

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEINEMENT SUPERIEUR

ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DE BATNA

FACULTE DES SCIENCES DE L'INGENIEUR

MEMOIRE

Présenté au

DEPARTEMENT D'ELECTRONIQUE

Pour l'obtention du diplôme de

MAGISTER EN MICROELECTRONIQUE

Option : IC Design

Par

Samir BARRA

Ingénieur, institut d'Electronique-Université de Batna

Intitulé

Conception et layout d'une machine d'états

à base de latch C²MOS à technologie

CMOS 0.35μm

Devant le jury constitué de :

Dr. HOBAR Farida

Prof. U. Constantine

Président

Dr. BOUGUECHAL Nour-Eddine

Prof. U. Batna

Rapporteur

Dr. DIBI Zohir

M.C. U. Batna

Examineur

Dr. HAFDI Zoubeida

M.C. U. Batna

Examineur

Dr. BENHAYA Abdelhamid

M.C. U. Batna

Examineur

Dédicace

A mes chers parents

A mes frères

A mes sœurs

A l'esprit de mon frère Farid

A mes Amis

Je dédie ce mémoire

Remerciements

Bien que cette thèse porte mon nom, elle est véritablement le fruit de la coopération de plusieurs personnes, qui m'ont encadré, supporté, guidé tout au long de ces mois. Ce sont ces personnes que je tiens ici à remercier.

*Mon directeur de thèse le professeur **Nour-Eddine BOUGUECHAL** le doyen de la faculté des sciences de l'ingénieur à l'université de Batna qui est été mon principal guide au cours de cette aventure. Je retiens du professeur BOUGUECHAL ses grandes qualités d'encadrement et plus généralement ses qualités humaines. Il a été une personne essentielle pour mon initiation et mon introduction au monde de la microélectronique.*

*Je tiens particulièrement à remercier **Mr. Otto Manck**, directeur général de la société MAZ, et professeur à l'université technique de Berlin en Allemagne, où nous avons effectué une grande partie de ce travail, pour la confiance qu'il m'a témoignée en m'accueillant dans son équipe à Berlin et pour ce captivant sujet de recherche proposé.*

*Mes vifs remerciements à **Mme. Farida HOBAR**, professeur à l'université de Constantine, pour avoir accepté de présider le jury et à bien voulu prendre le temps de s'intéresser au sujet du présent mémoire.*

*De même à **Mr. Zohir DIBI**, docteur et chef du département d'électronique de l'université de Batna, pour son aide, ces conseils ont été d'une grande valeur, aussi m'a-t-il honoré de faire partie du jury.*

*Je tiens à remercier vivement **Mme. Zoubida HAFDHI**, docteur et chargé de cours à l'institut d'électronique à l'université de Batna, pour m'avoir honoré par sa présence au jury.*

*Je remercie également **Mr Benhaya ABDELHAMID**, chargé de cours à l'institut d'électronique, pour son aide, pour l'honneur qu'il me fait en figurant dans la composition du jury.*

*Je remercie également **Mr Boris, Radouane Jeghader, et Hafiane Lamine** pour leurs conseils et leurs aides pendant le stage. Sans oublier les échanges très nombreux et très importants que j'ai eus avec mes collègues, **Souhil, A-Ghani, Hakim**. Nous avons commencé ensemble, travaillé ensemble et terminé « presque » ensemble.*

Enfin, je dois une dette certaine à ma famille et à mes parents, qui ont été mes plus fidèles supporteurs et qui m'ont aidé à traverser cette période.

Résumé

Les portes logiques dynamiques sont utilisés pour réduire la taille du layout, réduire la dissipation de puissance, augmenter la vitesse, et pour diminuer la complexité. L'idée fondamentale derrière la logique dynamique est d'utiliser l'entrée capacitive du transistor MOSFET pour stocker une charge et pour se rappeler ainsi un niveau logique pour un usage ultérieur. Le C²MOS est une famille logique combine la conception de la logique statique avec la synchronisation réalisée en employant des signaux d'horloge. Une machine d'états finis comporte une logique combinatoire et des registres. Le registre C²MOS est employé pour la conception d'une machine d'états dans la technologie 0.35μm, le registre C²MOS est un registre ingénieux basé sur le concept maître esclave, composé de deux latches C²MOS. Nous avons exploité l'outil Cadence Virtuoso pour concevoir le circuit de la machine d'états et pour dessiner leur layout .

Abstract

Dynamic logic gates are used to reduce layout size and power dissipation, to decrease complexity, and to increase speed. The basic idea behind dynamic logic is to use the capacitive input of the MOSFET to store a charge and thus remember a logic level for use later. Clocked CMOS is a logic family that combines static logic design with the synchronization achieved by using clock signals. Finite State Machine (FSM) consists of combinational logic and registers, the C²MOS register is an ingenious based on concept master-slave register, composed by two clocked dynamic latches, insensitive to clock overlap.

