

DE BEAUX CHRONOGRAMMES AVEC WAVEDROM

FABIEN MARTEAU

[Responsable FPGA chez Armadeus Systems, membre actif du Front de libération des FPGA]

MOTS-CLÉS : CHRONOGRAMME, WAVEDROM, JAVASCRIPT, JSON



WaveDrom est un outil de dessin de chronogrammes écrit en JavaScript/HTML/CSS. Il permet de décrire ses chronogrammes avec une syntaxe simple en JSON. Une bibliothèque JavaScript va ensuite convertir la structure JSON en une image SVG affichable dans un navigateur. Il est également possible d'utiliser une application « hors-ligne » pour générer ses images en SVG, PNG, JPG, etc.

Le chronogramme est l'outil de prédilection de l'électronicien pour documenter le fonctionnement de son système numérique ou analogique. Un simple chronogramme vaut souvent mille discours et descriptions.

Hélas, le dessin de chronogrammes se révèle vite être une activité dévoreuse de temps, pour un rendu souvent discutable. Habituellement pour dessiner des chronogrammes on utilise **LibreOffice Draw**, **DIA** ou **Xfig** qui peuvent permettre de faire des rendus relativement corrects au prix d'un long effort de dessin. Certains se servent même d'un tableur en utilisant astucieusement l'option d'encadrement des cellules pour faire les créneaux de leurs signaux, d'autres abandonnent l'ordinateur pour le papier la règle et le stylo. Pour versionner, bonjour les dégâts...

Les formats de fichiers de tous ces logiciels étant du genre « binaire », le gestionnaire de version ne fera que ré-enregistrer le fichier à chaque commit, et il sera impossible de visualiser correctement les modifications effectuées.

Partant de ce constat, Aliaksei Chapyzenka a cherché à développer un langage permettant de décrire ses chronogrammes au format texte qui soit convertible en une belle image. Pour convertir cette description

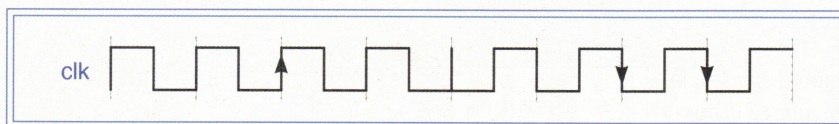


Fig. 4 : Troisième exemple.

Les signaux d'horloges changent d'état au cours d'une période de temps. C'est la raison pour laquelle il est prévu des symboles particuliers (voir figure 4) :

```
{ signal: [{ name: "clk", wave: "p.Ppn.N." } ] }
```

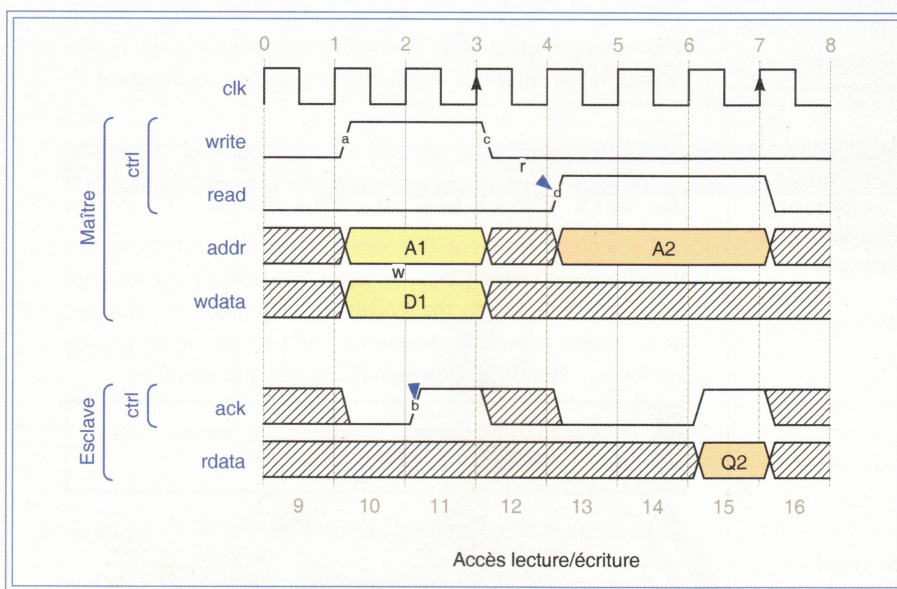


Fig. 5 : Exemple un peu plus complet de ce qu'il est possible de faire avec WaveDrom.

