



# Méthodologie de conception de circuits numériques (MET)

Cours Systèmes numérique

**Hes-so** VALAIS  
WALLIS

 Haute Ecole d'Ingénierie  
Hochschule für Ingenieurwissenschaften

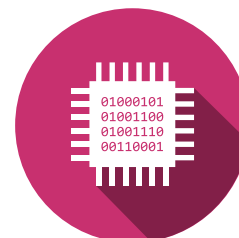
Orientation : Informatique et systèmes de communication (ISC)

Cours : Systèmes numérique (DiD)

Auteur : Christophe Bianchi, François Corthay, Pierre Pompili, Silvan Zahno

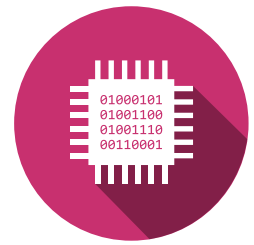
Date : 25 août 2022

Version : v2.1



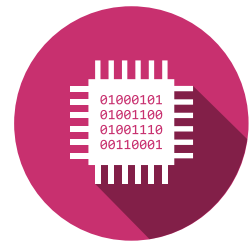
## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Modèle de développement : Diagramme en V</b>	<b>3</b>
2.1	Principe du diagramme en V	3
2.2	Diagramme en V appliqué au développement de circuits numériques	3
2.3	Documentation et prévision des activités	4
<b>3</b>	<b>Phase de spécification</b>	<b>5</b>
3.1	Cahier des charges	5
3.2	Document de spécification	5
3.3	Moyens mis en oeuvre	5
<b>4</b>	<b>Phase de conception</b>	<b>7</b>
4.1	Décomposition fonctionnelle	7
4.2	Document d'architecture	7
4.3	Design détaillé	7
4.3.1	Outil	7
4.3.2	Méthodologie	8
4.3.3	Règles générales de design	8
4.4	Systèmes synchrones ou asynchrones	9
4.4.1	Délais de portes	9
4.4.2	Recommandation : conception insensible aux délais de portes logiques	9
4.4.3	Règles relatives à la logique séquentielle et à la notion de délais de porte	9
4.4.3.1	Règle N°1	9
4.4.3.2	Règle N°2	10
4.4.3.3	Règle N°3	10
4.4.3.4	Règle N°4	11
4.4.3.5	Règle N°5	11
4.4.3.6	Règle N°6	11
4.4.3.7	Règle N°7	12
4.4.3.8	Règle N°8	12
4.4.4	Règles liées au comportement électrique des éléments logiques	12
4.4.4.1	Règle N°9	12
4.4.5	Conclusion	13
<b>5</b>	<b>Phase de vérification et validation</b>	<b>14</b>
5.1	La validation	14
5.2	La vérification	14
5.3	Technique de validation et vérification	14
<b>6</b>	<b>Phase d'intégration</b>	<b>17</b>
6.1	Résultats	17
6.2	Moyens mis en oeuvre	17
	<b>Acronymes</b>	<b>19</b>



# 1 Introduction

Ce document donne les lignes directrices de la procédure de développement de circuits numériques partant de sa définition donnée sous la forme d'un cahier des charges pour aboutir à son intégration et à sa validation dans un système complet.



## 2 Modèle de développement : Diagramme en V

### 2.1 Principe du diagramme en V

Parmi les nombreux modèles de développement existant, le diagramme en V est sans nul doute le plus employé. Ces caractéristiques principales sont les suivantes :

- Chaque étape est caractérisée par des activités dont le but est d'élaborer un produit intermédiaire.
- Chaque fin d'étape est matérialisée par un événement où s'exerce une activité de contrôle destinée à détecter et à éliminer le plus tôt possible les anomalies et imprécisions. Le passage à l'étape suivante est conditionné par le résultat du contrôle.
- Autant que possible, les retours en arrière se limitent à l'étape immédiatement antérieure ; il est recommandé qu'un produit approuvé ne puisse être modifié que par une procédure formelle de modification.

### 2.2 Diagramme en V appliqué au développement de circuits numériques

La Figure 1 montre un modèle de diagramme en V appliqué au processus de développement de circuits numériques. Ce diagramme met en évidence les phases de développement suivantes :

- La phase de spécification qui regroupe toutes les activités de définitions et d'analyses préliminaires (faisabilité)
- La phase de conception qui regroupe les activités d'architecture (découpage en bloc) et de design
- La phase de validation qui se compose des tests unitaires, des tests fonctionnels et des test de performances.
- La phase d'intégration qui aboutit à la validation du produit dans son environnement d'utilisation.

Ce regroupement en phase qui permet de limiter le nombre des réunions et les aller et retour de documents est très largement pratiqué dans les entreprises.