The C²MOS register, are used to design an FSM in 0.35μm CMOS technology, we exploit Cadence Virtuoso tools to design the state machine circuit, and to draw their full custom layout

SOMMAIRE

Introduction générale

Organisation de la thèse.....	3
-------------------------------	---

Chapitre I : Le transistor MOS

I.1. Introduction	5
I.2. La structure du transistor MOSFET	5
I.3. Le fonctionnement du transistor MOSFET	7
I.3.1. L'analyse qualitative	7
I.3.1.1. La création de la zone de la déplétion.....	7
I.3.1.2. La formation du canal	8
I.3.1.3. Le transistor MOSFET comme une résistance variable.....	9
I.3.1.4. Le pincement du canal	9
I.4. La dérivation des caractéristiques d'I/V	11
1.4.1. La densité de la charge du canal :	11
1.4.2. Le courant du drain.....	12
1.4.3. Les régions de triode et de saturation	14
1.4.4. La modulation de la longueur du Canal.....	15
1.4.5. L'effet du Body	16
1.5. Le transistor PMOS	17
1-6. La technologie CMOS.....	17
I.7. Conclusion.....	19

Chapitre II : Les circuits logiques CMOS

II.1. Introduction.....	21
II.2. Les portes logiques statiques.....	21
II.2.1. L'inverseur CMOS.....	22
II.2.1.1. Les caractéristiques de transfert.....	22
II.2.1.2. Le point de commutation	23
II.2.2. Les portes complémentaires CMOS	23
II.2.2.2. La porte NAND à deux entrées.....	25
II.2.2.3. La porte NOR à deux entrées.....	25
II.2.2.4. Le synthèse des portes complexes CMOS	26
II.2.3. La porte de transmission TG	26
II.3. Les portes logiques dynamiques	27
II.3.1. Le concept des circuits logiques dynamiques.....	27
II.3.2. Le principe fondamental de la logique dynamique.....	27
II.3.3. Les familles des circuits logiques dynamiques.....	28
II.3.3.1. La logique C ² MOS.....	29
II.3.3.2. La logique PE.....	29
II.3.3.3. La logique Domino	30
II.3.3.4. La logique NP	30
II.4. Les circuits logiques séquentiels.....	31
II.4.1. Les bascules	31
II.4.1.1. La bascule RS	31

II.4.1.2. Les bascules synchronisées.....	32
1. La bascule JK	33
2. La bascule D	33
3. La bascule JK maître esclave.....	34
II.4.2. Latches contre registres.....	35
II.4.2.1 Les registres dynamiques.....	36
1. Le registre dynamique de porte de transmission	36
2. Le registre dynamique C ² MOS	37
II.5. La machine d'états finies	38
II.5.1. Les modèles de la machine d'états finis (FSM) :	39
II.5.2. L'Analyse des circuits séquentiels.....	39
II.5.3. Le synthèse des circuits séquentiels.....	44
II.6. Conclusion	45

Chapitre III : Conception assistée par ordinateur des circuits numériques

III.1. Introduction.....	47
III.2. Les outils CAO de la microélectronique	48
III.2.1. Le langage VHDL.....	48
III.2.2. PSpice	49
III.2.3. Cadence.....	50
III.2.4. Les simulateurs numériques	50
III.2.5. Le simulateur ModelSim	50
III.3. Le flot de conception générique des circuits intégrés.....	51
III.3.1. Idée de conception	51
III.3.2. Description comportementale et simulation	52
III.3.2.1. Description comportementale.....	52
III.3.2.2. Description structurale.....	52
III.3.2.3. Description schématique.....	52
III.3.2.4. Simulation.....	53
III.3.3. Description Layout	53
III.3.3.1. Les cellules standards	53
III.3.3.2. Le compactage.....	53
III.3.4. LVS.....	54
III.3.4. DRC	54
III.3.6. Les fichiers CIF, GDS II.....	54
III.4. Les ASIC et les circuits logiques programmables.....	55
III.4.1. Les ASIC	55
1. A la demande.....	56
2. Prédifusés	56
3. Pré-caractérisés.....	57
4. A réseau structuré	57
III.4.2. Les circuits logiques programmables	58
1. SPLD	58
a. PROM.....	59
b. PLA	59
c. PAL	59
2. CPLD.....	60
2.1. FPGAs.....	60
III.5. Conclusion.....	61

Chapitre IV : Conception d'une machine d'états à base de latches C²MOS

IV.1. Introduction	63
IV.2. La plateforme Cadence Virtuoso	63
IV.2.1. Virtuoso Schematic Editor	64
IV.2.2. Virtuoso Spectre Circuit Simulator	64
IV.2.3. Virtuoso Layout Editor	64
IV.3. La conception du circuit de machine d'états	64
IV.3.1. La spécification	65
IV.3.2. Le diagramme des états	65
IV.3.3. La structure de la machine d'état	66
IV.3.3.1. Le bloc de la machine d'état	66
IV.3.3.2. La machine d'état	68
IV.3.4. Le fonctionnement du circuit de la machine d'états	69
IV.3.4.1. Le latche C ² MOS	69
IV.3.4.2 La porte logique complexe C ² MOS	69
IV.3.4.3. Le registre C ² MOS	70
IV.4. La simulation	71
IV.4.1. Le circuit de test bench	72
IV.4.2. Les résultats de la simulation des signaux de transition	73
IV.4.3. Les résultats de la simulation des états de la machines d'états	74
IV.4.4. Les résultats de la simulation du courant de fuite	75
IV.5. Le layout	75
IV.5.1. La conception physique des masques layout	76
IV.5.2. Le layout compact et les cellules standards	77
IV.5.3. Le dessin du layout Full-Custom utilisant Virtuoso-XL	78
IV.5.4. Design Rule Check (DRC)	78
IV.5.5. Layout Versus Schematic (LVS)	79
IV.6. Conclusion	80

