

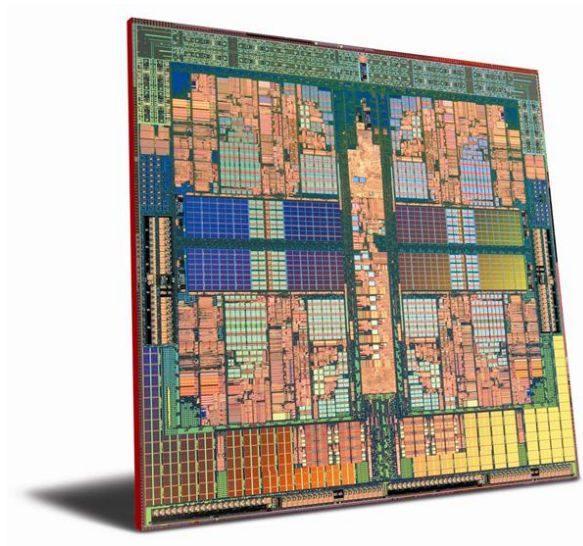
Durand-Estebe Paul
Hippert Simon
Janin Cédric
Percia-Mehlem Caio

4^{ème} année AE
Groupe A2

EMCA

Rapport de projet

Réalisation d'un Timer de microprocesseur



Le 7 janvier 2011

Table des matières

Introduction	1
I. Structure logique globale.....	2
II. Les principaux blocs constitutifs du Timer	6
1. Bascule D	6
2. Multiplexeur	7
3. Contrôleur de modulo	8
4. PWM	9
III. Simulation globale.....	10
Conclusion	14

Introduction

Les 3 dernières séances de travaux pratiques du cours d'EMCA (Etude et Modélisation des Composants Actifs) ont fait l'objet d'un mini-projet, ayant pour but de mettre en application nos connaissances en micro-électronique. Les travaux ont été réalisés à l'aide des logiciels MicroWind et Dsch, avec lesquels nous avons souhaité concevoir un périphérique de microcontrôleur du type Timer.

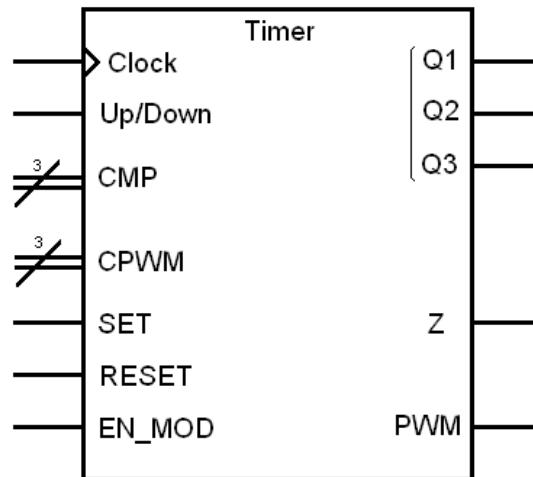
Les caractéristiques de ce Timer sont :

- 3 bits
- Comptage
- Décomptage
- Modulo réglable
- Set / Reset
- Sortie PWM

Ces caractéristiques ne représentent bien évidemment pas l'ensemble des fonctionnalités que peuvent proposer les Timers actuels, elles constituent néanmoins une base de travail solide dans un but d'apprentissage.

I. Structure logique globale

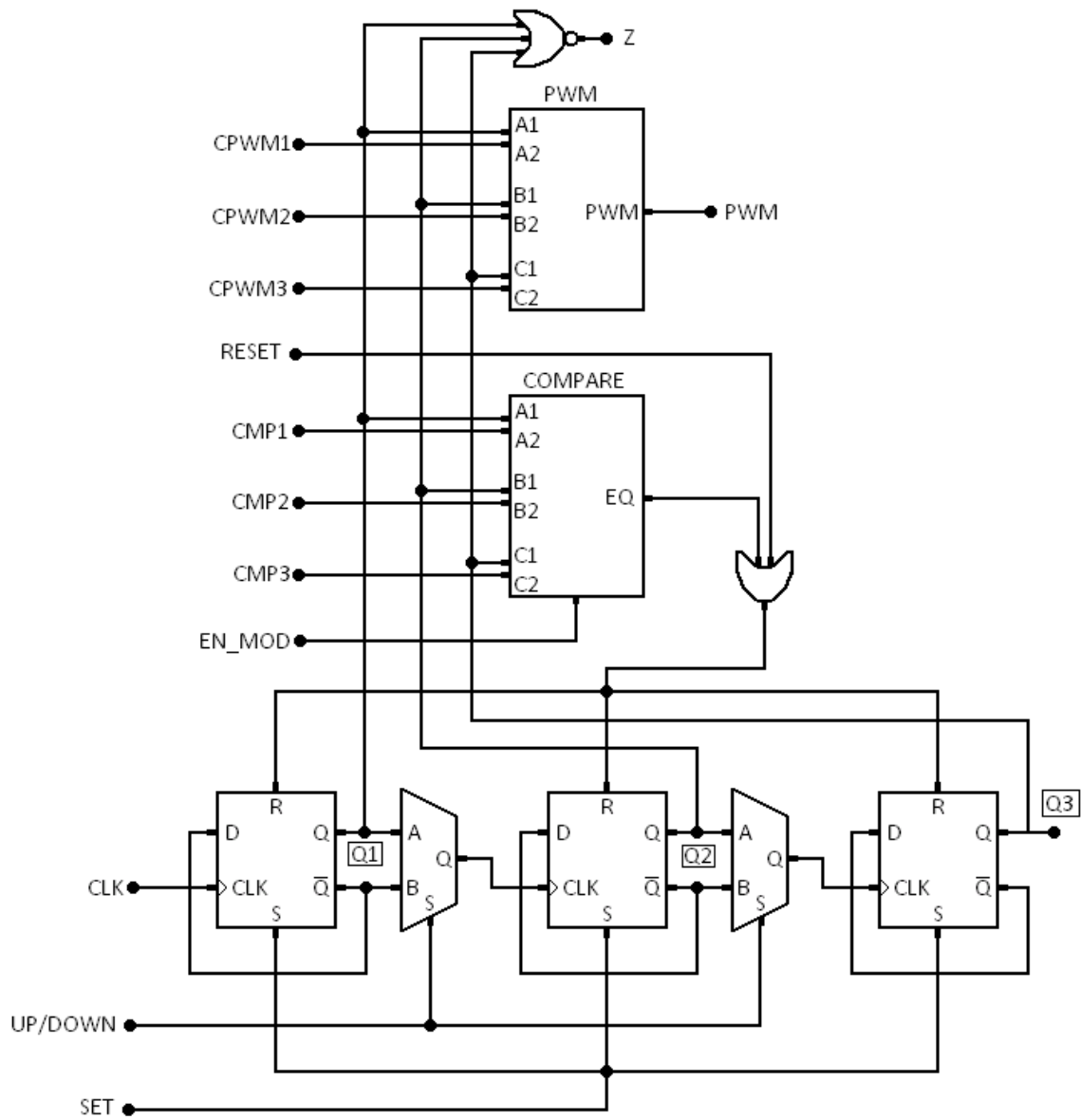
D'un point de vue externe, on peut voir notre circuit intégré comme un composant à part entière de la forme :



