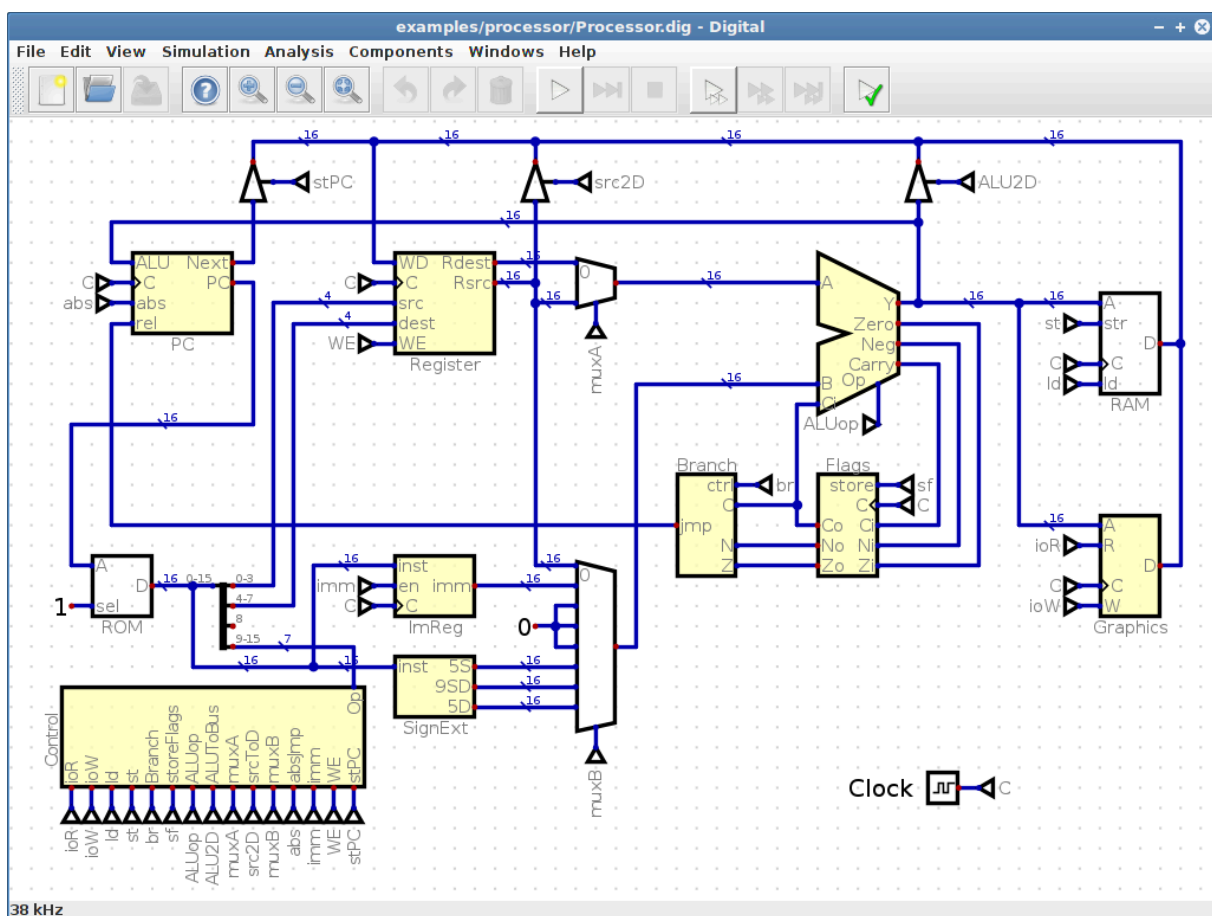


# Digital



Révision: v0.31  
Date: 2024-09-03 12:57

# Table des matières

## A Général

1. Digital .....	6
1.1. Introduction .....	6
1.2. Premiers pas .....	6
1.3. Câblage .....	14
1.4. Design hiérarchique .....	14
2. Simulation .....	18
2.1. Délais de propagation .....	18
3. Analyse .....	18
3.1. Analyse et synthèse de circuits .....	18
3.2. Expression .....	19
3.3. Diagrammes d'états .....	19
4. Matériel .....	19
4.1. GAL16v8 et GAL22v10 .....	19
4.2. ATF150xAS .....	19
4.3. Exportation en VHDL ou Verilog .....	20
5. Formes personnalisées .....	20
6. Circuits génériques .....	21
7. Tests contrôlés par script .....	21
8. Foire aux questions .....	22
9. Raccourcis claviers .....	24

## B Paramètres

## C Interface de ligne de commande

## D Composantes

1. Logique	
1.1. Et .....	31
1.2. Non-Et .....	31
1.3. Ou .....	32
1.4. Non-Ou .....	33
1.5. XOR .....	33
1.6. NXOR .....	34
1.7. Non .....	35
1.8. Table de recherche .....	35
2. E/S	
2.1. Sortie .....	36
2.2. DEL .....	37
2.3. Entrée .....	37
2.4. Horloge .....	38
2.5. Bouton .....	39
2.6. Interrupteur DIP .....	39
2.7. Sonde .....	40
2.8. Graphique de données .....	40
2.9. Graphique de données avec déclenchement .....	41
3. E/S - Affichages	
3.1. DEL-RVB .....	41

3.2. DEL avec deux connexions. ....	42
3.3. Bouton avec DEL .....	42
3.4. Affichage 7 segments .....	43
3.5. Affichage 7 segments Hex .....	44
3.6. Affichage 16 segments .....	44
3.7. Ampoule .....	45
3.8. Matrice de DELs .....	45
4. E/S - Mécanique .....	
4.1. Encodeur rotatif .....	46
4.2. Moteur pas-à-pas, unipolaire .....	47
4.3. Moteur pas-à-pas, bipolaire .....	47
5. E/S - Périphériques .....	
5.1. Clavier .....	48
5.2. Terminal .....	49
5.3. Telnet .....	49
5.4. Écran VGA .....	50
5.5. MIDI .....	51
6. Câblage .....	
6.1. Masse .....	52
6.2. Tension d'alimentation .....	52
6.3. Valeur constante .....	53
6.4. Tunnel .....	53
6.5. Séparateur/Fusionneur .....	53
6.6. Pilote .....	54
6.7. Pilote, sélection inversée .....	55
6.8. Délai .....	56
6.9. Résistance de tirage .....	56
6.10. Résistance de rappel .....	57
6.11. Pas connecté .....	57
7. Plexeurs .....	
7.1. Multiplexeur .....	57
7.2. Démultiplexeur .....	58
7.3. Décodeur .....	59
7.4. Sélecteur de bit .....	59
7.5. Codeur de priorité .....	60
8. Bascules .....	
8.1. Bascule RS .....	60
8.2. Bascule RS, avec horloge .....	61
8.3. Bascule JK .....	62
8.4. Bascule D .....	63
8.5. Bascule T .....	64
8.6. Bascule JK asynchrone .....	65
8.7. Bascule D asynchrone .....	66
8.8. Bascule monostable .....	67
9. Mémoire - RAM .....	
9.1. RAM, ports séparés .....	68
9.2. RAM de bloc, ports séparés .....	69
9.3. RAM, port bidirectionnel .....	70
9.4. RAM, sélection de puce .....	71
9.5. Banc de registres .....	72

9.6. RAM, port double .....	73
9.7. RAM, async. ....	74
9.8. RAM graphique .....	74
10. Mémoire - EEPROM .....	
10.1. EEPROM .....	75
10.2. EEPROM, ports séparés .....	76
11. Mémoire .....	
11.1. Registre .....	77
11.2. ROM .....	78
11.3. ROM, ports séparés .....	79
11.4. Compteur .....	81
11.5. Compteur prédéfini .....	82
11.6. Générateur de nombre aléatoire .....	83
12. Arithmétique .....	
12.1. Addition .....	84
12.2. Soustraction .....	84
12.3. Multiplication .....	85
12.4. Division .....	86
12.5. Décaleur .....	86
12.6. Comparateur .....	87
12.7. Négation .....	88
12.8. Extension de signe .....	88
12.9. Compteur de bits .....	89
13. Interrupteurs .....	
13.1. Interrupteur .....	89
13.2. Interrupteur bidirectionnel .....	90
13.3. Relais .....	90
13.4. Relais bidirectionnel .....	91
13.5. FET de type P .....	92
13.6. FET de type N .....	93
13.7. Fusible .....	93
13.8. Diode à l'alimentation .....	94
13.9. Diode à la masse .....	94
13.10. FET de type P à grille flottante .....	95
13.11. FET de type N à grille flottante .....	96
13.12. Porte de transmission .....	96
14. Divers .....	
14.1. Test .....	97
15. Divers - Décoration .....	
15.1. Texte .....	97
15.2. Rectangle .....	98
16. Divers - Générique .....	
16.1. Initialisation générique .....	98
16.2. Code .....	99
17. Divers - VHDL/Verilog .....	
17.1. Externe .....	99
17.2. Fichier externe .....	100
17.3. Commande de broche .....	101
18. Divers .....	
18.1. Alimentation .....	102

---

18.2. Séparateur bidirectionnel .....	102
18.3. Réinitialisation .....	103
18.4. Pause .....	103
18.5. Arrêt .....	104
18.6. Minuterie asynchrone .....	104

## **E Bibliothèque**

## A Général

### 1. Digital

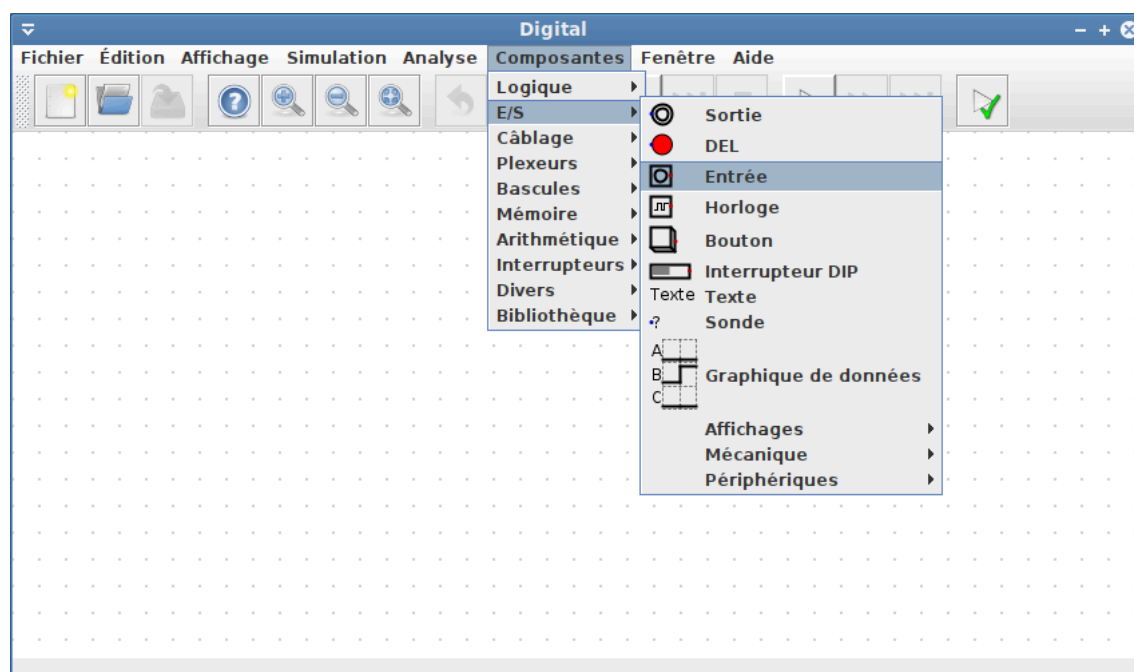
#### 1.1. Introduction

Digital est un simulateur simple utilisé pour simuler un circuit numérique. Les portes logiques sont connectées ensemble avec des fils et le comportement du circuit créé peut être simulé. Les utilisateurs peuvent interagir avec la simulation soit en appuyant sur des boutons ou en fixant des valeurs sur les entrées du circuit.

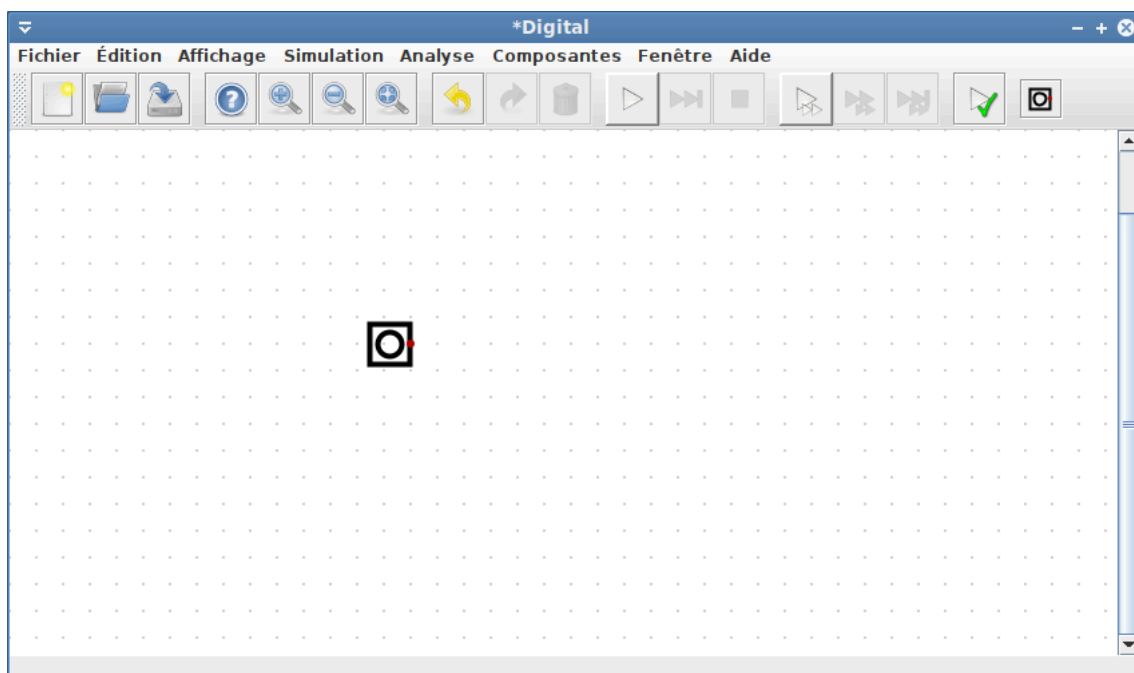
De cette manière, la plupart des circuits de base utilisés en électronique peuvent être construits et simulés. Dans le dossier *exemples*, les utilisateurs peuvent parcourir des exemples qui incluent un exemple fonctionnel d'un processeur Harvard 16 bits à cycle unique.

Le simulateur a deux modes de fonctionnement: le mode d'édition et de simulation. Dans le mode d'édition, les modifications au circuit peuvent être effectuées. Les utilisateurs peuvent ajouter ou connecter des composantes. Dans ce mode, la simulation est désactivée. Le mode de simulation est activé en appuyant sur le bouton *Exécuter* dans la barre d'outils. Durant le lancement du circuit, celui-ci est analysé pour vérifier le bon fonctionnement. S'il y a des erreurs, un message approprié est affiché et les composantes ou les fils affectés sont mis en évidence. Si le circuit ne contient pas d'erreurs, la simulation est activée. Vous pourrez alors interagir avec la simulation en cours.

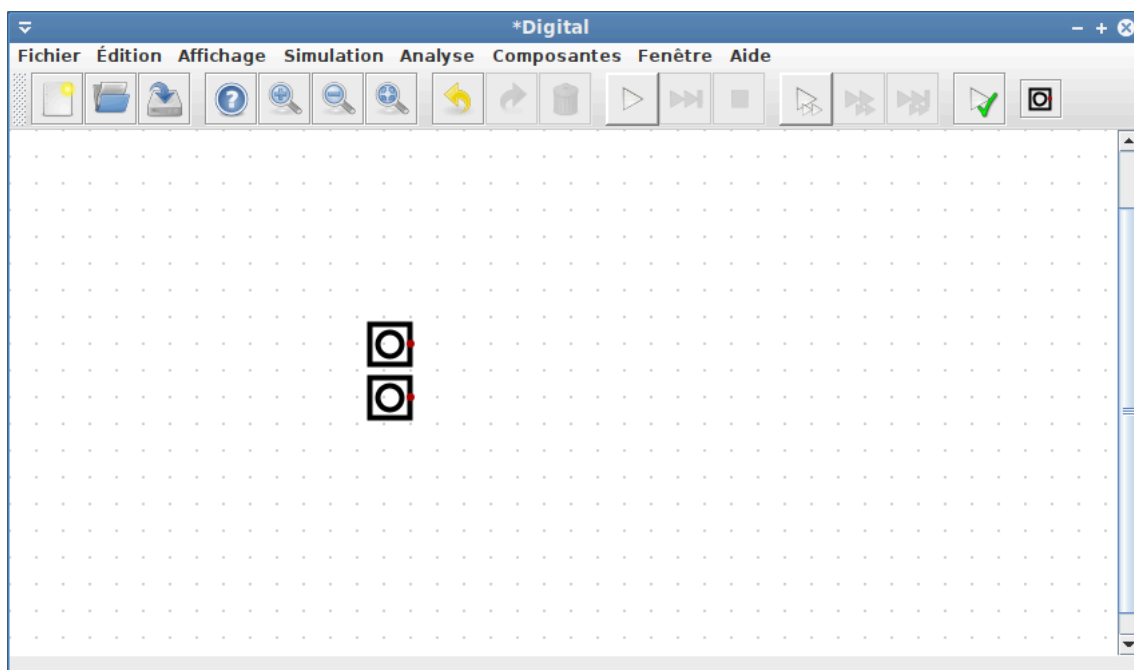
#### 1.2. Premiers pas



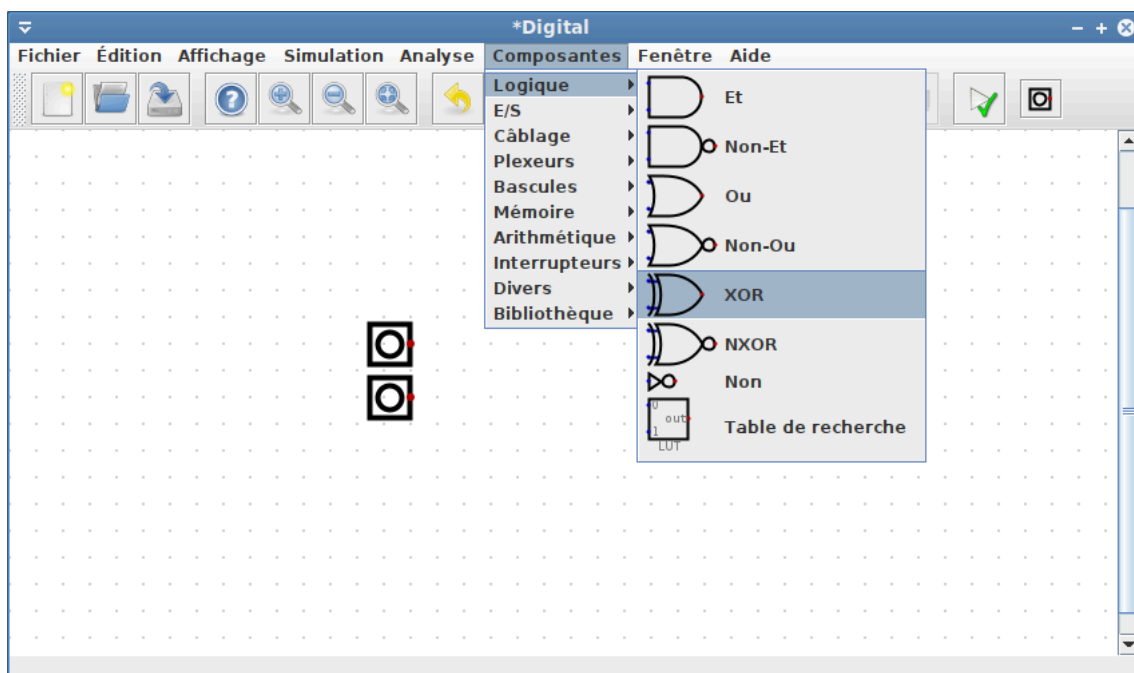
Comme premier exemple, un circuit sera construit avec une porte Ou-Exclusif, aussi appelée XOR. Depuis la fenêtre principale, un menu *Composantes* permet de sélectionner diverses composantes. Ensuite, elles peuvent être placées sur le volet de dessin. Ce processus peut être interrompu n'importe quand avec la touche Échap. Commencez par sélectionner une composante d'entrée. Celle-ci pourra plus tard servir à interagir avec le circuit avec la souris.



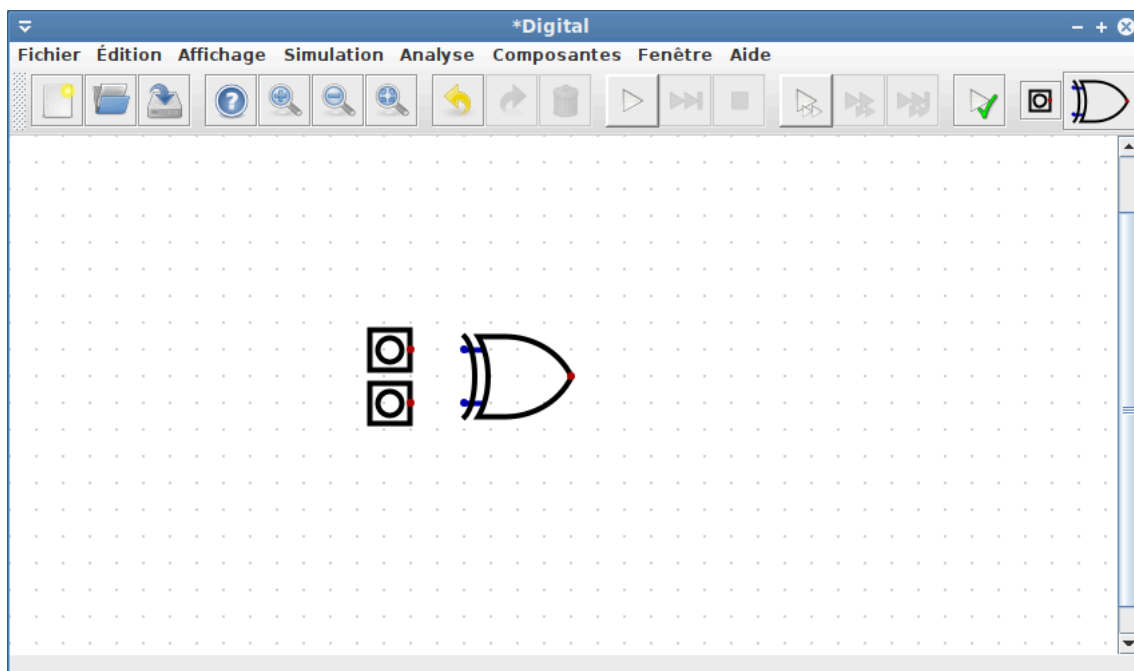
Après l'avoir sélectionnée, la première entrée peut être placée sur le volet de dessin. Le point rouge de la composante d'entrée sert à connecter la composante à un fil, qui sera ajouté plus tard. La couleur rouge indique une sortie. Cela veut dire que le port définit une valeur de signal et peut alimenter un fil.



De la même façon, une deuxième entrée est ajoutée. Il vaut mieux la placer directement sous la première.

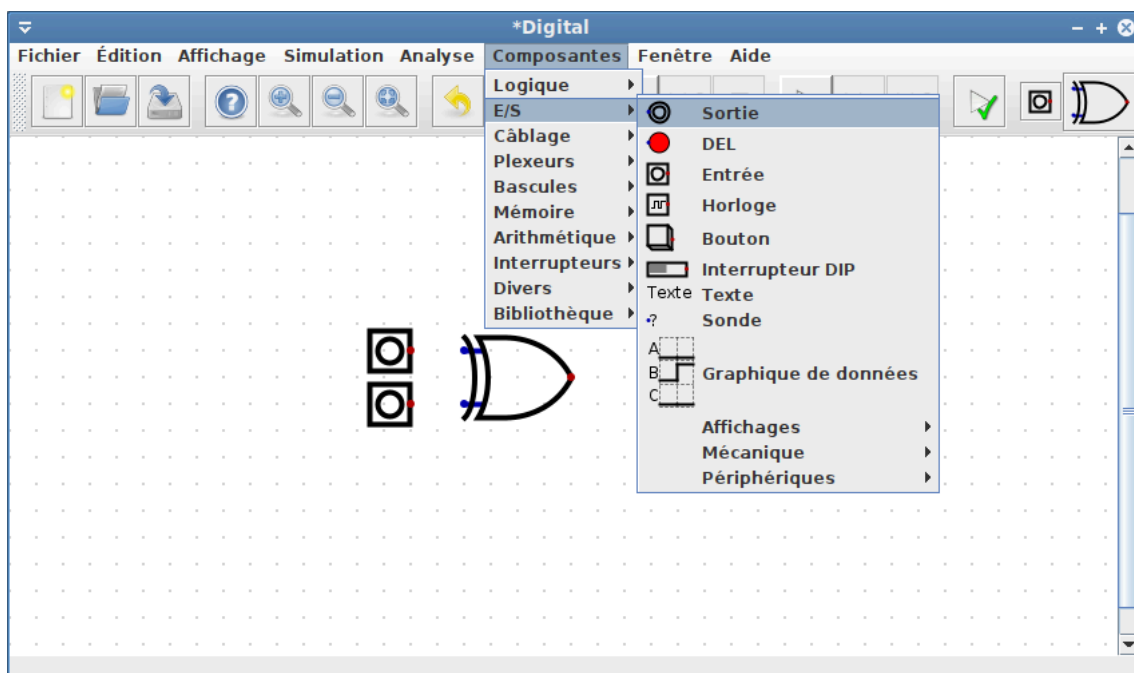


Après avoir ajouté les entrées, la porte Ou-Exclusif est sélectionnée. Cette porte représente la fonction logique en tant que telle.