Conclusion générale

Annexe A	83
Bibliographie	88

Introduction générale

Introduction générale

La technologie utilisée pour la production d'une puce VLSI moderne est en évolution depuis plus de quarante ans, et à bien des égards, elle évolue à un rythme exponentiel. Inventé en 1947, le transistor est à l'origine de la majorité des appareils électroniques, depuis la radio jusqu'à l'ordinateur. Aujourd'hui, il fait partie de presque toutes les technologies utilisées dans l'industrie, la médecine et l'information. Le transistor est un composant électronique formé par la jonction de cristaux de germanium, de silicium ou d'autres matériaux semblables (semi-conducteurs). Seul, le transistor peut amplifier l'énergie électrique et interrompre rapidement le passage du courant électrique à haute vitesse, tout en consommant lui-même une quantité moindre d'énergie.

L'interconnexion de plusieurs transistors, tous construits sur la même surface cristalline, permet de fabriquer de puissants circuits intégrés pouvant traiter de l'information codée électriquement. Ce sont ces circuits intégrés qui donnent leur puissance aux ordinateurs modernes, qu'il s'agisse d'un ordinateur personnel, d'un appareil électroménager ou d'un réseau de communication. À l'heure actuelle, il nous est possible d'intégrer plusieurs millions de transistors sur un seul morceau de cristal de silicium. Bien que le processus de fabrication d'un tel circuit VLSI soit remarquable en soi, les tâches liées à la conception du circuit et à la mise à l'essai de la puce manufacturée pour détecter les éventuels défauts. Il est donc essentiel, pour mieux connaître le monde de la microélectronique, de bien comprendre les concepts fondamentaux de la conception, de la mise en application et des essais des circuits VLSI.

Concevoir un circuit intégré ASIC, avec ses millions de transistors et ses interconnexions encore plus nombreuses, n'est pas une mince tâche. Aucune erreur ne peut être tolérée. Pour obtenir les résultats désirés, il est essentiel d'avoir accès à de puissants ordinateurs, appelés postes de travail, ainsi qu'à une panoplie de programmes d'automatisation de conception de circuits électroniques (CAO) fonctionnant sur ces postes de travail. Ces gros programmes peuvent générer automatiquement des circuits à partir de descriptions fonctionnelles axées sur le langage, simuler le comportement du circuit, fixer l'emplacement des transistors et le routage des éléments interconnectés à proximité de la puce et veiller à ce que tous les défauts possibles puissent être détectés durant les essais. L'utilisation de ces outils CAO se fait selon un processus de conception qu'on applique de façon rigoureuse afin de ne pas brûler les étapes. Pendant la conception, on peut consacrer de

nombreux mois à la simulation afin de déceler les erreurs, pour ensuite appliquer des mesures correctives et reprendre la simulation, mais cela augmente les chances de fabriquer dès le départ une puce qui fonctionnera correctement.

Les niveaux d'abstraction typiquement utilisés dans la conception des circuits ASIC, selon l'ordre de l'augmentation d'abstraction sont, le dispositif, le circuit, la porte, module fonctionnel (par exemple, additionneur) et le niveau du système (par exemple, processeur). Un dispositif de semi-conducteur est une entité avec un comportement très complexe. Aucun concepteur de circuit ne considérera toujours sérieusement les équations de physique de semi-conducteurs régissant le comportement du dispositif en concevant une porte numérique. Au lieu de cela il emploiera un modèle simplifié qui décrit en juste proportion le comportement d'entrée-sortie du transistor. Cette philosophie de conception a été favorisée par l'apparition de la conception assistée par ordinateur (CAO) des circuits intégrés numériques; sans elle la complexité de la conception courante n'était pas réalisable. Les outils de conception incluent la simulation aux divers niveaux de complexité, vérification de conception, génération de layout, et synthèse de conception.

Afin de simplifier la conception d'un circuit ASIC (une sonde ultrasonique), des méthodes de conception sont utilisées, telles que la division de la puce en plusieurs sous-circuits plus petits, la réutilisation de sous-circuits éprouvés et la reprise d'éléments de circuit pour former un sous-circuit plus complexe. L'un des sous-circuits de ce circuit ASIC, est une machine d'états. La conception du circuit de la machine d'états se base sur le concept du latch C^2MOS . Les circuits synchrones C^2MOS fonctionnent dans ce qui s'appelle le domaine de temps. Le temps dans un circuit synchrone est quantifié et discret. L'information est gênée pour couler librement par l'état tenant des éléments ce qui ont laissé l'information propager à travers seulement aux frontières d'un intervalle discret de temps.

Cette thèse a deux objectifs principaux. Le premier est de présenter au lecteur les principes de la conception des circuits numériques ASIC. Le deuxième est d'implémenter une méthodologie de CAO des circuits logiques CMOS pour concevoir une machine d'états, exploitant la plate forme Cadence Virtuoso.

Organisation de la thèse

Dans le chapitre I, on s'intéressera à l'état de l'art de transistor MOS. Ensuite, une étude de sa structure et son fonctionnement permet de comprendre les paramètres de base du transistor MOS qui sont indispensables dans la conception des circuits logiques CMOS. Le concept de base de la technologie CMOS est discuté à la fin de ce chapitre.

Le chapitre II, est consacré à la conception des portes logiques statiques et dynamiques. On présentera la conception des circuits séquentiels et en particulier le concept des circuits C²MOS, divers aspects de conception des bascules, latches et registres seront discutés. Enfin on conclura ce chapitre par une étude théorique des la machines d'états.

