

# Etude et conception d'un filtre numérique RIF à réponse impulsionnelle programmable

## Synthèse logique, analyse des résultats et validation

### Utilisation du Design Compiler (Synopsys) à travers l'interface Design Vision

#### 1. Configuration de l'outil de synthèse

La configuration de l'outil Design Vision en vue de la synthèse est réalisée grâce au fichier «`config_ASIC`» du répertoire «`config`» qui contient les chemins vers les exécutables et le serveur de licences. Ce répertoire contient aussi le fichier «`.synopsys_dc.setup`» du répertoire «`synth`» qui contient un ensemble de commandes et qui est lu par Design Vision lors de son exécution. Il est utilisé et lors de l'exécution de chaque commande liée à la lecture d'une description, compilation et optimisation. Les liens entre bibliothèques logiques et physiques sont spécifiés dans ce fichier.

#### 2. Lancer l'outil "Design Vision"

Pour lancer l'outil de synthèse logique Design Vision, il suffit d'entrer la commande suivante dans un terminal (sans le « & ») dans le répertoire «`synth`» :

```
design_vision
```

#### 3. Utilisation des menus de l'interface Design Vision

La suite du document présente **quelques** menus accessibles à travers cette interface graphique.

##### Menu "File" :

- **Analyze** - lire la source HDL, vérifier la syntaxe HDL et créer les formats objet de bibliothèque HDL. Utiliser la bibliothèque `CORE`. La liste des fichiers doit être ordonnée en fonction de la hiérarchie du circuits (niveaux inférieurs ou "feuilles" avant les niveaux plus élevés).
- **Elaborate** - créer un design en cellules génériques indépendantes de la technologie à partir d'un format HDL intermédiaire (objets générés par la commande `Analyze`). Utiliser la bibliothèque `CORE` et le design `FILTER(A)`.
- **Save** - sauvegarde en format interne sous forme de base de données (défaut `.ddc`).
- **Save As** - sauvegarde en format quelconque (choix multiple: VHDL, Verilog, Edif, ...)

Une fois le design compilé et élaboré, deux icônes deviennent visibles dans la fenêtre de travail. Elles ont la forme d'une porte logique (correspondant à la structure interne du circuit) respectivement d'un circuit à plusieurs entrées/sorties (correspondant à la vue externe). Dans la fenêtre `Hier`, sélectionner un composant et cliquer ensuite sur ces icônes afin de parcourir la

hiérarchie du composants sélectionné en observant les portes logiques, les pins d'entrée/sortie, les interconnexions.

A noter : `Read` permet de lire le fichier source dans un format VHDL, Verilog, ... L'option `Read` est équivalente aux commandes `Analyze` suivi d'`Elaborate` mais elle sera utilisée seulement pour les fichiers de bases de données `.ddc`.

Une fois l'élaboration faite plusieurs autres menus deviennent visibles.

### Menu « Attributes » :

Ce menu est utilisé pour imposer des valeurs/variables aux objets sélectionnés. **Placez vous sur le circuit "top" (FILTER) pour appliquer une contrainte à l'ensemble du circuit.**

- **Specify Clock** - préciser, pour le signal choisi comme signal d'horloge, la période (en ns) et les instants des fronts (montant et descendant) à l'intérieur de la période. Avant d'utiliser ce menu, sélectionner le port sur l'icône ou le schéma du composant `FILTRE` : le nom du port doit apparaître en grisé dans la fenêtre "`Specify clock`" qui s'ouvre. Eventuellement, vous pouvez également utiliser cette commande équivalente :

```
create_clock -name "clock" -period 10 -waveform {0.000 5.000} {CLK}
```

### - Operating Environments

- **Input Delays** - préciser les Rising/Falling Edges d'un signal d'entrée choisi par rapport au signal d'horloge,
- **Operating Conditions** - possibilité éventuelle de préciser les conditions de fonctionnement (Température, tension, et variation du processus de fabrication : PVT). Si aucun choix n'est fait, l'outil prend en compte les conditions TYPICAL de fonctionnement. Un mauvais PVT peut engendrer des slacks négatifs.
- **Wire load** - possibilité éventuelle de préciser un modèle RC pour les fils d'interconnexions. Si aucun choix n'est fait, l'outil fait une estimation de ces paramètres selon la surface du circuit.

- **Optimisation Constraints** - sert à préciser des contraintes de fonctionnement autres que les «Operating Conditions», «Wire load», «Clock». Par exemple le sous-menu :

- **Design Constraints** - pour la surface (mettre toujours 0), pour la consommation dynamique ou statique, ou le fanout. On se contentera de n'imposer que des contraintes de surface.
- **Timing Constraints** - pour les temps de propagation entre n'importe quelle entrée et sortie.
- **Optimisation directives** - possibilité éventuelle d'imposer des contraintes globales sur tout le circuit ou sur une cellule particulière sélectionnée, comme par exemple mettre à plat la description, rajouter des pads... .

### Menu « Design » :

Il permet d'effectuer la synthèse, des optimisations, des analyses du design et de produire des rapports, en lançant d'autres outils Synopsys (estimation de timing, de puissance, etc...) :

- **Compile Design** - vérifie la cohérence du design, réalise un mappage technologique sur la bibliothèque cible, avec éventuellement les options de compilations suivantes (les plus importantes) :

- **Top level** - optimisation du plus haut niveau hiérarchique sans toucher aux blocks internes.
- **Ungroup all** - mise à plat du design.
- **Scan** - identification et remplacement des bascules scan (sera utilisé dans une étape ultérieure du projet).
- **Incremental mapping** - permet de repartir de la version courante en cas d'optimisations successives.