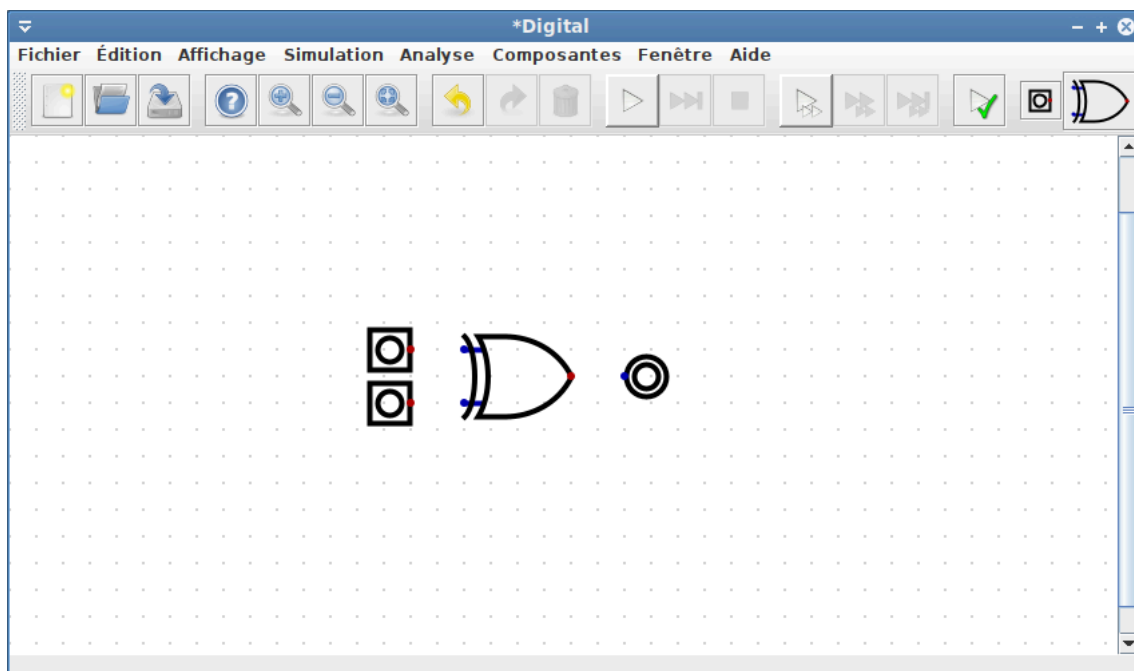


Cette porte peut ensuite être également ajoutée au circuit. Il vaut mieux la placer d'une manière à ce que le câblage subséquent soit le plus simple possible. Les points bleus indiquent les terminaux d'entrées de la porte.

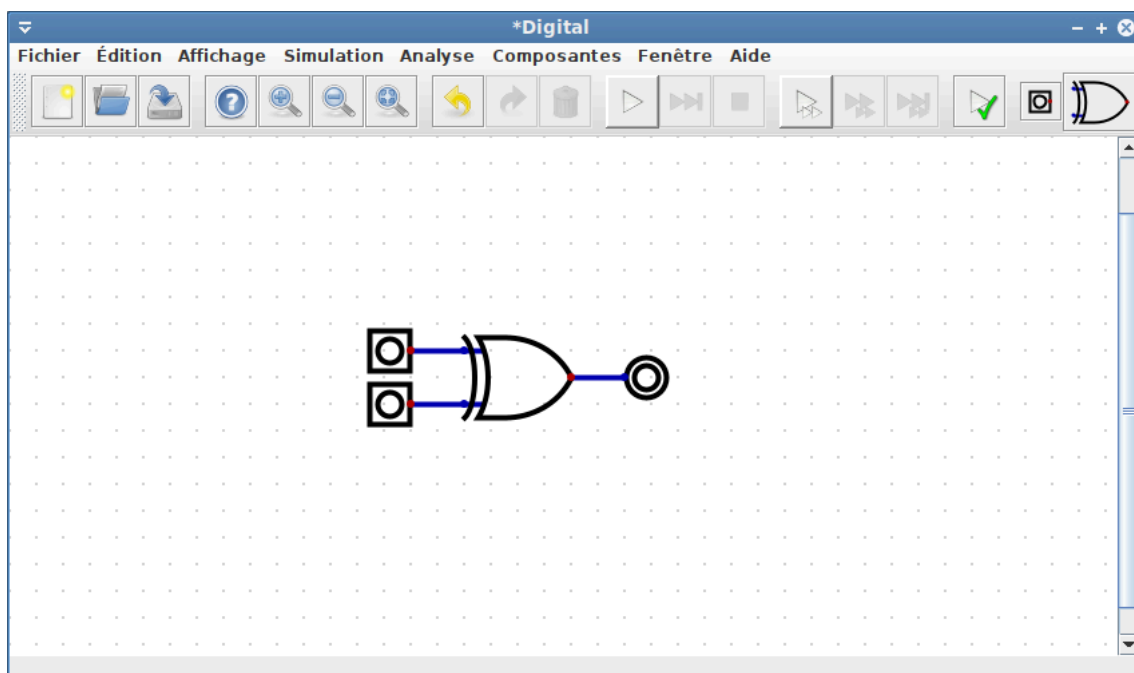




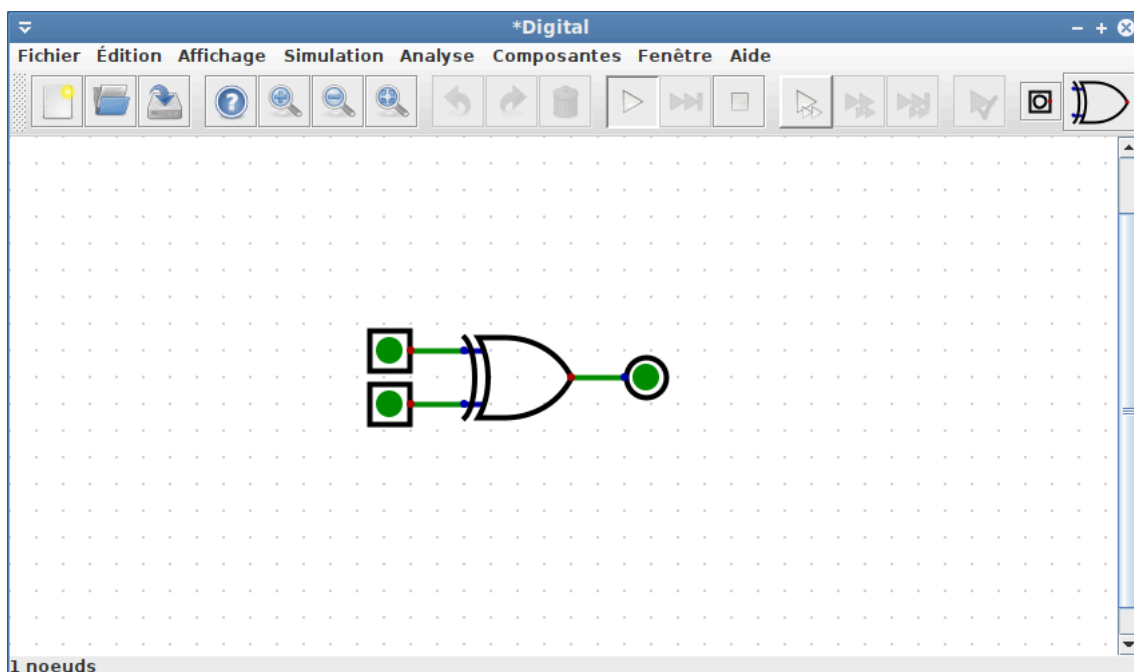
Maintenant, sélectionnez une sortie qui pourra être utilisée pour afficher l'état d'un signal, ou éventuellement servir à passer des signaux à un circuit intégré.



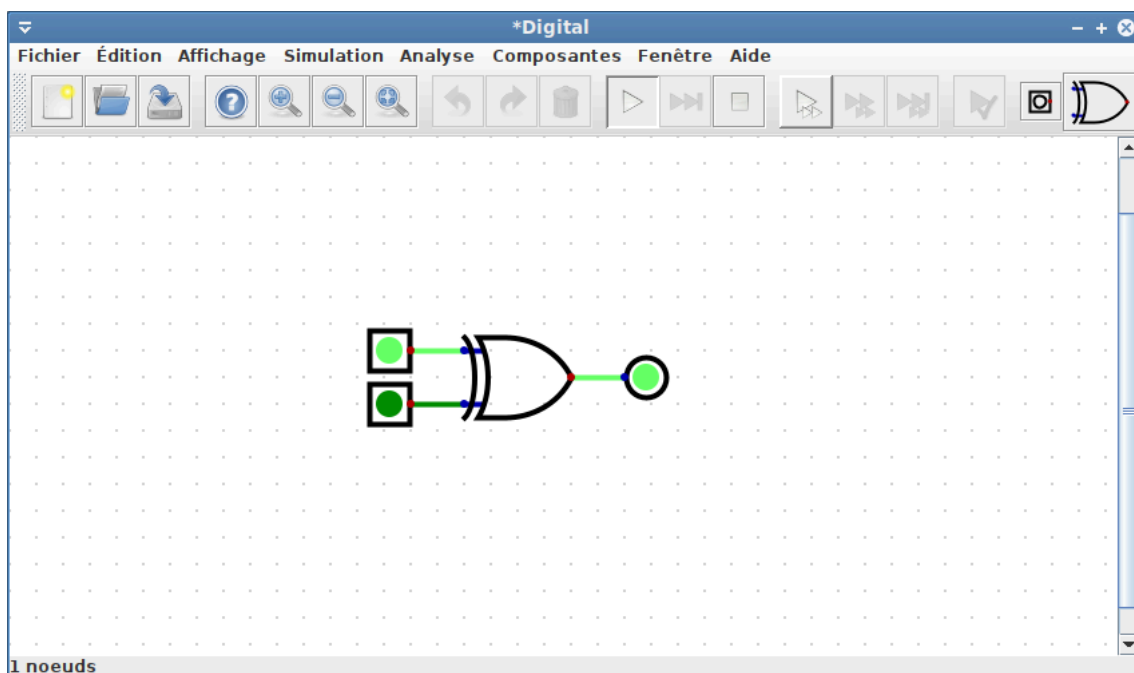
Elle est placée de façon à pouvoir être connectée facilement. La sortie a elle aussi un point bleu qui indique un terminal d'entrée. C'est à cet endroit que la valeur du signal entre et est ensuite exportée.



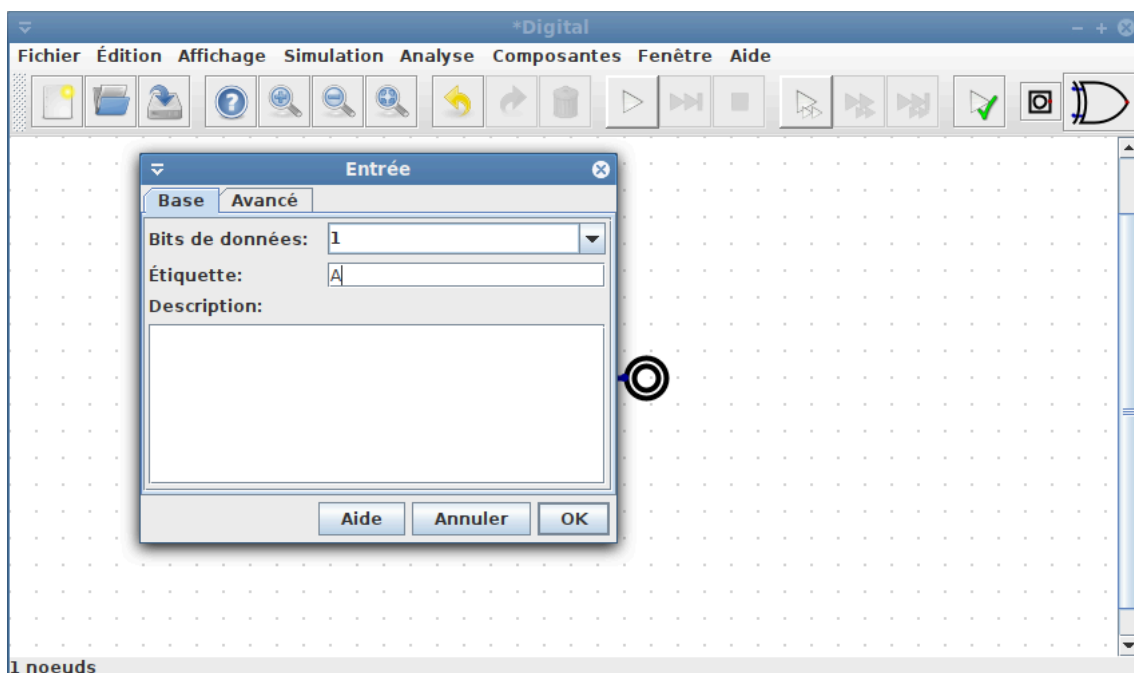
Après que toutes les composantes ont été sélectionnées et sont en place, utilisez la souris pour créer des connexions entre les points bleus et les points rouges. Assurez-vous qu'il y ait toujours exactement un point rouge connecté à n'importe quel nombre de points bleus. La seule exception à cette règle est lors de l'utilisation d'entrées à trois états. Dans ce cas particulier, il est possible de dévier de la règle et interconnecter plusieurs points rouges. Lorsque tous les fils ont été connectés, le circuit est complet.



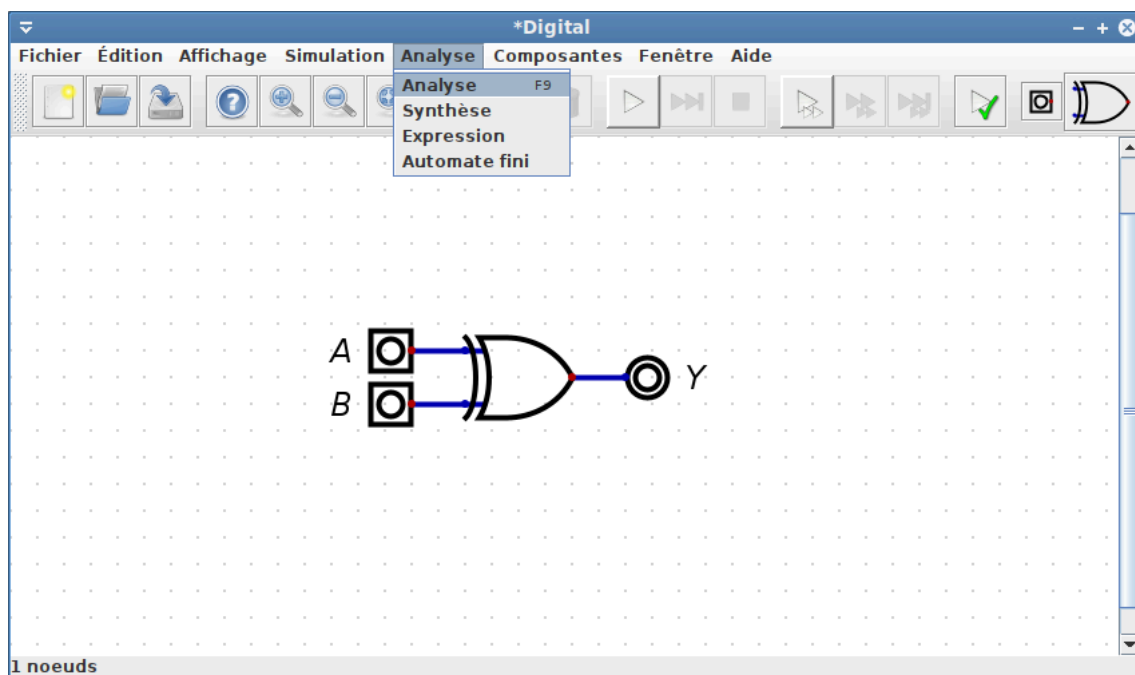
L'interaction avec le circuit est possible lorsque la simulation a commencé. Cela peut être fait avec le bouton Play situé dans la barre d'outils. Après avoir lancé la simulation, la couleur des fils, des entrées et des sorties change. Le vert pâle indique un '1' logique, tandis que le vert foncé indique un '0' logique. Dans la figure ci-dessus, tous les fils ont une valeur de '0'.



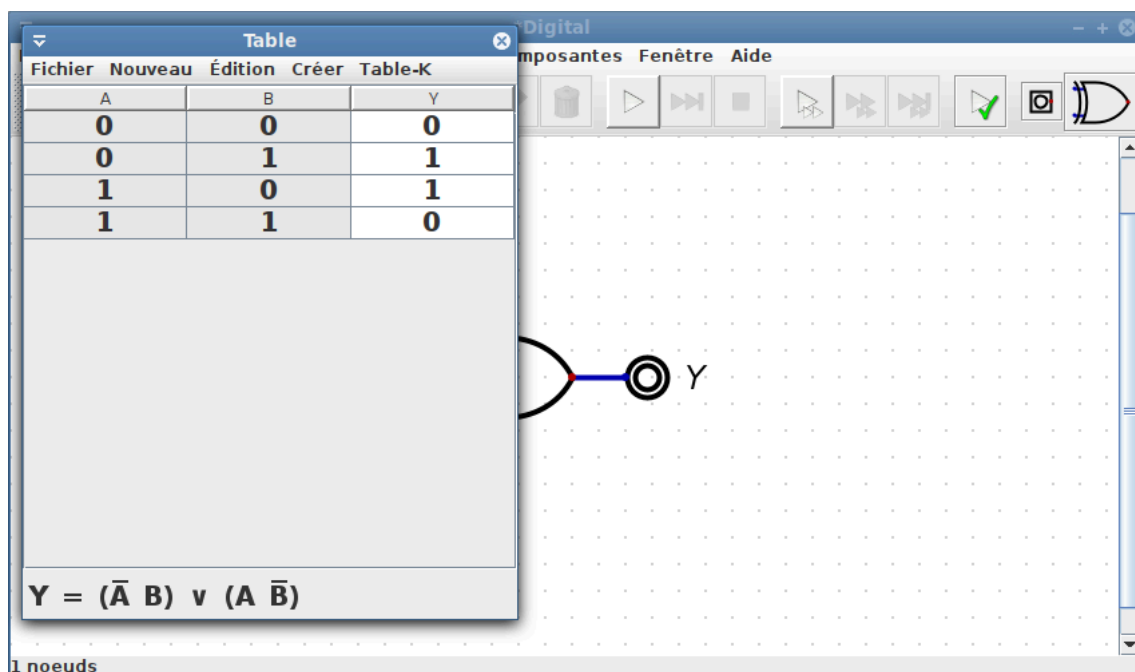
En cliquant avec la souris, les entrées peuvent être changées. Puisque la simulation est maintenant active, la sortie change selon l'état des entrées. Le circuit se comporte comme une porte Ou-Exclusif comme attendu.



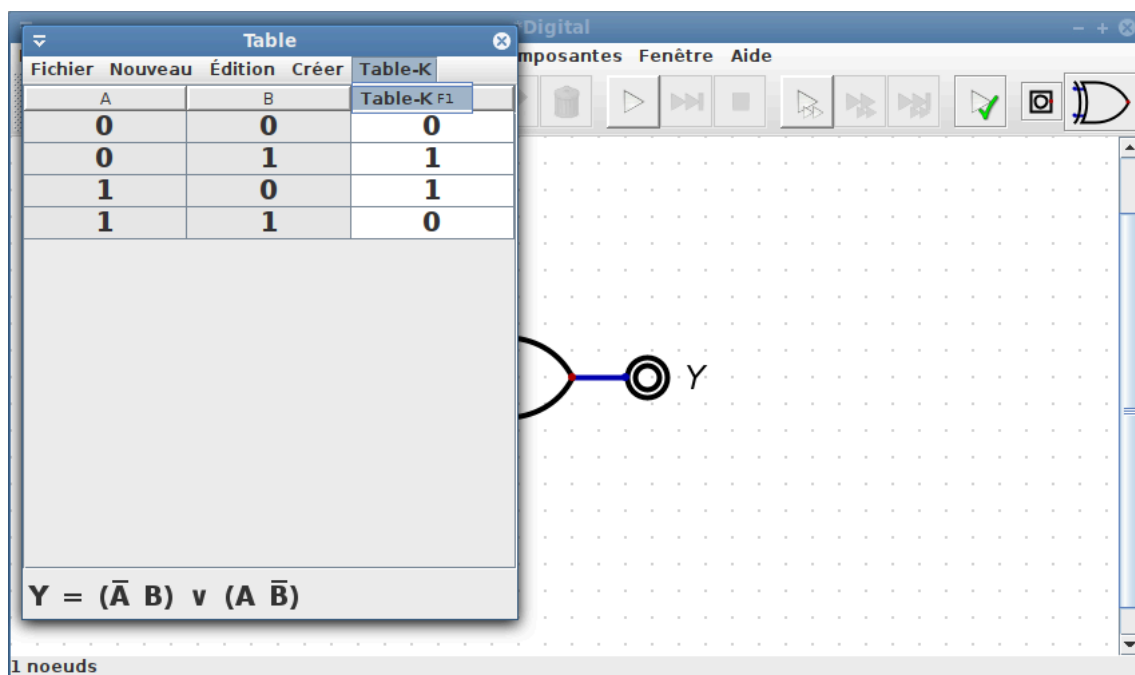
Pour améliorer le circuit davantage, la simulation doit d'abord être interrompue. La façon la plus facile de le faire est d'appuyer sur le bouton Arrêt dans la barre d'outils. En cliquant sur une composante avec le bouton droit de la souris (Ctrl+Clic sur macOS), un dialogue s'ouvre et affiche les propriétés de la composante. Une étiquette 'A' peut être définie pour cette première entrée avec le dialogue.



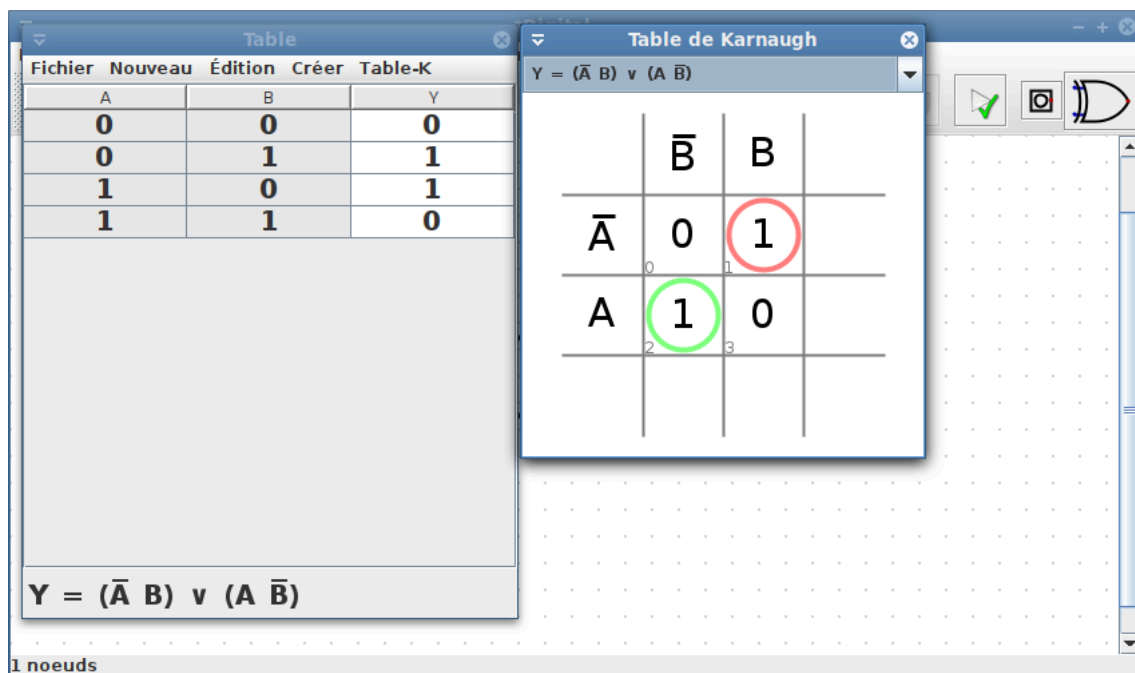
De la même façon, les étiquettes pour les autres entrées et la sortie peuvent être définies. Le menu *Analyse* contient un élément du même nom. Cette fonction effectue une analyse du circuit en cours. Cependant, il est seulement possible de la faire si toutes les entrées et sorties sont étiquettes correctement.



La table de vérité du circuit simulé apparaît dans une nouvelle fenêtre. Sous la table, vous pouvez voir l'expression algébrique associée au circuit. S'il y a plusieurs expressions possibles, une fenêtre séparée apparaîtra, montrant toutes les expressions possibles.



Le dialogue de la table a une entrée *Table-K* dans son menu principal. Celle-ci permet d'afficher la table de vérité sous la forme d'une table de Karnaugh.



Au haut de ce dialogue, une liste déroulante est présente, ce qui permet de sélectionner l'expression qui devrait s'afficher dans la table de Karnaugh. De cette manière, il est possible par exemple d'afficher plusieurs expressions algébriques équivalentes. Cependant, dans cet exemple, une seule expression minimale existe. La table de vérité peut aussi être modifiée en cliquant sur la table de Karnaugh.

### 1.3. Câblage

Toutes les composantes doivent être connectées à l'aide de fils. Il n'est pas possible de connecter deux composantes en les plaçant l'une à côté de l'autre.