Le chapitre III décrira quelques CAO utilisés dans la conception des circuits logiques ASIC, ainsi il présentera le flot de conception générique du circuit numérique.

Enfin le dernier chapitre, illustre la structure de la machine d'états, son principe de fonctionnement, et les étapes de sa conception tel que la spécification, la simulation, le dessin des masques 'Layout'.

Chapitre I : Le transistor MOS

I.1. Introduction

Aujourd'hui le domaine de la microélectronique est dominé par un type de dispositif appelé transistor MOSFET. Conçu dans les années 30 mais réalisé dans les années 60, les transistors MOSFET (également appelés les dispositifs MOS) offrent les seules propriétés qui ont menés à la révolution de l'industrie de semi-conducteur. Cette révolution a abouti aux microprocesseurs ayant 100 millions de transistors, des puces de mémoire contenant des milliards de transistors, et des circuits de communication sophistiquées fournir une capacité énorme de traitement de signal. Dans ce chapitre nous présenterons l'état de l'art du transistor MOSFET, et nous analyserons la structure et le fonctionnement des MOSFETs.

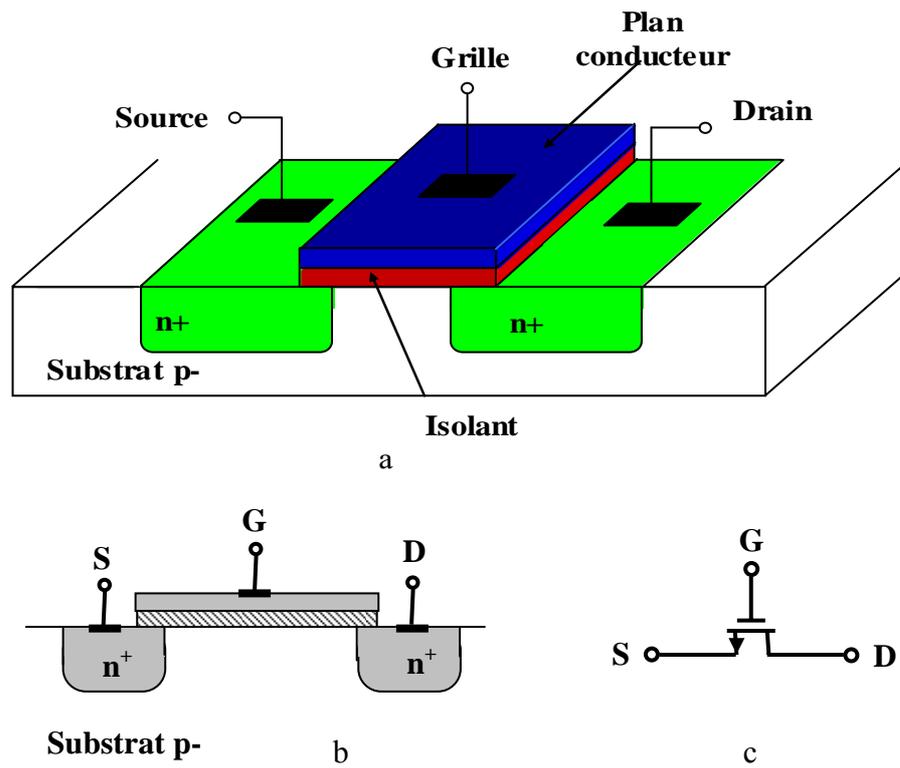
I.2. La structure du transistor MOSFET

La structure du transistor MOSFET es montrée dans la figure 1.1(a) .La plaque conductrice supérieure appelée la "grille" (G) réside sur une couche diélectrique mince (isolateur), qu'elle-même est déposée sur une autre couche sous jacente de silicium type p "substrat".

Pour permettre au courant de traverser le matériel de silicium, deux contacts sont attachés au substrat à travers deux régions n fortement dopés, parce que la connexion directe du métal au substrat ne produirait pas un bon contact "ohmique". Ces deux bornes s'appellent la "source" (S) et le "drain" (D) pour indiquer que le premier peut fournir des porteurs de charge et le dernier les absorbent.

La figure 1.1 (a) indique que le dispositif est symétrique par rapport S et D ; c.-à-d, dépendant aux tensions appliquées au dispositif, l'une ou l'autre de ces deux bornes peut drainer les porteurs de charge de l'autre. Comme sera expliqué dans la section 1.3, avec la source, le drain type n et le substrat type p, ce transistor fonctionne avec des électrons plutôt que des trous et s'appelle donc un dispositif MOS type n (NMOS).

Le type p (PMOS) contrepartie est étudié dans la section 1.5. Pour la simplicité nous dessinons le dispositif comme montré dans la figure. 1.1 (b). Le schéma 1.1 (c) représente le symbole de circuit pour un transistor NMOS, où la flèche signifie la borne source [1].