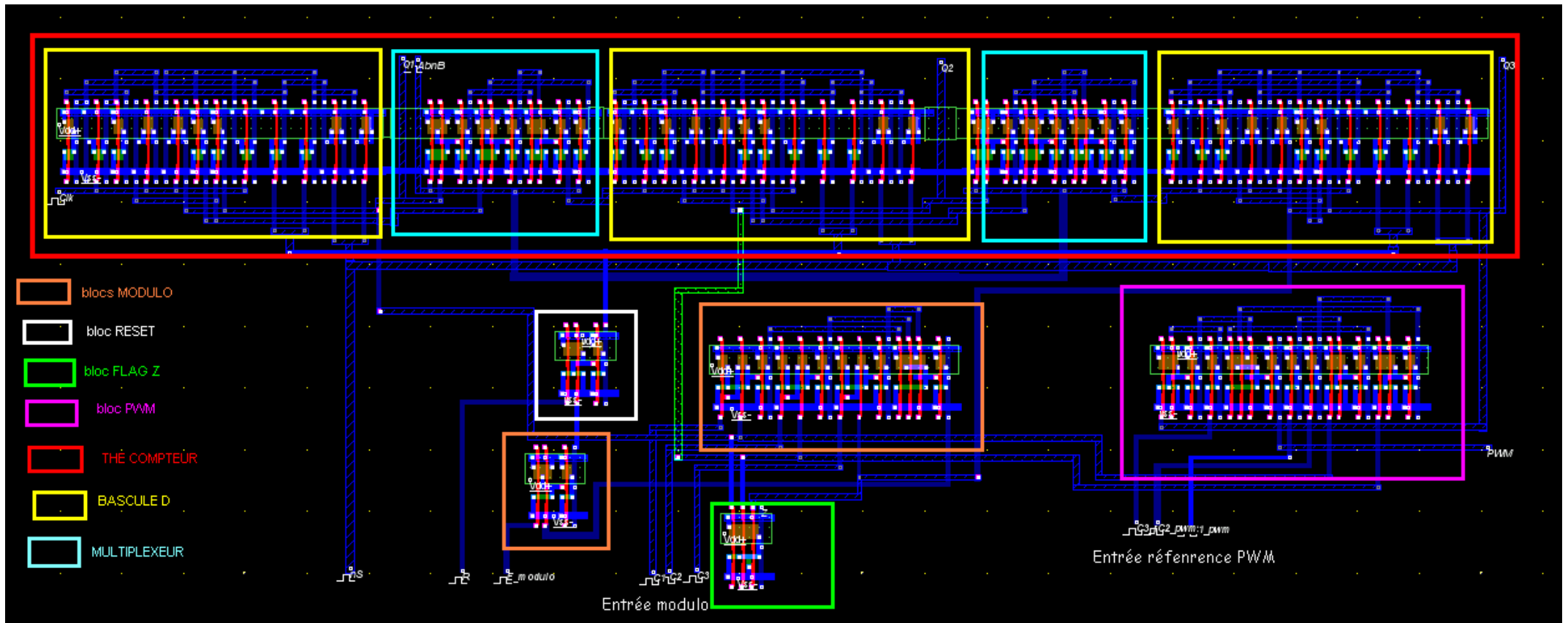
Détail des ports :

- Clock : entrée de l'horloge, active sur front montant.
- Up/Down : sélection compteur/décompteur. 0 -> compteur, 1 -> décompteur.
- CMP : valeur de modulo, sur 3 bits.
- CPWM : valeur de seuil pour génération de signal PWM.
- Set : mise à « 111 » du compteur, actif au niveau bas.
- Reset : mise à « 000 » du compteur, actif au niveau haut.
- EN_MOD : activation du modulo variable, actif au niveau haut.
- Q1-Q2-Q3 : sortie du compteur. Bit de poids faible : Q1.
- Z : flag Zéro. Vaut 1 quand la sortie du compteur vaut « 000 ».
- PWM : sortie de signal PWM.

Schéma logique global :



Layout (ci-après) :



Processus de conception :

Un périphérique Timer est un système complexe composé de plusieurs sous-systèmes simples. La partie fondamentale d'un tel périphérique est le comptage, qui, une fois associée à une horloge, lui permet de mesurer le temps.

Dans cette logique nous avons commencé le projet par l'implémentation d'un compteur 3 bits à l'aide de bascules D. Nous avons obtenu la structure CMOS de la bascule D à partir de son schéma électronique proposé page 292 de « Basics of CMOS Cell Design ». Ce schéma a ensuite été amélioré afin d'implémenter la fonction de Set.

Dans un second temps, nous avons souhaité ajouter la fonctionnalité de décomptage. Pour cela nous avons décidé d'intercaler un multiplexeur entre deux bascules D. Ceux-ci possèdent deux voies d'entrées, une sortie, et une voie de sélection.