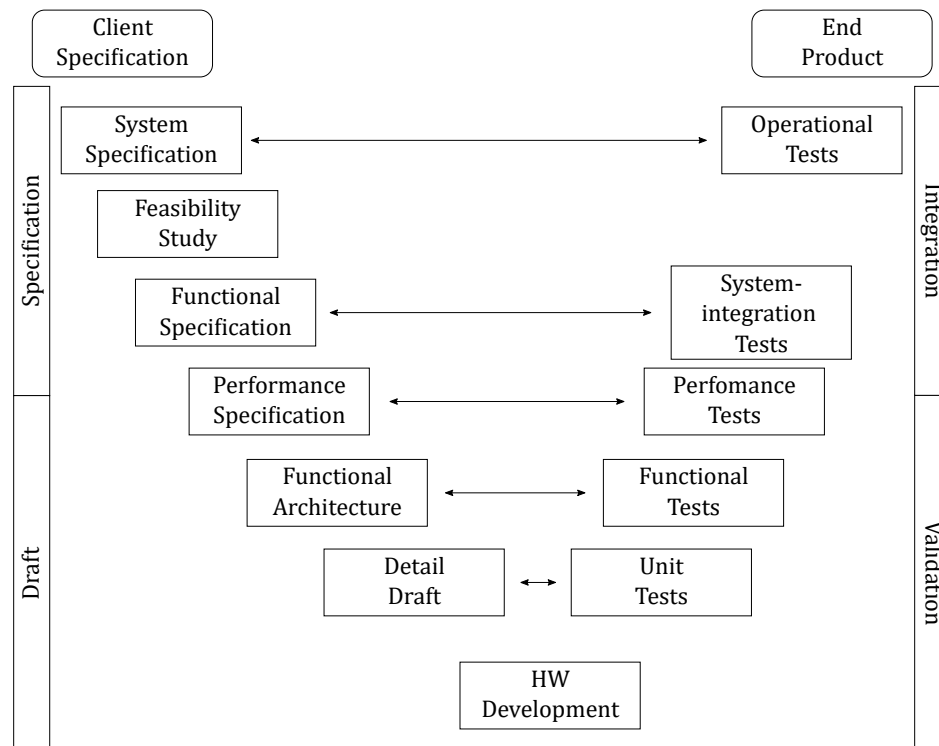
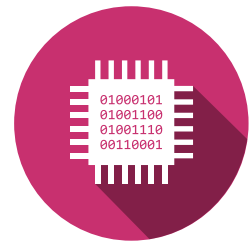


FIGURE 1 – Diagramme de développement en V

### 2.3 Documentation et prévision des activités

La réussite de cette technique est toutefois liée aux éléments suivants :

- la réalisation d'une documentation par anticipation : pour tout produit, la rédaction de la documentation doit précéder sa réalisation.
- Une bonne prévision des activités futures en ce qui concerne les études prévues et la validation

Un élément fondamental du schéma en V est constitué par les flèches horizontales qui traduisent un mode de fonctionnement standard qui se retrouve à tous les étages du développement : la phase qui se trouve dans la branche de gauche doit élaborer les procédures de test et validation qui seront exécutées dans la phase située au même niveau dans la branche de droite. Cette procédure évite de tomber dans le piège classique qui consiste à valider le résultat par rapport à ce qui a été fait et non pas par rapport à ce qui aurait dû être fait. Son application permet notamment :

- Une diminution du coût total de développement par une meilleure planification (moins de pertes de temps)
- Un déplacement du temps consommé vers les activités de spécification et conception (pré-test) qui arrivent à représenter jusqu'à 50% du temps total de développement.



## 3 Phase de spécification

La phase de démarrage d'un projet est une phase délicate car elle implique un transfert de savoir entre le client et l'équipe projet chargée de la réalisation du produit. Les deux parties (quand il n'y en a pas plus) ont peu de chances de se connaître et de se comprendre d'emblée ; elles n'ont généralement pas le même univers culturel, ni les mêmes attentes vis à vis du projet. Un dialogue organisé est donc indispensable pour obtenir une homogénéité de vues suffisante entre ces parties. La recherche de qualité et l'organisation du développement du projet sont impossibles si les attentes du client n'ont pas été établies de façon complète et non ambiguë.

Il est donc important, au début du projet de définir clairement les attentes fonctionnelles et non fonctionnelles du client. Cette étape est l'étape de spécification des besoins. Il s'agit avant tout d'organiser et de régler le dialogue entre clients et réalisateurs et d'aboutir à une expression complète et non ambiguë de ce que doit faire le circuit.

### 3.1 Cahier des charges

Il est important de bien distinguer le document de spécification, du cahier des charges. Le cahier des charges est un document rédigé par le client seul. Il identifie l'ensemble de ses besoins (le pourquoi faire) sans le plus souvent faire mention d'une quelconque solution électronique. Le document de spécification à l'inverse formalise les futurs services que devra pouvoir rendre le circuit. En cela il décrit d'un point de vue utilisateur, une solution matérielle aux besoins exprimés dans le cahier des charges.

Il est important de noter que la non-existence d'un cahier des charges en début de projet impose aux développeurs un travail supplémentaire lors de la phase de spécification : l'identification des besoins. Car un client évalue toujours le succès d'un projet non pas à la remise d'un circuit superbe et fiable mais à la prise en compte de ses besoins dans le matériel et à la valeur ajoutée qu'il va lui procurer.

### 3.2 Document de spécification

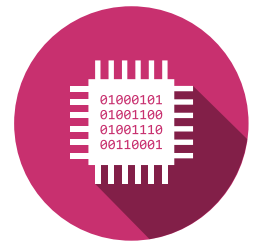
Le document de spécification est certainement le document le plus délicat et le plus long à établir. Il doit apporter une vision complète et non ambiguë de l'ensemble du circuit. Les différents points à aborder sont les suivants :

- Spécifications générales : nom du circuit, caractéristiques électriques (courant, charge, niveau, etc...), environnement opérationnel
- Spécifications fonctionnelles : entrées/sorties, interface avec d'autres blocs fonctionnels, description des fonctionnalités.
- Spécifications de performances : chronogrammes spécifiques, fréquence d'utilisation, consommation, taille

Il est à noter que ce document de spécification doit traduire une vision utilisateur du circuit et qu'il doit rester lisible pour le client qui en est l'utilisateur.