**Figure 1.1: La structure de transistor MOSFET(a), Une vue de côté (b),
Le symbole de circuit (c)**

Avant d'étudier le fonctionnement du transistor MOSFET, nous examinons le type de matières employés dans le dispositif. La grille doit servir de bon conducteur et a été réalisé par le métal (aluminium) dans les premières générations de la technologie MOS. Cependant, on l'a découvert que le silicium non cristallin "polysilicium" ou simplement "poly" avec un fort dopage (pour une faible résistivité) expose une meilleure fabrication et des excellentes propriétés physiques. Les transistors MOSFETs d'aujourd'hui utilisent des grilles de polysilicium. La couche diélectrique serrée entre la grille et le substrat joue un rôle critique dans le fonctionnement des transistors et elle est créée en accroissant le bioxyde de silicium (où simplement "oxyde") au dessus de la surface de silicium.

Les régions n^+ s'appellent parfois source et/ou drain de "diffusion," se rapportant à une méthode de fabrication employée aux jours têt de la microélectronique. Nous devrions également remarquer que ces régions forment ont faite des diodes avec le substrat de type p (figure. 1.2). Comme le fonctionnement approprié du transistor sera expliqué plus tard, il exige que ces jonctions demeurent polarisées en inverse. Ainsi, seulement la capacité de région de déplétion liée aux deux diodes doit être prise en considération. La figure 2.2 montre certaines dimensions de dispositif dans l'état de l'art d'aujourd'hui de la technologie MOS. L'épaisseur de l'oxyde est dénotée par t_{ox} .

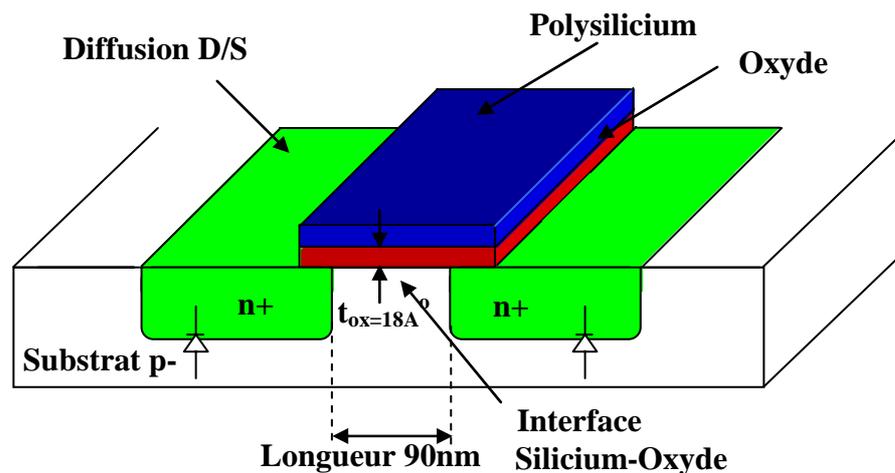


Figure 1.2 : La dimension typique des transistors MOSFET d'aujourd'hui

I.3. Le fonctionnement du transistor MOSFET

Cette section traite une multitude de concepts liés aux transistors MOSFET.

I.3.1. L'analyse qualitative

Notre étude des structures simples montrées dans les figures.1.1 suggère que le transistor MOSFET puisse conduire le courant entre la source et le drain si un canal des électrons est créé par l'application d'une tension de grille suffisamment positive. De plus, nous comptons que la magnitude du courant peut être commandée par la tension de la grille. Notre analyse confirmera en effet ces conjectures tout en indiquant d'autres effets subtils dans le dispositif. Noter que la borne grille ne passe aucun courant (de basse fréquence) comme elle est isolée du canal par l'oxyde [1].

I.3.1.1. La création de la zone de la déplétion

D'abord considérons l'arrangement montré dans figure. 1.3 (a), où la source et le drain sont à la masse et la tension de la grille est changée. Ce circuit ne semble pas particulièrement utile mais il nous donne beaucoup de perspicacité.

Comme V_G s'augmente, la charge positive sur la grille doit être reflétée par la charge négative dans le substrat. Tandis que les électrons sont attirés à une interface, en réalité, un autre phénomène précède la formation du canal. À mesure que V_G augmente de zéro, la charge positive sur la grille repousse les trous dans le substrat, ainsi exposant les ions négatifs et créant de ce fait une région déplétion figure. 1.3(b).

Noter que le dispositif agit toujours en tant qu'une charge condensateur charge positive sur la grille est reflétée par la charge négative dans le substrat mais aucun canal de charge mobile est créé encore. Ainsi, aucun courant ne peut découler de la source vers le drain. Nous disons que le transistor MOSFET est bloqué.

I.3.1.2. La formation du canal

Qu'est ce qui se passe quand V_G augmente? Pour refléter la charge sur la grille, beaucoup des ions négatifs sont exposés et la région de déplétion sous l'oxyde devient plus profonde. Est ce que ça veut dire que le transistor ne fonctionnera jamais?! Heureusement, si V_G devient suffisamment positif, des électrons libres sont attirés à l'interface d'oxyde de silicium, formant un canal conducteur figure. 1.3(c).

Nous disons que le transistor MOSFET est fonctionné. Le potentiel de la grille auquel le canal commence à apparaître s'appelle la "tension de seuil," V_{TH} , et tomber dans la gamme de 300 mV à 500 mV. Noter que les électrons sont aisément fournis par les régions n^+ de la source et de drain, et ils n'ont pas besoin d'être fournis par le substrat.

Il est intéressant de reconnaître, que la borne grille du transistor MOSFET ne passe aucun courant (de basse fréquence). La grille reposant sur l'oxyde, reste isolée des autres bornes et simplement fonctionne comme un condensateur plat.