Ensuite, afin de pouvoir régler la valeur maximum de comptage, nous avons développé un bloc de test d'égalité de deux mots de 3 bits, celui-ci contrôlant la fonctionnalité de Reset des bascules D (fonctionnalités que l'on avait anticipées lors du choix de bascule au début du projet).

Un bloc semblable a aussi été pensé, celui-ci surveillant l'inégalité de deux mots de 3 bits, et ayant pour but de générer alors un signal PWM entièrement réglable.

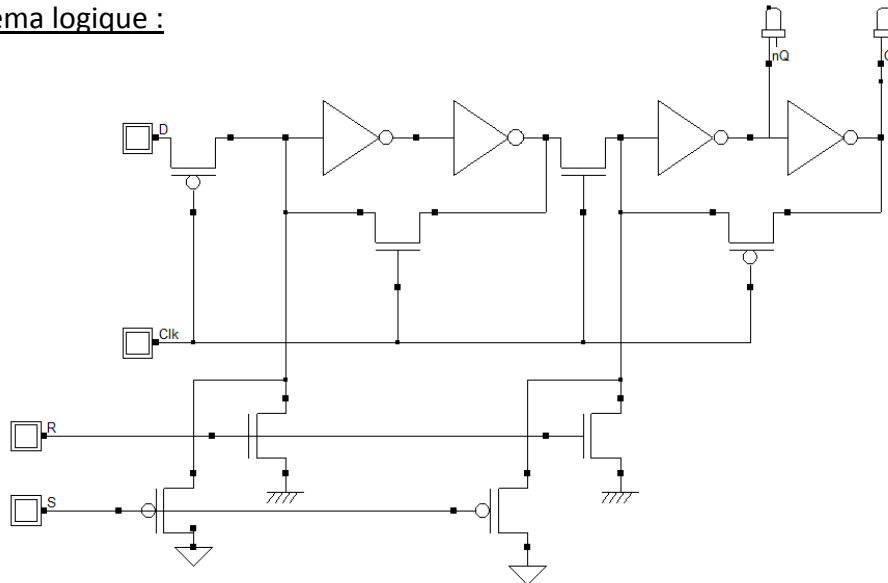
Pour finir, afin de coller au plus près des besoins d'un microcontrôleur, une simple logique combinatoire détecte une sortie nulle du Timer, un « flag Z ». Celui-ci peut être utile au microcontrôleur pour le déclenchement d'interruptions.

II. Les principaux blocs constitutifs du Timer

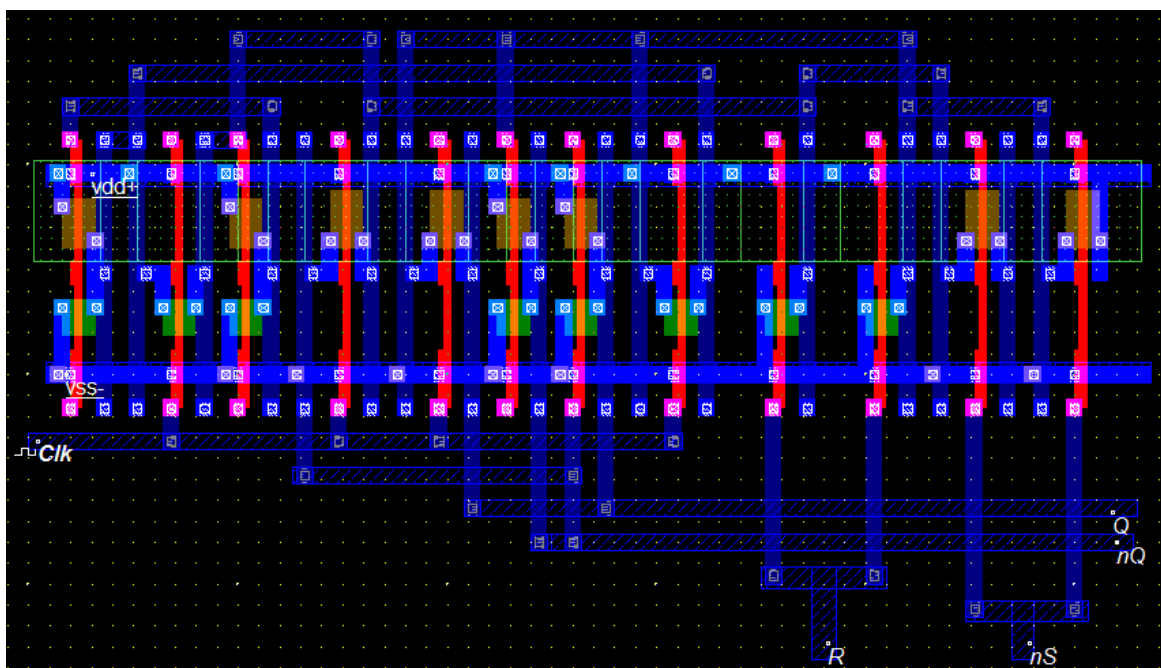
1. Bascule D

Nous utilisons des bascules D, actives sur front montant d'horloge, avec les fonctionnalités de Set et de Reset. Mises en cascade, 3 bascules créent un compteur 3 bits asynchrone qui sera la base de notre circuit. Les fonctionnalités de Set et Reset sur chaque bascule permettra le forçage à 1 ou à 0 du compteur.