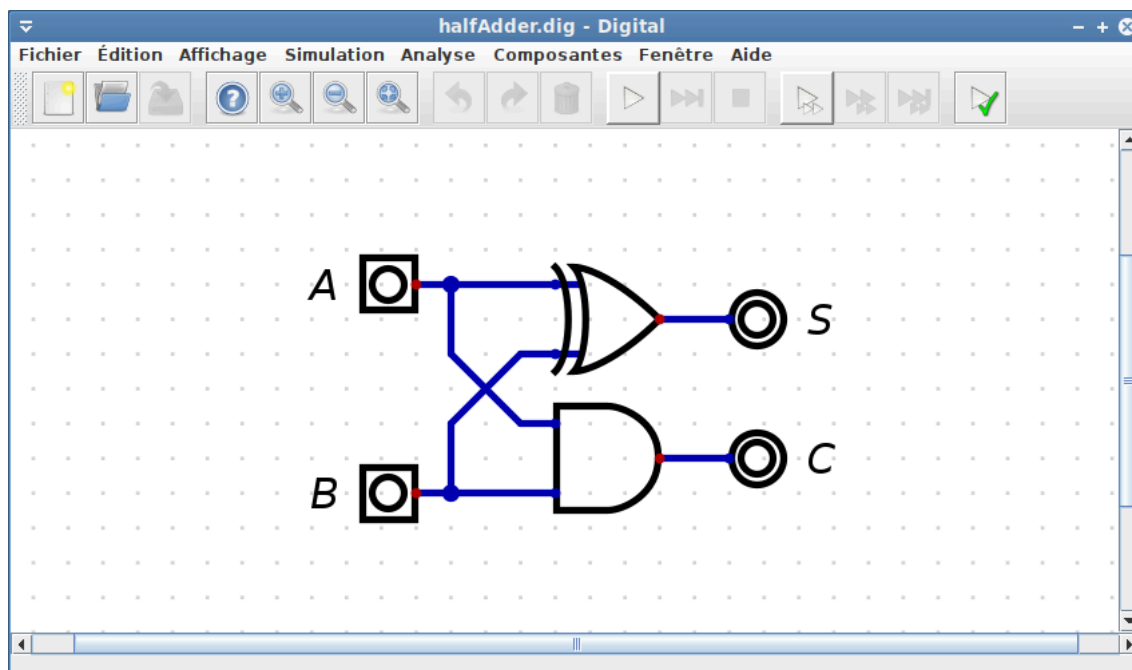
De plus, des connexions existent seulement entre les extrémités d'un fil et la composante. Si une broche d'une composante est placée au milieu d'un fil, aucune connexion n'est faite entre la composante et le fil. Ainsi, un fil doit vraiment se terminer à chaque broche qui doit être connectée. Même quand la composante de tunnel est utilisée, il doit y avoir un fil entre la broche et le tunnel.

La composante doit être sélectionnée avec une sélection rectangulaire afin de la déplacer en même temps ses fils. Pour déplacer une composante sans ses fils, sélectionnez la composante avec un clic de la souris et déplacez-la.

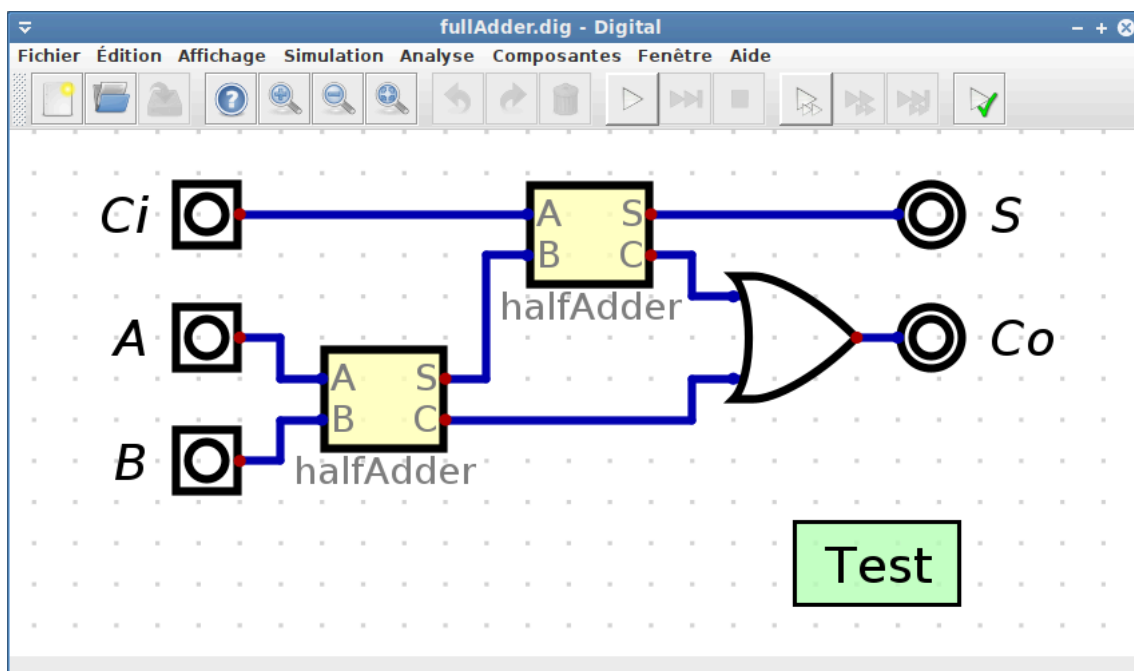
En faisant Ctrl+Clic, une seule section d'un fil peut être sélectionnée, pour la déplacer ou la supprimer. Si la touche D est enfoncée en dessinant un fil, un fil diagonal peut être dessiné. La touche S permet de séparer un segment de ligne en deux segments.

### 1.4. Design hiérarchique

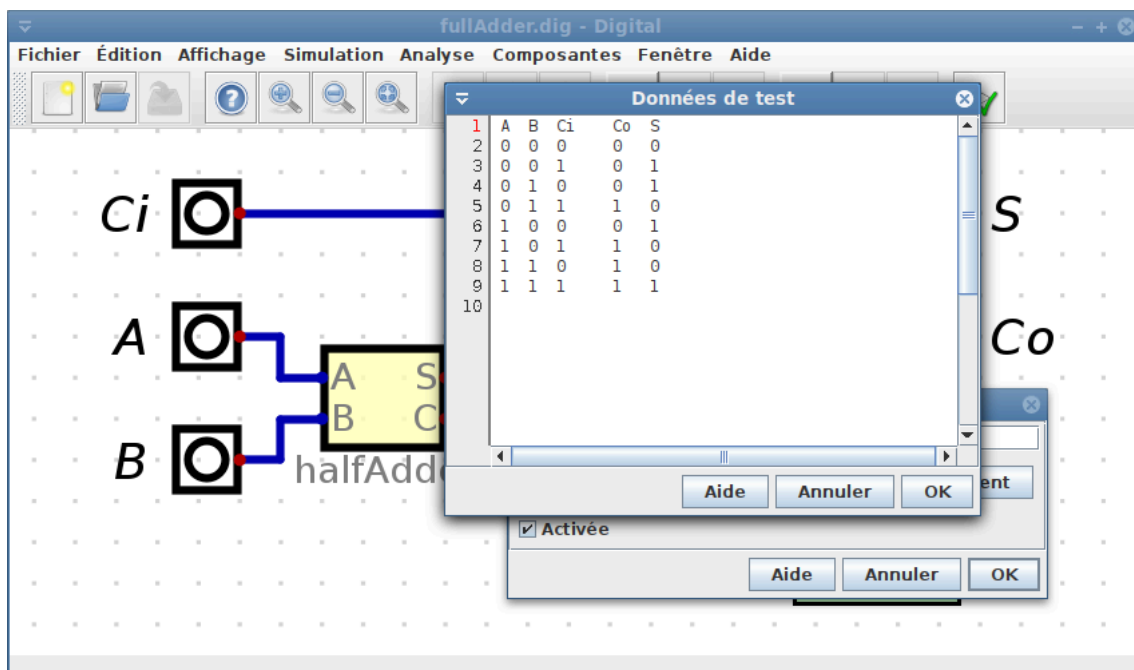
Si un circuit complexe doit être construit, cela peut rapidement devenir très frustrant. Pour garder le cap, les différentes parties du circuit peuvent être stockées dans des fichiers différents. Ce mécanisme permet aussi d'utiliser un sous-circuit qui a été créé une seule fois à de multiples reprises dans un autre circuit. Cette approche possède également l'avantage que les fichiers peuvent être stockés indépendamment les uns des autres dans un système de gestion des versions pour pouvoir faire un suivi des modifications.



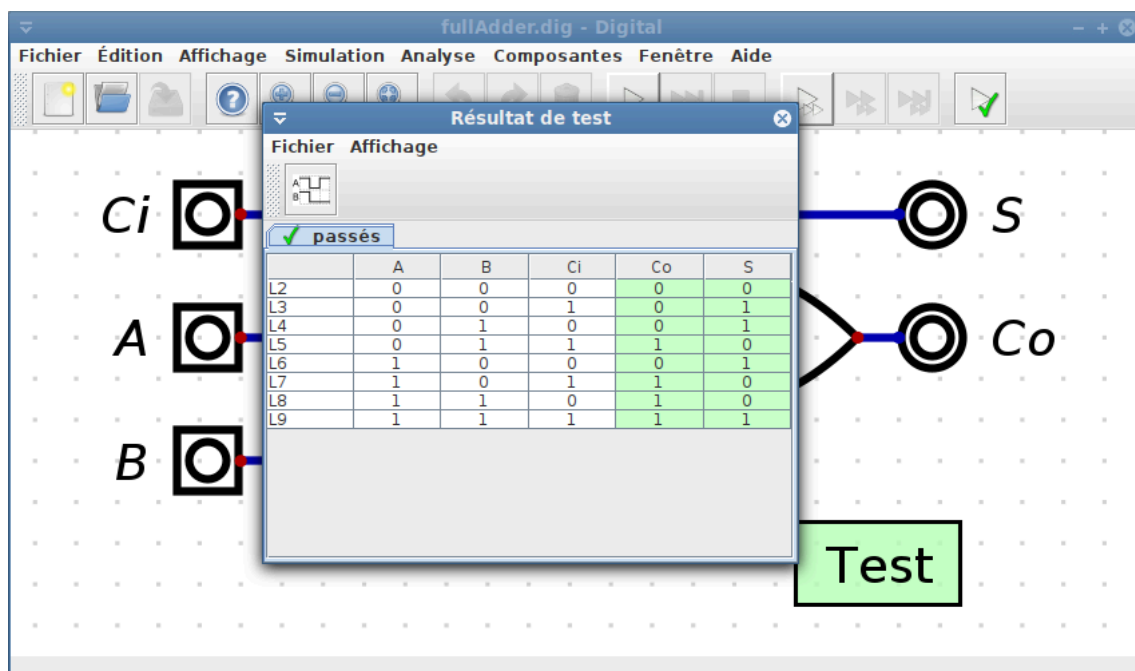
Comme exemple, considérez un additionneur 4 bits: d'abord, un demi-additionneur simple est construit. Celui-ci consiste en une porte XOR et une porte ET. La somme deux des bits 'A' et 'B' est donnée par les sorties 'S' et 'C'. Le circuit est stocké dans le fichier *halfAdder.dig*.



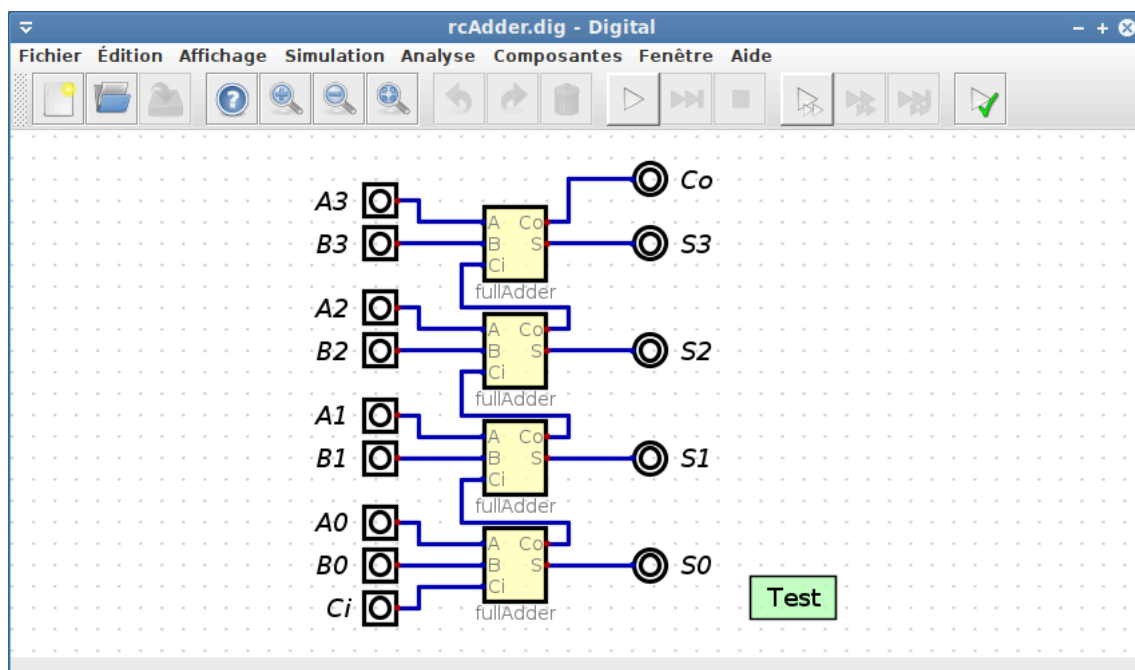
À partir de deux demi-additionneurs, un additionneur complet peut être construit. Pour ce faire, créez un nouveau fichier et enregistrez-le sous le nom *fullAdder.dig* dans le même dossier que le demi-additionneur. Le demi-additionneur peut ensuite être ajouté au circuit grâce le menu *Composantes* → *Personnalisé*. L'ordre des broches du boîtier du demi-additionneur peut être réarrangé à partir du circuit du demi-additionneur dans le menu *Édition* → *Ordonner les entrées* et *Édition* → *Ordonner les sorties*. L'additionneur complet additionne trois bits 'A', 'B' et 'Ci' et donne la somme aux sorties 'S' et 'Co'.



Pour vérifier le bon fonctionnement de l'additionneur complet, un test devrait être ajouté. Dans le test, une table de vérité est stockée et devrait correspondre à celle du circuit. Ainsi, il est possible de déterminer automatiquement si celui-ci fonctionne.

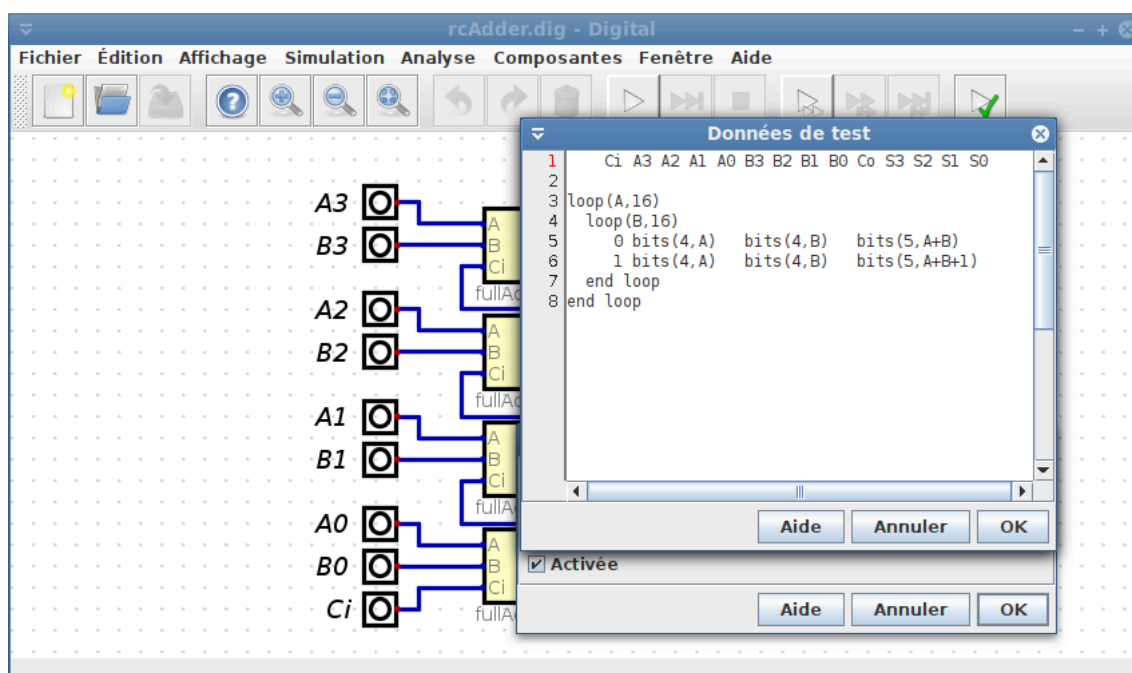


Les tests peuvent être exécutés à partir de l'éditeur de test ou avec le bouton de test dans la barre d'outils. Les cellules en vert dans la table de vérité indiquent que la sortie du circuit correspond à la table de vérité donnée pour ce test.



Maintenant, les additionneurs complets peuvent être assemblés pour former ce qu'on appelle un additionneur à propagation de retenue. Dans ce cas-ci, la sortie de retenue d'une addition est propagée comme retenue d'entrée de l'addition du bit suivant, comme c'est le cas pour une addition sur papier. Cet additionneur 4 bits devrait être testé pour vérifier son bon fonctionnement. Pour cette raison, un test a été inséré.





Ce test effectue un test à 100%, ce qui est seulement possible avec le circuit relativement simple: toutes les 512 combinaisons sont appliquées au circuit et la sortie est vérifiée. La première ligne liste les signaux d'entrée et de sortie. Sous celle-ci, les valeurs des entrées à appliquer et les valeurs de sorties à vérifier sont spécifiées dans une ligne, comme dans une table de vérité. Dans cet exemple toutefois, 512 lignes sont requises. Les entrer à la main serait une tâche pénible et susceptible aux erreurs. Il est plus facile et plus fiable de générer automatiquement les lignes nécessaires. Pour ainsi faire, les variables  $A$  et  $B$  sont chacune traversées de 0 à 15. Les valeurs respectives de  $A$  et de  $B$  sont ensuite affectées aux entrées ' $A[n]$ ' et ' $B[n]$ '. On regarde ensuite si le circuit émet la valeur  $A+B$ . Puis, le même test est repris mais avec le bit de retenue mis, auquel cas le résultat doit être  $A+B+1$ . Les détails de la syntaxe des tests sont fournis dans le dialogue d'aide.

Si un circuit est intégré dans un autre circuit, seulement le nom du fichier du sous-circuit est stocké dans le circuit, pas le circuit lui-même. Les fichiers correspondants aux sous-circuits doivent donc pouvoir être trouvés sur le système durant le déroulement de la simulation. Afin de permettre les différentes méthodes de travail des utilisateurs aussi bien que possible et tout de même éviter une administration complexe des chemins d'importation, une approche inhabituelle est implantée.

Seulement le nom du fichier des circuits intégrés est stocké dans le fichier des circuits, pas le chemin complet. Si un fichier doit être ouvert, tous les dossiers et les sous-dossiers sont scannés pour trouver un fichier au nom correspondant. Si un tel fichier est trouvé, il est importé. Ce procédé dépend seulement du nom du fichier à lire, pas de son emplacement. Ainsi, un message d'erreur est affiché s'il y a plusieurs fichiers du même nom dans différents sous-dossiers, puisque cela donne lieu à des ambiguïtés.

Une structure de projet appropriée ressemble donc à ceci: le circuit principal est situé dans son propre dossier. Tous les circuits importés doivent être situés dans le même dossier ou un de ses sous-dossiers. Tous les circuits ont des noms différents, donc il ne doit pas avoir de circuits avec un nom identique dans différents dossiers.

## 2. Simulation

### 2.1. Délais de propagation

Pendant la simulation, chaque porte logique possède un délai de propagation. Chaque composante trouvée dans la bibliothèque possède le même délai, peu importe sa complexité. Le porte ET possède donc le même délai qu'un multiplicateur. Les seules exceptions sont les diodes, les interrupteurs et les séparateurs utilisés pour créer des bus de données. Ces dernières composantes n'ont aucun délai.

S'il est nécessaire de simuler une porte, par exemple le multiplicateur, avec un délai de propagation supérieur, une composante de délai peut être insérée dans le circuit juste après la sortie du multiplicateur.

Si le circuit est inclus dans un autre circuit, le circuit inclus garde le même comportement temporel. Donc si un circuit complexe est inclus, celui-ci possédant un grand délai de propagation dû au fait que les signaux d'entrées doivent passer à travers trois portes avant d'atteindre la sortie, ce comportement est conservé après l'inclusion du circuit. Il n'y a pas de délai additionnel dû à l'inclusion du circuit. Si les entrées du circuit inclus ne possèdent pas toutes le même délai, alors c'est aussi le cas lorsque ce circuit est inclus dans un autre. Bref, en général, inclure un circuit dans un autre ne modifie pas son comportement temporel. Au contraire, celui-ci se comporte exactement de la même façon que si les composantes qui composent le circuit avaient été insérées directement.

## 3. Analyse

### 3.1. Analyse et synthèse de circuits

Un circuit peut être analysé grâce au menu *Analyse*. Une table de vérité est générée pour les circuits purement combinatoires. La table de vérité peut être éditée au besoin. Puis, il est possible de générer un nouveau circuit après l'édition de cette table de vérité.

En plus des circuits purement combinatoires, il est également possible d'analyser et de générer des circuits séquentiels. Au lieu d'une simple table de vérité, ce qu'on appelle une table de transition d'états est créée. Chaque bascule apparaît alors du côté des entrées et des sorties de la table. Dans cette table, on trouve sur le côté droit le prochain état qui se produira au prochain coup d'horloge. Cet état dépend de l'état actuel des bascules comme on peut les trouver sur le côté gauche de la table. Pour qu'une analyse soit possible, les bascules doivent être nommées.

La convention d'appellation suivante s'applique: le prochain état d'un bit, sur le côté droit de la table, est indiqué par le suffixe 'n+1'. L'état actuel correspondant est indiqué par le suffixe 'n'. S'il y a une variable d'état nommée 'A', alors 'An' indique l'état actuel et 'An+1' l'état suivant. Ainsi, si jamais une table de vérité suit le même modèle sur son côté gauche et droit, alors on suppose que la table est une table de transition d'états, et un circuit séquentiel est généré au lieu d'un circuit combinatoire.

Il est important de noter que le circuit à analyser ne peut contenir que des éléments combinatoires et les bascules D et JK. Si par exemple une bascule est implantée avec des portes Non-Ou, alors elle ne sera pas reconnue comme une bascule. Il n'est donc pas possible d'analyser un tel circuit.

## 3.2. Expression

Avec le menu *Expression*, il est possible d'entrer une fonction booléenne à partir de laquelle un circuit peut être généré.

## 3.3. Diagrammes d'états

Un éditeur pour les diagrammes d'états est disponible dans le menu *Automate fini*. Il permet la création graphique d'automates finis en dessinant les états et leurs transitions. Ainsi, différentes sorties peuvent être mises pour différents états. En donnant des transitions avec des conditions, des signaux d'entrée peuvent être générés. En donnant des valeurs de sortie aux transitions, il est aussi possible de définir des machines de Mealy.

L'automate fini défini de cette façon peut être automatiquement transféré vers une table de transition d'états, qui, en une étape de plus, peut générer un circuit qui implante l'automate initial. La simulation de ce circuit est ensuite lancée et l'état actuel peut être suivi dans le diagramme d'états.

## 4. Matériel

### 4.1. GAL16v8 et GAL22v10

Dans le menu de génération de circuit à partir de la table de vérité, il y a aussi des fonctions qui permettent de générer des fichiers JEDEC. Ce format de fichier particulier décrit le plan des fusibles d'un PLD. Ce fichier JEDEC peut ensuite être écrit dans un PLD correspondant avec un programmeur approprié. Pour le moment, seuls les circuits du type *GAL16v8* et *GAL22v10* ou des appareils avec des plans de fusible compatibles sont pris en charge.

### 4.2. ATF150xAS

Les puces de la famille *ATF150x* sont des CPLDs simples avec jusqu'à 128 macrocellules. Elles sont disponibles dans des boîtiers PLCC, ce qui les rend adaptées aux exercices de laboratoire: si une puce est détruite durant les exercices, elle peut simplement être remplacée. De plus, avec le *ATDH1150USB*, un programmeur facile d'utilisation et abordable est disponible. Ce programmeur est en mesure de programmer les puces *ATF150x* en utilisant l'interface JTAG. Une plaque d'évaluation appropriée (*ATF15XX-DK3-U*) est aussi disponible. Le logiciel *ATMISP*, qui est disponible sur le site d'ATMEL/Microchip est requis pour programmer les puces.

Malheureusement, les détails des plans de fusibles ne sont pas publiquement disponibles, alors aucun ajusteur approprié à cette puce ne peut être intégré à Digital, comme c'est possible avec les puces *GAL16v8* et *GAL22v10*.

Ainsi, les ajusteurs *fit150[x].exe* fournis par ATMEL doivent être utilisés. Ces programmes créent un fichier JEDEC à partir d'un fichier *TT2* approprié, qui peut ensuite être programmé sur la puce. Digital lance l'ajusteur automatiquement à chaque fois qu'un fichier *TT2* est créé. Pour cette raison, le chemin vers l'exécutable *fit150[n].exe* doit être spécifié dans les paramètres. Le fichier *JEDEC* créé peut être ouvert et programmé directement avec *ATMISP*.

Pour des raisons légales, l'ajusteur *fit1502.exe* ne peut pas être distribué avec Digital. Cependant, il peut être trouvé dans le dossier *WinCupl\Fitters* après avoir installé *WinCupl*. *WinCupl* est disponible sur le site d'ATMEL/Microchip. Sur des systèmes Linux, les ajusteurs peuvent aussi être exécutés par Digital si *wine* est installé.