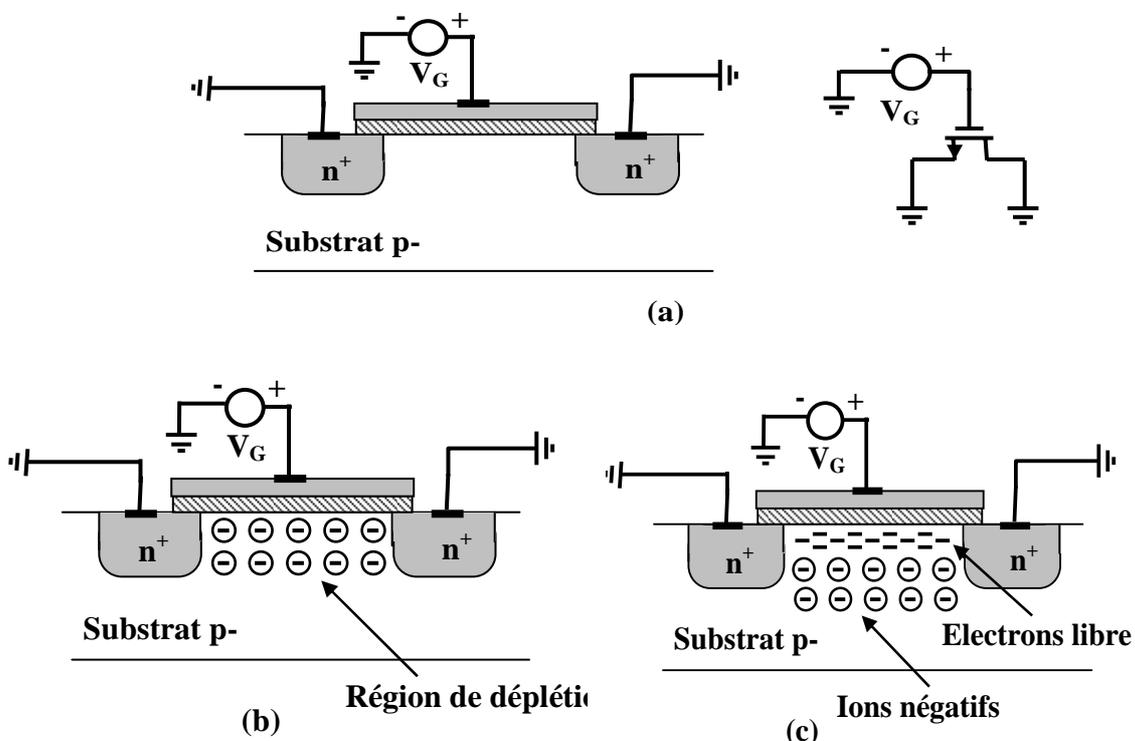


Figure 1.3 : Le transistor MOSFET avec la tension de la grille (a), La formation de la région de déplétion (b), La formation du canal (c).

I.3.1.3. Le transistor MOSFET comme une résistance variable

Le canal conducteur entre S et D peut être vue comme résistance. En outre, puisque la densité des électrons dans le canal doit augmenter pendant que V_G devient plus positif, la valeur de cette résistance change avec la tension de la grille. Conceptuellement illustré dans la figure. 1.4, tellement cette résistance dépendante de la tension, ce qui prouve son utilisation extrêmement dans les circuits analogiques et numériques.

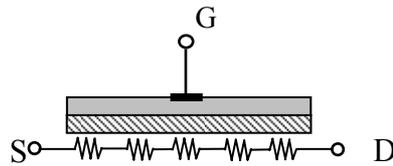


Figure 1.4 : Le transistor MOSFET vu comme résistance dépendante de la tension

I.3.1.4. Le pincement du canal

Notre étude qualitative du transistor MOSFET implique jusqu'ici que le dispositif agit en tant que résistance dépendante de la tension si la tension de grille excède V_{TH} . Cependant, en réalité le transistor fonctionne comme source de courante si la tension de drain est suffisamment positive. Pour comprendre cet effet, nous faisons deux observations :

- (1) pour former un canal, la différence de potentielle entre la grille et l'interface d'oxyde de silicium doit excéder V_{TH}

- (2) si la tension de drain reste plus élevée que la tension de source, puis la tension à chaque point le long du canal par rapport a la masse augmente quand nous allons de la source vers le drain. Illustré dans la figure. 1.5 (a), cet effet résulte de la chute graduelle de la tension le long de la résistance du canal. Puisque la tension de la grille est constante (parce que la grille est conductrice mais aucun courant ne passe dans n'importe quelle direction), et puisque le potentiel à l'interface d'oxyde du silicium monte de la source au drain, la différence de potentielle entre la grille et l'interface d'oxyde de silicium diminue le long de l'axe des abscisses figure. 1.5 (b). La densité des électrons dans le canal suit la même tendance, diminuant à un minimum à $x = L$.