### 3.3 Moyens mis en oeuvre

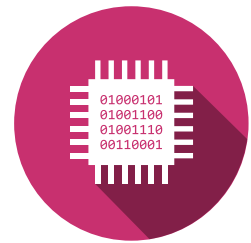
L'essentiel du travail de spécification est réalisé à travers des dialogues entre l'équipe de développement et l'ensemble des personnes concernées par le projet. Pour être efficaces, ces dia-



logues doivent être organisés, préparés par l'équipe de développement. Des techniques de réunions peuvent être utilisées avec profit pour optimiser le temps consacré par les clients à ces réunions.

Si une revue conjointe terminale est indispensable pour valider la spécification en fin de phase, il faut certainement en prévoir plusieurs pendant la durée de la phase de spécification afin d'éviter de partir sur une fausse route et de s'y perdre. Il n'est ainsi pas rare de faire deux ou trois documents préparatoires partiels avant de proposer le document définitif.

Tous les documents générés par cette phase doivent être validés par le client du projet, ce qui implique qu'ils soient rédigés dans un langage clair et compréhensible. Cela ne signifie pas qu'il est interdit d'utiliser des techniques de modélisation mais cette utilisation doit être expliquée et comprise par le client ; ainsi, le choix des techniques dépend étroitement de la culture du client et du type de problème à résoudre.



## 4 Phase de conception

Le présent chapitre donne quelques recommandations sur la marche à suivre pour concevoir un circuit numérique à partir de ses spécifications.

### 4.1 Décomposition fonctionnelle

Dans un premier temps il est utile pour l'ingénieur, partant des spécifications, de décomposer son circuit sous la forme d'un ensemble de blocs fonctionnels. Chaque bloc représentera dans la mesure du possible les différentes parties physiques du circuit dédiées à des fonctions propres (glue logique, sélection, compteur, séquençement, lecture de data, filtrage de data,...) ou à des interfaces bien spécifiées (entrées digitales, sorties digitales, **Random Access Memory (RAM)**, **First In First Out (FIFO)**, ...).

Chaque bloc peut à son tour être divisé en différents modules selon la complexité qu'il engendre. Ainsi un bloc dédié au séquençement de l'ASIC peut se diviser en plusieurs compteurs, machines d'état ou groupes de logiques combinatoires.

Enfin, pour des raisons de clarté, chaque module peut être divisé en sous-modules. Il est à noter que, partant d'une fonction logique globale, un nombre de 3 à 5 niveaux paraît approprié pour satisfaire à la fois aux besoins de simplification du circuit (ensemble de blocs fonctionnels plus simples) et à la clarté des schémas (nombre de portes logiques sur un schéma).

### 4.2 Document d'architecture

Chaque bloc, module ou sous-module doit être décrit avant le commencement du 'design' par ses entrées, ses sorties et son fonctionnement global sous une forme rédactionnelle ou visuelle. Cette décomposition en blocs fonctionnels sert de base à l'élaboration du document d'architecture du circuit numérique.

Le document d'architecture donne la découpe du circuit numérique en blocs fonctionnels. Son contenu est le suivant :

- Description succincte du circuit numérique et de son environnement
- Liste des entrées - sorties et attribution des pins
- Découpage du circuit en blocs fonctionnels
- Estimation de la taille du circuit (nombre de portes équivalentes)
- Estimation de la consommation
- Liste des fonctions critiques
- Stratégie de test, contrôlabilité et observabilité des blocs fonctionnels et attribution des **Input Output (I/O)** de test.

### 4.3 Design détaillé

#### 4.3.1 Outil

Installée sur un **Program Counter (PC)**, l'application HDL Designer de Mentor Graphics permet d'accéder à tous les outils nécessaires pour le dessin de schéma ou la description sous forme **Very High Speed Integrated Circuit Hardware Description Language (VHDL)**.





Dans le cas d'une saisie schématique, la librairie de composants utilisés est fournie en général par le fondeur du circuit.

### 4.3.2 Méthodologie

Tous les schémas doivent être placés dans une même librairie. Le nom donné à chaque schéma doit être dans la mesure du possible représentatif de la fonction réalisée ou représentatif du nom des blocs auxquels il se rattache (lettres d'abréviation).

Les fonctions sont réalisées à partir du niveau logique le plus bas. Chaque fonction logique est simulée de manière indépendante par l'intermédiaire de l'outil Modelsim.

Pour permettre l'assemblage des différents modules d'un même bloc, chaque module est édité sous la forme d'un symbole. Ces symboles sont par la suite assemblés dans un nouveau schéma qui à son tour sera édité sous la forme d'un symbole et placé dans un niveau logique supérieur, etc...

Un bloc peut également être décrit par un code **VHDL**. Si ce bloc doit être intégré dans un design schématique, il devra pour cela être édité sous la forme d'un symbole et placé dans ce niveau schématique supérieur. Le même outil de simulation sera utilisé.

La Figure 2 donne un exemple de décomposition arborescente.