### 4.3. Exportation en VHDL ou Verilog

Un circuit peut être exporté en VHDL ou en Verilog. Le fichier généré contient une description complète du circuit. Le code VHDL généré a été testé avec Xilinx Vivado et le simulateur VHDL open source ghdl. Le code Verilog a été testé avec le simulateur Icarus Verilog.

Si un circuit contient des tests, les données de test sont utilisées pour générer un banc de test HDL. Cela permet de vérifier le bon fonctionnement du circuit dans la simulation HDL.

Les fichiers additionnels requis par certaines plaquettes peuvent être créés. Pour l'instant, seulement la plaquette BASYS3 et les plaquettes Mimas et Mimas V2 sont prises en charge. Un fichier de contraintes est créé, qui contient l'affectation des broches. La description des broches peut être trouvée dans la fiche technique des plaquettes, et doit être entrée comme numéro de broche pour les entrées et les sorties.

Pour une plaquette BASYS3, si la fréquence de l'horloge du circuit est basse, un diviseur de fréquence est intégré au code HDL pour diviser l'horloge de la plaquette en conséquence. Si la fréquence sélectionnée dans le circuit dépasse 4.7MHz, l'unité MMCM du Artix-7 est utilisée pour la génération de l'horloge. Cela permet de s'assurer que les ressources fournies par le FPGA pour la distribution de l'horloge sont utilisées. Cela permet au processeur inclus en exemple de fonctionner à 20MHz, et si vous pouvez faire sans l'instruction de multiplication, 30MHz est aussi possible.

Si un circuit doit s'exécuter sur une plaquette BASYS3, un nouveau projet peut être créé dans Vivado. Le fichier VHDL généré et le fichier de contraintes doivent être ajoutés au projet. Une fois que le programme a été créé, le flux de bits peut être généré et le gestionnaire de matériel peut être utilisé pour programmer la plaquette BASYS3.

Pour créer le fichier de contraintes requis en plus du fichier HDL, la plaquette correspondante doit être configurée dans les paramètres. Dans le champ "Configuration de chaîne d'outils", le fichier XML correspondant peut être sélectionné. Les configurations disponibles peuvent être trouvées dans le dossier *examples/hdl* et ont l'extension de fichier *.config*. Si la configuration est intégrée avec succès, un menu additionnel s'affiche, ce qui rend disponibles les fonctions spécifiques à la plaquette.

## 5. Formes personnalisées

Même si Digital possède quelques options pour déterminer l'apparence d'un circuit lorsque celui est intégré dans un autre, dans certains cas il peut être utile d'utiliser une forme particulière pour un sous-circuit. Un exemple de cela est pour la représentation de l'ALU dans le processeur inclus dans les exemples. Ce chapitre explique comment définir une telle forme pour un circuit.

Digital ne fournit pas d'éditeur pour créer une forme particulière. À la place, un petit détour est nécessaire pour créer des formes de circuit: d'abord, le circuit qui est à représenter par une forme particulière est ouvert. Ensuite, un modèle SVG est créé pour le circuit. Dans ce modèle, le circuit est représenté par un simple rectangle. Le modèle contient aussi toutes les broches du circuit, représentées par des cercles bleus (entrées) et rouges (sorties). Pour voir quel cercle appartient à quelle broche du circuit, vous pouvez regarder l'ID du cercle dans les propriétés de l'objet. L'ID a la forme *pin:[name]* ou *pin+: [name]*. Dans la seconde variante, une étiquette sera fournie à la broche lorsque le modèle sera réimporté dans Digital. Si vous ne voulez pas de telle étiquette, le + peut être supprimé.

Le fichier SVG peut maintenant être édité. Le programme open source le plus adapté est Inkscape, qui est disponible gratuitement. Les broches peuvent être déplacées librement, mais elles seront déplacées à la jonction de grille la plus proche lors de la réimportation.

Si un fichier SVG existant doit être utilisé, il est plus facile d'ouvrir le modèle créé et de coller la forme existante dans le modèle avec le copier/coller.

Après que le fichier a été enregistré, il peut être importé avec Digital. Le fichier est lu et toute l'information nécessaire est extraite et stockée dans le circuit. Le fichier SVG n'est plus nécessaire pour pouvoir utiliser la nouvelle forme du circuit.

Une dernière remarque: le format SVG est un format de fichier très puissant et flexible. Il peut être utilisé pour décrire des graphiques extrêmement complexes. Cependant, l'importeur de Digital n'est pas en mesure d'importer tous les fichiers SVG possibles sans erreurs. Si un fichier ne peut pas être importé, ou s'il n'apparaît pas comme attendu, il est possible que certains ajustements soient requis pour atteindre le résultat escompté.

## 6. Circuits génériques

Il arrive qu'un circuit doive être créé afin d'être utilisé sous différentes variantes. Par exemple, on peut imaginer un compteur particulier qui doit être utilisé avec différentes largeurs binaires. S'il fallait créer un circuit indépendant pour 4, 5 et 6 bits, l'entretien de ces circuits serait difficile dans le futur, puisqu'on devrait toujours travailler sur chaque variante séparément, même si ceux-ci sont identiques mis à part un paramètre, la largeur binaire.

Pour empêcher cela, il est possible de créer un sous-circuit générique qui pourra être paramétré. Pour cela, la case "Circuit générique" peut être cochée dans les paramètres spécifiques au circuit. Par la suite, le dialogue d'attributs pour chaque composante de ce circuit contiendra un champ additionnel nommé "paramétrisation générique". Dans ce champ, un code de programme peut être entré, celui-ci servant à changer les paramètres de la composante. Chaque paramètre a un nom et peut être modifié comme un attribut du champ *this*. Le nom des paramètres peut être trouvé dans le dialogue d'aide de la composante. Pour changer la largeur binaire d'un additionneur, la ligne *this.Bits=1;* peut être utilisée.

Cependant, cela en tant que tel ne permet pas encore de paramétrer le circuit. Il est encore nécessaire de pouvoir accéder aux paramètres établis lorsque le circuit est utilisé. Ceci est fait grâce au champ "args". Si vous voulez accéder à la largeur binaire de l'extérieur, vous pouvez écrire: *this.Bits=args.largeurBinaire;*. Le nom de l'argument, *largeurBinaire* ici, est arbitraire. Si le sous-circuit est utilisé, cet argument doit être mis.

Si le circuit est utilisé et le dialogue d'attributs du circuit intégré est ouvert, celui-ci possède également un champ "paramétrisation générique". Ici, la largeur binaire désirée peut être mise avec l'instruction *largeurBinaire:=5;*.

De cette manière cependant, aucun fil ne peut être supprimé ou ajouté. Il existe tout de même un truc pour réaliser plusieurs circuits. Ce truc consiste à remplacer un circuit par un autre, selon les arguments. Pour ce faire, la fonction *setCircuit([Name])* est disponible. Si elle est appelée dans la définition du sous-circuit, le circuit à insérer peut être remplacé par un autre circuit. Cela permet une définition récursive d'un circuit. Comme dans d'autres langages de programmation, il faut veiller à avoir une condition de fin appropriée.

Le dossier *examples/generic* contient un exemple d'un compteur en code Gray pour lequel la largeur binaire peut être configurée. Dans cet exemple, le compteur est construit en ajoutant récursivement plus de bits à un circuit initial jusqu'à ce que le nombre de bits requis soit atteint.

## 7. Tests contrôlés par script

Si les étudiants doivent compléter des exercices avec Digital, il peut être pratique que les circuits soumis par les étudiants soient vérifiés automatiquement par un processus automatique.

Pour effectuer cette vérification, Digital peut être appelé à partir d'une ligne de commande. Cet appel se fait comme suit:

```
java -cp Digital.jar CLI test [fichier à tester] [-tests [fichier optionnel contenant les tests]]
```

Si uniquement le fichier à tester est spécifié, alors les tests dans ce fichier sont exécutés. De cette façon, les tests créés par les étudiants eux-mêmes peuvent être exécutés.

Si un second fichier est spécifié, les tests seront pris dans le second fichier et le premier circuit sera testé avec ces tests. Le second fichier contiendra habituellement la solution, dans laquelle les tests sont complets et corrects. Le circuit contenu dans le second fichier en tant que tel est ignoré. Seulement les tests qu'il contient sont utilisés.

Dans le but de tester un circuit avec un exemple de solution, le nom des signaux des entrées et des sorties dans les deux circuits doit concorder.

## 8. Foire aux questions

### Comment déplacer un fil?

Sélectionnez une des extrémités du fil avec une sélection rectangulaire, puis déplacer ce point avec la souris. Un fil peut aussi être sélectionné avec Ctrl+Clic.

### Comment supprimer un fil?

Sélectionnez une des extrémités du fil et appuyez sur *Suppr* ou cliquez sur la corbeille. Un fil peut aussi être sélectionné avec Ctrl+Clic.

### Comment déplacer une composante avec tous les fils connectés?

Sélectionnez la composante avec une sélection rectangulaire. La sélection doit inclure la composante en entier. Puis, déplacer la composante et ses fils avec la souris.

### Il y a une composante qui n'est pas connectée à un fil, même si les broches sont sur le fil.

Une broche est seulement connectée à un fil si le fil possède une extrémité sur cette broche.

### Si les noms des broches d'un circuit sont trop longs, les noms ne sont plus lisibles lorsque le circuit est intégré. Qu'est-ce que je peux faire?

La largeur du bloc peut être augmentée dans le menu *Édition* → *Paramètres spécifiques au circuit*.

### Les broches dans un circuit intégré ont un ordre peu optimal. Comment le changer?

L'ordre des broches peut être changé dans le menu *Édition* → *Ordonner les entrées* ou *Édition* → *Ordonner les sorties*.

### Lorsque la simulation a débuté, les fils deviennent gris. Qu'est-ce que cela signifie?

Les couleurs vert pâle et vert foncé sont utilisées pour représenter les états haut et bas. Le gris signifie que le fil est dans un état de haute impédance.

### J'ai une table de vérité. Comment calculer les expressions booléennes minimales?

Dans le menu *Analyse*, sélectionnez l'élément *Synthétiser*. Ensuite, entrez la table de vérité. Au bas de la fenêtre, vous trouverez les expressions booléennes correspondantes. Si vous entrez plus d'une variable dépendante, une nouvelle fenêtre apparaît dans laquelle toutes les expressions se trouvent.

**J'ai entré une table de vérité, mais plus d'une équation booléenne est affichée.****Laquelle est la bonne?**

La minimisation d'une équation booléenne peut donner plusieurs équations qui décrivent la même fonction. Digital les affiche toutes, et celles-ci créent toutes la même table de vérité. Il peut y avoir des différences en fonction de si la table de vérité contient des cas indéfinis (les X).

**J'ai une table de vérité. Comment créer un circuit qui représente cette table de vérité?**

Dans le menu *Analyse*, sélectionnez l'élément *Synthétiser*. Ensuite, entrez la table de vérité. Vous pouvez éditer la table en utilisant les menus *Nouveau* ou *Édition*. Dans le menu *Créer*, vous pouvez créer un circuit avec l'élément *Circuit*.

**Comment éditer le nom d'un signal dans la table de vérité?**

Faites un clic droit sur le nom du signal dans l'en-tête de la table pour éditer son nom.

**J'ai une équation booléenne. Comment créer le circuit?**

Dans le menu *Analyse*, sélectionnez l'élément *Expression*, puis entrez l'équation.

**Comment créer une table de vérité à partir d'une équation booléenne?**

Dans le menu *Analyse*, sélectionnez l'élément *Expression*, puis entrez l'équation. Ensuite, créer le circuit et utiliser l'élément *Analyse* du menu *Analyse* pour créer la table de vérité.

**Comment créer un fichier JEDEC à partir d'un circuit donné?**

Dans le menu *Analyse*, sélectionnez l'élément *Analyse*. Puis, dans la nouvelle fenêtre, un sous-menu *Périphérique* dans le menu *Créer* permet de sélectionner le bon périphérique.

**Lors de création d'un fichier JEDEC, comment affecter un numéro broche à un signal?**

Dans les entrées et les sorties correspondantes, vous pouvez entrer un numéro de broche dans le dialogue des attributs de cette broche.

**J'ai créé un fichier JEDEC. Comment le programmer sur un GAL16v8 ou un GAL22v10?**

Pour programmer une telle plaquette, du matériel de programmation spécialisé est requis.

**J'ai créé un circuit et je veux l'utiliser dans d'autres circuits. Comment faire sans avoir à copier le fichier encore et encore dans chaque dossier approprié?**

Le circuit peut être enregistré dans le dossier "lib". Il sera alors disponible dans tous les autres circuits.

## 9. Raccourcis claviers

<b>Espace</b>	Lance ou arrête la simulation.
<b>F6</b>	Affiche le dialogue du tableau des mesures.
<b>F7</b>	Exécuter jusqu'à une pause.
<b>F8</b>	Exécuter les tests.
<b>C</b>	Un seul pas d'horloge (fonctionne seulement lors d'une simulation où il n'y a qu'une horloge).
<b>V</b>	Calculer le résultat d'un seul pas.
<b>B</b>	Exécuter tous les pas jusqu'à ce que le circuit se soit stabilisé. Si une composante de pause est présente, alors jusqu'à une pause.
<b>F9</b>	Analyse du circuit.
<b>CTRL-A</b>	Tout sélectionner.
<b>CTRL-X</b>	Couper les composantes sélectionnées dans le presse-papier.
<b>CTRL-C</b>	Copier les composantes sélectionnées dans le presse-papier
<b>CTRL-V</b>	Insérer des composantes à partir du presse-papier.
<b>CTRL-D</b>	Reproduire la sélection en cours sans utiliser le presse-papier.
<b>R</b>	Lors de l'insertion, faire pivoter la composante.
<b>L</b>	Insérer la dernière composante une autre fois.
<b>T</b>	Insérer un nouveau tunnel.
<b>CTRL-N</b>	Nouveau circuit.
<b>CTRL-O</b>	Ouvrir un circuit.
<b>CTRL-S</b>	Enregistrer le circuit.
<b>CTRL-Z</b>	Annuler la dernière modification.
<b>CTRL-Y</b>	Rétablir la dernière modification annulée.
<b>P</b>	Programmer une diode ou un FGFET.
<b>D</b>	Lors du dessin d'un fil, changer pour le mode diagonal.
<b>F</b>	Lors du dessin d'un fil, changer l'orientation.
<b>S</b>	Séparer un fil en deux fils.
<b>Échap</b>	Interrompre l'action en cours.
<b>Suppr</b>	Supprimer les composantes sélectionnées.
<b>Retour arr.</b>	Supprimer les composantes sélectionnées.
<b>+</b>	Augmente le nombre d'entrées de la composante sous la souris. Si utilisé sur des constantes, la valeur est incrémentée.
<b>-</b>	Réduit le nombre d'entrées de la composante sous la souris. Si utilisé sur des constantes, la valeur est décrémentée.



<b>CTRL +</b>	Zoom avant.
<b>CTRL -</b>	Zoom arrière.
<b>F1</b>	Ajuster à la fenêtre.
<b>F5</b>	Afficher ou cacher l'arborescence des composantes.

## B Paramètres

Le texte suivant décrit les paramètres disponibles dans le simulateur.

### Paramètres

Les paramètres globaux du simulateur spécifient, entre autres, la langue, la forme des symboles à utiliser, ou les chemins des outils externes.

#### Attributs

Utiliser les formes IEEE 91-1984

Utiliser les formes IEEE 91-1984 au lieu des formes rectangulaires.

Langue

Langue de l'interface. Prend effet seulement après un redémarrage.

Format

Format des expressions à l'écran

Thème de couleur

Thème de couleur

Couleurs personnalisées

Couleurs personnalisées

Afficher l'arborescence des composantes au démarrage.

Si coché, l'arborescence des composantes est affichée au démarrage.

Afficher la grille

Afficher la grille dans la fenêtre principale.

Afficher le nombre de fils sur un bus.

ATTENTION: La valeur est seulement mise à jour lorsque la simulation débute.

Surlignement des fils

Si coché, les lignes de fils sont surlignées lorsque la souris est par-dessus.

Utiliser la touche égal

Utiliser la touche égal au lieu de la touche plus. Cela est toujours utile si le caractère plus n'est pas une touche primaire, mais une affectation secondaire de la touche égal, par exemple sur les claviers américains et français.

Increase numbers in identifiers when copying.

If set, in components whose label ends with a number, this number is incremented when copying.

Afficher le dialogue pour renommer automatiquement les tunnels.

Si coché, un dialogue pour renommer automatiquement les tunnels du même nom sera affiché après qu'un tunnel ait été renommé.

Bibliothèque

Dossier qui contient la bibliothèque avec des sous-circuits prédéfinis. Contient par exemple les composantes de la série 74xx. Vous pouvez aussi ajouter vos propres circuits en les stockant à cet emplacement. Il faut s'assurer que le nom des fichiers dans ce dossier et ses sous-dossiers sont uniques.

Bibliothèque Java

Un fichier JAR qui contient des composantes additionnelles implantées en Java.

Ajusteur ATF15xx

Chemin vers l'ajusteur pour le ATF15xx. Entrer un répertoire qui contient les fichiers fit15xx.exe fournis par Microchip (anciennement ATMEL).

**ATMISP**

Chemin du fichier exécutable ATMISP.exe. Si coché, le programme ATMISP sera exécuté automatiquement!

**GHDL**

Chemin du fichier ghdl exécutable. Seulement nécessaire si vous voulez utiliser ghdl pour simuler des composantes définies avec VHDL.

**IVerilog**

Chemin du fichier exécutable d'Icarus Verilog. Seulement nécessaire si vous voulez utiliser IVerilog pour simuler des composantes définies avec Verilog.

**Configuration de la chaîne d'outils**

Sert à configurer l'intégration d'une chaîne d'outils. Permet d'exécuter des outils externes, par exemple pour programmer un FPGA ou autre.

**Taille de la police des menus [%]**

Taille des polices utilisées dans le menu en pourcentage de la valeur par défaut.

**Utiliser les clics de souris macOS.**

Utiliser Ctrl+Clic au lieu du clic droit.

**Permettre la connexion à distance**

Si coché, un port TCP/IP est ouvert, à travers lequel le contrôle du simulateur est possible.

**Numéro de port**

Le port sur lequel le serveur à distance est ouvert.

**Paramètres spécifiques au circuit**

Les paramètres spécifiques au circuit affectent le comportement du circuit présentement ouvert. Par exemple, la forme qui représente le circuit quand il est intégré dans d'autres circuits. Ces paramètres sont stockés avec le circuit.

**Attributs****Étiquette**

Le nom de cet élément.

**Largeur**

Largeur du symbole si le circuit est utilisé comme une composante dans un autre circuit.

**Couleur d'arrière-plan**

La couleur d'arrière-plan du circuit lorsqu'il est intégré dans un autre circuit. Elle n'est pas utilisée pour les boîtiers DIL.

**Description**

Une courte description de cet élément et son utilisation.

**Empêcher la modification**

Verrouille le circuit. Il est possible de configurer les diodes et les FGF-FETs.

**Forme**

La forme à utiliser pour la représentation du circuit lorsque celui-ci est intégré. Dans le mode "Simple", les entrées sont affichées à gauche et les sorties à droite dans un rectangle simple. Avec "Disposition", la position des entrées et des sorties ainsi que leur orientation dans le circuit détermine la position des broches. Avec "Boîtier DIL", un boîtier DIL est utilisé pour afficher le circuit. Les numéros de broches des entrées et des sorties déterminent la position des broches sur le boîtier.

**Forme personnalisée**

Importation d'un fichier SVG

**Hauteur**

Hauteur du symbole si le circuit est utilisé comme une composante dans un autre circuit.

**Nombre de broches DIL**

Nombre de broches. Un zéro signifie que le nombre de broches est déterminé automatiquement.

**Contenu des ROM.**

Contenu de toutes les ROMs utilisées.

**Afficher les valeurs de mesures au début de la simulation**

Lorsque la simulation débute, un tableau avec les valeurs mesurées est affiché.

**Afficher le graphique de mesures au début de la simulation**

Lorsque la simulation est débutée, un graphique avec les valeurs mesurées est affiché.

**Afficher le graphique de mesures en mode pas-à-pas au début de la simulation**

Lorsque la simulation est débutée, un graphique avec les valeurs mesurées en mode pas-à-pas est affiché. Tous les changements de portes sont inclus dans le graphique.

**Nombre maximal d'étapes à afficher**

Le nombre maximal de valeurs stockées. Si le nombre maximal est atteint, les valeurs plus vieilles disparaissent.

**Précharger le mémoire de programme au démarrage.**

Lors de la simulation d'un processeur qui utilise une composante de RAM comme mémoire de programme, il est difficile de lancer le processeur parce que le contenu de la RAM est toujours initialisé avec des zéros au début de la simulation. Cette option permet de charger les données dans la mémoire de programme au démarrage. La mémoire de programme dans le circuit doit être identifiée comme telle.

**Fichier de programme**

Fichier qui devrait être chargé dans la mémoire de programme au début de la simulation.

**Utiliser big endian lors de l'importation**

Utiliser l'ordre des octets big endian ("gros-boutiste") lors de l'importation

**Ignorer lors de l'exportation en Verilog/VHDL**

Ignore la génération des composants internes au circuit lors de l'exportation en Verilog ou VHDL. Les références au circuit sont gardées, permettant de fournir une implémentation différente.

**Circuit générique**

Permet de créer un circuit générique.

**Détection d'oscillation**

Nombre de délais équivalents au temps de propagation d'une porte logique à partir duquel une oscillation est détectée si le circuit n'est pas encore stable.

**In case of oscillations, continue with random values.**

If this option is set, the behavior of the simulation becomes unpredictable and extremely slow if there is an oscillation!

## C Interface de ligne de commande

`java -cp Digital.jar CLI`

`test -circ [String] [-tests [String]] [-allowMissingInputs] [-verbose]:`

Le premier nom de fichier spécifie le nom du circuit à tester. Si un second nom de fichier est spécifié, les tests sont exécutés à partir de ce fichier. Sinon, les tests sont exécutés à partir du premier fichier.

Options:

`-circ [String(def: )]`

Nom du fichier à être testé.

`[-tests [String(def: )]]`

Nom du fichier avec les tests.

`[-allowMissingInputs(def: false)]`

Permet l'omission d'entrées dans le circuit qui sont définies dans le test. Cela peut être utile s'il y a plusieurs solutions qui peuvent dépendre d'entrées différentes.

`[-verbose(def: false)]`

Si activé, la table des valeurs est émise en cas d'erreur.

`svg -dig [String] [-svg [String]] [-ieee] [-LaTeX] [-pinsInMathMode] [-hideTest] [-noShapeFilling] [-smallIO] [-noPinMarker] [-thinnerLines] [-highContrast] [-monochrome]:`

Peut être utilisé pour créer un fichier SVG à partir d'un circuit.

Options:

`-dig [String(def: )]`

Le nom de fichier du circuit.

`[-svg [String(def: )]]`

Le nom du fichier SVG à créer.

`[-ieee(def: false)]`

Utiliser les symboles IEEE.

`[-LaTeX(def: false)]`

Le texte est inséré en notation LaTeX. Inkscape est requis pour un traitement ultérieur.

`[-pinsInMathMode(def: false)]`

Pour les étiquettes de broches, utiliser le mode math même s'elles ne contiennent aucun indice.

`[-hideTest(def: false)]`

Cacher les tests

`[-noShapeFilling(def: false)]`

Les polygones ne sont pas remplis.

`[-smallIO(def: false)]`

Les entrées et les sorties sont représentées comme de petits cercles.

`[-noPinMarker(def: false)]`

Les marqueurs de broches bleus et rouges sur les symboles sont omis.

`[-thinnerLines(def: false)]`

Si coché, les lignes sont dessinées un peu plus minces.

`[-highContrast(def: false)]`

Les fils et le texte des broches sont affichés en noir.

`[-monochrome(def: false)]`

Seulement des nuances de gris sont utilisées.

`stats -dig [String] [-csv [String]]:`

Créer un fichier CSV qui contient les statistiques du circuit. Toutes les composantes sont listées dans le fichier CSV.

Options:

`-dig [String(def: )]`

Nom de fichier du circuit.

`[-csv [String(def: )]]`

Nom du fichier CSV à créer. Si cette option est manquée, la table de vérité est écrite à la sortie standard.

`run -dig [String]:`

Runs a circuit headless.

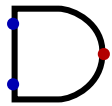
Options:

`-dig [String(def: )]`

File name of the circuit.

## D Composantes

### 1. Logique



#### 1.1. Et

Porte ET binaire. Retourne une valeur haute lorsque les deux entrées ont une valeur haute. Il est également possible d'utiliser des bus à plusieurs bits comme entrées et sortie. Dans ce cas, une opération bit à bit est effectuée. Cela équivaut à connecter chaque paire de bits d'entrée à une porte ET 1-bit et émettre le résultat correspondant pour chaque bit de sortie. Exportable en VHDL/Verilog.

##### Entrées

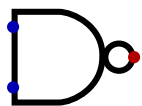
- In\_1  
La valeur d'entrée 1 pour l'opération logique.
- In\_2  
La valeur d'entrée 2 pour l'opération logique.

##### Sorties

- out  
Retourne le résultat de l'opération logique.

##### Attributs

- Bits de données  
Le nombre de bits utilisés pour les données.
- Nombre d'entrées  
Le nombre d'entrées utilisées. Chaque entrée doit être connectée.
- Entrées inversées  
Sélectionner les entrées qui doivent être inversées.
- Rotation  
L'orientation de l'élément dans le circuit.
- Forme large  
Utilise une forme plus large pour visualiser la porte.



#### 1.2. Non-Et

Une combinaison du ET et du NON. Retourne 0 lorsque toutes les entrées sont mises à 1. Si une des entrées est à 0 alors la sortie est mise à 1. Il est également possible d'utiliser des bus à plusieurs bits comme entrées. Dans ce cas, l'opération s'applique à chaque bit de l'entrée. Exportable en VHDL/Verilog.