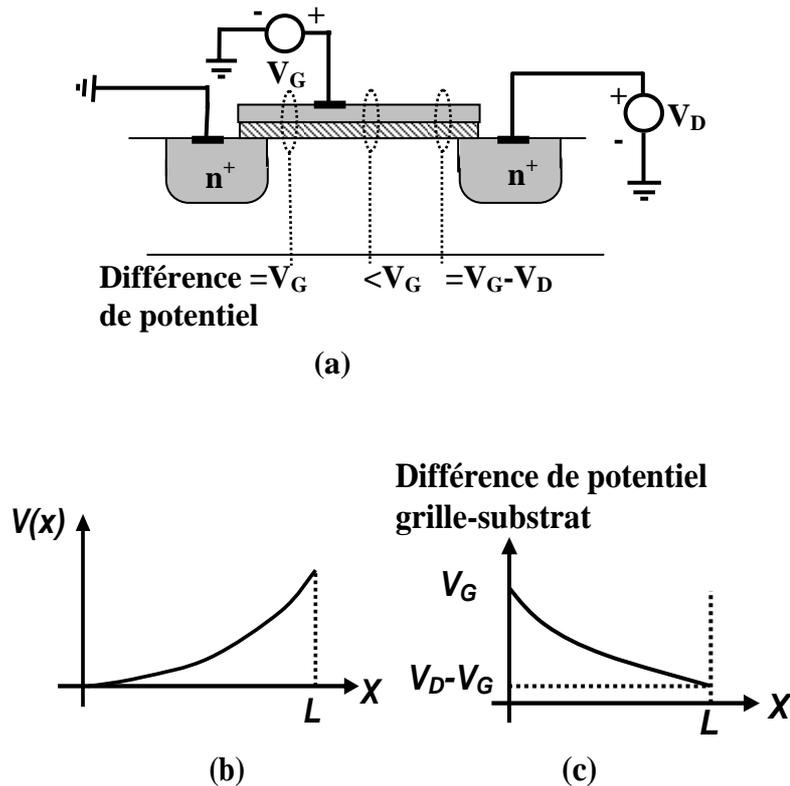


Figure 1.5 : La variation de potentielle du canal (a), La différence de tension de grille-substrat le long du canal (b)

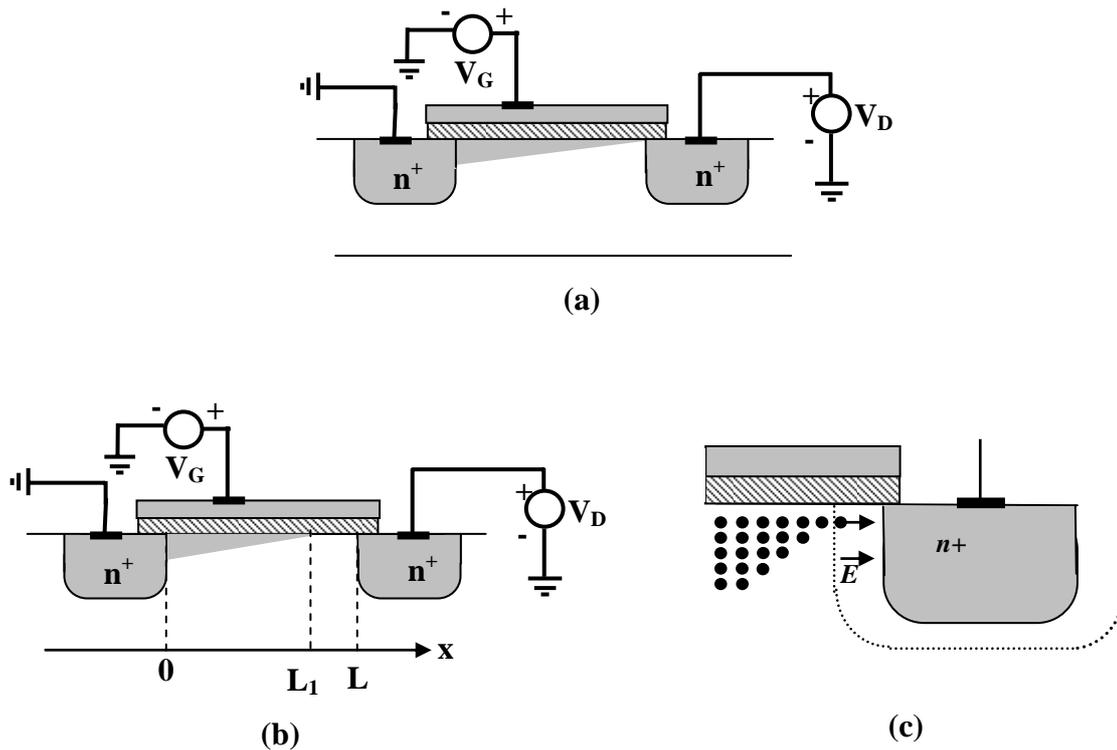
De ces observations, nous concluons cela, si la tension de drain doit être assez haute pour produire $V_G - V_D \leq V_{TH}$, alors le canal cesse d'exister près du drain. Nous disons que la différence du potentielle de grille - substrat n'est pas suffisante à $x = L$ pour attirer des électrons et le canal "est pincé " figure 1.6 (a).

Qu'est ce qui se produit si V_D s'augmente encore plus haut que $V_G - V_{TH}$?

Puisque $V(x)$ va maintenant de 0 en $x=0$ à $V_D > V_G - V_{TH}$ en $x=L$, la différence de tension entre la grille et le substrat chute à V_{TH} à un certain point $L_1 < L$ figure. 1.6 (b). Le dispositif ne contient donc aucun canal entre L_1 et L .

Et est ce que ça veut dire que le transistor ne peut pas conduire le courant ?

