne de commande imposant une valeur fixe à  $T_{ON}$  (**param Ton 1000u**) et la troisième ligne de commande imposant une variation de valeur à  $T_{ON}$  (**step param Ton list 400u 800u 1200u 1600u**). Précédemment, avec LTspice IV, il suffisait de

remplacer un **point-virgule** en début de commande par un **point** pour rendre la commande active ou le contraire pour la rendre inactive. Avec la version XVII, vous avez deux possibilités.

Si vous cliquez droit sur la première ligne qui est en commentaire (**point-virgule** au début du texte), vous obtenez l'ouverture de la fenêtre suivante (figure 1.16).

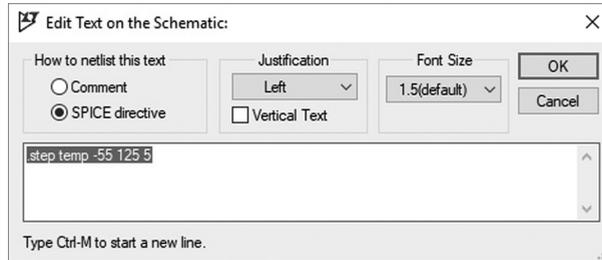


Figure 1.16

Vous voyez immédiatement la contradiction, la commande est précédée d'un **point-virgule**, elle est donc inactive et pourtant **SPICE directive** indique que cette commande doit être active. Si vous faites **OK**, qu'est-ce qui va prévaloir ? C'est le **point-virgule** qui impose que la commande soit inactive.

Vous pouvez d'ailleurs sortir de cette fenêtre par **OK** ou par **Cancel**, l'action sera la même : la commande restera inactive, ce que vous pouvez vérifier immédiatement en cliquant droit de nouveau sur cette même première ligne de commande, vous retrouvez exactement le même contenu de la fenêtre avec la même ambiguïté (figure 1.17).

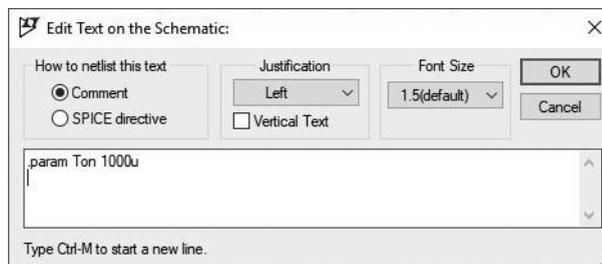


Figure 1.17

En revanche, si vous modifiez la commande en remplaçant le **point-virgule** par un **point** et que vous validez par **OK**, la commande sera alors active.

Mais attention : si vous modifiez la commande en remplaçant le **point-virgule** par un **point** et que vous cliquez gauche sur **Comment**, alors votre ligne de commande débutera par un point, ce qui semble indiquer que la commande est valide, mais il n'en sera rien. Pour LTspice ce sera un commentaire, ce que vous pourrez vérifier en cliquant droit sur cette commande. Cela est dangereux, car cela peut occasionner des erreurs dans l'exécution d'une simulation. Si le nombre de com-

## 1.6 Machine d'état arbitraire

mandes sur le schéma est important, il pourra être difficile de retrouver la ou les commandes que vous pensez et que vous voyez (grâce au point) actives alors qu'elles ne le sont pas, ce qui empêchera leur exécution au moment du lancement de l'exécution.

Nous vous engageons à relire ce paragraphe et à faire quelques essais, car cette ambiguïté de commande/commentaire peut être une source d'erreur parfois difficile à retrouver.

**Notre conseil :** toujours faire coïncider la présence du **point-virgule** et de l'affectation **Comment** ainsi que celle du **point** et de l'affectation **SPICE Directive**.

## 1.5 Accès plus facile au diagramme de l'œil

Avec l'éditeur d'affichage des courbes (oscilloscope), pour changer l'échelle des axes horizontal ou vertical, il faut placer le curseur sur l'axe et cliquer droit, la fenêtre qui s'affiche offre une possibilité inédite (figure 1.18).