Schéma logique :



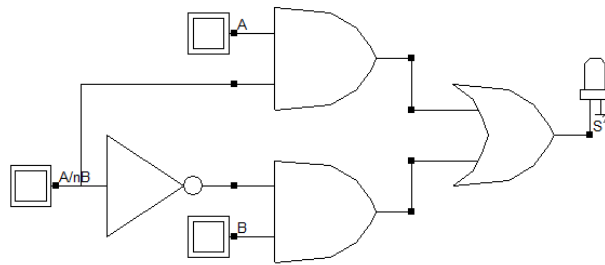
Layout généré :



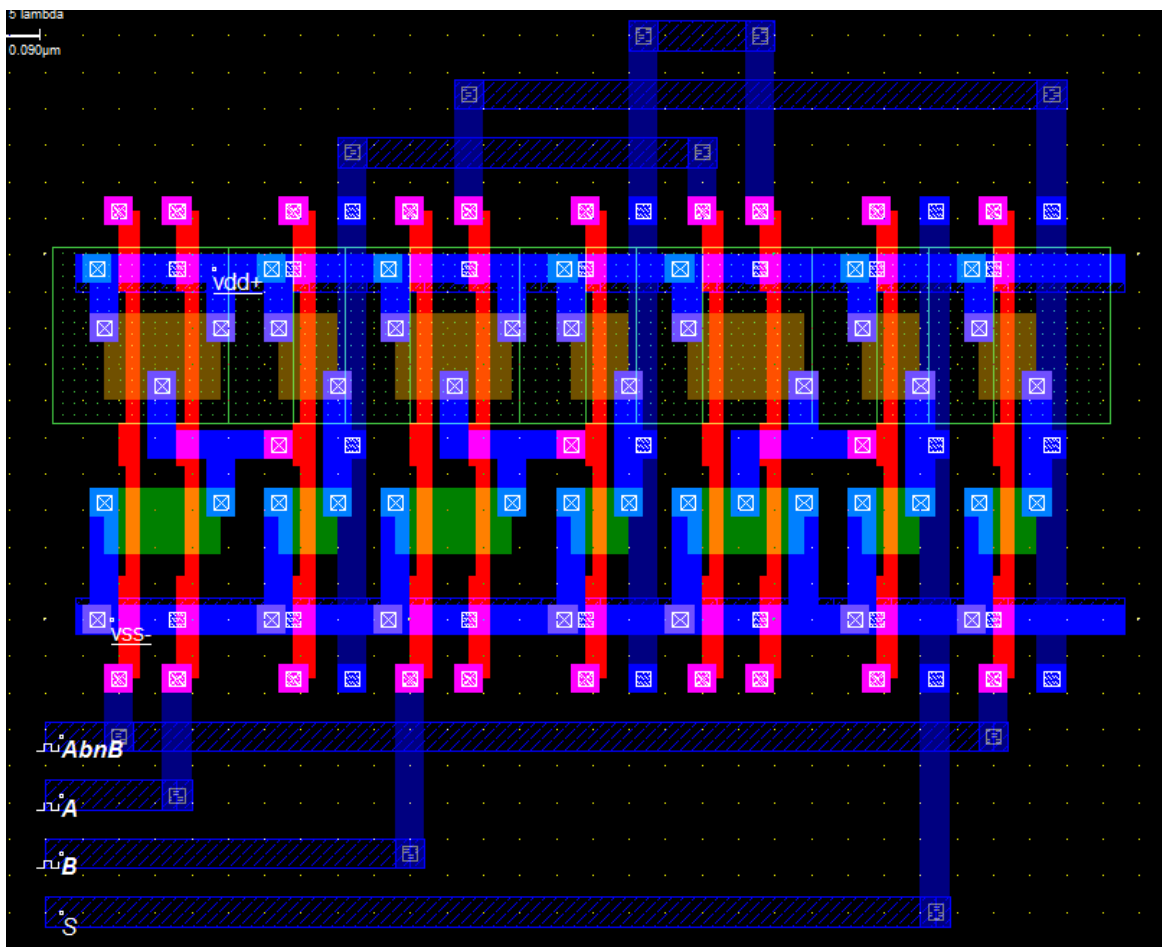
2. Multiplexeur

Le multiplexeur est intercalé entre chaque bascule. Il va sélectionner la sortie Q ou nQ comme entrée d'horloge de la bascule suivante. Lorsque l'horloge est attaquée par le signal Q, on obtient un décompteur. Lorsqu'elle est attaquée avec le signal nQ, c'est un compteur. Les deux multiplexeurs du circuit sont contrôlés par l'entrée « Up/Down » de la puce.

Schéma logique :



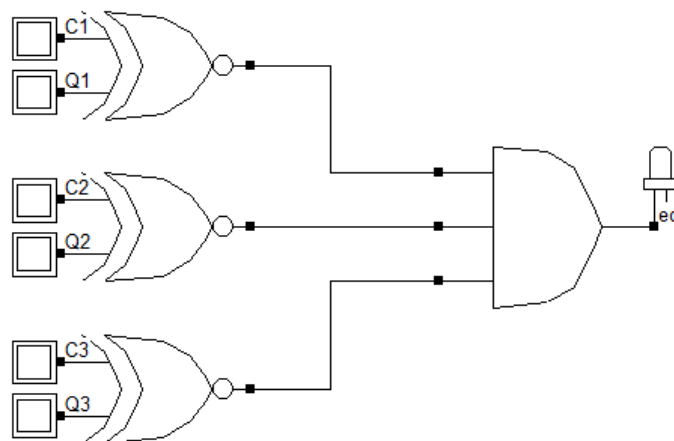
Layout généré :



