

## ElecAna3

### M2 SESI

## TP PLL, partie 1 : Étude d'une PLL tout-numérique. Modèle Matlab

Une PLL est modélisé par le modèle Simulink *vlsi\_single\_pll*. Pour initialiser le modèle (définir les valeurs des paramètres), il faut exécuter le script Matlab *pll\_Nbit\_vlsi\_init.m*

**TDC.** Le TDC est réalisé par une chaîne des délais (fig. 1). Les paramètres du TDC sont :

- Le délai d'un délai élémentaire de la ligne de délai,  $\tau = 30 \text{ pS}$
- Un vecteur donnant le nombre de délai élémentaire dans chaque segment de la ligne (par défaut, 1 délai par segment), Vecteur  $TDC(1:16)=1$

Le TDC génère un signal entier allant de 0 à 14.

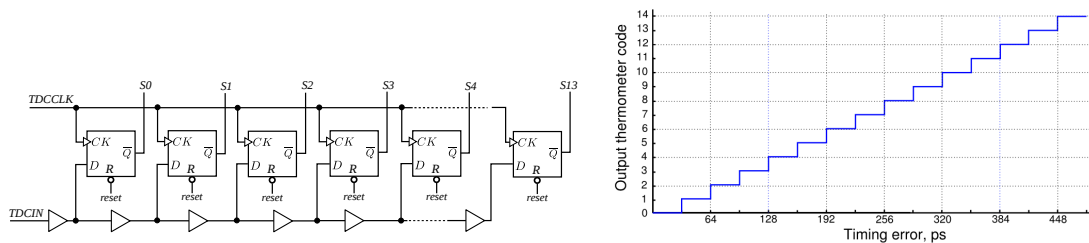
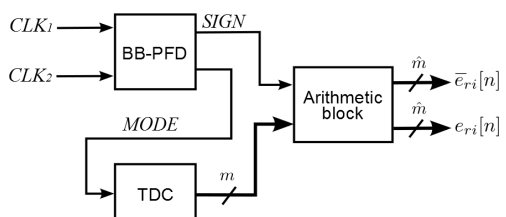


Figure 1. Architecture et la caractéristique du TDC

**Comparateur de phase.** Le comparateur de phase est composé d'un comparateur de phase Bang-Bang détectant le signe de l'erreur de phase et du TDC mesurant la valeur absolue de l'erreur de phase. Les deux valeurs sont combinées de sorte à ce que la sortie du bloc change de la manière suivante :

-15, -14, ..., -2, -1, 1, 2, ..., 15.

de sorte à ce que à aucun moment la sortie n'est égale à zéro : pour les petites erreurs de phase, on connaît toujours au moins le signe de l'erreur.



**Filtre.** Le filtre est réalisé par un circuit numérique dont l'architecture est donnée fig. 2. Cette architecture prend en compte la quantification des signaux numériques : la largeur des bus véhiculant les signaux.

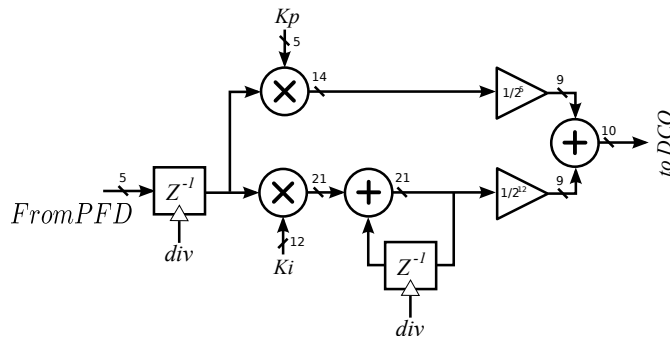


Figure 2. Architecture du filtre.