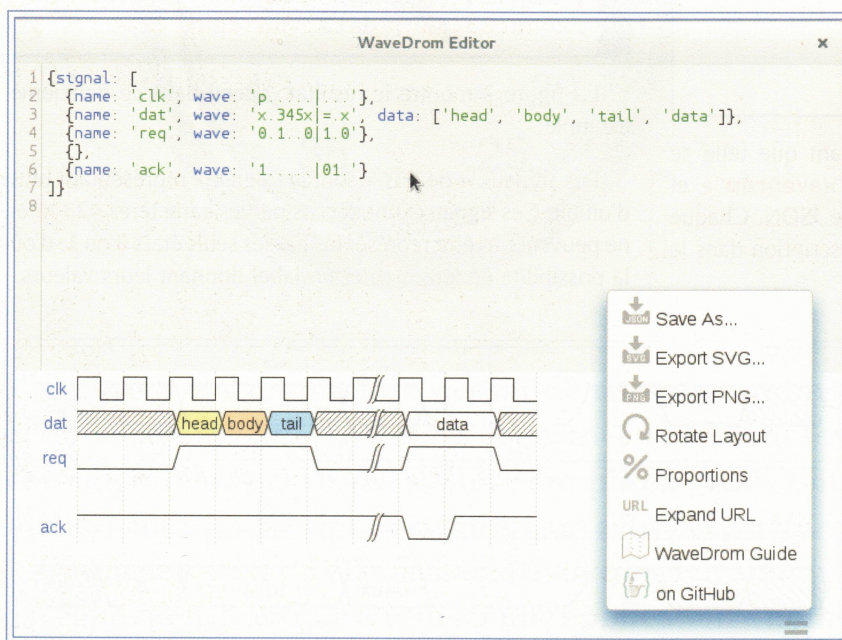


Fig. 6 : Exécution de l'application WaveDrom.

Le **p** représente une période d'horloge active sur front montant (positif), et le **n** sur front descendant (négatif). Il est possible d'ajouter une flèche pour indiquer le front en utilisant la majuscule et l'utilisation du **.** répètera la période d'horloge décrite au caractère précédent.

Nous n'avons ici qu'effleuré la syntaxe WaveDrom, il est aussi possible d'ajouter des titres, sous-titres et labels à des groupes de signaux. Il est également possible de tracer des flèches, d'intégrer des formules LaTeX et de rendre dynamique ou paramétrable le chronogramme grâce à l'insertion de JavaScript.

3. DIFFÉRENTES MANIÈRES D'ÉDITER

Pour le chaland voulant simplement tester rapidement WaveDrom, il suffit de se rendre sur le site officiel du projet, une zone d'édition/rendu s'y trouve tout simplement [2].

Maintenant, si l'on veut faire des rendus WaveDrom en mode déconnecté, une application est aussi disponible [3] sous la forme d'un paquet **tar.gz** ou **zip** qu'il suffit de décompresser et de lancer avec la commande (voir figure 6) :

```
$ ./wavedrom-editor
```

Une extension développée par votre serveur est également disponible pour **mediawiki**. Une fois installée, il suffit d'encadrer sa prose WaveDrom par des balises :

```
<wavedrom></wavedrom>
```

Et enfin, la base, quand on veut intégrer ses chronogrammes dans ses documents LaTeX et autres : une version en ligne de commandes nommée **wavedrom-cli** [4] est bien sûr disponible pour convertir un fichier **texte.js** en png/svg/jpg/...

Cette version en ligne de commandes demeure en JavaScript pur (**wavedrom-cli.js**), et il est donc nécessaire d'utiliser

un logiciel tiers pour l'exécuter (voir **phantomjs** [5]). Par exemple, pour convertir un fichier WaveDrom nommé **wb_ack.js** en image au format PNG on utilisera la commande suivante :

```
$ phantomjs /opt/wavedrom/wavedrom-cli.js  
-i wb_ack.js -p wb_ack.png
```

ATTENTION !

Pour la version ligne de commandes de WaveDrom, il est nécessaire d'exécuter du JavaScript en mode « console ». Le site officiel recommande phantomjs :

```
$ wget https://bitbucket.org/ariya/phantomjs/  
downloads/phantomjs-2.1.1-linux-x86_64.tar.bz2  
$ tar -jxvf phantomjs-2.1.1-linux-x86_64.tar.bz2  
$ vim ~/.bashrc  
#phantomjs  
export PATH="/opt/phantomjs-2.1.1-linux-x86_64/  
bin":$PATH
```

CONCLUSION

WaveDrom permet de réaliser de beaux chronogrammes relativement facilement. Tous les amateurs de Latex/Markdown/... approuveront la description au format texte. Wavedrom est peu dépendant de logiciels installés sur sa machine dans la mesure où l'on utilise la version web, et il s'insère très bien dans les documents web ou les présentations du type **impress.js**. Et enfin, WaveDrom permet d'avoir une excellente finition de ses illustrations dans sa documentation. Ce principe de description graphique via une structure JSON a également été étendu pour réaliser des schémas de logique combinatoire [6]. ■

RÉFÉRENCES

- [1] Le tutoriel officiel de wavedrom :
<http://wavedrom.com/tutorial.html>
- [2] Un éditeur en ligne permettant de tester wavedrom avec un simple navigateur et une connexion internet : <http://wavedrom.com/editor.html>
- [3] Version « application » de wavedrom, à installer sur son ordinateur : <https://github.com/wavedrom/wavedrom.github.io/releases>
- [4] Version ligne de commandes, pour intégrer dans un Makefile par exemple :
<https://github.com/wavedrom/cli>
- [5] Pour la ligne de commandes, il est nécessaire d'utiliser un logiciel pour exécuter le JavaScript :
<http://phantomjs.org/>
- [6] Comment utiliser WaveDrom pour faire des schémas : <http://wavedrom.com/tutorial2.html>