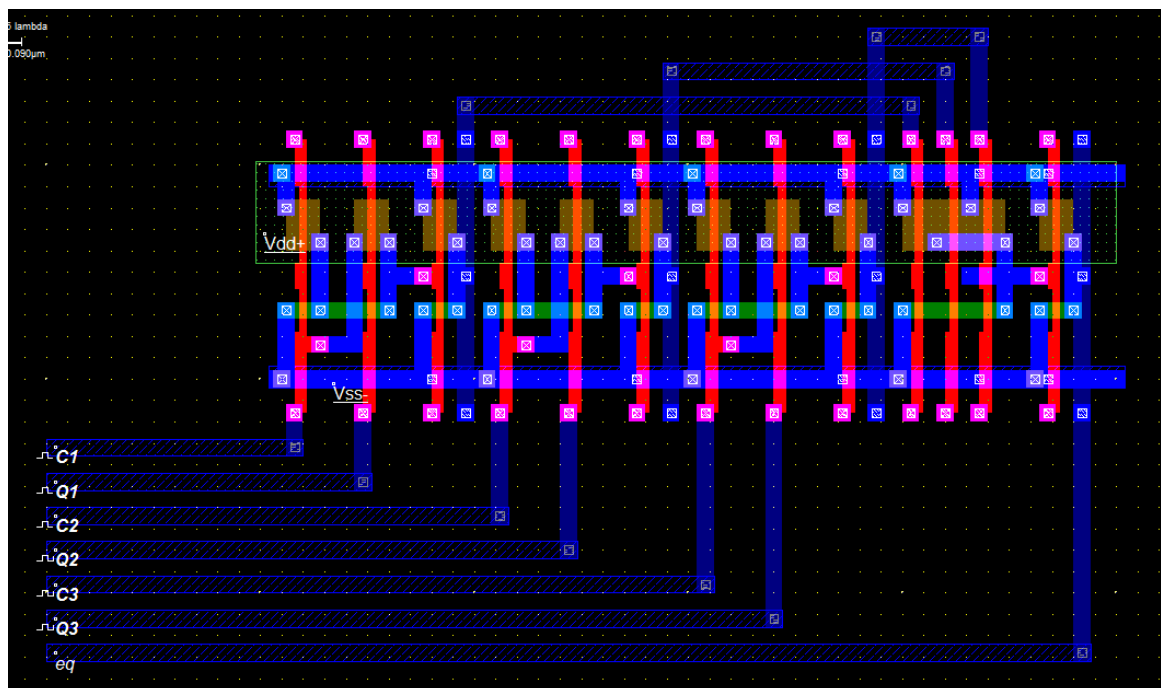
3. Contrôleur de modulo

Le contrôleur de modulo permet de régler la valeur maximum de comptage. Pour cela la sortie est comparée à 3 bits (CMP, ici C1, C2, C3). Une fois cette valeur égale à la valeur courante du compteur, l'égalité déclenche un Reset du compteur. Pour que cela fonctionne, il faut être en mode « comptage » (Up/Down = 0) et activer le modulo (EN_MOD = 1).

Schéma logique :



Layout généré :



4. PWM

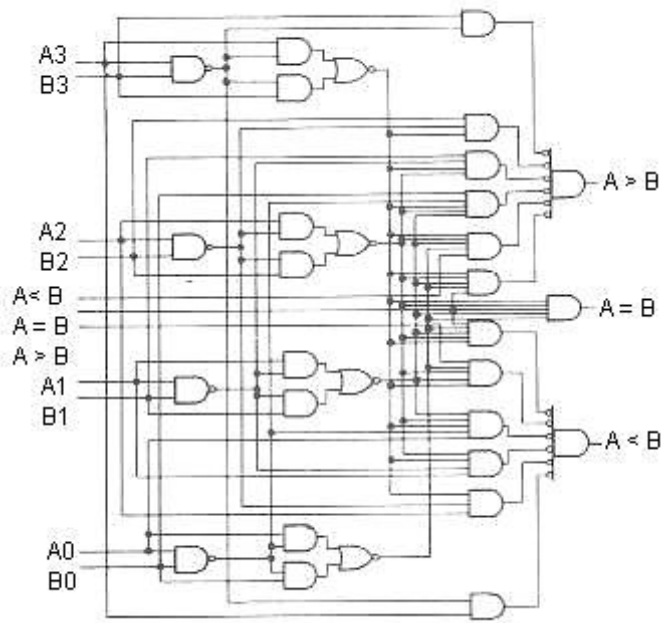
Le module PWM compare les 3 bits du compteur (Q1 à Q3) à 3 bits de référence (CPWM). Tant que la valeur du compteur est inférieure à la référence, la sortie du PWM est à 1. Sinon elle passe à 0.

Par exemple, pour une valeur de référence à 2, si le compteur est en modulo 8, on aura un rapport cyclique de sortie du PWM à 25%.

Implémentation :

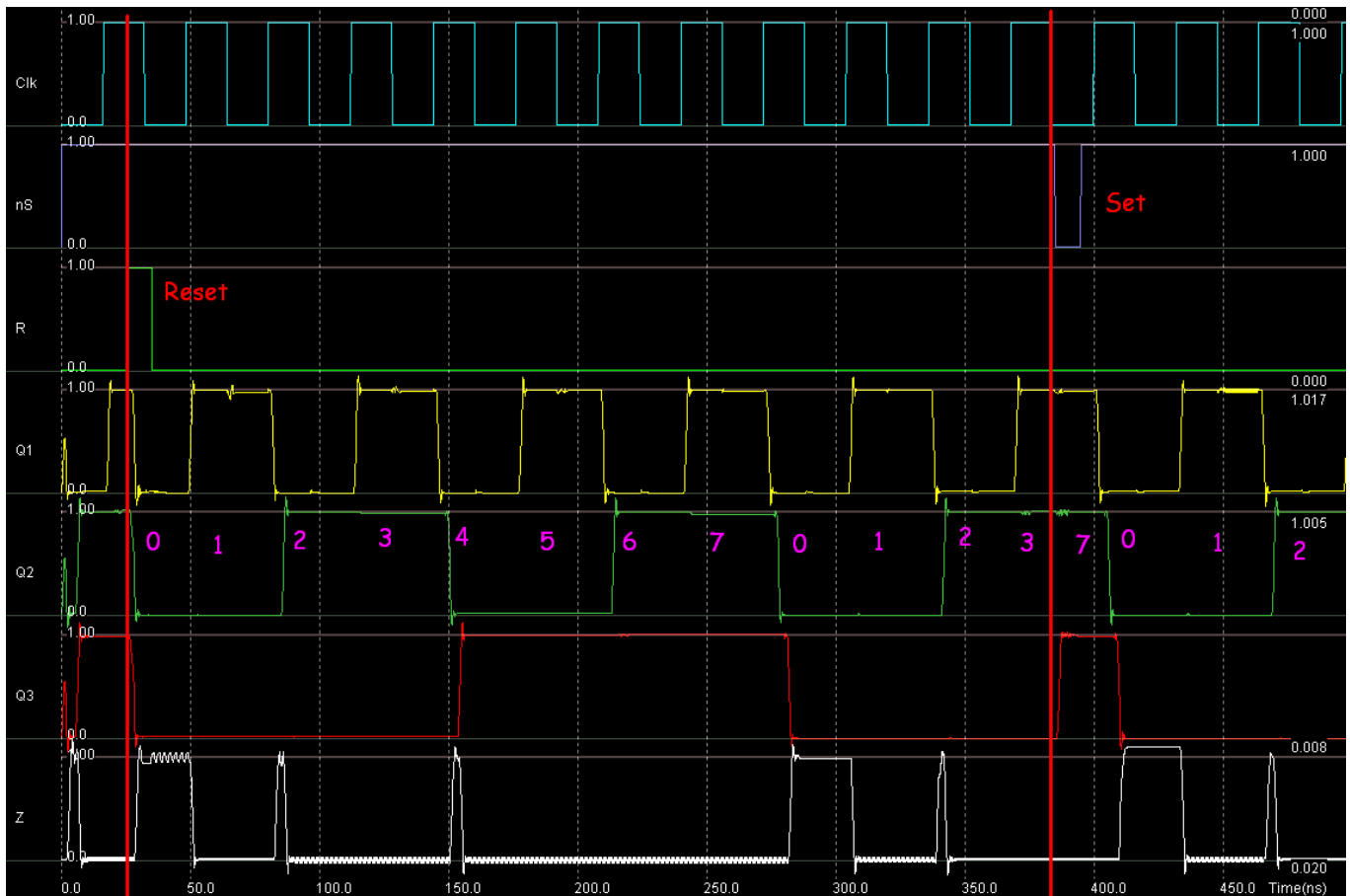
Dans un premier temps, nous avons créé un module pour assurer cette fonction. Hélas, nous avons omis quelques aspects de logique élémentaire et de ce fait, notre comparateur, trop simpliste, s'est avéré défectueux.