Les paramètres du filtre sont :

- Les coefficients  $K_{prop\_num}$ ,  $K_{prop\_den}$ ,  $K_{int\_num}$ ,  $K_{int\_den}$ . Ainsi, les coefficients réels sont les rapports  $K_{prop\_num}/K_{prop\_den}$  et  $K_{int\_num}/K_{int\_den}$ .
- Nombre de bit à l'entrée du filtre, à l'entrée et en sortie de la branche proportionnelle

**DCO.** Le DCO est un oscillateur commandée par code, dont la fréquence est donnée par :

$$f = f0\_dco + Y \times gain\_dco, \text{ où}$$

$f0\_dco$  est la fréquence de DCO à code d'entrée nul,  $gain\_dco$  est le gain de DCO (le pas de fréquence),  $Y$  est le code d'entrée. Le DCO est de 10 bits, c.a.d.,  $Y$  peut prendre des valeurs de 0 à  $2^{10}-1$ .

Le script est configuré pour que la fréquence initiale du DCO définie par la valeur initiale de l'intégrateur soit différente de la fréquence du signal d'entrée.

### Exercice.

1) Lancez la simulation avec les valeurs des paramètres définis par dans le script *pll\_Nbit\_vlsi\_init.m* ( $K_{prop\_num}=20$ ,  $K_{int\_num}=400$ ). Observez les sorties : la sortie du comparateur de phase donnant l'erreur de phase entre le signal d'entrée et le signal de DCO, et l'évolution de la fréquence du DCO.

2) Pour comprendre le rôle des branches proportionnelles et intégrales, annulez d'abord  $K_{prop\_num}$ , ensuite  $K_{int\_num}$  (en remettant  $K_{prop\_num}=20$ ), et simulez les 2 configurations.

3) Remettez les valeurs à  $K_{prop\_num}=20$ ,  $K_{int\_num}=400$  ; simulez de nouveau. multipliez  $K_{int\_num}$  par 2. Comment change le comportement de la PLL ?

4) Remettez les valeurs à  $K_{prop\_num}=20$ ,  $K_{int\_num}=400$  ; simulez de nouveau. multipliez  $K_{prop\_num}$  par 2. Comment change le comportement de la PLL ?

5) Pour la dernière configuration, quelles sont les bornes maximales et minimales de l'erreur de phase en secondes en régime synchronisé ? Faites une mesure sur quelques périodes.

6) **Question TD (calcul sur papier).** Estimez le temps de synchronisation en fréquence, connaissant les paramètres de la PLL.

7) **Question TD.** Écrivez l'équation donnant la fonction de transfert de la PLL dans le domaine  $z$ .

## TP PLL, partie 2 : Étude d'un comparateur de phase à 1 bit (bang-bang) au niveau transistors.

Dans cette deuxième partie nous proposons d'étudier le fonctionnement d'un comparateur de phase à 1 bit appelé « comparateur de phase Bang-Bang ».

Son fonctionnement, expliqué en cours et en thèse de doctorat de E. Zianbetov, est basé sur la machine d'état donné ci-dessous.

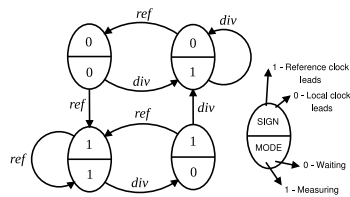


Figure 3

Pour minimiser l'impact des métastabilités, cette machine d'état est réalisée à l'aide d'un circuit un peu plus sophistiquée décrit dans l'article

Tierno, J. A., Rylyakov, A. V., & Friedman, D. J. (2008). A wide power supply range, wide tuning range, all static CMOS all digital PLL in 65 nm SOI. Solid-State Circuits, IEEE Journal of, 43(1), 42-51.

Pour étudier ce circuit, nous vous proposons son modèle décrit au niveau de transistor sous Cadence.