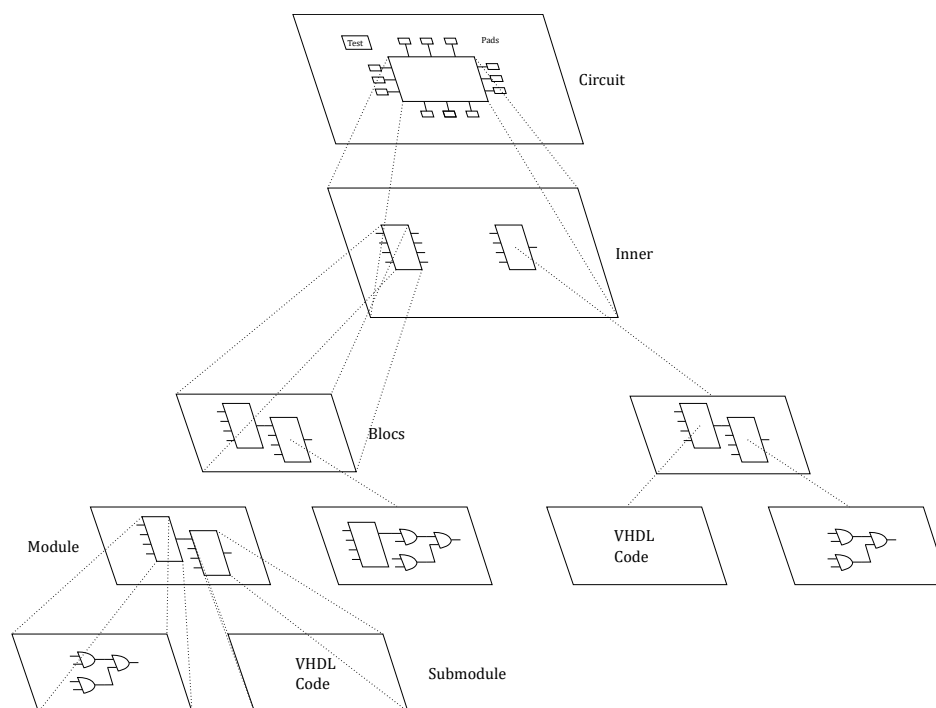
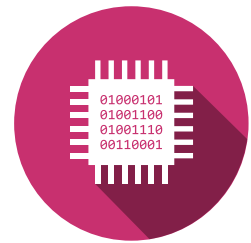


FIGURE 2 – Décomposition arborescente

### 4.3.3 Règles générales de design

Lors du design détaillé du circuit numérique les règles générales suivantes sont à prendre en considération.

- mise en forme : un schéma se lisant de gauche à droite, les entrées se situent en général sur la gauche et les sorties sur la droite. Pour des raisons de visibilité, il est quelquefois



préférable de laisser une entrée ou une sortie proche de la porte ou du bloc qui lui est attribué.

- nom des signaux
  - les noms des signaux sont écrits en minuscule et composés de lettres ou de chiffres uniquement. Seuls les bus utilisent les caractères spéciaux.
  - un signal actif à l'état bas peut être reconnaissable par le préfix N

## 4.4 Systèmes synchrones ou asynchrones

### 4.4.1 Délais de portes

Les délais de portes logiques sont à prendre en compte au cours de la conception.

Le comportement temporel d'une porte logique révèle un délai entre le moment où les entrées varient et celui où les sorties présentent l'état correspondant. Ce délai dépend de la technologie utilisée et, dans une même technologie, elle varie passablement d'une porte logique à l'autre. Le délai d'une porte logique dépend aussi de sa charge.

Du fait des délais de portes logiques, il est possible que la sortie d'un bloc logique se trouve pendant un court instant à un état qui ne correspond pas à la séquence des états d'entrée. Ce phénomène, appelé aléa, correspond à la situation où la sortie d'un bloc logique présente deux transitions presque simultanées entre deux états consécutifs de sortie identiques.

L'aléa est dû au phénomène de temps de propagation (ou temps de retard de porte) : la variation d'un signal d'entrée affecte la sortie par deux ou plusieurs chemins parallèles auxquels sont associés des retards différents.

### 4.4.2 Recommandation : conception insensible aux délais de portes logiques

Le délai de porte logique n'est pas un paramètre fiable d'un système. Il est donc chaudement recommandé de concevoir des systèmes de manière la plus insensible possible aux délais de portes logiques. Ceci se fait notamment en utilisant une logique synchrone.

Rappelons qu'un système logique synchrone est un système où les signaux d'horloge de toutes les bascules sont reliées au même signal de commande.

### 4.4.3 Règles relatives à la logique séquentielle et à la notion de délais de porte

Les systèmes logiques asynchrones sont difficiles à concevoir et encore plus difficiles à tester et à qualifier. Il est donc recommandé de se tenir le plus possible aux règles suivantes :

**4.4.3.1 Règle N°1** Toute logique séquentielle doit être synchrone et il faut s'efforcer d'utiliser dans la mesure du possible une horloge externe unique active sur un seul flanc. L'élément fondamental de ces systèmes séquentiels est la bascule D (flip-flop) ou E (Enable).

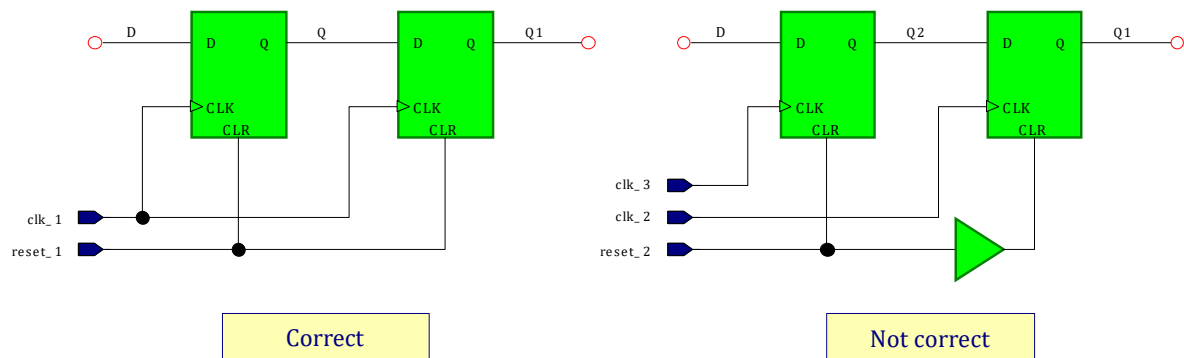
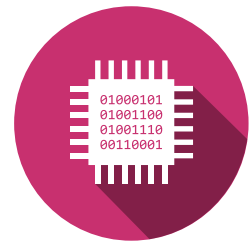