### Entrées

In\_1

La valeur d'entrée 1 pour l'opération logique.

In\_2

La valeur d'entrée 2 pour l'opération logique.

### Sorties

out

Retourne le résultat de l'opération logique.

### Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Nombre d'entrées

Le nombre d'entrées utilisées. Chaque entrée doit être connectée.

Entrées inversées

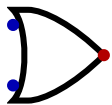
Sélectionner les entrées qui doivent être inversées.

Rotation

L'orientation de l'élément dans le circuit.

Forme large

Utilise une forme plus large pour visualiser la porte.



## 1.3. Ou

Porte OU binaire. Retourne un 1 lorsqu'au moins une entrée est mise à 1. Il est également possible d'utiliser des bus à plusieurs bits comme entrées et sortie. Dans ce cas, une opération bit à bit est effectuée. Cela équivaut à connecter chaque paire de bits d'entrée à une porte OU 1-bit et émettre le résultat correspondant pour chaque bit de sortie. Exportable en VHDL/Verilog.

### Entrées

In\_1

La valeur d'entrée 1 pour l'opération logique.

In\_2

La valeur d'entrée 2 pour l'opération logique.

### Sorties

out

Retourne le résultat de l'opération logique.

### Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Nombre d'entrées

Le nombre d'entrées utilisées. Chaque entrée doit être connectée.



Entrées inversées

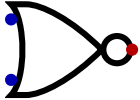
Sélectionner les entrées qui doivent être inversées.

Rotation

L'orientation de l'élément dans le circuit.

Forme large

Utilise une forme plus large pour visualiser la porte.



#### 1.4. Non-Ou

Une combinaison du OU et du NON. Retourne 0 lorsqu'au moins une entrée est mise à 1. Si toutes les entrées sont à 1 alors la sortie est à 0. Il est également possible d'utiliser des bus à plusieurs bits comme entrées. Dans ce cas, l'opération s'applique à chaque bit de l'entrée. Exportable en VHDL/Verilog.

Entrées

In\_1

La valeur d'entrée 1 pour l'opération logique.

In\_2

La valeur d'entrée 2 pour l'opération logique.

Sorties

out

Retourne le résultat de l'opération logique.

Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Nombre d'entrées

Le nombre d'entrées utilisées. Chaque entrée doit être connectée.

Entrées inversées

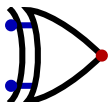
Sélectionner les entrées qui doivent être inversées.

Rotation

L'orientation de l'élément dans le circuit.

Forme large

Utilise une forme plus large pour visualiser la porte.



#### 1.5. XOR

Aussi appelée Ou-exclusif. Si deux entrées sont utilisées, la sortie est à 0 si les deux entrées sont égales. Sinon la sortie est mise à 1. Si plus de deux entrées sont utilisées, il se comporte que des portes XOR en cascade.  $(A \text{ XOR } B \text{ XOR } C = (A \text{ XOR } B) \text{ XOR } C)$ .

Il est également possible d'utiliser des bus à plusieurs bits comme entrées. Dans ce cas, l'opération s'applique à chaque bit de l'entrée. Exportable en VHDL/Verilog.

#### Entrées

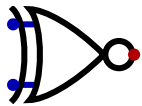
- In\_1  
La valeur d'entrée 1 pour l'opération logique.
- In\_2  
La valeur d'entrée 2 pour l'opération logique.

#### Sorties

- out  
Retourne le résultat de l'opération logique.

#### Attributs

- Bits de données  
Le nombre de bits utilisés pour les données.
- Nombre d'entrées  
Le nombre d'entrées utilisées. Chaque entrée doit être connectée.
- Entrées inversées  
Sélectionner les entrées qui doivent être inversées.
- Rotation  
L'orientation de l'élément dans le circuit.
- Forme large  
Utilise une forme plus large pour visualiser la porte.



### 1.6. NXOR

Une combinaison du XOR et du NON. Les entrées sont combinées avec l'opération XOR. Le résultat de cette opération est ensuite inversé. Il est également possible d'utiliser des bus à plusieurs bits comme entrées. Dans ce cas, l'opération s'applique à chaque bit de l'entrée. Exportable en VHDL/Verilog.

#### Entrées

- In\_1  
La valeur d'entrée 1 pour l'opération logique.
- In\_2  
La valeur d'entrée 2 pour l'opération logique.

#### Sorties

- out  
Retourne le résultat de l'opération logique.

#### Attributs

- Bits de données  
Le nombre de bits utilisés pour les données.
- Nombre d'entrées  
Le nombre d'entrées utilisées. Chaque entrée doit être connectée.

**Entrées inversées**

Sélectionner les entrées qui doivent être inversées.

**Rotation**

L'orientation de l'élément dans le circuit.

**Forme large**

Utilise une forme plus large pour visualiser la porte.

**1.7. Non**

Inverse la valeur d'entrée. Un 1 devient un 0 et un 0 devient un 1. Il est également possible d'utiliser des bus à plusieurs bits comme entrées. Dans ce cas, l'opération s'applique à chaque bit de l'entrée. Exportable en VHDL/Verilog.

**Entrées**

in

L'entrée de la porte NON.

**Sorties**

out

La valeur inversée de l'entrée.

**Attributs****Bits de données**

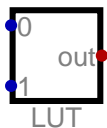
Le nombre de bits utilisés pour les données.

**Rotation**

L'orientation de l'élément dans le circuit.

**Forme large**

Utilise une forme plus large pour visualiser la porte.

**1.8. Table de recherche**

Trouve la valeur de sortie à partir d'une table stockée. Aussi appelée "lookup table". Cette porte permet donc d'imiter toutes les portes combinatoires. Exportable en VHDL/Verilog.

**Entrées**

0

Entrée 0. Cette entrée en combinaison avec toutes les autres définit l'adresse de sortie de la valeur stockée à retourner.

1

Entrée 1. Cette entrée en combinaison avec toutes les autres définit l'adresse de sortie de la valeur stockée à retourner.

## Sorties

out

Retourne la valeur stockée à l'adresse fixée par les entrées.

## Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Nombre d'entrées

Le nombre d'entrées utilisées. Chaque entrée doit être connectée.

Étiquette

Le nom de cet élément.

Données

Les valeurs stockées dans cet élément.

Rotation

L'orientation de l'élément dans le circuit.

## 2. E/S



### 2.1. Sortie

Peut être utilisé pour afficher un signal de sortie dans un circuit. Cet élément peut aussi servir à connecter un circuit à un circuit intégré. Dans ce cas, la connexion est bidirectionnelle. Il peut aussi servir à assigner un numéro de broche pour la génération de code de CPLD ou FPGA. Exportable en VHDL/Verilog.

## Entrées

in

Cette valeur est utilisée pour la sortie.

## Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Étiquette

Le nom de cet élément.

Description

Une courte description de cet élément et son utilisation.

Rotation

L'orientation de l'élément dans le circuit.

Format de nombre

Le format utilisé pour afficher les nombres.

Décimales (virgule fixe)

Nombre de décimales binaires pour le format à virgule fixe.

**Numéro de broche**

Numéro de cette broche. Utilisé pour la représentation d'un circuit dans un boîtier DIL et lors de l'affectation des broches lors de la programmation d'un CPLD. S'il y a plusieurs bits, tous les numéros de broche peuvent être spécifiés dans une liste séparée par des virgules.

**Afficher dans le graphique de mesures**

Affiche la valeur dans le graphique de mesures

**Forme petite**

Si coché, une forme plus petite sera utilisée.

**2.2. DEL**

Une DEL (ou LED) peut être utilisée pour visualiser une valeur de sortie. L'entrée accepte un seul bit. La lumière est allumée quand l'entrée est mise à 1.

**Entrées**

in

Entrée de la DEL. La DEL est allumée si l'entrée est mise à 1.

**Attributs****Étiquette**

Le nom de cet élément.

**Couleur**

La couleur de l'élément.

**Rotation**

L'orientation de l'élément dans le circuit.

**Taille**

La taille de la forme de cet élément dans le circuit.

**Afficher dans le graphique de mesures**

Affiche la valeur dans le graphique de mesures

**2.3. Entrée**

Peut être utilisé pour manipuler interactivement avec la souris un signal d'entrée dans un circuit. Cet élément peut aussi servir à connecter un circuit à un circuit intégré. Dans ce cas, la connexion est bidirectionnelle. Il peut aussi servir à assigner un numéro de broche pour la génération de code de CPLD ou FPGA. Exportable en VHDL/Verilog.

**Sorties**

out

Donne la valeur liée à cette entrée.

**Attributs****Bits de données**

Le nombre de bits utilisés pour les données.

**Étiquette**

Le nom de cet élément.

**Description**

Une courte description de cet élément et son utilisation.

**Rotation**

L'orientation de l'élément dans le circuit.

**Par défaut**

Cette valeur est mise lorsque le circuit débute. Un "Z" signifie un état de haute impédance.

**Entrée à trois états**

Si coché, l'entrée pourra être mise dans un état de haute impédance. Pour une entrée, c'est aussi disponible si "Z" est choisi comme valeur par défaut.

**Pas de sortie à zéro.**

Évite la sortie à zéro. Cela est particulièrement pratique pour monter des circuits de relais. Peut seulement être coché si l'état de haute impédance est permis.

**Format de nombre**

Le format utilisé pour afficher les nombres.

**Décimales (virgule fixe)**

Nombre de décimales binaires pour le format à virgule fixe.

**Numéro de broche**

Numéro de cette broche. Utilisé pour la représentation d'un circuit dans un boîtier DIL et lors de l'affectation des broches lors de la programmation d'un CPLD. S'il y a plusieurs bits, tous les numéros de broche peuvent être spécifiés dans une liste séparée par des virgules.

**Afficher dans le graphique de mesures**

Affiche la valeur dans le graphique de mesures

**Forme petite**

Si coché, une forme plus petite sera utilisée.



## 2.4. Horloge

Un signal d'horloge. Il est possible de le contrôler avec une horloge temps réel. Selon la complexité du circuit, la fréquence de l'horloge atteinte pourrait être moins que la valeur sélectionnée. Si la fréquence est supérieure à 50Hz, la représentation graphique du circuit ne sera plus mise à jour à chaque cycle d'horloge, donc les couleurs de fils ne pourront plus être mises à jour à chaque changement de valeur. Si l'horloge temps réel est désactivée, l'horloge peut être contrôlée par des clics de la souris. Elle peut aussi servir à assigner un numéro de broche pour la génération de code de CPLD ou FPGA. Exportable en VHDL/Verilog.

**Sorties****C**

Oscille entre 0 et 1 avec la fréquence d'horloge sélectionnée.

**Attributs****Étiquette**

Le nom de cet élément.

Partir l'horloge temps réel

Si coché, l'horloge de temps d'exécution est partie quand le circuit débute.

Fréquence/Hz

La fréquence utilisée pour l'horloge temps réel.

Rotation

L'orientation de l'élément dans le circuit.

Numéro de broche

Numéro de cette broche. Utilisé pour la représentation d'un circuit dans un boîtier DIL et lors de l'affectation des broches lors de la programmation d'un CPLD. S'il y a plusieurs bits, tous les numéros de broche peuvent être spécifiés dans une liste séparée par des virgules.

Forme petite

Si coché, une forme plus petite sera utilisée.



## 2.5. Bouton

Un simple bouton-poussoir qui revient à son état original lorsqu'il est relâché.

Sorties

out

Le signal de sortie du bouton.

Attributs

Étiquette

Le nom de cet élément.

Actif bas

Si coché, la sortie est basse lorsque la composante est active.

Associer au clavier

Le bouton est associé au clavier. Pour utiliser les touches de direction, utilisez UP, DOWN, LEFT ou RIGHT comme étiquette.

Rotation

L'orientation de l'élément dans le circuit.

Afficher dans le graphique de mesures

Affiche la valeur dans le graphique de mesures



## 2.6. Interrupteur DIP

Interrupteur DIP simple (DIP switch) qui peut émettre des valeurs hautes ou basses.

Sorties

out

La valeur de sortie pour l'interrupteur

Attributs

**Étiquette**

Le nom de cet élément.

**Description**

Une courte description de cet élément et son utilisation.

**Rotation**

L'orientation de l'élément dans le circuit.

**Sortie haute**

La valeur par défaut de l'interrupteur DIP lorsque la simulation débute.



## 2.7. Sonde

Une mesure qui peut être affichée dans un graphique de données ou un tableau de mesures. Cette composante peut être utilisée pour observer facilement des valeurs dans des circuits embarqués. N'affecte pas la simulation.

**Entrées**

in

La valeur mesurée.

**Attributs****Étiquette**

Le nom de cet élément.

**Mode d'affichage**

Définit si la valeur ou un compteur doit être affiché.

**Rotation**

L'orientation de l'élément dans le circuit.

**Format de nombre**

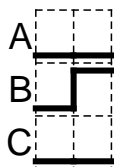
Le format utilisé pour afficher les nombres.

**Décimales (virgule fixe)**

Nombre de décimales binaires pour le format à virgule fixe.

**Afficher dans le graphique de mesures**

Affiche la valeur dans le graphique de mesures



## 2.8. Graphique de données

Affiche un graphique de données à l'intérieur de la zone du circuit. Il est possible de tracer des cycles d'horloge complets ou le changement d'une seule porte. N'affecte pas la simulation.

**Attributs****Afficher les étapes pas-à-pas**

Afficher toutes les étapes pas-à-pas dans le graphique.



Nombre maximal d'étapes à afficher

Le nombre maximal de valeurs stockées. Si le nombre maximal est atteint, les valeurs plus vieilles disparaissent.

Aligner sur la grille

Si coché, la composante reste alignée sur la grille.



## 2.9. Graphique de données avec déclenchement

Affiche un graphique des valeurs mesurées, excepté que les valeurs sont stockées seulement lorsque le signal d'entrée change. Le stockage prend place lorsque le circuit s'est stabilisé. Le déclenchement ne débute pas les mesures comme un vrai oscilloscope, mais plutôt chaque signal de déclenchement stocke la valeur des signaux affichés à un moment donné. La seule entrée directe est celle du déclenchement. Les entrées les sorties du circuit, les bascules, les registres et les sondes peuvent être utilisés comme signaux. Cela peut être activé dans les paramètres de chacune de ces composantes.

Entrées

T

Un changement à cette entrée provoque le stockage des valeurs mesurées

Attributs

Étiquette

Le nom de cet élément.

Déclenchement

Condition de déclenchement pour l'enregistrement des données

Nombre maximal d'étapes à afficher

Le nombre maximal de valeurs stockées. Si le nombre maximal est atteint, les valeurs plus vieilles disparaissent.

## 3. E/S - Affichages



### 3.1. DEL-RVB

Une DEL RVB pour laquelle la couleur est établie par trois entrées. Un canal de couleur est associé à chaque entrée.

Entrées

R

Le canal de couleur rouge.

G

Le canal de couleur verte.

B

Le canal de couleur bleue.

## Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Étiquette

Le nom de cet élément.

Rotation

L'orientation de l'élément dans le circuit.

Taille

La taille de la forme de cet élément dans le circuit.

**3.2. DEL avec deux connexions.**

DEL avec des connexions pour la cathode et l'anode. La DEL s'allume si l'anode est connectée à un signal haut et l'anode à un signal bas. Cette LED ne peut pas être utilisée comme une résistance de tirage. Elle agit comme un élément d'affichage uniquement. La résistance affichée sert seulement à symboliser la résistance qui sert normalement à limiter le courant.

## Entrées

A

La connexion de l'anode pour la DEL.

C

La connexion de la cathode pour la DEL.

## Attributs

Étiquette

Le nom de cet élément.

Couleur

La couleur de l'élément.

Rotation

L'orientation de l'élément dans le circuit.

Afficher dans le graphique de mesures

Affiche la valeur dans le graphique de mesures

**3.3. Bouton avec DEL**

Un simple bouton-poussoir qui revient à son état original lorsqu'il est relâché. Le bouton a également une DEL qui peut être allumée avec un signal d'entrée.

### Entrées

in

L'entrée contrôlant la DEL.

### Sorties

out

Le signal de sortie du bouton.

### Attributs

Étiquette

Le nom de cet élément.

Actif bas

Si coché, la sortie est basse lorsque la composante est active.

Associer au clavier

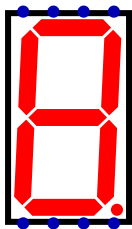
Le bouton est associé au clavier. Pour utiliser les touches de direction, utilisez UP, DOWN, LEFT ou RIGHT comme étiquette.

Couleur

La couleur de l'élément.

Rotation

L'orientation de l'élément dans le circuit.



## 3.4. Affichage 7 segments

Affichage 7 segments, chaque segment a sa propre entrée.

### Entrées

a

Cette entrée contrôle la ligne horizontale du haut.

b

Cette entrée contrôle la ligne verticale du haut à droite.

c

Cette entrée contrôle la ligne verticale du bas à droite.

d

Cette entrée contrôle la ligne horizontale du bas.

e

Cette entrée contrôle la ligne verticale du bas à gauche.

f

Cette entrée contrôle la ligne verticale du haut à gauche.

g

Cette entrée contrôle la ligne horizontale du milieu.

dp

Cette entrée contrôle le point décimal.

## Attributs

## Couleur

La couleur de l'élément.

## Connexion commune

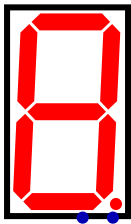
Si coché, une entrée de cathode ou d'anode commune est aussi simulée.

## Commun

Type de connexion commune.

## Durée de rémanence

Spécifie la durée de la lumière rémanente. Plus la valeur est élevée, plus la rémanence est longue.



### 3.5. Affichage 7 segments Hex

Affichage 7 segments avec une entrée hexadécimale de 4 bits.

## Entrées

## d

La valeur de cette entrée est visualisée sur l'affichage.

## dp

Cette entrée contrôle le point décimal.

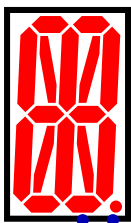
## Attributs

## Couleur

La couleur de l'élément.

## Taille

La taille de la forme de cet élément dans le circuit.



### 3.6. Affichage 16 segments

L'entrée de la DEL a 16 bits qui contrôlent chaque segment. La deuxième entrée contrôle le point décimal.

**Entrées**

led

Bus 16 bits pour contrôler les DELs.

dp

Cette entrée contrôle le point décimal.

**Attributs**

Couleur

La couleur de l'élément.

Taille

La taille de la forme de cet élément dans le circuit.

**3.7. Ampoule**

Une ampoule avec deux connexions. Si le courant circule, l'ampoule s'allume! La direction du courant n'a pas d'importance. La lampe s'allume quand les entrées ont des valeurs différentes. Ainsi, l'ampoule se comporte de façon similaire à une porte XOR.

**Entrées**

A

Connexion

B

Connexion

**Attributs**

Étiquette

Le nom de cet élément.

Couleur

La couleur de l'élément.

Rotation

L'orientation de l'élément dans le circuit.



Matrice de DELs

**3.8. Matrice de DELs**

Une matrice de DELs. Les DELs sont affichées dans une fenêtre séparée. Les DELs d'une colonne de l'affichage sont contrôlées par un mot. Dans une autre entrée, la colonne courante est sélectionnée. Ainsi, un affichage multiplexé est réalisé. Les DELs peuvent être allumées indéfiniment dans la simulation pour empêcher l'affichage de clignoter.

## Entrées

### r-data

L'état de la DEL de chaque rangée pour une colonne. Chaque bit de ce mot représente l'état d'une rangée de la colonne courante.

### c-addr

Le numéro de la colonne qui est sélectionnée pour l'autre entrée.

## Attributs

### Étiquette

Le nom de cet élément.

### Lignes

Spécifie le nombre de lignes en spécifiant le nombre de bits du mot de la ligne.

### Bits d'adressage des colonnes

Pour adresser des colonnes individuelles. Trois bits donnent huit colonnes.

### Couleur

La couleur de l'élément.

### Éviter le clignotement

Il n'est pas possible d'augmenter la fréquence assez pour que le clignotement disparaisse. Afin de supprimer le clignotement quand même, une "rémanence" peut être activée pour les LEDs avec cette option. Si coché, les LEDs restent allumées, même si une des broches change vers un état de haute impédance. Cela simule une fréquence au-delà de la fréquence critique de fusion du clignotement.

### Rotation

L'orientation de l'élément dans le circuit.

## 4. E/S - Mécanique



### 4.1. Encodeur rotatif

Roulette tournante avec encodeur rotatif. Sert à détecter les mouvements de rotation.

## Sorties

### A

signal de l'encodeur A

### B

signal de l'encodeur B

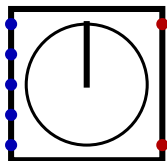
## Attributs

### Étiquette

Le nom de cet élément.

### Rotation

L'orientation de l'élément dans le circuit.



## 4.2. Moteur pas-à-pas, unipolaire

Moteur pas-à-pas unipolaire avec deux interrupteurs de position limite. Permet le fonctionnement à pas complet, à demi-pas et à une phase (wave).

### Entrées

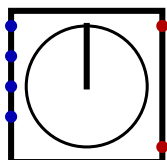
- P0  
Phase 0
- P1  
Phase 1
- P2  
Phase 2
- P3  
Phase 3
- com  
Connexion de bobine centrale commune

### Sorties

- S0  
Interrupteur de position limite 0, devient 1 lorsque l'angle du moteur est de  $0^\circ$ .
- S1  
Interrupteur de position limite 1, devient 1 lorsque l'angle du moteur est de  $180^\circ$ .

### Attributs

- Étiquette  
Le nom de cet élément.
- Inverser la sortie  
Si coché la sortie est inversée.
- Rotation  
L'orientation de l'élément dans le circuit.



## 4.3. Moteur pas-à-pas, bipolaire

Moteur pas-à-pas bipolaire avec deux interrupteurs de position limite. Permet le fonctionnement à pas complet, à demi-pas et à une phase (wave).

## Entrées

- A+  
Bobine A, positif
- A-  
Bobine A, négatif
- B+  
Bobine B, positif
- B-  
Bobine B, négatif

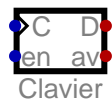
## Sorties

- S0  
Interrupteur de position limite 0, devient 1 lorsque l'angle du moteur est de 0°.
- S1  
Interrupteur de position limite 1, devient 1 lorsque l'angle du moteur est de 180°.

## Attributs

- Étiquette  
Le nom de cet élément.
- Inverser la sortie  
Si coché la sortie est inversée.
- Rotation  
L'orientation de l'élément dans le circuit.

# 5. E/S - Périphériques



## 5.1. Clavier

Un clavier peut être utilisé pour entrer du texte. Cette composante enregistre l'entrée dans un tampon qui peut être lu à la sortie. Une fenêtre séparée est ouverte pour entrer le texte.

### Entrées

- C  
Signal d'horloge. Un front montant supprime le caractère le plus vieux du tampon.
- en  
Si activée, la sortie D est active et un caractère est émis. Cette entrée active également la lecture du signal d'horloge.

### Sorties

- D  
Le dernier caractère tapé, ou zéro si aucun caractère n'est disponible. La sortie correspond à une valeur de caractère en Java de 16 bits.
- av  
Cette sortie indique que des caractères sont disponibles. Elle peut permettre de déclencher une interruption.



## Attributs

## Étiquette

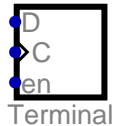
Le nom de cet élément.

## Entrées inversées

Sélectionner les entrées qui doivent être inversées.

## Rotation

L'orientation de l'élément dans le circuit.



## 5.2. Terminal

Des caractères ASCII peuvent être écrits dans ce terminal. Ce terminal s'ouvre dans sa propre fenêtre pour visualiser la sortie.

## Entrées

D

Les données à écrire dans le terminal

C

Signal d'horloge. Un front montant écrit la valeur à l'entrée dans la fenêtre du terminal.

en

Une valeur haute à cette entrée active le signal d'horloge.

## Attributs

## Caractères par ligne

Le nombre de caractères affichés dans une seule ligne.

## Lignes

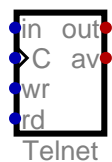
Le nombre de lignes à afficher.

## Étiquette

Le nom de cet élément.

## Rotation

L'orientation de l'élément dans le circuit.



## 5.3. Telnet

Permet une connexion Telnet au circuit. Il est possible de recevoir et de transmettre des caractères via Telnet.

## Entrées

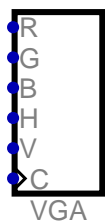
- in  
Les données à transmettre
- C  
Signal d'horloge
- wr  
Si cette entrée est active, l'octet d'entrée est transmis.
- rd  
Si cette entrée est active, l'octet reçu est émis.

## Sorties

- out  
Valeur de sortie
- av  
Cette sortie indique que des données sont présentes lorsqu'elle est haute.

## Attributs

- Étiquette  
Le nom de cet élément.
- Mode Telnet  
Si activé, les commandes de contrôle Telnet sont évaluées. De plus, le serveur envoie les commandes SGA et ECHO. Si cette option est désactivée, le serveur est un simple serveur TCP.
- Port  
Le port ouvert par le serveur.
- Rotation  
L'orientation de l'élément dans le circuit.



## 5.4. Écran VGA

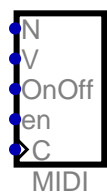
Analyse les signaux vidéo entrants et affiche les graphiques correspondants. Puisque la simulation ne peut pas s'exécuter en temps réel, le signal d'horloge de pixel est nécessaire en plus du signal vidéo.

## Entrées

- R  
La composante rouge
- G  
La composante verte
- B  
La composante bleue
- H  
Le signal de synchronisation horizontal
- V  
Le signal de synchronisation vertical
- C  
Le signal d'horloge de pixel

## Attributs

- Étiquette  
Le nom de cet élément.
- Rotation  
L'orientation de l'élément dans le circuit.



## 5.5. MIDI

Utilise le système MIDI pour jouer des notes.