**Pour la première synthèse ne cocher aucune option.**

- **Check Design** - vérifie les erreurs de structure et les violations de timing

- **Reports Design** - générer des rapports sur toutes les valeurs imposées au design
- **Report Constraints** - choix multiple, vérifier les contraintes
- **Report Ports, Cells, Nets, Clocks, Area** - obtenir des informations sur le résultat de la synthèse.
- **Report Power** - obtenir des résultats sur la puissance consommée (types et valeurs).  
Attention : considérer « NETS and CELLS ». Faire "show nets histogram".  
Cocher aussi « traverse hierarchy at all levels »

### Menu "Timing":

- **Check Timing** - vérifie seulement le timing du circuit, rapport en cas de violation.
- **Report Timing Paths** - rapport des chemins de propagation du plus long au plus court, choix multiples, laisser toutes les options par défaut
- **Path Slack** - créer des histogrammes des slacks (marge d'un chemin de propagation par rapport au chemin critique). Cliquer sur une histogramme, dans la fenêtre à droite on voit les chemins et les valeurs des slacks.
- **Net capacitance** - créer des histogrammes des capacités de nœuds. Cliquer sur une histogramme, dans la fenêtre à droite on voit les valeurs des capacités.
- **Path Profile View** - visualiser le chemin critique avec le temps de propagation des portes se trouvant sur le chemin. Cliquer sur Flat dans la fenêtre de visualisation.

Pour voir effectivement le chemin critique sur le design, sélectionner le chemin dans l'histogramme, aller dans le menu View, et Highlight Selected. Alternativement on peut aussi utiliser le menu contextuel de la touche droite de la souris.

On peut aussi visualiser le chemin critique en le bon path le dans le menu Select -> Path[...]

## Validation après synthèse

Le résultat de la synthèse doit être sauvé sous la forme d'une netlist (VHDL ou Verilog), mais VHDL est conseillé à cette étape du projet), et d'un fichier SDF (Standard Delay Format). Le VHDL peut être sauvegardé directement à partir de l'interface graphique, alors que le SDF doit être généré à la ligne de commande avec la commande write\_sdf.

La fichier VHDL doit ensuite être compilé dans la bibliothèque lib\_synth:

```
vlib ~/Project/libs/lib_synth
vcom -work lib_synth filter_synth.vhdl
```

A partir de là, le testbench écrit dans la phase précédente du projet peut être ré-utilisé, en modifiant simplement la bibliothèque déclarée en haut du fichier `bench` pour utiliser l'entité `FILTER` ou `TOP` compilée dans la bibliothèque "`lib_synth`" à partir de la description de la netlist.

Sous Modelsim, il faudra lancer la simulation du menu `start simulation` pour pouvoir paramétrer correctement l'outil. Pour que les cellules de base puissent être simulées, il est nécessaire de rajouter la bibliothèque de cellules (`c35_CORELIB`) dans l'onglet `libraries`. L'onglet `SDF` est aussi utilisé pour ajouter les résultats de l'analyse de timing (static timing analysis) réalisée par l'outil de synthèse. Pour chaque fichier, il faut aussi spécifier le domaine (la « `region` ») d'application, qui doit correspondre au composant synthétisé (i.e. le filtre). Pour le testbench fourni, ce sera donc `/bench_filter/DUT`. L'analyse des fichiers SDF peut générer énormément de messages : il est donc conseillé de cocher les cases « `disable SDF warnings` » et « `reduce SDF errors to warnings` ».

## Ce qu'il faut faire à cette étape

### Analyse des résultats de synthèse

Voici quelques suggestions sur les analyses pouvant être réalisées. Toute autre analyse est possible ... et laissée à votre sagacité !

1 - Faites une première synthèse (hiérarchique), qui inclut l'ensemble du projet (filtre+uC). Analyser le chemin critique. Quelles remarques pouvez-vous faire ?

2 - Faites ensuite une synthèse du filtre seul. Le nombre de bascules correspond-t-il à ce que vous attendiez ? Quel codage est utilisé pour la machine à états ? Quelle logique est générée pour le multiplieur, l'accumulateur, ... ?

Attention au bloc `MULT`, vérifier qu'il a bien été synthétisé et mappé sur la bibliothèque `CORELIB` et qu'il n'a pas été maintenu sur une bibliothèque générique, auquel cas la synthèse n'est pas complète.

3 - Faites différentes synthèses, en indiquant des contraintes diverses, notamment pour la fréquence d'horloge. Analysez les conséquences sur la surface, sur la répartition des "slacks", sur le chemin critique et sur la consommation (statique et dynamique). Quel bloc consomme le plus ? Quel bloc définit le chemin critique ? Tester l'effet des conditions d'opérations (modèles de timing utilisés pour le STA) sur le chemin critique, la taille du circuit...

3 - Sauvegarder en format VHDL et sdf pour effectuer les simulations/validations après synthèse.

34- Réalisez une synthèse finale avec des contraintes de surface 0, conditions d'opérations `TYPICAL` et un signal d'horloge de 10ns de période. Vérifiez la surface du système et le temps de propagation. Regardez les slacks et la puissance totale dissipée. Quel est le bloc limitant ? Quelles techniques devraient être mises en œuvre pour améliorer la rapidité du système ? Mettre en œuvre ces techniques.