FIGURE 3 – Règle 1

**4.4.3.2 Règle N°2** Il est interdit d'utiliser le temps de transit des éléments logiques pour engendrer des impulsions en comptant sur la différence de temps de propagation entre une ligne directe et un réseau série d'éléments logiques :

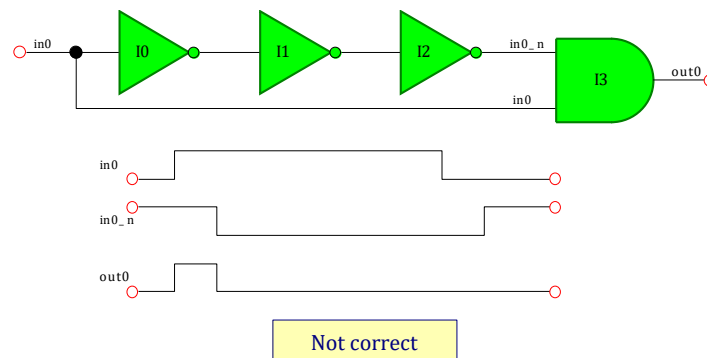


FIGURE 4 – Règle 2

**4.4.3.3 Règle N°3** Il est préférable d'amener directement les signaux d'horloge sur les bascules et de ne pas les contrôler par une logique combinatoire de manière à éviter tout aléa sur ces entrées. Il est nécessaire par ailleurs de s'assurer que les signaux présents sur les entrées synchronisées par l'horloge respectent bien les temps de prépositionnement (setup) et de maintien (hold) par rapport à l'horloge.

**Remarques :** le strict respect de la règle N°3 peut s'avérer préjudiciable lorsqu'une minimisation du nombre de portes ou de la consommation est recherchée. Ainsi, les diviseurs de fréquence asynchrones nécessitent bien moins de matériel que les diviseurs de fréquence synchrones. S'ils sont utilisés avec une logique qui travaille à une fréquence plus haute, il est important de resynchroniser les sorties des diviseurs avec le signal de fréquence supérieure. Certains blocs d'entrée ou de sortie nécessitent un fonctionnement asynchrone pour réagir rapidement à des conditions extérieures. Il s'agit alors de limiter au maximum la partie asynchrone du système. Pour les parties asynchrones d'un système, il est important d'analyser intensivement le comportement temporel du système et de concevoir des blocs fonctionnels sans retard de porte.

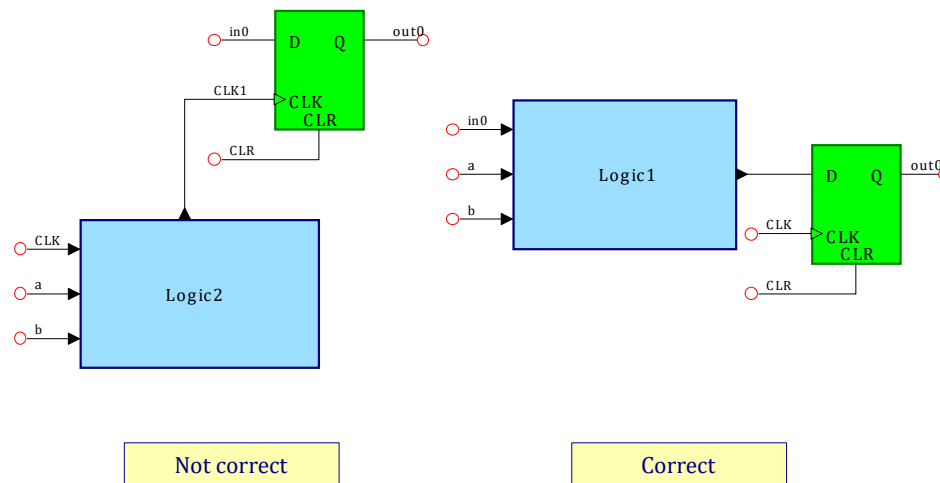
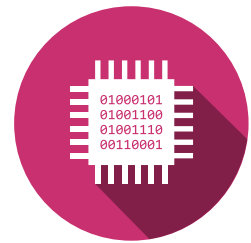


FIGURE 5 – Règle 3

**4.4.3.4 Règle N°4** Il ne faut pas amener des signaux avec des aléas sur les entrées asynchrones des bascules. Les signaux de Set et Reset ne doivent donc pas être issus de décodages purement combinatoires.

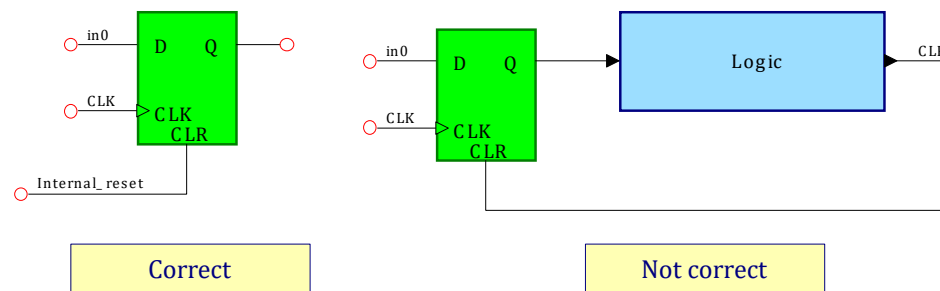


FIGURE 6 – Règle 4

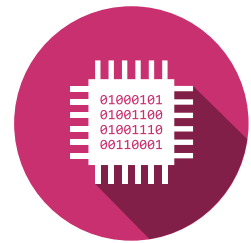
**4.4.3.5 Règle N°5** Toute machine séquentielle doit pouvoir être placée dans un état connu après la mise sous tension ou au début d'une simulation. Pour cela il est nécessaire d'utiliser les entrées asynchrones Set et Reset des bascules. Ces entrées ne doivent en principe pas être utilisées pour satisfaire à la fonctionnalité du circuit mais uniquement pour garantir leur testabilité.