## Entrées

- N  
Note
- V  
Volume
- OnOff  
Si activée, cela se traduit par un appui sur une touche (événement de touche enfoncée), sinon, cela se traduit par un relâchement de la touche (événement de touche relâchée).
- en  
Active la composante
- C  
Signal d'horloge

## Attributs

- Étiquette  
Le nom de cet élément.
- Canal MIDI  
Sélectionne le canal MIDI à utiliser.

### Instrument MIDI

L'instrument MIDI à utiliser

### Permettre le changement de programme

Ajoute une nouvelle entrée PC (program change). Si cette entrée est haute, la valeur à l'entrée N est utilisée pour changer le programme (l'instrument).

### Rotation

L'orientation de l'élément dans le circuit.

## 6. Câblage



### 6.1. Masse

Une connexion à une masse (mise à la terre). La sortie est toujours zéro. Exportable en VHDL/Verilog.

#### Sorties

out

La sortie est toujours 0.

#### Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Étiquette

Le nom de cet élément.

Rotation

L'orientation de l'élément dans le circuit.



### 6.2. Tension d'alimentation

Une connexion à une tension d'alimentation. La sortie est toujours un. Exportable en VHDL/Verilog.

#### Sorties

out

La sortie est toujours 1.

#### Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Étiquette

Le nom de cet élément.

Rotation

L'orientation de l'élément dans le circuit.

1•

### 6.3. Valeur constante

Une composante qui retourne une valeur donnée comme une constante. Cette valeur peut être mise dans le dialogue des attributs. Exportable en VHDL/Verilog.

#### Sorties

out

Retourne la valeur donnée comme une constante.

#### Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Valeur

La valeur de la constante.

Rotation

L'orientation de l'élément dans le circuit.

Format de nombre

Le format utilisé pour afficher les nombres.

Décimales (virgule fixe)

Nombre de décimales binaires pour le format à virgule fixe.



### 6.4. Tunnel

Connecte des composantes sans un fil. Tous les tunnels ayant le même nom de réseau sont connectés ensemble. Fonctionne uniquement de manière locale, il est donc impossible de connecter des circuits différents. Les tunnels sans nom sont ignorés sans produire d'erreurs. Exportable en VHDL/Verilog.

#### Entrées

in

La connexion au tunnel.

#### Attributs

Nom du réseau

Tous les réseaux avec un nom identique sont connectés ensemble.

Rotation

L'orientation de l'élément dans le circuit.



### 6.5. Séparateur/Fusionneur

Sépare ou crée un amas de fils ou un bus de données de plus d'un bit. Avec un bus il est par exemple possible de générer des connexions 16 bits sans avoir à tracer 16 fils individuels.

Les 16 connexions peuvent être fusionnées en un seul fil. Le séparateur possède une direction, ce qui veut dire qu'il ne peut transmettre les signaux que dans une direction. Exportable en VHDL/Verilog.

#### Entrées

0-3

Les bits d'entrée 0-3.

4-7

Les bits d'entrée 4-7.

#### Sorties

0-7

Les bits de sortie 0-7.

#### Attributs

##### Séparation de l'entrée

S'il y a par exemple quatre bits, deux bits et encore deux bits à être utilisés comme entrées, cela peut être configuré avec "4,2,2". Le nombre indique le nombre de bits. Pour plus de facilité, l'astérisque peut être utilisé: 16 bits peuvent ainsi être séparés avec "[Bits]\*[Nombre]", par exemple "1\*16" donnera 16 bits individuels. Il est également possible de spécifier les bits à utiliser dans n'importe quel ordre. Par exemple, "4-7,0-3" utilisera les bits 4-7 et 0-3. Cette notation permet n'importe quel arrangement de bits. Les entrées doivent être spécifiées complètement et sans ambiguïté.

##### Séparation de la sortie

S'il y a par exemple quatre bits, deux bits et encore deux bits à être utilisés comme entrées, cela peut être configuré avec "4,2,2". Le nombre indique le nombre de bits. Pour plus de facilité, l'astérisque peut être utilisé: 16 bits peuvent ainsi être séparés avec "[Bits]\*[Nombre]", par exemple "1\*16" donnera 16 bits individuels. Il est également possible de spécifier les bits à utiliser dans n'importe quel ordre. Par exemple, "4-7,0-3" utilisera les bits 4-7 et 0-3. Cette notation permet n'importe quel arrangement de bits. Les bits de sortie peuvent être émis plusieurs fois: "0-7,1-6,4-7".

##### Rotation

L'orientation de l'élément dans le circuit.

##### Miroir

Reflète la composante dans le circuit.

##### Étalement

Configure l'étalement des entrées et des sorties dans le circuit.



## 6.6. Pilote

Un pilote peut être utilisé pour connecter une valeur de signal à un autre fil. Le pilote est contrôlé par l'entrée "sel" (sélection). Si l'entrée "sel" est basse, la sortie est dans un état de haute impédance. Si l'entrée "sel" est haute, la sortie est mise à la valeur d'entrée. Exportable en VHDL/Verilog.

### Entrées

in

La valeur d'entrée du pilote.

sel

Si la valeur est 1, la sortie est mise à la valeur d'entrée. Si la valeur est 0, la sortie est dans un état de haute impédance.

### Sorties

out

Si l'entrée "sel" est haute, la valeur de l'entrée est donnée à cette sortie. Si l'entrée "sel" est basse, la sortie est dans un état de haute impédance.

### Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Inverser la sortie

Si coché la sortie est inversée.

Inverser la position du sélecteur

Cette option permet de déplacer l'entrée du sélecteur du côté opposé du plexeur.

Rotation

L'orientation de l'élément dans le circuit.



## 6.7. Pilote, sélection inversée

Un pilote peut être utilisé pour connecter une valeur de signal à un autre fil. Le pilote est contrôlé par l'entrée "sel" (sélection). Si l'entrée "sel" est haute, la sortie est dans un état de haute impédance. Si l'entrée "sel" est basse, la sortie est mise à la valeur d'entrée. Exportable en VHDL/Verilog.

### Entrées

in

La valeur d'entrée du pilote.

sel

Si la valeur est 0, la sortie est mise à la valeur d'entrée. Si la valeur est 1, la sortie est dans un état de haute impédance.

### Sorties

out

Si l'entrée "sel" est haute, la valeur de l'entrée est donnée à cette sortie. Si l'entrée "sel" est basse, la sortie est dans un état de haute impédance.

### Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Inverser la sortie

Si coché la sortie est inversée.

Inverser la position du sélecteur

Cette option permet de déplacer l'entrée du sélecteur du côté opposé du plexeur.

Rotation

L'orientation de l'élément dans le circuit.



## 6.8. Délai

Retarde un signal par une valeur ajustable de temps de propagation équivalent d'une porte logique. Toutes les autres composantes dans Digital ont un délai de temps de propagation d'une durée de 1. Cette composante peut servir à réaliser n'importe quel temps de propagation nécessaire.

Entrées

in

Entrée du signal qui doit être retardé.

Sorties

out

L'entrée du signal retardée d'un temps de propagation équivalent à celui d'une porte logique.

Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Durée

Temps de délais avec comme unités le temps de propagation équivalent d'une porte logique.

Rotation

L'orientation de l'élément dans le circuit.



## 6.9. Résistance de tirage

Si un réseau n'est pas dans un état de haute impédance (Hi-Z), cette résistance tire le réseau à une valeur haute. Dans n'importe quel autre cas, cette composante n'a aucun effet.

Sorties

out

Un signal haut "faible"

Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Rotation

L'orientation de l'élément dans le circuit.





### 6.10. Résistance de rappel

Si un réseau n'est pas dans un état de haute impédance (Hi-Z), cette résistance tire le réseau à une valeur basse (à la masse). Dans n'importe quel autre cas, cette composante n'a aucun effet.

Sorties

out  
Un signal bas "faible"

Attributs

Bits de données  
Le nombre de bits utilisés pour les données.  
Rotation  
L'orientation de l'élément dans le circuit.



### 6.11. Pas connecté

Cette composante peut être utilisée pour mettre un fil en état de haute impédance. Si l'entrée d'une porte logique est mise à haute impédance, la valeur lue est indéfinie. À noter qu'en réalité, dans plusieurs situations, cet état de haute impédance tend à endommager les composants ou à causer une consommation excessive de courant.

Sorties

out  
Cette sortie émet toujours l'état hi-Z.

Attributs

Bits de données  
Le nombre de bits utilisés pour les données.

## 7. Plexeurs



### 7.1. Multiplexeur

Une composante qui utilise la valeur de l'entrée "sel" (sélection) pour décider quelle valeur d'entrée est émise à la sortie. Exportable en VHDL/Verilog.

### Entrées

sel

Cette entrée sert à sélectionner l'entrée qui est émise.

in\_0

La valeur d'entrée 0 du multiplexeur.

in\_1

La valeur d'entrée 1 du multiplexeur.

### Sorties

out

La valeur de l'entrée sélectionnée.

### Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Nombre de bits du sélecteur

Nombre de bits utilisés pour l'entrée du sélecteur.

Inverser la position du sélecteur

Cette option permet de déplacer l'entrée du sélecteur du côté opposé du plexeur.

Rotation

L'orientation de l'élément dans le circuit.



## 7.2. Démultiplexeur

Une composante qui émet la valeur d'entrée à une de ses sorties. Les autres sorties sont mises à la valeur par défaut. Exportable en VHDL/Verilog.

### Entrées

sel

Cette entrée sélectionner la sortie à utiliser.

in

La valeur de cette entrée est donnée à la sortie sélectionnée.

### Sorties

out\_0

Valeur de sortie 0.

out\_1

Valeur de sortie 1.

### Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Nombre de bits du sélecteur

Nombre de bits utilisés pour l'entrée du sélecteur.

Inverser la position du sélecteur

Cette option permet de déplacer l'entrée du sélecteur du côté opposé du plexeur.

**Rotation**

L'orientation de l'élément dans le circuit.

**Par défaut**

Cette valeur est mise lorsque le circuit débute. Pour un démultiplexeur, cette valeur est mise pour les sorties non sélectionnées.

**7.3. Décodeur**

Une sortie sélectionnée est à 1, les autres sorties sont à 0. Exportable en VHDL/Verilog.

**Entrées****sel**

Cette entrée sélectionne la sortie activée. La sortie activée est mise à 1. Les autres sorties sont mises à 0.

**Sorties****out\_0**

Valeur de sortie 0. Cette sortie est à 1 lorsqu'elle est sélectionnée par l'entrée "sel".

**out\_1**

Valeur de sortie 1. Cette sortie est à 1 lorsqu'elle est sélectionnée par l'entrée "sel".

**Attributs****Nombre de bits du sélecteur**

Nombre de bits utilisés pour l'entrée du sélecteur.

**Inverser la position du sélecteur**

Cette option permet de déplacer l'entrée du sélecteur du côté opposé du plexeur.

**Rotation**

L'orientation de l'élément dans le circuit.

**7.4. Sélecteur de bit**

Sélectionne un seul bit à partir d'un bus de données d'entrée. Exportable en VHDL/Verilog.

**Entrées****in**

Le bus d'entrée

**sel**

Cette entrée sélectionne le bit

**Sorties****out**

Le bit sélectionné.

**Attributs**

Nombre de bits du sélecteur

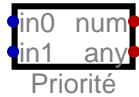
Nombre de bits utilisés pour l'entrée du sélecteur.

Inverser la position du sélecteur

Cette option permet de déplacer l'entrée du sélecteur du côté opposé du plexeur.

Rotation

L'orientation de l'élément dans le circuit.



## 7.5. Codeur de priorité

Si une des entrées est mise, sa valeur est émise. Si plusieurs entrées sont mises en même temps, la plus grande valeur est émise. Exportable en VHDL/Verilog.

Entrées

in0

La valeur d'entrée 0 du codeur de priorité.

in1

La valeur d'entrée 1 du codeur de priorité.

Sorties

num

Valeur de l'entrée choisie.

any

Si cette sortie est active, au moins une des deux entrées est mise.

Attributs

Étiquette

Le nom de cet élément.

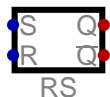
Nombre de bits du sélecteur

Nombre de bits utilisés pour l'entrée du sélecteur.

Rotation

L'orientation de l'élément dans le circuit.

## 8. Bascules



### 8.1. Bascule RS

Une composante pour stocker un seul bit. Fournit les fonctions "set" et "reset" pour mémoriser ou réinitialiser le bit stocké. Si les deux entrées sont à un, les deux sorties seront aussi à un. Si les deux entrées sont mises à zéro en même temps, l'état final est aléatoire.

## Entrées

S

L'entrée "set" pour mémoriser.

R

L'entrée "reset" pour réinitialiser.

## Sorties

Q

Retourne la valeur stockée.

 $\neg Q$ 

Retourne l'inverse de la valeur stockée.

## Attributs

Étiquette

Le nom de cet élément.

Entrées inversées

Sélectionner les entrées qui doivent être inversées.

Rotation

L'orientation de l'élément dans le circuit.

Miroir

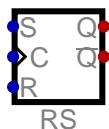
Reflète la composante dans le circuit.

Par défaut

Cette valeur est mise lorsque le circuit débute. Pour un démultiplexeur, cette valeur est mise pour les sorties non sélectionnées.

Utiliser comme valeur de mesure

Si coché, la valeur est une valeur de mesure et s'affiche dans le graphique et le tableau de données. De plus, une étiquette doit être spécifiée afin d'identifier la valeur.



## 8.2. Bascule RS, avec horloge

Une composante pour stocker un seul bit. Fournit les fonctions "set" et "reset" pour mémoriser ou réinitialiser le bit stocké. Si les deux entrées (S, R) sont actives au front montant de l'horloge, l'état final est aléatoire.

## Entrées

S

L'entrée "set" pour mémoriser.

C

Le signal d'horloge. Un front montant amorce une transition d'état.

R

L'entrée "reset" pour réinitialiser.

## Sorties

Q

Retourne la valeur stockée.

 $\neg Q$ 

Retourne l'inverse de la valeur stockée.

## Attributs

Étiquette

Le nom de cet élément.

Entrées inversées

Sélectionner les entrées qui doivent être inversées.

Rotation

L'orientation de l'élément dans le circuit.

Miroir

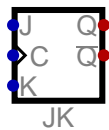
Reflète la composante dans le circuit.

Par défaut

Cette valeur est mise lorsque le circuit débute. Pour un démultiplexeur, cette valeur est mise pour les sorties non sélectionnées.

Utiliser comme valeur de mesure

Si coché, la valeur est une valeur de mesure est s'affiche dans le graphique et le tableau de données. De plus, une étiquette doit être spécifiée afin d'identifier la valeur.



## 8.3. Bascule JK

À la possibilité de stocker ( $J=K=0$ ), mémoriser ( $J=1, K=0$ ), réinitialiser ( $J=0, K=1$ ) ou basculer ( $J=K=1$ ) la valeur stockée. Un changement d'état se produit seulement lors d'un front montant du signal d'horloge C. Exportable en VHDL/Verilog.

## Entrées

J

L'entrée "set" de la bascule, pour mémoriser.

C

Le signal d'horloge. Un front montant amorce un changement d'état.

K

L'entrée "reset" de la bascule, pour réinitialiser.

## Sorties

Q

Retourne la valeur stockée.

 $\neg Q$ 

Retourne l'inverse de la valeur stockée.

## Attributs

**Étiquette**

Le nom de cet élément.

**Entrées inversées**

Sélectionner les entrées qui doivent être inversées.

**Rotation**

L'orientation de l'élément dans le circuit.

**Miroir**

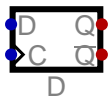
Reflète la composante dans le circuit.

**Par défaut**

Cette valeur est mise lorsque le circuit débute. Pour un démultiplexeur, cette valeur est mise pour les sorties non sélectionnées.

**Utiliser comme valeur de mesure**

Si coché, la valeur est une valeur de mesure est s'affiche dans le graphique et le tableau de données. De plus, une étiquette doit être spécifiée afin d'identifier la valeur.

**8.4. Bascule D**

Une composante utilisée pour stocker une valeur. La valeur sur l'entrée D est stockée lors du front montant du signal d'horloge C. La largeur de bit peut être sélectionnée ce qui permet de stocker plusieurs bits. Exportable en VHDL/Verilog.

**Entrées**

**D**

L'entrée du bit à stocker.

**C**

Signal d'horloge pour stocker une valeur. La valeur à l'entrée D est stockée sur un front montant de cette entrée.

**Sorties**

**Q**

Retourne la valeur stockée.

**¬Q**

Retourne l'inverse de la valeur stockée.

**Attributs**

**Bits de données**

Le nombre de bits utilisés pour les données.

**Étiquette**

Le nom de cet élément.

**Entrées inversées**

Sélectionner les entrées qui doivent être inversées.

**Rotation**

L'orientation de l'élément dans le circuit.

**Miroir**

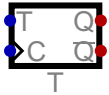
Reflète la composante dans le circuit.

**Par défaut**

Cette valeur est mise lorsque le circuit débute. Pour un démultiplexeur, cette valeur est mise pour les sorties non sélectionnées.

**Utiliser comme valeur de mesure**

Si coché, la valeur est une valeur de mesure est s'affiche dans le graphique et le tableau de données. De plus, une étiquette doit être spécifiée afin d'identifier la valeur.

**8.5. Bascule T**

Stocke un seul bit. L'état bascule sur un front montant de l'entrée C.

**Entrées**

T

Active la fonction de bascule.

C

Signal d'horloge. Un front montant fait basculer la sortie si l'entrée T est à 1.

**Sorties**

Q

Retourne la valeur stockée.

$\neg Q$

Retourne l'inverse de la valeur stockée.

**Attributs****Étiquette**

Le nom de cet élément.

**Entrée d'activation**

Si coché, une entrée d'activation (T) est disponible.

**Entrées inversées**

Sélectionner les entrées qui doivent être inversées.

**Rotation**

L'orientation de l'élément dans le circuit.

**Miroir**

Reflète la composante dans le circuit.

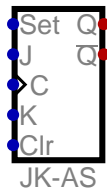
**Par défaut**

Cette valeur est mise lorsque le circuit débute. Pour un démultiplexeur, cette valeur est mise pour les sorties non sélectionnées.

**Utiliser comme valeur de mesure**

Si coché, la valeur est une valeur de mesure est s'affiche dans le graphique et le tableau de données. De plus, une étiquette doit être spécifiée afin d'identifier la valeur.





## 8.6. Bascule JK asynchrone

À la possibilité de stocker ( $J=K=0$ ), mémoriser ( $J=1, K=0$ ), réinitialiser ( $J=0, K=1$ ) ou basculer ( $J=K=1$ ) la valeur stockée. Un changement d'état se produit seulement lors d'un front montant du signal d'horloge C. Il y a deux entrées additionnelles pour mémoriser et réinitialiser immédiatement, sans signal d'horloge. Exportable en VHDL/Verilog.

### Entrées

#### Set

Une entrée "set" asynchrone, pour mémoriser. Une valeur de 1 à cette entrée fixe la sortie de la bascule.

#### J

L'entrée "set" de la bascule, pour mémoriser.

#### C

Le signal d'horloge. Un front montant amorce un changement d'état.

#### K

L'entrée "reset" de la bascule, pour réinitialiser.

#### Clr

Une entrée "clear" asynchrone, pour effacer. Une valeur de 1 à cette entrée réinitialise la sortie de la bascule.

### Sorties

#### Q

Retourne la valeur stockée.

#### $\neg Q$

Retourne l'inverse de la valeur stockée.

### Attributs

#### Étiquette

Le nom de cet élément.

#### Entrées inversées

Sélectionner les entrées qui doivent être inversées.

#### Rotation

L'orientation de l'élément dans le circuit.

#### Miroir

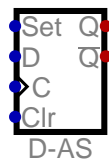
Reflète la composante dans le circuit.

#### Par défaut

Cette valeur est mise lorsque le circuit débute. Pour un démultiplexeur, cette valeur est mise pour les sorties non sélectionnées.

#### Utiliser comme valeur de mesure

Si coché, la valeur est une valeur de mesure est s'affiche dans le graphique et le tableau de données. De plus, une étiquette doit être spécifiée afin d'identifier la valeur.



## 8.7. Bascule D asynchrone

Une composante utilisée pour stocker une valeur. La valeur sur l'entrée D est stockée lors du front montant du signal d'horloge C. Il y a deux entrées additionnelles pour mémoriser et réinitialiser immédiatement, sans signal d'horloge. La largeur de bit peut être sélectionnée ce qui permet de stocker plusieurs bits. Exportable en VHDL/Verilog.

### Entrées

#### Set

Une entrée "set" asynchrone, pour mémoriser. Si la valeur est à 1, tous les bits de sortie sont mis à 1.

#### D

L'entrée du bit à stocker.

#### C

Signal d'horloge pour stocker une valeur. La valeur à l'entrée D est stockée sur un front montant de cette entrée.

#### Clr

Une entrée "clear" asynchrone, pour effacer. Si la valeur est à 1, tous les bits de sortie sont mis à 0.

### Sorties

#### Q

Retourne la valeur stockée.

#### $\neg Q$

Retourne l'inverse de la valeur stockée.

### Attributs

#### Bits de données

Le nombre de bits utilisés pour les données.

#### Étiquette

Le nom de cet élément.

#### Entrées inversées

Sélectionner les entrées qui doivent être inversées.

#### Rotation

L'orientation de l'élément dans le circuit.

#### Miroir

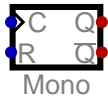
Reflète la composante dans le circuit.

#### Par défaut

Cette valeur est mise lorsque le circuit débute. Pour un démultiplexeur, cette valeur est mise pour les sorties non sélectionnées.

#### Utiliser comme valeur de mesure

Si coché, la valeur est une valeur de mesure est s'affiche dans le graphique et le tableau de données. De plus, une étiquette doit être spécifiée afin d'identifier la valeur.



## 8.8. Bascule monostable

Une bascule monostable est mise à 1 au front montant du signal d'horloge. Après un délai configurable, la bascule est réinitialisée automatiquement. La bascule monostable est redéclenchable. Elle peut seulement être utilisée s'il y a exactement une composante horloge dans le circuit. Cette composante horloge est utilisée comme temps de base pour mesurer le délai.

### Entrées

- C  
Le signal d'horloge. Un front montant fixe la sortie de la bascule monostable.
- R  
L'entrée "reset". Une valeur haute réinitialise la bascule monostable.

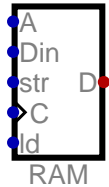
### Sorties

- Q  
Sortie
- $\neg Q$   
Inverse de la sortie

### Attributs

- Étiquette  
Le nom de cet élément.
- Largeur d'impulsion  
La largeur d'impulsion est mesurée en cycles d'horloge.
- Entrées inversées  
Sélectionner les entrées qui doivent être inversées.
- Rotation  
L'orientation de l'élément dans le circuit.
- Miroir  
Reflète la composante dans le circuit.
- Par défaut  
Cette valeur est mise lorsque le circuit débute. Pour un démultiplexeur, cette valeur est mise pour les sorties non sélectionnées.
- Utiliser comme valeur de mesure  
Si coché, la valeur est une valeur de mesure est s'affiche dans le graphique et le tableau de données. De plus, une étiquette doit être spécifiée afin d'identifier la valeur.

## 9. Mémoire - RAM



### 9.1. RAM, ports séparés

Un module de RAM avec des entrées séparées pour stocker une valeur et émettre une valeur stockée. Exportable en VHDL/Verilog.

#### Entrées

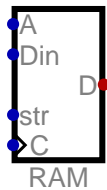
- A  
L'adresse à laquelle lire et écrire.
- Din  
La valeur à stocker dans la RAM.
- str  
Si cette entrée est active et lorsque le signal d'horloge est haut, la valeur est stockée.
- C  
Signal d'horloge
- Id  
Si cette entrée est active, la sortie est activée et la valeur devient visible à la sortie.

#### Sorties

- D  
La valeur de sortie

#### Attributs

- Bits de données  
Le nombre de bits utilisés pour les données.
- Bits d'adressage  
Nombre de bits utilisés pour l'adresse.
- Étiquette  
Le nom de cet élément.
- Rotation  
L'orientation de l'élément dans le circuit.
- Format de nombre  
Le format utilisé pour afficher les nombres.
- Décimales (virgule fixe)  
Nombre de décimales binaires pour le format à virgule fixe.
- Mémoire de programme  
Fait de cette ROM une mémoire de programme. Ainsi, elle devient accessible par un éditeur externe.



## 9.2. RAM de bloc, ports séparés

Un module de RAM avec des entrées séparées pour stocker une valeur et émettre une valeur stockée. Cette RAM met sa sortie à jour uniquement sur un front montant du signal d'horloge. Cela permet de l'utiliser dans un FPGA. Exportable en VHDL/Verilog.

### Entrées

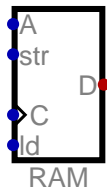
- A  
L'adresse à laquelle lire et écrire.
- Din  
La valeur à stocker dans la RAM.
- str  
Si cette entrée est active et lorsque le signal d'horloge est haut, la valeur est stockée.
- C  
Signal d'horloge

### Sorties

- D  
La valeur de sortie

### Attributs

- Bits de données  
Le nombre de bits utilisés pour les données.
- Bits d'adressage  
Nombre de bits utilisés pour l'adresse.
- Étiquette  
Le nom de cet élément.
- Rotation  
L'orientation de l'élément dans le circuit.
- Mémoire de programme  
Fait de cette ROM une mémoire de programme. Ainsi, elle devient accessible par un éditeur externe.



### 9.3. RAM, port bidirectionnel

Un module de RAM avec un port bidirectionnel pour stocker une valeur et émettre une valeur stockée.

#### Entrées

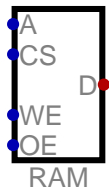
- A  
L'adresse à laquelle lire et écrire.
- str  
Si cette entrée est active et lorsque le signal d'horloge est haut, la valeur est stockée.
- C  
Signal d'horloge
- ld  
Si cette entrée est active, la sortie est activée et la valeur devient visible à la sortie.