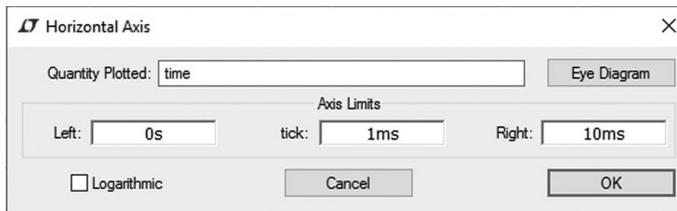


Figure 1.18

Un bouton donne un accès direct au diagramme de l'œil, qu'il était difficile d'activer précédemment (figure 1.19).

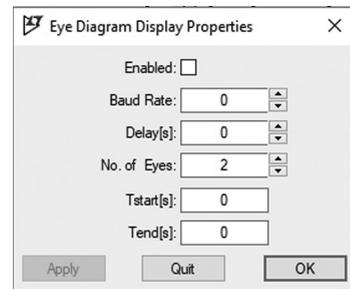


Figure 1.19

## 1.6 Machine d'état arbitraire

La nouvelle version LTspice XVII propose désormais une commande supplémentaire **.mach** (abréviation de **.machine**) permettant d'implanter **une machine d'état arbitraire** avec une syntaxe propre à LTspice XVII.

Voici l'exemple donné par Mike Engelhardt permettant de réaliser un diviseur de fréquence par 2 avec une remise à zéro. Sur le schéma on commence par placer deux sources de tension qui génèrent, d'une part le signal d'horloge que l'on doit diviser par deux et d'autre part un signal correspondant à une commande de remise à zéro ainsi qu'une résistance reliée à la masse. La sortie se faisant en courant, elle sert de convertisseur de courant/tension pour obtenir une sortie classique en tension (figure 1.20).

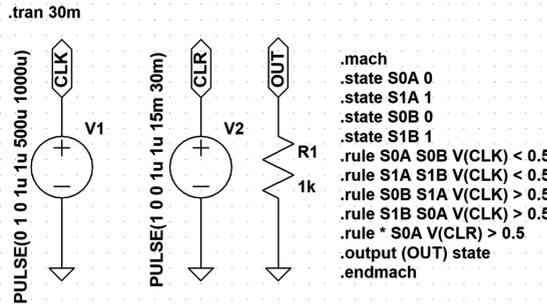


Figure 1.20

Les règles pour utiliser cette machine d'état arbitraire sont décrites ci-après.

1) La description de la machine d'état est saisie sous la forme de commandes (commençant par un point). Elles sont encadrées par les commandes **.mach** (au début) et **.endmach** (à la fin). Le caractère '\*' en tant que valeur de <old state> correspond à n'importe quel état précédent. Les règles qui comportent ce signe sont vérifiées en premier, car elles définissent l'état initial.

2) L'ordre des instructions placées entre **.mach** et **.endmach** est indifférent. Le programme ne fait aucune différence si on permute les lignes de commande comme c'est le cas dans tous les **langages déclaratifs**, par opposition au langage procédural tel que le langage C. Il n'existe qu'une exception concernant le premier état déclaré qui est l'état initial. Les règles sont vérifiées par le compilateur, dans l'ordre d'apparition.

```
.mach[in] [<tripdt>]
```

où

- ▶ **.mach** est un mot réservé qui indique le début d'un programme.
- ▶ **tripdt** est une tolérance temporelle optionnelle par exemple un délai.
- ▶ **.endmach[in]** est un mot réservé qui indique la fin du programme.

3) La commande **.state** permet d'attribuer une valeur (**value**) à une variable d'état à laquelle est donné un nom (**name**). Dans la machine d'état arbitraire, cette variable est appelée **état**, elle peut être mentionnée par son nom (**name**) dans l'expression d'une sortie. C'est ce qui correspond aux déclarations de variables en langage C.

```
.state <name> <value>
```

où

- ▶ **.state** est le mot réservé qui permet d'attribuer une valeur (**value**) à un état dont le nom est **name**.
- ▶ **name** est le nom de la variable.
- ▶ **value** est la valeur donnée à cette variable.

4) Il peut y avoir autant de règles (**rule**) conditionnelles que vous le souhaitez. Une seule règle est exécutée par pas d'exécution du programme.

**.rule <old state> <new state> <condition>**

où

- ▶ **.rule** est le mot réservé indiquant le début d'une règle.
- ▶ **old state** est l'état précédent.
- ▶ **new state** est le nouvel état.
- ▶ **condition** est la condition à remplir pour que l'état précédent (**old state**) soit remplacé par le nouvel état (**new state**).