**4.4.3.6 Règle N°6** La période minimale de l'horloge d'une machine synchrone devra être calculée comme suit :

$$T_{min} \leq T_{ClkQ_{max}} + T_{QD_{max}} + T_{skew} - T_{setup_{max}} \quad (1)$$

où :

- $T_{ClkQ}$  est le temps de retard entre le front de l'horloge et la sortie de la bascule Q,
- $T_{QD_{max}}$  représente le temps de propagation de la plus longue chaîne combinatoire entre une sortie Q de dispositif séquentiel et une entrée D de dispositif séquentiel sensibles au même flanc de la même horloge
- $T_{skew}$  représente la dispersion sur les chemins d'horloge aboutissant aux entrées clock des dispositifs séquentiels
- $T_{setup}$  est le temps de setup min des dispositifs séquentiels



**Remarques** : pour augmenter la vitesse de fonctionnement d'un circuit, il est possible d'insérer des registres de synchronisation dans les grandes chaînes logiques combinatoires.

**4.4.3.7 Règle N°7** il est nécessaire de synchroniser les signaux d'entrée d'un système à l'aide de bascules D. Le montage ci-après permet de filtrer les aléas et les états métastables qui peuvent se présenter lorsqu'un signal passe d'un système asynchrone à un système synchrone.

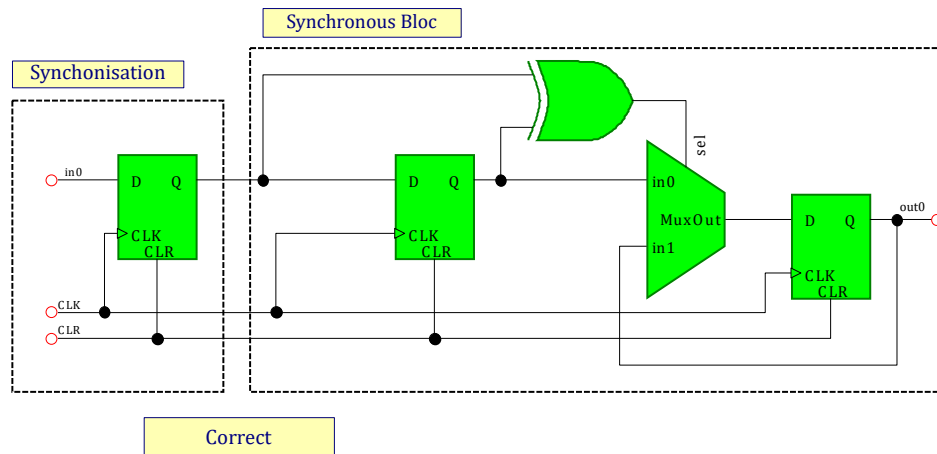


FIGURE 7 – Règle 7

**4.4.3.8 Règle N°8** La disparition du signal d'initialisation interne

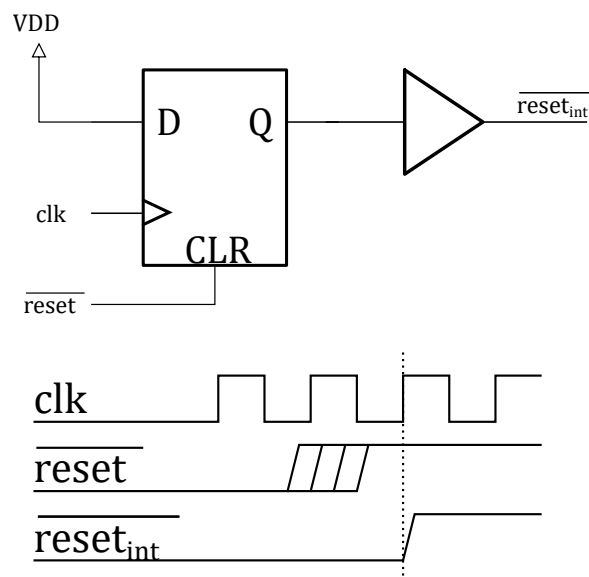


FIGURE 8 – Règle 8

#### 4.4.4 Règles liées au comportement électrique des éléments logiques

**4.4.4.1 Règle N°9** Ne pas trop charger les sorties des portes logiques : pour les signaux importants, estimer le fan-out de la porte qui les génère et le fan-in des portes que celle-ci doit piloter.