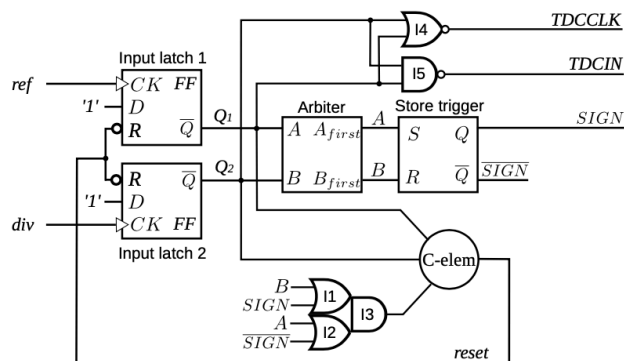


Figure 4

Ce circuit est modélisé dans la technologie 45 nm CMOS décrite dans le Process Design Kit GPDK fourni avec Cadence. La tension d'alimentation (et le niveau logique haut) est 1 V. Il s'agit d'un PDK fictif, créé pour les objectifs d'enseignement, mais ayant toutes les propriétés et caractéristiques d'un vrai PDK industriel. (Son avantage est ne pas être soumis à la NDA, et dont nous pouvons diffuser librement les développements et les bibliothèques qui l'utilisent.).

Pour tester le circuit, nous vous proposons la séquence de test suivant :

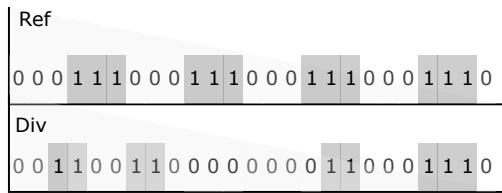


Figure 5

Ces séquences sont définies au niveau de deux sources appliquées aux entrées ref et div. Chaque séquence est caractérisée par un mot binaire, chaque bit ayant une durée égale (5 nS dans cet exemple).

Ainsi, on a la séquence de sortie « idéale » qui doit être :

« Div premier » à partir du bit 3

« Div premier » à partir du bit 7 »

« Ref premier » à partir du bit 16

Au bit 22, « Ref » et « Div » affichent un front montant simultané. La valeur n'est donc pas définie. Cette situation donnera lieu à la métastabilité dont la résolution l'on étudiera dans les détails.

### Manipulations à effectuer.

Pour préparer : copiez le répertoire

- Sous Cadence, ouvrir la cellule BBPFD, vue Schematic fourni par l'enseignant. Observer le schéma et comparer avec le schéma de la Figure 4.
  - Lancer l'outil ADE L et charger l'état (state) « test\_BBPFD » fourni par l'enseignant.
  - Lancez la simulation.
  - Observez l'évolution de la valeur des signaux affichés « ref », « div » et « sign ».
- 1) Interprétez la valeur du bit « sign » : quels événements sont codés par la valeur « 1 » (« div premier » ou « ref premier ») ?
  - 2) Visualisez les entrées et les sorties de la cellule de Muller (C-element) et le signal *reset*.
  - 3) Observez l'événement correspondant à l'arrivée simultanée du front montant aux deux entrées (bit 22).
    - a. Afficher tous les signaux : Q1, Q2, A0, B0, A, B, sign, not\_sign, reset. On constate que la métastabilité n'arrive pas : le système s'établit à l'état correspondant à « sign=0 ». Ceci est dû à des délais infinitésimaux créant un déséquilibre.
    - b. En « jouant » avec la valeur de la variable *dt*, (à gauche de la fenêtre de ADE), obtenez une situation où les signaux A0 et B0 passent par un état métastable et où l'arbitre joue le rôle de filtre de métastabilité. Pour quelle valeur de *dt* cela arrive (il faut chercher dans des picosecondes) ?
  - 4) Dans le cas où il n'y a pas de métastabilité, quel est le délai de ce comparateur de phase (le temps qui s'écoule entre l'arrivée d'un des événements qui définissent le signe, ref ou div, et l'établissement de la valeur du signe).
  - 5) Montrez que cette porte est capable de déterminer le signe de l'erreur de fréquence. Pour cela, appliquer aux entrées ref et div des sources rectangulaires de fréquences de 40 MHz et 50 MHz (testez ref=40 MHz, div =50 MHz et vice versa), et observez la valeur de la variable signe. Vous utiliserez les sources de tension de type vpulse, avec la durée des fronts de 100 ps.