Voici le schéma du comparateur que nous aurions dû implémenter (sortie A<B de ce comparateur 7485). Or, ce fut avec grand désarroi que nous manquâmes de temps pour réaliser un module PWM fonctionnel, au vu du trop grand nombre de portes logiques à implémenter.



III. Simulation globale

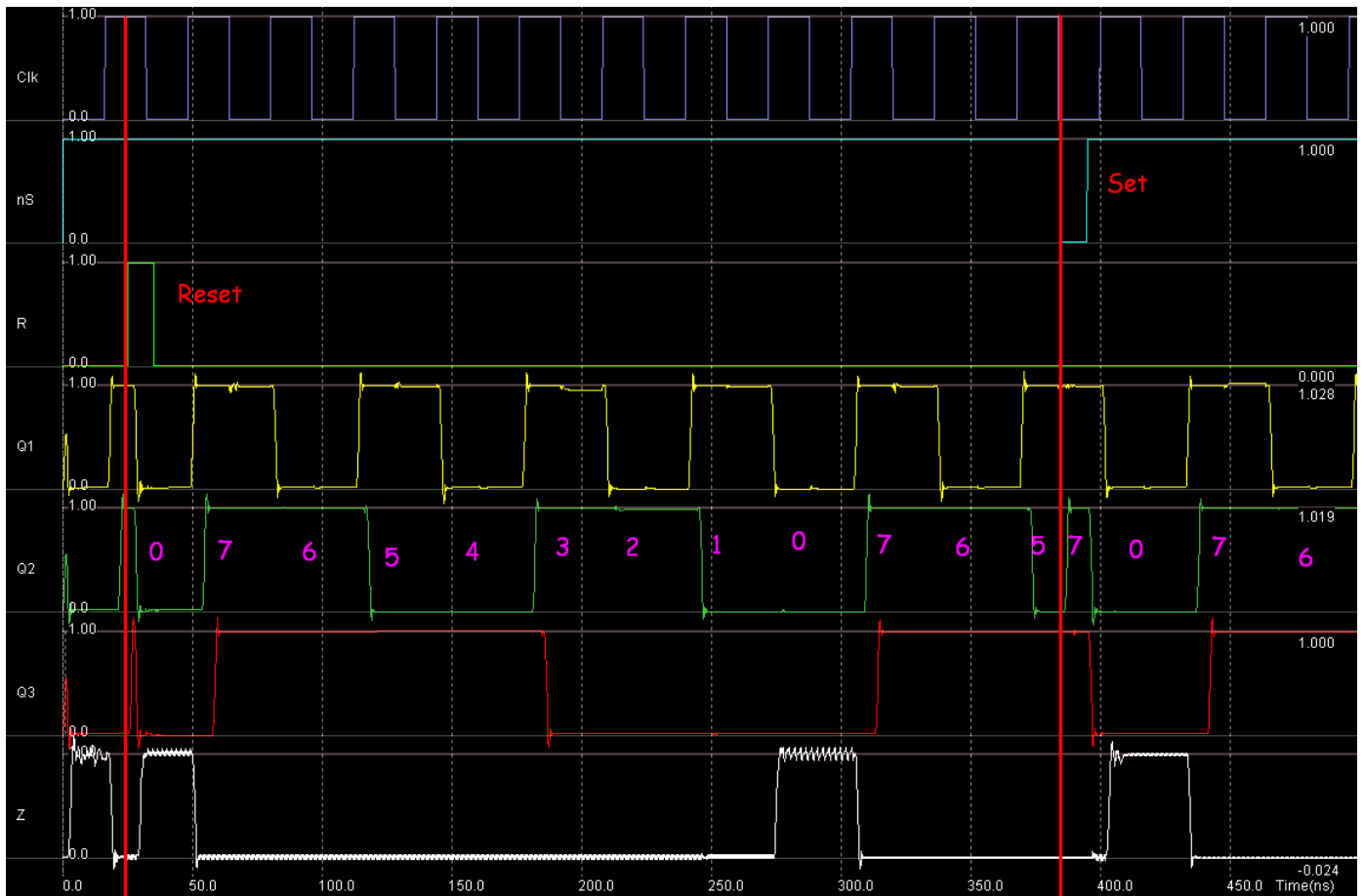
Chronogramme Compteur :



Le compteur est en fonctionnement simple, sans modulo. Il compte donc de 0 à 7 à une fréquence d'environ 30 GHz. On observe le Reset asynchrone en début de chronogramme qui réinitialise les bits Q à 0 et le Set asynchrone en fin de chronogramme qui met les bits Q à 1.