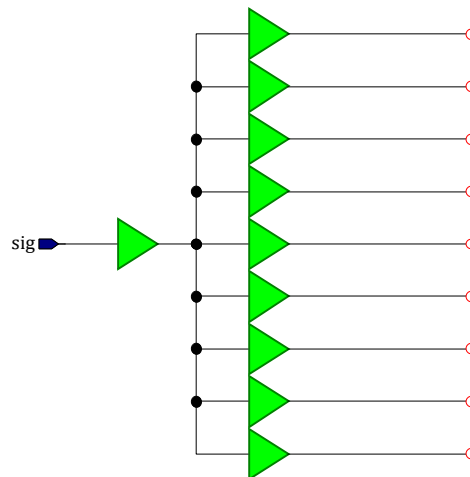
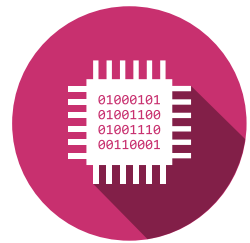
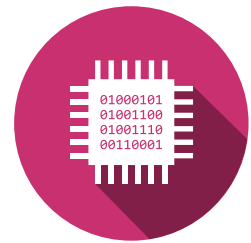


FIGURE 9 – Règle 9

#### 4.4.5 Conclusion

La meilleure solution pour réaliser un système logique complexe réside dans l'utilisation d'une logique synchrone. La logique synchrone ne permet pas d'atteindre des vitesses de traitement de l'information aussi élevées que la logique asynchrone, mais elle permet une plus grande flexibilité à la conception et, surtout, elle permet d'analyser le fonctionnement d'un système avec plus de maîtrise que ce n'est le cas pour la logique asynchrone.



## 5 Phase de vérification et validation

Toutes les études s'accordent sur un point. Une erreur coûte d'autant plus cher à corriger qu'elle est détectée tard dans le cycle de vie. Il est donc essentiel de tout mettre en oeuvre pour détecter les erreurs le plus tôt possible. Idéalement, il est souhaitable de détecter et de corriger une erreur dès la fin de l'étape dans laquelle elle est apparue. Les activités de détection et de correction des erreurs sont appelées activités de Vérification et Validation.

### 5.1 La validation

Les activités de Validation visent à vérifier que le produit en cours de développement correspond bien au produit attendu par le client. Elles essayent de répondre à la question : construisons nous LE bon produit ? La présentation de prototypes au client en phase de spécification, les revues conjointes ou la validation par le client sont des exemples de techniques de Validation.

### 5.2 La vérification

Les activités de Vérification cherchent à assurer que les produits en sortie d'une étape de développement sont bien conformes aux produits et normes en entrée de l'étape. En d'autres termes, elles essayent de répondre à la question : construisons nous UN bon produit ? Les revues, les activités de mise au point, l'inspection de schéma ou code sont quelques exemples de technique de Vérification. Une vérification ne nécessite pas, le plus souvent, une participation des clients.

### 5.3 Technique de validation et vérification

Il existe de très nombreuses techniques de vérification et validation. Certaines sont dites statiques car elles s'appliquent à des documents (comme la revue). D'autres sont qualifiées de dynamiques car elles nécessitent la simulation de tout ou partie du code VHDL (comme le test unitaire).

Une fois la démarche de développement arrêtée, il faut donc pour chacune des étapes qui ont été identifiées, se poser la question suivante : que faire pour s'assurer que les produits en sortie de cette étape sont d'une part conformes aux attentes du client (Validation) et d'autre part pertinents et cohérents par rapport aux objectifs fixés pour l'étape et aux produits en entrée de l'étape (Vérification).

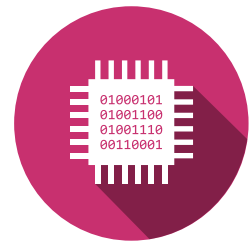
Il est important de prévoir et de mettre en évidence dans le Planning de projet toutes les activités de VV qui sont envisagées.

Pour des projets de taille modeste le minimum est de prévoir pour la **Vérification** :

- les simulations des fonctions vhdL ou schématique développées
- La revue de la documentation de conception par des audits compétents sur les points techniques abordés
- La relecture croisée du code vhdL par les différents membres de l'équipe

Pour la **Validation** prévoir au minimum :

- La ou les simulations effectuées pour valider le design du circuit dans son entier. Pour cela, il est nécessaire d'élaborer à partir des spécifications une matrice de conformité comportant l'ensemble des points à contrôler. Pour chaque point, il s'agira d'indiquer par quelle simulation et à quel endroit de celle-ci la vérification a eu lieu. Un exemple d'une telle matrice



est donné à la page suivante.

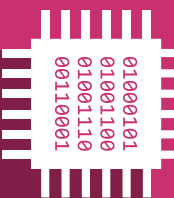
- La revue conjointe de la documentation de spécification par, d'une part, des audits compétents dans le domaine d'application abordé, et d'autre part, des représentants de chaque catégorie de client (futurs utilisateurs, décideurs, etc.)
- Des tests de validation internes (uniquement interne à l'équipe de développement) et externes (tests de recette chez et avec les clients).

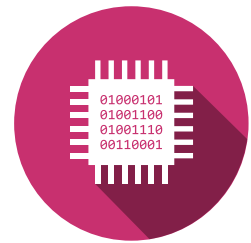
L'usage d'autres techniques est également conseillé : réalisation de prototypes en phase de spécification (Validation), analyse du code à l'aide d'outils dédiés, etc. Parmi toutes les techniques de V&V, les techniques de simulation et de revues sont les plus efficaces et les plus pratiquées dans les situations réelles de réalisation.



Numéro d'identification l'exigence fonctionnel	Description	Testbench & Si- mulation Setup	Instant de validation	Méthode de vali- dation	État de validation	Annexes page
(1)	(2)	(3)	(4)	(5)	(6)	(7)

- (1) Chaque spécification doit être identifiée par un numéro unique : DSxx-YYXXX
  - où DSxx est le numéro du document de spécification (DS pour Design Specification)
  - où YY représente le numéro de chapitre (1, 2, 3, ...,11,...)
  - où XXX représente le numéro de l'exigence fonctionnelle (005, 010, 015, 020, 021, 030, 035 ...)
- (2) Mots clés de l'exigence fonctionnelle à valider
- (3) Identification du testbench utilisé et du setup de simulation
- (4) Instant de validation de l'exigence dans la simulation.
- (5) Méthode de validation (analyse, contrôle visuel, impression, fichier de résultats, validation automatique, ...)
- (6) État de validation(OK, Not OK, valeur mesurée)
- (7) Numéro d'annexe si nécessaire.





