

TP4 : Débogage du circuit Am2901

1. Objectifs
2. Le débogage
3. Circuit Am2901
4. Compte-Rendu

Objectifs

Le but de cette séance est de vous apprendre à déboguer le modèle comportemental d'un circuit qui sera réutilisé dans les TP suivants. L'intérêt est donc double:

- Vous apprendre, par la pratique, les méthodes de débogage.
- Vous inciter à analyser en détail les fonctionnalités du circuit Am2901.

Le débogage

Le débogage comprend 5 étapes:

- Reconnaître l'existence d'un bogue.
 - ◆ Cette étape peut être effectuée de manière passive: on attend qu'un bogue arrive ou active: on utilise d'importants jeux de test pour valider l'application. Les développeurs couplent en général les deux méthodes.
- Isoler la source d'un bogue.
 - ◆ C'est l'étape la plus difficile. La source du dysfonctionnement n'est pas forcément la même que la source des symptômes. La méthode la plus courante consiste à tester de manière itérative: on vérifie les fonctionnalités les unes après les autres.
- Identifier la cause d'un bogue.
 - ◆ Une fois le problème localisé il est important d'en déterminer la cause. En effet on peut constater qu'une valeur est fautive mais l'origine de cette valeur peut provenir d'une autre portion de l'application à tester.
- Déterminer le correctif pour le bogue.
 - ◆ Pour l'écriture du correctif, la difficulté dépend fortement du type de bogue. Cette étape est simple dans le cas d'une erreur d'inattention, elle peut s'avérer très complexe quand le bogue est lié à une mauvaise architecture de l'application.
- Corriger et tester l'application.
 - ◆ Il faut ici vérifier que le bogue est bien supprimé par le correctif, et vérifier que ce même correctif n'apporte pas de nouveaux bogues. Il est pour cela préférable d'avoir des batteries de tests anti régression.

Circuit Am2901

Vous devez, au cours de cette séance, analyser et corriger les deux modèles comportementaux qui vous sont fournis. (deux fichiers .vbe par binôme). Dans chaque modèle, un bug a été délibérément introduit, et le travail consiste donc à localiser, puis à corriger ce bug. Pour vous aider dans ce travail, vous trouverez la DATA SHEET "officielle" du circuit Am2901 en bas de la page.

La validation devra être réalisée par simulation sous Asimut, à l'aide de stimuli générés avec Genpat. Pour obtenir la description d'un AMD Am2901, tapez la commande suivante :

```
/users/enseig/trncomun/TP/2006/TP1/Fichiers/Fournis/get2901.Linux > amd.vbe
```

Remarque : Le modèle comportemental du circuit Am2901 que vous allez récupérer présente quelques différences par rapport à la spécification définie par la DATA SHEET, mais ces différences ne constituent pas des bugs :

- La RAM de 16 mots de 4 bits est modélisée par un banc de registres à triple accès (un accès en écriture, deux accès en lecture). Les écritures se font sur front montant du signal d'horloge ck.
- Les drapeaux np et ng sortent la valeur '1' pour les opérations logiques.

Compte-Rendu

Il vous est demandé un rapport au format .pdf. Vous joindrez tous vos tests en annexe. Votre méthodologie sera évaluée en priorité sur les résultats. Par exemple, un groupe ayant identifié le bogue d'un seul des deux AMD Am2901 mais de manière intelligible recevra bien plus de points qu'un groupe ayant trouvé les bogues au hasard sans la moindre méthode.