On observe également le flag Z qui se lève lorsque la valeur du compteur est 0. On s'aperçoit qu'il se lève brièvement lors de certains changements de valeur (par exemple lors du passage de 1 à 2 et de 3 à 4). Ce phénomène est dû au temps de propagation du signal à travers les bascules qui ne commutent pas en même temps. Cela paraît gênant, mais vu la très faible durée de ces derniers (de l'ordre de la nanoseconde), en fonctionnement à plus basse fréquence, ces pics sont négligeables (on pourrait même rajouter un filtre passe bas pour les enlever).

Chronogramme Décompteur :



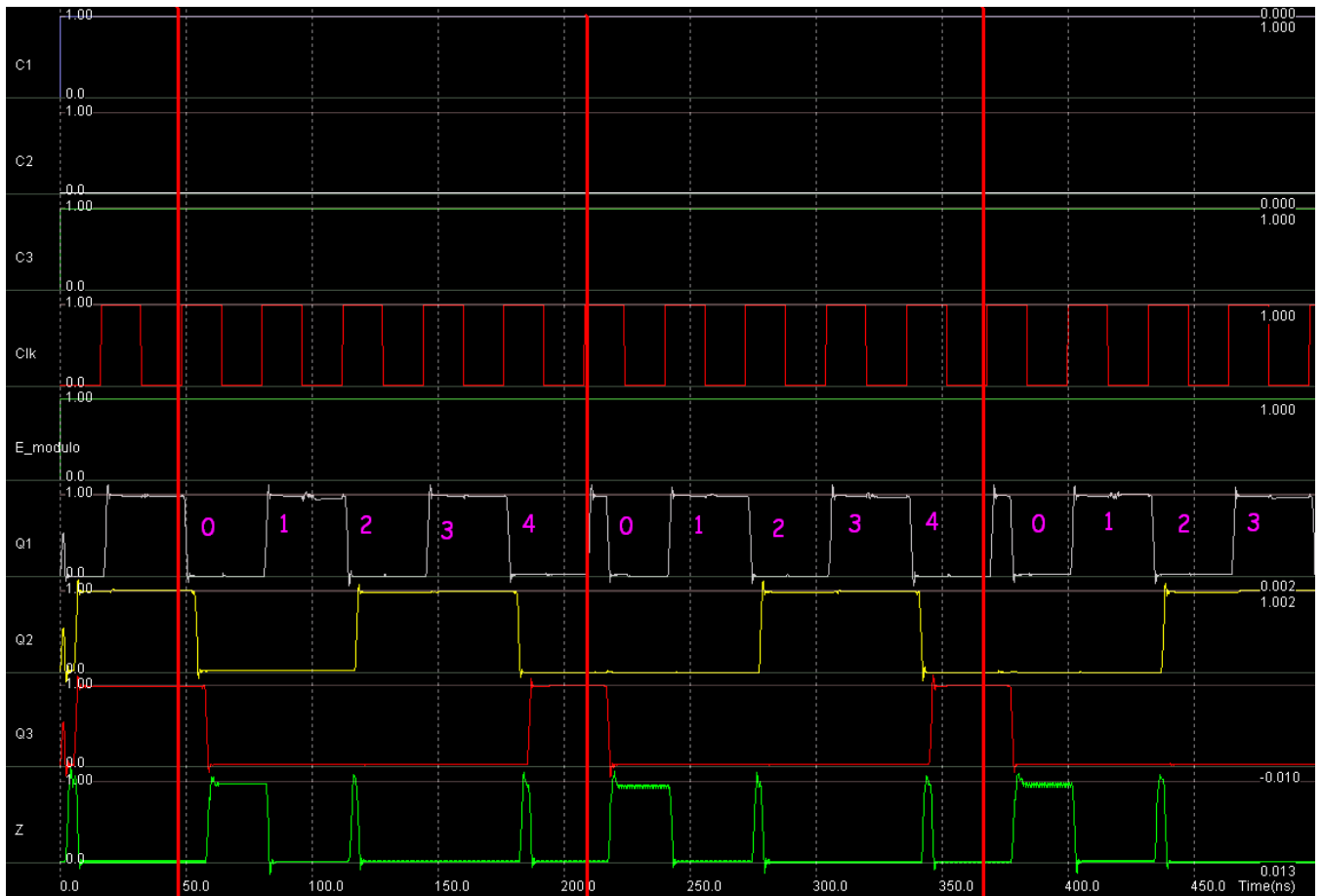
Le décompteur est ici aussi en fonctionnement simple. Il décompte donc de 7 à 0 à une fréquence d'environ 30 GHz. On observe le Reset asynchrone en début de chronogramme qui réinitialise les bits Q à 0.

Notre compteur ne permet pas d'utiliser correctement le Set en fonctionnement décomptage car la structure que nous avons choisi inverse la valeur des bits après leur forçage. Cela vient du fait qu'en décomptage, c'est la sortie Q (et non nQ en comptage) qui est branchée sur l'entrée CLK de la bascule suivante. Ainsi, après un set, les bits bascule de 111 à 000 avant de recommencer à décompter (si on pouvait forcer les bits à une autre valeur, par exemple 101, le compteur aurait inversé à 010 avant de continuer à décompter).

On a donc un état indésirable qui dure une période d'horloge. La fonction Set n'est donc pas utilisable en décomptage.

Remarque : ce phénomène est aussi présent au Reset mais n'est pas gênant puisque après 000 il vient naturellement 111 en décomptage.