## 6 Phase d'intégration

Cette phase regroupe toutes les étapes de la branche de droite du schéma en V depuis la conception détaillée jusqu'à la fin du projet. Les groupes de développement doivent organiser cette phase afin que l'enchaînement vérifications et validations unitaires - intégration - validation système soit effectivement réalisé. En effet, c'est le respect sérieux de cet enchaînement qui permet de garantir la conformité et la qualité du produit.

La réalisation de cette phase est d'autant plus aisée que les phases antérieures ont été bien faites : si les phases de spécification des besoins et de conception générale ont prévu les mécanismes de V&V et d'intégration, le respect de cette séquence est un simple travail de routine assez rapide.

### 6.1 Résultats

Les productions attendues de cette phase ne sont pas seulement constituées de rapports, on doit aussi trouver un ensemble de sources et d'exécutables qui constituent l'application opérationnelle ; il y a donc un ensemble de rendus à livrer sur un support de données adéquat.

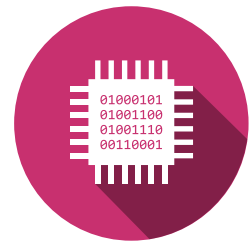
Les rendus sous forme de dossiers sont :

- Un rapport de conception détaillé qui fait une description des activités de conception détaillée, codage et vérification, validations unitaires, intégration et validation finale. Les points qui doivent ressortir de ce rapport sont le sérieux des validations faites et les éléments qui vont favoriser la reprise ultérieure du projet par une autre équipe.
- Un Manuel Utilisateur (MU) : c'est le document qui est remis au client, utilisateur final du matériel. Il doit décrire, l'installation, le paramétrage et le mode d'emploi du matériel. En fait ce document est rédigé tout au long du projet puisque c'est une reprise de la maquette élaborée en phase de conception générale qui était elle-même une reprise de la Description Fonctionnelle du Produit (DFP) élaborée en phase de spécification des besoins.

### 6.2 Moyens mis en oeuvre

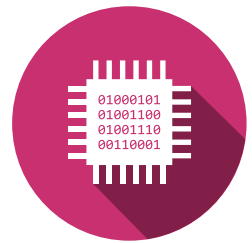
Cette phase d'intégration correspond le plus souvent à une période de travail à plein temps du groupe. Outre la maîtrise des moyens techniques, deux classes de problèmes sont couramment observées durant cette période :

- Les problèmes de délais : quoiqu'on fasse, la date butoir approchant, les retards accumulés dans les phases précédentes vont apparaître au grand jour. Ceci a le plus souvent pour effet de conduire les participants du groupe à la panique et à négliger en premier lieu les règles de qualité et d'organisation ... ce qui ne fait qu'empirer le mal. Il n'y a qu'une solution à ce problème : anticiper, ANTICIPER, ANTICIPER, ANTICIPER. C'est durant la phase de spécification des besoins que les délais d'un projet se tiennent ou non : les développeurs et, particulièrement les enseignants maîtres d'œuvre, doivent être particulièrement vigilants lors du démarrage du projet (les quinze jours qui manquent à la fin sont souvent les quinze jours perdus au début !!). C'est aussi la pratique sérieuse du suivi de projet qui peut éviter d'arriver à des situations catastrophiques en permettant, par exemple, de redéfinir à temps les objectifs.
- La gestion du travail de groupe : si beaucoup des réflexions des phases précédentes peuvent et/ou doivent se faire collectivement dans le groupe, cette phase est, par excellence, celle du travail individuel éclaté. Le travail ne peut être réalisé dans les temps qu'en étant géré en



parallèle. Or l'addition de travaux individuels ne fait pas forcément une somme de résultats au bout du compte : il faut y ajouter de l'organisation pour que ça fonctionne. Il est donc nécessaire de prévoir une structure de répartition des tâches et des temps de synchronisation pour les membres du groupe. Plus le groupe est nombreux, plus l'organisation doit être rigoureuse et plus le temps nécessaire à la synchronisation est long. Ne pas oublier de prévoir explicitement ce temps dans les plannings. Pour un groupe de 5 à 6 personnes qui travaillent à temps complet sur un projet, un point rapide quotidien (café du matin ou pause du soir par exemple) et une réunion structurée hebdomadaire semblent constituer une solution réaliste.

Sur le point de vue technique, un élément important est la maîtrise technique : chaque développeur doit maîtriser tous les outils (de développement, de documentation, de gestion de projet) qui sont utilisés dans le projet. La phase d'intégration n'est pas une phase d'apprentissage. Si, comme c'est le cas dans presque tous les projets, des techniques nouvelles sont utilisées, il a dû y avoir, en parallèle avec le développement des phases précédentes, des étapes d'apprentissage des outils nécessaires. Si un outil n'est pas maîtrisé, dans la plupart des cas, il vaut mieux renoncer à l'utiliser et revenir à des techniques, peut-être moins spectaculaires, mais maîtrisées.



## Acronymes

**FIFO** First In First Out. [7](#)

**I/O** Input Output. [7](#)

**PC** Program Counter. [7](#)

**RAM** Random Access Memory. [7](#)

**VHDL** Very High Speed Integrated Circuit Hardware Description Language. [7](#), [8](#)