5) Si la règle (**rule**) comporte un astéris (« \* ») alors elle est lue en premier et est appliquée immédiatement et **New state** devient le nouvel état quel que soit l'état précédent (**old state**).

**.rule \* <new state> <condition>**

6) L'instruction **.output** applique le résultat **expression** à la borne **node** sous forme d'un courant que l'on peut transformer en tension à l'aide d'une résistance. Vous pouvez ajouter une capacité en parallèle sur la résistance pour donner une constante de temps aux transitions. **Expression** peut être la combinaison d'états combinatoires et/ou logiques.

**.output (node) <expression>**

où

- ▶ **.output** est un mot réservé pour appliquer le résultat d'une expression à une borne de sortie.
- ▶ **node** est la borne de sortie où expression est appliquée.
- ▶ **expression** est la combinaison d'états combinatoires et/ou logiques aussi complexe que vous le souhaitez.

### Exemple du diviseur par deux

Les quatre premières lignes définissent les variables ainsi que leurs valeurs initiales :

**.state S0a 0**

**.state S0b 0**

**.state S1a 1**

**.state S1b 1**

Les quatre lignes suivantes définissent les changements d'état en fonction d'une condition rencontrée à l'entrée **CLK**. L'ordre dans lequel elles sont placées n'a aucune incidence sur le résultat. Par exemple pour la première ligne, si **V(CLK) < 0,5 V** est vrai, alors l'ancien état **S0a** devient le nouvel état **S0b** et ainsi de suite pour les autres lignes.

```
.rule S0a S0b V(CLK) < 0.5
```

```
.rule S0b S1a V(CLK) > 0.5
```

```
.rule S1a S1b V(CLK) < 0.5
```

```
.rule S1b S0a V(CLK) > 0.5
```

Si  $V(CLR) > .5$  est vrai, alors S0a devient le nouvel état quel que soit l'état précédent.

```
.rule * S0a V(CLR) > .5
```

La valeur du courant de sortie à l'endroit de l'étiquette **Etat** est celle de l'état **state**

```
.output (Etat) state
```

Afin d'illustrer la description de cette machine d'état, voici une représentation sous la forme d'un diagramme d'état, quel que soit l'état de départ  $SX = 0a, 0b, 1a$  ou  $1b$ , les règles que chaque cas doit respecter sont les suivantes (figure 1.21).

Et voici sur l'oscilloscope de LTspice XVII, les traces que vous obtiendrez (figure 1.22).

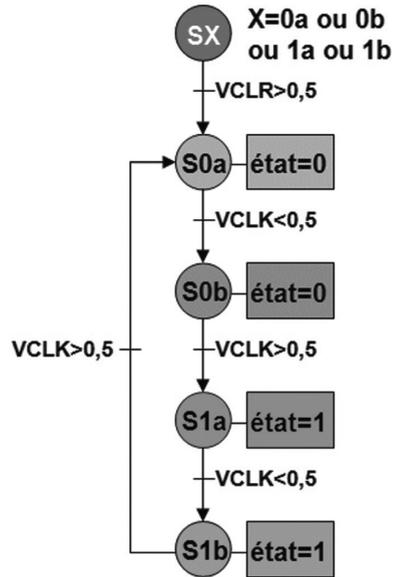


Figure 1.21

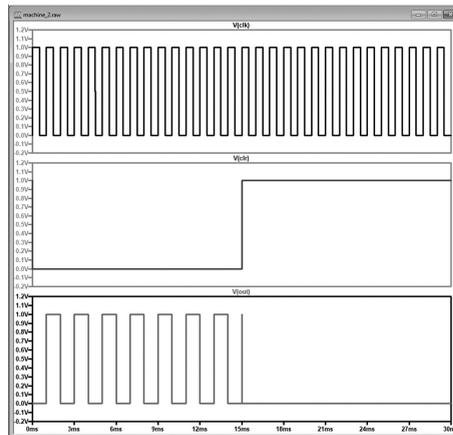


Figure 1.22

## 1.7 Petit retour sur LTspice IV

Malgré les nombreux aménagements, améliorations et ajouts apportés à LTspice XVII, si vous gardez une préférence pour LTspice IV, sachez que cette ancienne version reste toujours téléchargeable en choisissant l'option **LTspice pour Windows XP** au moment du téléchargement sur le **site Analog Devices**.