Modulo 5 :



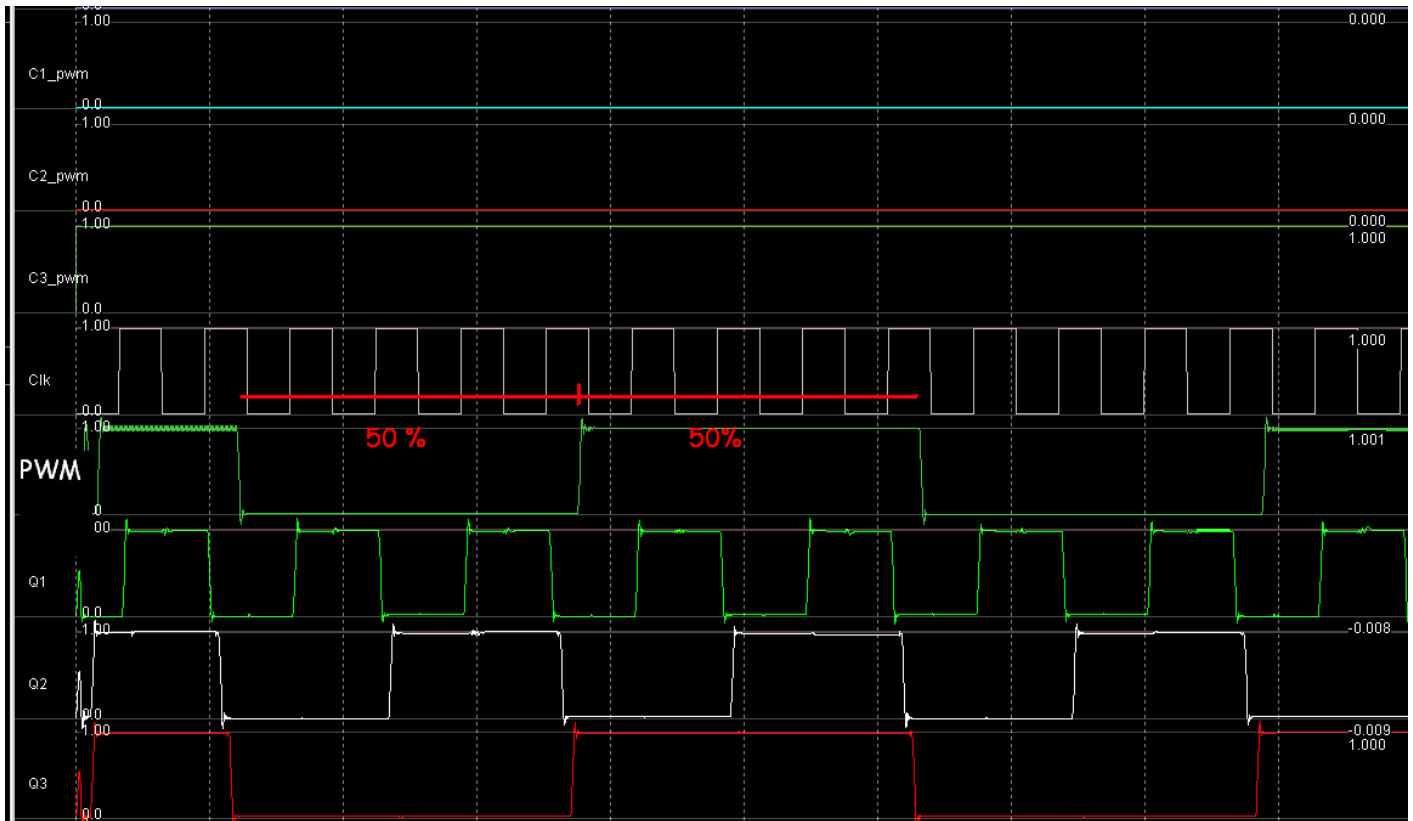
Le compteur est en fonctionnement avec un modulo 5, il compte donc de 0 à 4. Pour activer ce mode, $E_modulo = 1$ et $CMP=101$.

On voit que le compteur repasse à 0 tous les 5 fronts d'horloge.

Lors de la réinitialisation il y a aussi un état indésirable ($Q = 5$) qui est aussi dû au temps de propagation du signal dans les portes logiques de notre bloc modulo. Mais cela n'est, ici, absolument pas gênant pour le fonctionnement puisque l'on a quand même 5 périodes d'horloge entre chaque réinitialisation.

PWM :

Comme expliqué précédemment, notre PWM n'est pas au point. Cependant, pour la valeur 4 (rapport cyclique 50% si il n'y a pas de modulo), le module marche exceptionnellement.



Description du fonctionnement attendu :

La sortie PWM est à 1 lorsque la valeur du compteur est supérieure à la valeur de référence CPWM et à 0 sinon. Ceci permet donc, en jouant sur le modulo du compteur et sur la valeur de référence, de générer une large possibilité de rapports cycliques différents en sortie.

La fréquence du PWM sera égale à la fréquence de l'horloge divisée par le modulo du compteur.

Conclusion

Tout d'abord, ce projet nous a permis de mettre en application nos connaissances en logique séquentielle et combinatoire, afin de concevoir le circuit logique de notre puce. Dans un second temps, nous avons appris à maîtriser les logiciels de conception MicroWind et Dsch ainsi que les méthodes d'élaboration de layout, en appliquant nos connaissances sur la physique des semi-conducteurs et la micro-électronique.

Nous sommes parvenus à obtenir un circuit intégré fonctionnel, répondant au cahier des charges que nous nous sommes fixés. Même si la dernière étape consistant à l'implémentation du bloc PWM n'a pas pu être réalisée, ceci n'a été qu'une question de temps et non de limite de connaissances.

Les limites de fonctionnement, vues précédemment, sont directement liées à un choix de conception initial : compteur asynchrone et non synchrone. Ceci crée une accumulation des retards, menant à des valeurs parasites lors du comptage et à pour principal effet de nécessiter un filtre passe-bas en sortie du compteur, et donc limiter la fréquence de fonctionnement globale.

Il aurait été intéressant, dans un second temps, de :

- Finaliser la partie PWM ;
- Filtrer la sortie et déterminer une fréquence maximum de fonctionnement ;
- Implémenter les Input/Output.