#### Sorties

- D  
Une connexion bidirectionnelle de valeur

#### Attributs

- Bits de données  
Le nombre de bits utilisés pour les données.
- Bits d'adressage  
Nombre de bits utilisés pour l'adresse.
- Étiquette  
Le nom de cet élément.
- Rotation  
L'orientation de l'élément dans le circuit.
- Format de nombre  
Le format utilisé pour afficher les nombres.
- Décimales (virgule fixe)  
Nombre de décimales binaires pour le format à virgule fixe.
- Mémoire de programme  
Fait de cette ROM une mémoire de programme. Ainsi, elle devient accessible par un éditeur externe.



#### 9.4. RAM, sélection de puce

Un module de RAM avec un port bidirectionnel pour stocker une valeur et émettre une valeur stockée. Si l'entrée CS est basse, la composante est désactivée. Cela permet de créer une RAM plus grande à partir de RAM plus petite et un décodeur d'adresse. Le cycle d'écriture fonctionne comme suit: en activant l'entrée CS, la composante est activée. Un front montant à l'entrée WE verrouille l'adresse, et le front descendant suivant stocke la valeur.

##### Entrées

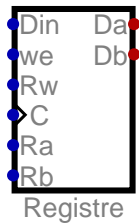
- A  
L'adresse à laquelle lire et écrire.
- CS  
Si cette entrée est active, la RAM est activée. Sinon la sortie est dans état de haute impédance.
- WE  
Si cette entrée est activée, la valeur est écrite à la RAM.
- OE  
Si cette entrée est active, la valeur stockée est émise.

##### Sorties

- D  
Une connexion bidirectionnelle de valeur.

##### Attributs

- Bits de données  
Le nombre de bits utilisés pour les données.
- Bits d'adressage  
Nombre de bits utilisés pour l'adresse.
- Étiquette  
Le nom de cet élément.
- Entrées inversées  
Sélectionner les entrées qui doivent être inversées.
- Rotation  
L'orientation de l'élément dans le circuit.
- Mémoire de programme  
Fait de cette ROM une mémoire de programme. Ainsi, elle devient accessible par un éditeur externe.



## 9.5. Banc de registres

Mémoire avec un port qui permet d'écrire et deux ports qui permettent de lire de la mémoire simultanément. Peut être utilisé pour implanter les registres d'un processeur. Deux registres peuvent être lus en même temps et un troisième peut être écrit. Exportable en VHDL/Verilog.

### Entrées

- Din  
La valeur à stocker dans la RAM.
- we  
Si cette entrée est active et lorsque le signal d'horloge est haut, la valeur est stockée.
- Rw  
Le registre dans lequel la valeur est écrite.
- C  
Signal d'horloge
- Ra  
Le registre qui est visible au port A.
- Rb  
Le registre qui est visible au port B.

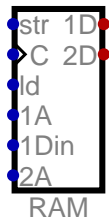
### Sorties

- Da  
Port de sortie A
- Db  
Port de sortie B

### Attributs

- Bits de données  
Le nombre de bits utilisés pour les données.
- Bits d'adressage  
Nombre de bits utilisés pour l'adresse.
- Étiquette  
Le nom de cet élément.
- Rotation  
L'orientation de l'élément dans le circuit.





## 9.6. RAM, port double

RAM avec un port qui sert à la fois à la lecture et l'écriture et un second port de lecture seulement. Ce second port peut être utilisé pour donner un accès à une logique graphique à partir du contenu de la mémoire. Ainsi, un processeur peut écrire à la RAM et une unité graphique peut y lire simultanément. Exportable en VHDL/Verilog.

### Entrées

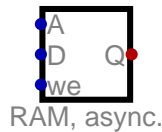
- str  
Si cette entrée est active et lorsque le signal d'horloge est haut, la valeur est stockée.
- C  
Signal d'horloge
- 1d  
Si cette entrée est active, la sortie est activée et la valeur est visible à la sortie 1D.
- 1A  
L'adresse à laquelle le port 1 est lu ou écrit.
- 1Din  
La valeur à stocker dans la RAM.
- 2A  
L'adresse utilisée pour lire du port 2.

### Sorties

- 1D  
Port de sortie 1.
- 2D  
Port de sortie 2

### Attributs

- Bits de données  
Le nombre de bits utilisés pour les données.
- Bits d'adressage  
Nombre de bits utilisés pour l'adresse.
- Étiquette  
Le nom de cet élément.
- Rotation  
L'orientation de l'élément dans le circuit.
- Mémoire de programme  
Fait de cette ROM une mémoire de programme. Ainsi, elle devient accessible par un éditeur externe.



### 9.7. RAM, async.

Aussi longtemps que WE est actif, les données sont stockées dans la RAM. Correspond à une RAM très simple, où les lignes d'adresse et de données sont connectées directement aux cellules de mémoire. Exportable en VHDL/Verilog.

#### Entrées

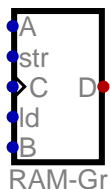
- A  
L'adresse à laquelle lire et écrire.
- D  
La valeur à stocker
- we  
Si cette entrée est activée, la valeur est écrite à la RAM lorsque A ou D changent.

#### Sorties

- Q  
La valeur de sortie

#### Attributs

- Bits de données  
Le nombre de bits utilisés pour les données.
- Bits d'adressage  
Nombre de bits utilisés pour l'adresse.
- Entrées inversées  
Sélectionner les entrées qui doivent être inversées.
- Étiquette  
Le nom de cet élément.
- Rotation  
L'orientation de l'élément dans le circuit.
- Mémoire de programme  
Fait de cette ROM une mémoire de programme. Ainsi, elle devient accessible par un éditeur externe.



### 9.8. RAM graphique

Sert à afficher une image matricielle (bitmap). Cet élément se comporte comme une RAM. De plus, il affiche son contenu sur un écran. Chaque pixel est représenté par une adresse en mémoire. La valeur stockée définit la couleur de son pixel à l'aide d'une palette de couleurs

fixe. Il y a deux tampons d'écran implantés pour permettre le changement de page. L'entrée B sélectionne le tampon à afficher. Ainsi, la taille totale de la mémoire est de  $dx * dy * 2$  mots. La palette de couleurs utilisée fonctionne comme suit: les indices 0-9 correspondent aux couleurs blanc, noir, rouge, vert, bleu, jaune, cyan, magenta, orange et rose. Les indices 32-63 correspondent des nuances de gris et les indices 64-127 correspondent 64 couleurs avec deux bits par canal de couleur. Cela permet une palette simple qui peut être adressée avec seulement 7 bits. Si l'architecture permet un indice 16 bits, à partir de l'indice 0x8000, un mode 5 bits avec plus de couleurs est utilisé, qui permet 32768 couleurs.

#### Entrées

- A  
L'adresse à laquelle lire et écrire.
- str  
Si cette entrée est active et lorsque le signal d'horloge est haut, la valeur est stockée.
- C  
Signal d'horloge
- ld  
Si cette entrée est active, la sortie est activée et la valeur devient visible à la sortie.
- B  
Sélectionne le tampon d'écran à afficher

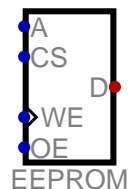
#### Sorties

- D  
Une connexion bidirectionnelle de valeur.

#### Attributs

- Bits de données  
Le nombre de bits utilisés pour les données.
- Étiquette  
Le nom de cet élément.
- Largeur en pixels  
La largeur de l'écran en pixels.
- Hauteur en pixels  
La hauteur de l'écran en pixels.
- Rotation  
L'orientation de l'élément dans le circuit.

## 10. Mémoire - EEPROM



### 10.1. EEPROM

Un module EEPROM avec un port bidirectionnel pour stocker une valeur et émettre une valeur stockée.

## Entrées

A

L'adresse à laquelle lire et écrire.

CS

Si cette entrée est active, l'EEPROM est activée. Sinon la sortie est dans un état de haute impédance.

WE

Si cette entrée est activée, la valeur est écrite à l'EEPROM.

OE

Si cette entrée est active, la valeur stockée est émise.

## Sorties

D

Une connexion bidirectionnelle de valeur.

## Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Bits d'adressage

Nombre de bits utilisés pour l'adresse.

Étiquette

Le nom de cet élément.

Entrées inversées

Sélectionner les entrées qui doivent être inversées.

Données

Les valeurs stockées dans cet élément.

Rotation

L'orientation de l'élément dans le circuit.

Format de nombre

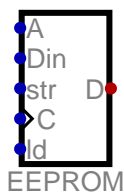
Le format utilisé pour afficher les nombres.

Décimales (virgule fixe)

Nombre de décimales binaires pour le format à virgule fixe.

Mémoire de programme

Fait de cette ROM une mémoire de programme. Ainsi, elle devient accessible par un éditeur externe.



## 10.2. EEPROM, ports séparés

Un module EEPROM avec des entrées séparées pour stocker une valeur et émettre une valeur stockée.

## Entrées

- A  
L'adresse à laquelle lire et écrire.
- Din  
La valeur à stocker dans le EEPROM.
- str  
Si cette entrée est active et lorsque le signal d'horloge est haut, la valeur est stockée.
- C  
Signal d'horloge
- ld  
Si cette entrée est active, la sortie est activée et la valeur devient visible à la sortie.

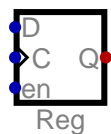
## Sorties

- D  
La valeur de sortie

## Attributs

- Bits de données  
Le nombre de bits utilisés pour les données.
- Bits d'adressage  
Nombre de bits utilisés pour l'adresse.
- Étiquette  
Le nom de cet élément.
- Données  
Les valeurs stockées dans cet élément.
- Rotation  
L'orientation de l'élément dans le circuit.
- Format de nombre  
Le format utilisé pour afficher les nombres.
- Décimales (virgule fixe)  
Nombre de décimales binaires pour le format à virgule fixe.
- Mémoire de programme  
Fait de cette ROM une mémoire de programme. Ainsi, elle devient accessible par un éditeur externe.

# 11. Mémoire



## 11.1. Registre

Une composante pour stocker des valeurs. La largeur de bit du mot peut être sélectionnée. Contrairement à une bascule D, le registre fournit une entrée qui active l'horloge. Exportable en VHDL/Verilog.

## Entrées

D

L'entrée du mot à stocker.

C

Signal d'horloge. Un front montant stocke la valeur à l'entrée D.

en

L'entrée "enable", pour activer. Il est seulement possible de stocker une valeur si cette entrée est active.

## Sorties

Q

Retourne la valeur stockée.

## Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Étiquette

Le nom de cet élément.

Entrées inversées

Sélectionner les entrées qui doivent être inversées.

Rotation

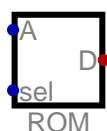
L'orientation de l'élément dans le circuit.

Pointeur d'instruction

Fait de ce registre un pointeur d'instruction (ou compteur ordinal). La valeur de ce registre est retournée à un assembleur externe pour marquer la ligne de code courante durant le débogage.

Utiliser comme valeur de mesure

Si coché, la valeur est une valeur de mesure est s'affiche dans le graphique et le tableau de données. De plus, une étiquette doit être spécifiée afin d'identifier la valeur.



## 11.2. ROM

Une composante de mémoire non volatile. Les données stockées peuvent être éditées dans le dialog d'attributs. Exportable en VHDL/Verilog.

## Entrées

A

Cette entrée définit l'adresse d'un mot à émettre à la sortie.

sel

Si cette entrée est active, la sortie est activée. Sinon la sortie est dans état de haute impédance.

## Sorties

### D

Le mot sélectionné si l'entrée "sel" est active.

## Attributs

### Bits de données

Le nombre de bits utilisés pour les données.

### Bits d'adressage

Nombre de bits utilisés pour l'adresse.

### Étiquette

Le nom de cet élément.

### Données

Les valeurs stockées dans cet élément.

### Rotation

L'orientation de l'élément dans le circuit.

### Format de nombre

Le format utilisé pour afficher les nombres.

### Décimales (virgule fixe)

Nombre de décimales binaires pour le format à virgule fixe.

### Mémoire de programme

Fait de cette ROM une mémoire de programme. Ainsi, elle devient accessible par un éditeur externe.

### Recharger au début

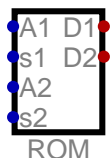
Recharger le fichier HEX à chaque fois que le modèle débute.

### Fichier

Fichier à charger dans la ROM.

### Utiliser big endian lors de l'importation

Utiliser l'ordre des octets big endian ("gros-boutiste") lors de l'importation



## 11.3. ROM, ports séparés

Une composante de mémoire non volatile. Les données stockées peuvent être éditées dans le dialog d'attributs.

## Entrées

A1

Cette entrée définit l'adresse du mot qui sera émis sur D1.

s1

Si cette entrée est haute, la sortie D1 est activée. Si elle est basse, la sortie est dans un état de haute impédance.

A2

Cette entrée définit l'adresse du mot qui sera émis sur D2.

s2

Si cette entrée est haute, la sortie D2 est activée. Si elle est basse, la sortie est dans un état de haute impédance.

## Sorties

D1

Le mot sélectionné si l'entrée s1 est haute.

D2

Le mot sélectionné si l'entrée s2 est haute.

## Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Bits d'adressage

Nombre de bits utilisés pour l'adresse.

Étiquette

Le nom de cet élément.

Données

Les valeurs stockées dans cet élément.

Rotation

L'orientation de l'élément dans le circuit.

Format de nombre

Le format utilisé pour afficher les nombres.

Décimales (virgule fixe)

Nombre de décimales binaires pour le format à virgule fixe.

Mémoire de programme

Fait de cette ROM une mémoire de programme. Ainsi, elle devient accessible par un éditeur externe.

Recharger au début

Recharger le fichier HEX à chaque fois que le modèle débute.

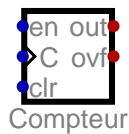
Fichier

Fichier à charger dans la ROM.

Utiliser big endian lors de l'importation

Utiliser l'ordre des octets big endian ("gros-boutiste") lors de l'importation





## 11.4. Compteur

Une composante simple pour compter. Le signal d'horloge incrémente le compteur. Peut être réinitialisée à 0 avec l'entrée "clr" (clear). Le nombre de bits est ajustable dans le dialogue des attributs. Exportable en VHDL/Verilog.

### Entrées

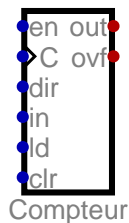
- en  
Si mise à 1 le compteur est activé!
- C  
Le signal d'horloge. Un front montant incrémente le compteur.
- clr  
Si mis à 1, réinitialise le compteur de façon synchrone.

### Sorties

- out  
Retourne la valeur comptée.
- ovf  
Indicateur de débordement. Si cette sortie est à 1 alors le compteur est à sa valeur maximale et l'entrée d'activation "en" est à 1.

### Attributs

- Bits de données  
Le nombre de bits utilisés pour les données.
- Entrées inversées  
Sélectionner les entrées qui doivent être inversées.
- Étiquette  
Le nom de cet élément.
- Rotation  
L'orientation de l'élément dans le circuit.
- Utiliser comme valeur de mesure  
Si coché, la valeur est une valeur de mesure est s'affiche dans le graphique et le tableau de données. De plus, une étiquette doit être spécifiée afin d'identifier la valeur.
- Pointeur d'instruction  
Fait de ce registre un pointeur d'instruction (ou compteur ordinal). La valeur de ce registre est retournée à un assembleur externe pour marquer la ligne de code courante durant le débogage.



### 11.5. Compteur prédéfini

Un compteur pour lequel la valeur peut être changée. De plus, une valeur maximale et une direction peuvent être spécifiés. Exportable en VHDL/Verilog.

#### Entrées

- en  
Si mis à 1, le compteur est activé!
- C  
Signal d'horloge. Un front montant incrémente ou décrémente le compteur.
- dir  
Spécifie la direction du compte. Un zéro signifie vers le haut (en ordre croissant).
- in  
Cette valeur est stockée dans le compteur lorsque l'entrée "ld" est activée.
- ld  
Si activée, la valeur dans l'entrée "in" est stockée dans le compteur au prochain signal d'horloge.
- clr  
Si mis à 1, réinitialise le compteur de façon synchrone.

#### Sorties

- out  
Retourne la valeur comptée.
- ovf  
Indicateur de débordement. Si cette sortie est à 1 et que l'entrée d'activation "en" est à 1 alors le compteur est à sa valeur maximale s'il compte vers le haut, ou à zéro s'il compte vers le bas.

#### Attributs

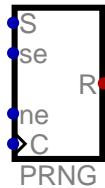
- Bits de données  
Le nombre de bits utilisés pour les données.
- Valeur maximale  
Si un zéro est entré, la valeur maximale est utilisée (tous les bits à un).
- Entrées inversées  
Sélectionner les entrées qui doivent être inversées.
- Étiquette  
Le nom de cet élément.
- Rotation  
L'orientation de l'élément dans le circuit.

Utiliser comme valeur de mesure

Si coché, la valeur est une valeur de mesure est s'affiche dans le graphique et le tableau de données. De plus, une étiquette doit être spécifiée afin d'identifier la valeur.

Pointeur d'instruction

Fait de ce registre un pointeur d'instruction (ou compteur ordinal). La valeur de ce registre est retournée à un assembleur externe pour marquer la ligne de code courante durant le débogage.



## 11.6. Générateur de nombre aléatoire

Peut être utilisé pour générer des nombres aléatoires. Lorsque la simulation débute, le générateur est réinitialisé afin que des nombres aléatoires différents soit générés à chaque simulation. Le générateur peut être initialisé durant la simulation pour modifier la graine utilisée pour générer la séquence de nombres pseudo-aléatoires.

Entrées

S

Nouvelle valeur de graine utilisée pour le générateur.

se

Si activée, le générateur de nombre aléatoire est réinitialisé avec la nouvelle valeur de graine au prochain front montant de l'horloge.

ne

Si activée, un nouveau nombre aléatoire est émis à la sortie au prochain front montant de l'horloge.

C

Signal d'horloge

Sorties

R

Émet un nombre pseudo-aléatoire.

Attributs

Bits de données

Le nombre de bits utilisés pour les données.

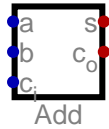
Étiquette

Le nom de cet élément.

Rotation

L'orientation de l'élément dans le circuit.

## 12. Arithmétique



### 12.1. Addition

Une composante pour des calculs d'addition simples. Additionne deux valeurs entières aux entrées A et B ( $A+B$ ). Le résultat est incrémenté de un si l'entrée de retenue est active. Exportable en VHDL/Verilog.

#### Entrées

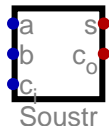
- a  
Première valeur à additionner.
- b  
Seconde valeur à additionner.
- c\_i  
Entrée de retenue, si active le résultat sera incrémenté de un.

#### Sorties

- s  
Le résultat de l'addition
- c\_o  
Indicateur de retenue. Si actif, il y a eu un débordement.

#### Attributs

- Étiquette  
Le nom de cet élément.
- Bits de données  
Le nombre de bits utilisés pour les données.
- Rotation  
L'orientation de l'élément dans le circuit.



### 12.2. Soustraction

Une composante pour des soustractions simples. Soustrait des nombres binaires aux entrées A et B ( $A-B$ ). Le résultat est décrémenté de 1 si l'entrée de retenue est active. Exportable en VHDL/Verilog.

### Entrées

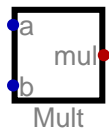
- a  
L'entrée A de la soustraction.
- b  
L'entrée B de la soustraction
- c\_i  
Entrée de retenue, si active le résultat sera décrémenté de 1.

### Sorties

- s  
Retourne le résultat de la soustraction.
- c\_o  
Émet un 1 si un débordement a eu lieu.

### Attributs

- Étiquette  
Le nom de cet élément.
- Bits de données  
Le nombre de bits utilisés pour les données.
- Rotation  
L'orientation de l'élément dans le circuit.



## 12.3. Multiplication

Une composante pour la multiplication. Multiplie les nombres entiers aux entrées A et B.  
Exportable en VHDL/Verilog.

### Entrées

- a  
L'entrée A de la multiplication
- b  
L'entrée B de la multiplication

### Sorties

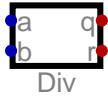
- mul  
La sortie pour le résultat de la multiplication.

### Attributs

- Étiquette  
Le nom de cet élément.
- Opération signée  
Si coché, l'opération est effectuée avec des valeurs signées (complément 2).
- Bits de données  
Le nombre de bits utilisés pour les données.

**Rotation**

L'orientation de l'élément dans le circuit.

**12.4. Division**

Une composante pour la division. Divise le nombre entier à l'entrée A par le nombre entier à l'entrée B. Si le diviseur est zéro, le nombre est divisé par un à la place. Pour la division signée, le reste est toujours positif.

**Entrées**

- a  
Dividende
- b  
Diviseur

**Sorties**

- q  
Quotient
- r  
Reste

**Attributs****Étiquette**

Le nom de cet élément.

**Bits de données**

Le nombre de bits utilisés pour les données.

**Opération signée**

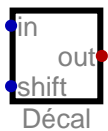
Si coché, l'opération est effectuée avec des valeurs signées (complément 2).

Le reste est toujours positif.

Si coché, le reste d'une division signée sera toujours positif.

**Rotation**

L'orientation de l'élément dans le circuit.

**12.5. Décaleur**

Une composante pour décaler des bits. Décale la valeur d'entrée par un nombre de bits donné par l'entrée "shift".

### Entrées

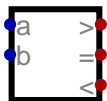
- in  
Entrée avec les bits à décaler.
- shift  
Entrée avec le nombre de bits à décaler.

### Sorties

- out  
Sortie de la valeur décalée.

### Attributs

- Étiquette  
Le nom de cet élément.
- Bits de données  
Le nombre de bits utilisés pour les données.
- L'entrée de décalage est signée  
Si coché, l'entrée de décalage est une valeur signée (complément 2).
- Direction  
Détermine la direction du décalage.
- Mode  
Mode du décaleur
- Rotation  
L'orientation de l'élément dans le circuit.



## 12.6. Comparateur

Une composante pour comparer des valeurs binaires. Compare les nombres binaires aux entrées A et B et met la sortie correspondante à 1. Exportable en VHDL/Verilog.

### Entrées

- a  
L'entrée A à comparer.
- b  
L'entrée B à comparer.

### Sorties

- >  
La sortie est active si A est plus grand que B.
- =  
La sortie est active si A est égal à B.
- <  
La sortie est active si A est plus petit que B.

### Attributs

- Étiquette  
Le nom de cet élément.

Bits de données

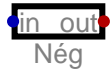
Le nombre de bits utilisés pour les données.

Opération signée

Si coché, l'opération est effectuée avec des valeurs signées (complément 2).

Rotation

L'orientation de l'élément dans le circuit.



## 12.7. Négation

Négation selon le complément 2. Exportable en VHDL/Verilog.

Entrées

in

L'entrée du mot à inverser selon le complément 2.

Sorties

out

Retourne le résultat de l'inversion selon le complément 2.

Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Rotation

L'orientation de l'élément dans le circuit.



## 12.8. Extension de signe

Augmente la largeur binaire d'une valeur signée en gardant le même signe que la valeur.

Si l'entrée est un seul bit, ce bit est répété sur tous les bits de sortie. Exportable en VHDL/Verilog.

Entrées

in

Valeur d'entrée. La largeur binaire d'entrée doit être plus petite que la largeur binaire de sortie!

Sorties

out

Valeur d'entrée étendue. La largeur binaire d'entrée doit être plus petite que la largeur binaire de sortie!

Attributs

Étiquette

Le nom de cet élément.



Largeur binaire d'entrée

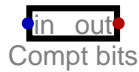
Le nombre de bits de sortie doit être plus grand que le nombre de bits d'entrée.

Largeur binaire de sortie

Le nombre de bits de sortie doit être plus grand que le nombre de bits d'entrée.

Rotation

L'orientation de l'élément dans le circuit.



## 12.9. Compteur de bits

Retourne le nombre de bits à 1 dans la valeur d'entrée.

Entrées

in

La valeur d'entrée pour laquelle les bits à 1 sont comptés.

Sorties

out

Retourne le nombre de bits à 1.

Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Rotation

L'orientation de l'élément dans le circuit.

## 13. Interrupteurs



### 13.1. Interrupteur

Interrupteur simple. Il n'y a pas de délai de transmission: un changement est propagé instantanément.

Sorties

A1

Une des connexions de l'interrupteur.

B1

Une des connexions de l'interrupteur.

Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Étiquette

Le nom de cet élément.

Nombre de pôles

Nombre de poles disponibles.

Fermé

Donne l'état initial de l'interrupteur.

Rotation

L'orientation de l'élément dans le circuit.

Miroir

Reflète la composante dans le circuit.

Interrupteur se comporte comme une entrée

Si le modèle est analysé, l'interrupteur se comporte comme une entrée, où "ouverte" correspond à '0' et "fermée" à '1'.



### 13.2. Interrupteur bidirectionnel

Interrupteur bidirectionnel. Il n'y a pas de délai de transmission: un changement est propagé instantanément.

Sorties

A1

Une des connexions de l'interrupteur.

B1

Une des connexions de l'interrupteur.

C1

Une des connexions de l'interrupteur.

Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Étiquette

Le nom de cet élément.

Nombre de pôles

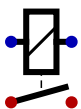
Nombre de poles disponibles.

Rotation

L'orientation de l'élément dans le circuit.

Miroir

Reflète la composante dans le circuit.



### 13.3. Relais

Un relais est un interrupteur qui peut être contrôlé par une bobine. Si le courant circule dans la bobine, l'interrupteur est ouvert ou fermé. Il n'y a pas de diode antiretour alors le sens du courant n'a pas d'importance. L'interrupteur est actionné si les deux entrées ont des valeurs différentes. Le relais se comporte de façon similaire à une porte XOR.

### Entrées

in1

Une des entrées qui contrôle le relais.

in2

Une des entrées qui contrôle le relais.

### Sorties

A1

Une des connexions de l'interrupteur.

B1

Une des connexions de l'interrupteur.

### Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Étiquette

Le nom de cet élément.

Nombre de pôles

Nombre de poles disponibles.

Relais normalement fermé.

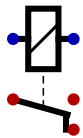
Si coché, le relais est fermé si l'entrée est basse.

Rotation

L'orientation de l'élément dans le circuit.

Miroir

Reflète la composante dans le circuit.



## 13.4. Relais bidirectionnel

Un relais est un interrupteur qui peut être contrôlé par une bobine. Si le courant circule dans la bobine, l'interrupteur est ouvert ou fermé. Il n'y a pas de diode antiretour alors le sens du courant n'a pas d'importance. L'interrupteur est actionné si les deux entrées ont des valeurs différentes. Le relais se comporte de façon similaire à une porte XOR.

### Entrées

in1

Une des entrées qui contrôle le relais.

in2

Une des entrées qui contrôle le relais.

## Sorties

- A1  
Une des connexions de l'interrupteur.
- B1  
Une des connexions de l'interrupteur.
- C1  
Une des connexions de l'interrupteur.

## Attributs

- Bits de données  
Le nombre de bits utilisés pour les données.
- Étiquette  
Le nom de cet élément.
- Nombre de pôles  
Nombre de poles disponibles.
- Rotation  
L'orientation de l'élément dans le circuit.
- Miroir  
Reflète la composante dans le circuit.

**13.5. FET de type P**

Transistor à effet de champ avec dopage de type P. Le substrat est connecté à l'alimentation et le transistor est simulé sans diode de corps.

## Entrées

- G  
Grille

## Sorties

- S  
Source
- D  
Drain

## Attributs

- Bits de données  
Le nombre de bits utilisés pour les données.
- Unidirectionnel  
Les transistors unidirectionnels propagent un signal seulement de la source au drain. Ils sont beaucoup plus rapides à simuler que les transistors bidirectionnels. Dans ce mode, puisqu'il n'y a pas de rétroaction du drain à la source, le transistor ne peut pas court-circuiter les fils connectés lorsqu'il est conducteur. Ainsi, ce mode est nécessaire pour simuler certains circuits CMOS.
- Étiquette  
Le nom de cet élément.

Rotation

L'orientation de l'élément dans le circuit.

Miroir

Reflète la composante dans le circuit.



### 13.6. FET de type N

Transistor à effet de champ avec dopage de type N. Le substrat est connecté à la masse et le transistor est simulé sans diode de corps.

Entrées

G

Grille

Sorties

D

Drain

S

Source

Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Unidirectionnel

Les transistors unidirectionnels propagent un signal seulement de la source au drain. Ils sont beaucoup plus rapides à simuler que les transistors bidirectionnels. Dans ce mode, puisqu'il n'y a pas de rétroaction du drain à la source, le transistor ne peut pas court-circuiter les fils connectés lorsqu'il est conducteur. Ainsi, ce mode est nécessaire pour simuler certains circuits CMOS.

Étiquette

Le nom de cet élément.

Rotation

L'orientation de l'élément dans le circuit.

Miroir

Reflète la composante dans le circuit.



### 13.7. Fusible

Un fusible utilisé pour construire une mémoire programmée une seule fois

Sorties

out1

Une des connexions de l'interrupteur.

out2

Une des connexions de l'interrupteur.

## Attributs

## Programmé

Si la diode est "sautée" ou "programmée". Pour un FET à grille flottante, la grille flottante est chargée. Il est aussi possible de changer cette option avec la touche [P].

## Rotation

L'orientation de l'élément dans le circuit.



### 13.8. Diode à l'alimentation

Une diode unidirectionnelle simplifiée, utilisée pour le tirage d'un fil à l'alimentation. Elle est utilisée pour créer un OU câblé. Donc il est nécessaire de connecter une résistance de rappel à la sortie de la diode. Dans la simulation la diode se comporte comme une porte active avec une table de vérité trivalente: si l'entrée est haute, la sortie l'est aussi. Dans tous les autres cas (l'entrée est basse ou Hi-Z), la sortie est dans un état de haute impédance. Donc deux diodes antiparallèles connectées peuvent se garder l'une et l'autre dans un état haut, ce qui n'est pas possible avec de vraies diodes. C'est une diode polarisée idéale: il n'y a pas de sans perte de tension à travers la diode.

## Entrées

in

Si l'entrée est haute alors la sortie l'est aussi. Dans tous les autres cas la sortie est dans état de haute impédance.

## Sorties

out

Si l'entrée est haute alors la sortie l'est aussi. Dans tous les autres cas la sortie est dans état de haute impédance.

## Attributs

## Programmé

Si la diode est "sautée" ou "programmée". Pour un FET à grille flottante, la grille flottante est chargée. Il est aussi possible de changer cette option avec la touche [P].

## Rotation

L'orientation de l'élément dans le circuit.



### 13.9. Diode à la masse

Une diode unidirectionnelle simplifiée, utilisée pour le rappel d'un fil à la masse. Elle est utilisée pour créer un ET câblé. Donc il est nécessaire de connecter une résistance de tirage à la sortie de la diode. Si l'entrée est basse, la sortie l'est aussi. Dans tous les autres cas (l'entrée est haute ou Hi-Z), la sortie est dans un état de haute impédance. Donc deux diodes antiparallèles connectées peuvent se garder l'une et l'autre dans un état bas, ce qui n'est pas possible avec de vraies diodes. C'est une diode polarisée idéale: il n'y a pas de sans perte de tension à travers la diode.

## Entrées

in

Si l'entrée est basse alors la sortie l'est aussi. Dans tous les autres cas la sortie est dans état de haute impédance.

## Sorties

out

Si l'entrée est basse alors la sortie l'est aussi. Dans tous les autres cas la sortie est dans état de haute impédance.

## Attributs

Programmé

Si la diode est "sautée" ou "programmée". Pour un FET à grille flottante, la grille flottante est chargée. Il est aussi possible de changer cette option avec la touche [P].

Rotation

L'orientation de l'élément dans le circuit.



### 13.10. FET de type P à grille flottante

Transistor à effet de champ avec dopage de type P avec grille flottante (floating gate). Le substrat est connecté à l'alimentation et le transistor est simulé sans diode de corps. S'il y a une charge stockée dans la grille flottante, le FET n'est pas conducteur même si la grille est basse.

## Entrées

G

Grille

## Sorties

S

Source

D

Drain

## Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Étiquette

Le nom de cet élément.

Programmé

Si la diode est "sautée" ou "programmée". Pour un FET à grille flottante, la grille flottante est chargée. Il est aussi possible de changer cette option avec la touche [P].

Rotation

L'orientation de l'élément dans le circuit.

Miroir

Reflète la composante dans le circuit.



### 13.11. FET de type N à grille flottante

Transistor à effet de champ avec dopage de type N avec grille flottante (floating gate). Le substrat est connecté à la masse et le transistor est simulé sans diode de corps. S'il y a une charge stockée dans la grille flottante, le FET n'est pas conducteur même si la grille est haute.

#### Entrées

G  
Grille

#### Sorties

D  
Drain  
S  
Source

#### Attributs

##### Bits de données

Le nombre de bits utilisés pour les données.

##### Étiquette

Le nom de cet élément.

##### Programmé

Si la diode est "sautée" ou "programmée". Pour un FET à grille flottante, la grille flottante est chargée. Il est aussi possible de changer cette option avec la touche [P].

##### Rotation

L'orientation de l'élément dans le circuit.

##### Miroir

Reflète la composante dans le circuit.



### 13.12. Porte de transmission

Une vraie porte de transmission est construite à partir de seulement deux transistors. Ainsi, elle est souvent utilisée pour économiser des transistors lors de l'implantation sur le silicium.

#### Entrées

S  
Entrée de contrôle  
 $\neg S$   
Entrée de contrôle inversée



#### Sorties

- A  
Entrée A
- B  
Entrée B

#### Attributs

- Bits de données  
Le nombre de bits utilisés pour les données.
- Rotation  
L'orientation de l'élément dans le circuit.

## 14. Divers

### Test

#### 14.1. Test

Décrit un test. Dans un test, vous pouvez décrire comment le circuit devrait se comporter. Il est ensuite possible de vérifier automatiquement si le comportement du circuit correspond réellement à cette description. Si ce n'est pas le cas, un message d'erreur est affiché. Un message d'aide dans l'éditeur de test décrit en détail la syntaxe utilisée pour les tests. Exportable en VHDL/Verilog.

#### Attributs

- Étiquette  
Le nom de cet élément.
- Données de test  
La description du test. Les détails de la syntaxe peuvent être trouvés dans le dialogue d'aide de l'éditeur de données de test.
- Activée  
Active ou désactive cette composante.

## 15. Divers - Décoration

### Texte

#### 15.1. Texte

Affiche du texte dans le circuit. N'affecte pas la simulation. Le texte peut être changé dans le dialogue d'attributs.

#### Attributs

- Description  
Une courte description de cet élément et son utilisation.
- Taille de police  
Détermine la taille de la police à utiliser pour ce texte.

**Rotation**

L'orientation de l'élément dans le circuit.

**Orientation**

Position de la coordonnée relativement au texte.

**Aligner sur la grille**

Si coché, la composante reste alignée sur la grille.

**Texte****15.2. Rectangle**

Affiche un rectangle dans le circuit. N'affecte pas la simulation. Si un trait d'union est d'utilisé comme titre, le titre n'est pas affiché.

**Attributs****Étiquette**

Le nom de cet élément.

**Largeur**

Largeur en unités de grille

**Hauteur**

Hauteur en unités de grille

**Taille de police**

Détermine la taille de la police à utiliser pour ce texte.

**Texte en dedans**

Place le texte à l'intérieur du rectangle.

**Texte en bas**

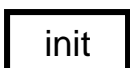
Place le texte au bas du rectangle.

**Text à droite**

Place le texte à droite du rectangle.

**Aligner sur la grille**

Si coché, la composante reste alignée sur la grille.

**16. Divers - Générique****16.1. Initialisation générique**

Code qui est exécuté pour simuler un circuit générique directement. Cette composante est requise si un circuit générique doit être simulé directement. Exportable en VHDL/Verilog.

**Attributs****Étiquette**

Le nom de cet élément.

**Activée**

Active ou désactive cette composante.

Paramétrisation générique  
Énoncés utilisés pour générer un circuit.

## Code

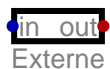
### 16.2. Code

Code qui est exécuté lorsqu'un circuit générique est rendu concret. Peut être utilisé, par exemple, pour ajouter des composantes ou des fils au circuit. Exportable en VHDL/Verilog.

Attributs

Paramétrisation générique  
Énoncés utilisés pour générer un circuit.

## 17. Divers - VHDL/Verilog



### 17.1. Externe

Une composante qui exécute un processus externe pour calculer la fonction logique. Peut être utilisé pour spécifier le comportement d'une composante en VHDL ou en Verilog. La simulation du comportement en tant que telle doit être faite avec un simulateur externe. À l'heure actuelle, seuls le simulateur de VHDL ghdl et le simulateur de Verilog Icarus Verilog sont pris en charge. L'étiquette de la composante doit correspondre au nom de l'entité ou du module! Exportable en VHDL/Verilog.

Entrées

in

Sorties

out

Attributs

Étiquette

Le nom de cet élément.

Largeur

Largeur du symbole si le circuit est utilisé comme une composante dans un autre circuit.

Entrées

Les entrées du processus externe. C'est une liste de noms de signaux séparée par des virgules. Pour chaque nom de signal, un nombre de bits séparé par un deux-points peut être spécifié. Les entrées d'un additionneur 8-bit pourraient ainsi être séparées avec "a:8,b:8,c\_in".

**Sorties**

Les sorties du processus externe. C'est une liste de noms de signaux séparée par des virgules. Pour chaque nom de signal, un nombre de bits séparé par un deux-points peut être spécifié. Les sorties d'un additionneur 8-bit pourraient ainsi être séparées avec "s:8,c\_out".

**Code du programme**

Le code du programme à exécuter par une application externe.

**Application**

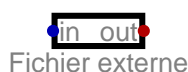
Définit l'application à utiliser.

**Options GHDL**

Options utilisées pour les étapes de traitement par GHDL.

**Options IVerilog**

Options utilisées pour toutes les étapes de traitement par IVerilog.

**17.2. Fichier externe**

Une composante qui exécute un processus externe pour calculer la fonction logique. Peut être utilisé pour spécifier le comportement d'une composante en VHDL ou en Verilog. La simulation du comportement en tant que telle doit être faite avec un simulateur externe. À l'heure actuelle, seuls le simulateur de VHDL ghdl et le simulateur de Verilog Icarus Verilog sont pris en charge. L'étiquette de la composante doit correspondre au nom de l'entité ou du module! Exportable en VHDL/Verilog.

**Entrées**

in

**Sorties**

out

**Attributs****Étiquette**

Le nom de cet élément.

**Largeur**

Largeur du symbole si le circuit est utilisé comme une composante dans un autre circuit.

**Entrées**

Les entrées du processus externe. C'est une liste de noms de signaux séparée par des virgules. Pour chaque nom de signal, un nombre de bits séparé par un deux-points peut être spécifié. Les entrées d'un additionneur 8-bit pourraient ainsi être séparées avec "a:8,b:8,c\_in".

**Sorties**

Les sorties du processus externe. C'est une liste de noms de signaux séparée par des virgules. Pour chaque nom de signal, un nombre de bits séparé par un deux-points peut être spécifié. Les sorties d'un additionneur 8-bit pourraient ainsi être séparées avec "s:8,c\_out".

**Code du programme**

Le fichier contenant le code du programme à exécuter par l'application externe

**Application**

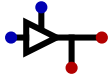
Définit l'application à utiliser.

**Options GHDL**

Options utilisées pour les étapes de traitement par GHDL.

**Options IVerilog**

Options utilisées pour toutes les étapes de traitement par IVerilog.



### 17.3. Commande de broche

Logique de commande pour une broche bidirectionnelle. Cette composante est nécessaire seulement dans un contexte de génération de VHDL ou de Verilog, dans le but de créer un port HDL bidirectionnel! Si vous ne voulez pas utiliser un port E/S bidirectionnel sur un FPGA, n'utilisez pas cette composante! L'élément de commande de broche ne peut pas être utilisé dans un circuit intégré, seulement dans le circuit du niveau supérieur! Exportable en VHDL/Verilog.

**Entrées**

wr

La valeur à émettre

oe

Active la sortie

**Sorties**

rd

La valeur à lire

pin

Le connecteur pour la broche en tant que telle. Seule une sortie peut être connectée ici.

**Attributs**

Bits de données

Le nombre de bits utilisés pour les données.

Rotation

L'orientation de l'élément dans le circuit.

Miroir

Reflète la composante dans le circuit.

## 18. Divers



### 18.1. Alimentation

N'a aucune fonction. S'assure que l'alimentation (VDD) et la masse (GND) sont connectées. Peut être utilisé dans les circuits 74xx pour générer les broches de l'alimentation, qui sont testées pour s'assurer que le câblage est correct.

#### Entrées

VDD

Doit être connecté à l'alimentation (VDD)!

GND

Doit être connecté à la masse (GND)!

#### Attributs

Étiquette

Le nom de cet élément.

Rotation

L'orientation de l'élément dans le circuit.



### 18.2. Séparateur bidirectionnel

Peut être utilisé pour les bus de données et simplifie particulièrement la construction de module de mémoire dans un boîtier DIL, car l'implantation du bus de données est simplifiée.

#### Entrées

OE

Lorsqu'activée, la valeur au terminal commun D est émise au bits de sortie D[i], sinon, les bits de sortie D[i] sont émis à la sortie commune D.

#### Sorties

D

La connexion de la valeur commune.

D0

Le bit de donnée 0 du séparateur de bus.

#### Attributs

Bits de données

Le nombre de bits utilisés pour les données.

Rotation

L'orientation de l'élément dans le circuit.

Miroir

Reflète la composante dans le circuit.

Étalement

Configure l'étalement des entrées et des sorties dans le circuit.



### 18.3. Réinitialisation

La sortie de cette composante est gardée haute lors de l'initialisation du circuit. Après que le circuit se soit stabilisé, la sortie devient basse. Si la sortie est inversée alors elle se comporte de la façon inverse. Exportable en VHDL/Verilog.

Sorties

Reset

Sortie de réinitialisation

Attributs

Étiquette

Le nom de cet élément.

Inverser la sortie

Si coché la sortie est inversée.

Rotation

L'orientation de l'élément dans le circuit.



### 18.4. Pause

Si cette composante est utilisée dans le circuit, le bouton "Exécuter jusqu'à une pause" entre les boutons "Commencer" et "Arrêter" est activé. Ce bouton exécute le circuit jusqu'à ce que cette composante reçoive un front montant. Cet élément peut aussi être utilisé pour déboguer en exécutant un circuit jusqu'à un point d'arrêt. Une instruction d'assembleur BRK (break) peut également être implantée. Cela permet à un programme de s'exécuter jusqu'à la prochaine instruction BRK. Cette fonction peut seulement être utilisée si l'horloge temps réel est désactivée!

Entrées

brk

Arrête la simulation haute vitesse si un front montant est détecté.

Attributs

Étiquette

Le nom de cet élément.

Activée

Active ou désactive cette composante.

Cycles d'attente

Si ce nombre de cycles est atteint dans signal de pause, une erreur est affichée.

#### Rotation

L'orientation de l'élément dans le circuit.



### 18.5. Arrêt

Un front montant à l'entrée arrête la simulation. A le même effet que d'appuyer sur le bouton "Arrêter" dans la barre d'outils.

#### Entrées

##### stop

Un front montant arrête la simulation.

#### Attributs

##### Étiquette

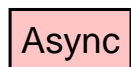
Le nom de cet élément.

##### Entrées inversées

Sélectionner les entrées qui doivent être inversées.

##### Rotation

L'orientation de l'élément dans le circuit.



### 18.6. Minuterie asynchrone

Permet la configuration d'une minuterie d'un circuit séquentiel asynchrone comme un pipeline de Muller. Le circuit doit être lancé en mode pas-à-pas et doit être en mesure d'atteindre un état stable au démarrage. Le circuit séquentiel peut alors être lancé interactivement ou avec une porte de réinitialisation. Il n'est pas permis d'utiliser une composante d'horloge ordinaire dans ce mode.

#### Attributs

##### Partir l'horloge temps réel

Si coché, l'horloge de temps d'exécution est partie quand le circuit débute.

##### Fréquence/Hz

La fréquence utilisée pour l'horloge temps réel.



## E Bibliothèque

- 27c801:** 8 Mbit (1Mb x 8) UV EPROM
- 28c010:** 1-Megabit (128K x 8) Paged Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 28c16:** 16K (2K x 8) Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 28c64:** 64K (8K x 8) Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 28c256:** 256K (32K x 8) Paged Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 28c512:** 512K-Bit (64K x 8) CMOS Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 7400:** quad 2-input NAND gate
- 7401:** quad 2-input NAND gate with open-collector outputs
- 7402:** quad 2-input NOR gate
- 7403:** quad 2-input NAND gate with open-collector outputs, different pinout than 7401
- 7404:** hex inverter
- 7405:** hex inverter, open-collector output
- 7406:** hex inverter buffer, open-collector output
- 7407:** hex buffer, open-collector output
- 7408:** quad 2-input AND gate
- 7409:** quad 2-input AND gate with open-collector outputs
- 7410:** triple 3-input NAND gate
- 7411:** triple 3-input AND gate
- 7412:** triple 3-input NAND gate with open-collector outputs
- 7413:** dual 4-input NAND gate, Schmitt trigger
- 7414:** hex inverter, Schmitt trigger
- 7415:** triple 3-input AND gate with open-collector outputs
- 7416:** hex inverter buffer, open-collector output, same as 7406
- 7417:** hex buffer, open-collector output, same as 7407
- 7420:** dual 4-input NAND gate
- 7421:** dual 4-input AND gate
- 7425:** dual 4-input NOR gate with strobe
- 7427:** triple 3-input NOR gate
- 7428:** quad 2-input NOR buffer
- 7430:** 8-input NAND gate
- 7432:** quad 2-input OR gate
- 7434:** hex buffer
- 7440:** dual 4-input NAND buffer
- 7442:** 4-line BCD to 10-line decimal decoder
- 7447:** BCD to 7-segment decoder, active low
- 7448:** BCD to 7-segment decoder, active high
- 7451:** 2-input/3-input AND-NOR gate
- 7454:** 2-3-2-3-line AND NOR gate
- 7455:** 2 wide 4-input AND-NOR gate
- 7458:** dual AND OR gate
- 7474:** dual D-flip-flop
- 7476:** dual J-K flip-flops with preset and clear
- 7480:** Gated Full Adder with Complementary Inputs and Complementary Sum Outputs

**7482:** 2-bit binary full adder  
**7483:** 4-bit binary full adder  
**7483Real:** 4-bit binary full adder, real gates  
**7485:** 4-bit comparator  
**7486:** quad 2-input XOR gate  
**7489:** 64-bit RAM  
**7490:** asynchronous two - five - decimal addition counter  
**7493:** 4-bit Binary Counter. Connect QA to CKB and clock on CKA for full 4-bit counter.  
Connect QB to R1 and QD to R2 for a BCD counter.  
**74107:** dual J-K flip-flops with clear  
**74109:** Dual J-NOT-K flip-flop with set and reset; positive-edge-trigger  
**74112:** Dual J-K negative-edge-triggered flip-flop, clear and preset  
**74116:** dual 4-bit D-type latches  
**74125:** Quadruple bus buffer gates with 3-state outputs (active low output enable)  
**74126:** Quadruple bus buffer gates with 3-state outputs (active high output enable)  
**74133:** 13-input NAND gate  
**74138:** 3-line to 8-line decoder/demultiplexer, inverted out  
**74139:** dual 2-line to 4-line decoder/demultiplexer  
**74147:** 10-line to 4-line priority encoder  
**74148:** 8-line to 3-Line priority encoder  
**74150:** 4-line to 16-line data selectors/multiplexers  
**74151:** 3-line to 8-line data selectors/multiplexers  
**74153:** dual 4-line to 1-line data selectors/multiplexers  
**74154:** 4-line to 16-line decoders/demultiplexers  
**74157:** quad 2-line to 1-line data selectors/multiplexers  
**74160:** decimal synchronous counter, async clear  
**74161:** hex synchronous counter, async clear  
**74162:** decimal synchronous counter  
**74162Real:** decimal synchronous counter, real gates  
**74163:** hex synchronous counter  
**74164:** 8-bit parallel-out serial shift register, asynchronous clear  
**74165:** parallel-load 8-bit shift register  
**74166:** 8-Bit Parallel-In/Serial-Out Shift Register  
**74173:** quad 3-state D flip-flop with common clock and reset  
**74174:** hex D-flip-flop  
**74175:** quad D-flip-flop  
**74181:** 4-bit arithmetic logic unit  
**74182:** look-ahead carry generator  
**74189:** 64-Bit Random Access Memory with 3-STATE Outputs  
**74190:** Presettable synchronous 4-bit bcd up/down counter  
**74191:** Presettable synchronous 4-bit binary up/down counter  
**74193:** Synchronous 4-Bit Up/Down Binary Counter with Dual Clock  
**74194:** 4-Bit Bidirectional Universal Shift Register  
**74194real:** 4-Bit Bidirectional Universal Shift Register, Databook implementation.  
**74198:** 8-bit shift register  
**74238:** 3-line to 8-line decoder/demultiplexer  
**74244:** octal 3-state buffer/line driver/line receiver  
**74245:** octal bus transceivers with 3-state outputs  
**74247:** BCD to 7-segment decoder, active low, tails on 6 and 9  
**74248:** BCD to 7-segment decoder, active high, tails on 6 and 9

**74253:** dual tri state 4-line to 1-line data selectors/multiplexers  
**74257:** quad 2-line to 1-line data selectors/multiplexers (3-state output)  
**74260:** dual 5-input NOR gate  
**74266:** quad 2-input XNOR gate with open collector outputs  
**74273:** octal D-type flip-flop with clear  
**74280:** 9 bit Odd-Even Parity Generator-Checker  
**74283:** 4-bit binary full adder, alternative pinning  
**74299:** 8-Input Universal Shift/Storage Register with Common Parallel I/O Pins  
**74373:** octal transparent latches  
**74374:** octal positive-edge-triggered flip-flops  
**74377:** Octal D Flip-Flop with enable  
**74381:** 4-Bit Arithmetic Logic Unit with high-speed expansion  
**74382:** 4-Bit Arithmetic Logic Unit with ripple carry output  
**74540:** octal buffer/line driver, inverted  
**74541:** octal buffer/line driver  
**74573:** octal transparent latches, different pinout compared to 74373  
**74574:** octal positive-edge-triggered flip-flops, different pinout compared to 74374  
**74590:** 8-bit binary counter with tri-state output registers  
**74595:** 8-Bit Shift Registers with 3-State Output Registers  
**74670:** 3-state 4-by-4 Register File  
**74682:** 8-bit digital comparator  
**74688:** 8-bit identity comparator  
**74779:** 8-Bit Bidirectional Binary Counter with 3-STATE Outputs  
**74804:** hex 2-input NAND gate <https://www.ti.com/lit/ds/symlink/sn74as804b.pdf>  
**74805:** hex 2-input NOR gate <http://www.ti.com/lit/ds/symlink/sn54as805b.pdf>  
**74808:** hex 2-input AND gate <http://www.ti.com/lit/ds/symlink/sn54as808b.pdf>  
**74832:** hex 2-input OR gate <http://www.ti.com/lit/ds/symlink/sn54as832b.pdf>  
**744002:** dual 4-input NOR gate  
**744017:** Johnson decade counter with 10 decoded outputs  
**744075:** triple 3-input OR gate  
**747266:** quad 2-input XNOR gate  
**7440105:** 4-Bit x 16-Word FIFO Register  
**A623308A:** 8K X 8 BIT CMOS SRAM  
**RAM32Bit:** A 32-bit memory that allows byte access and can handle non-aligned memory addresses.