Non, le dispositif encore conduit, Comme illustré dans figure 1.6 (c), une fois que les électrons atteignent l'extrémité du canal, ils éprouvent le champ électrique élevé dans la région de déplétion, entourant la jonction de drain et ils sont rapidement balayés à la borne de drain. Néanmoins, comme sera montré dans la prochaine section, la tension de drain n'affecte plus le courant d'une manière significative, et le transistor MOSFET agit en tant que source de courant constant semblable à un transistor bipolaire dans la région active [2].



**Figure 1.6 : Le pincement (a), La variation de longueur avec la tension de drain(b),
Le fonctionnement détaillé près du drain (c)**

1.4. La dérivation des caractéristiques d'I/V

Avec l'étude qualitative antérieure, nous pouvons maintenant formuler le comportement des transistors MOSFET en termes de leurs tensions des bornes.

1.4.1. La densité de la charge du canal :

Nos dérivations exigent une expression pour la charge de canal (c.-à-d, les électrons libres) par unité de longueur, également appelée " la densité de charge " $Q=C.V$, nous notons que si C est la capacité de la grille par unité de longueur et V la différence de tension entre la grille et le canal, alors Q est la densité de la charge désirée. On dénote la capacité de la grille par unité de superficie C_{ox} (exprimé en F/m^2 ou fF/m^2), nous écrivons $C=W.C_{ox}$ pour considérer la largeur du transistor figure. 1.7. De plus nous avons $V=V_{GS}-V_{TH}$ parce qu'aucune charge mobile n'existe pour $V_{GS}<V_{TH}$, (Ci-après, nous dénotons les deux tensions de la grille et du drain par rapport à la tension de la source), elle suit cela :

$$Q = W. C_{ox}(V_{GS} - V_{TH}) \quad (1.1)$$

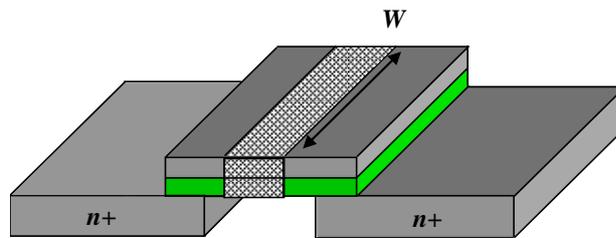


Figure 1.7 : Illustration de la capacité par unité de longueur.

La tension du canal se varie le long du transistor, et la densité de charge diminue allant de la source vers le drain. Ainsi, l'Eq. (1.1) est valide seulement près de la borne de source, où le potentiel de canal demeure de près de zéro. Comme il est montré dans fig. 1.8, nous dénotons le potentiel du canal à x par $V(x)$ et nous écrivons :

$$Q(x) = W \cdot C_{ox} \cdot [V_{GS} - V(x) - V_{TH}] \quad (1.2)$$

Notons que $V(x)$ va de zéro à V_D si le canal n'est pas pincé.

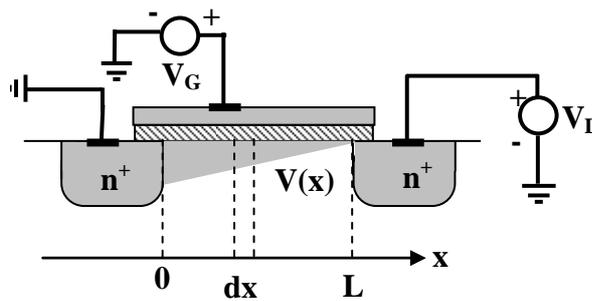


Figure 1.8 : Le dispositif qui illustre le calcul de courant de drain

1.4.2. Le courant du drain

Quelle est la relation entre la densité de charge mobile et le courant ?

Considérer une barre du semi-conducteur ayant un égal uniforme de densité de charge (par unité de longueur) égal à Q et portant un courant I figure. 1.9.

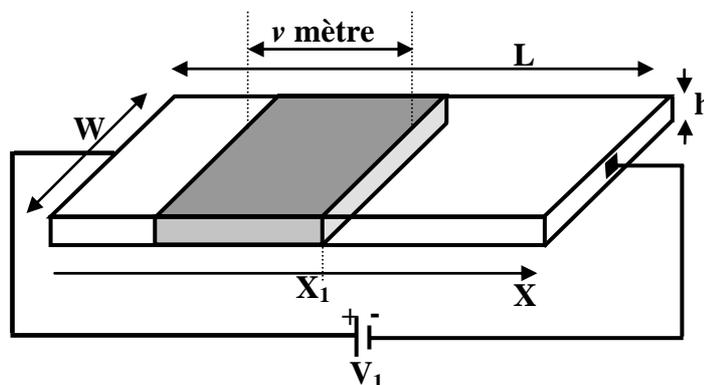


Figure 1.9 : La relation entre la densité de charge et le courant

Noter que :

1 - I est donné par toute la charge qui traverse une section de la barre dans une seconde.

2 - et si les porteurs se déplacent avec une vitesse de v m/s, alors la charge incluse dans v mètres le long de la barre, et traverse le secteur dans une second. Puisque la charge incluse dans v mètres est égal à $Q.v$, nous avons :

$$I = Q.v \quad (1.3)$$

Et on a

$$V = -\mu_n E; \quad (1.4)$$

$$V = +\mu_n \frac{dV}{dx} \quad (1.5)$$

Où dV/dx dénote la dérivée de la tension à un point donné. Nous combinons (1.2), (1.3), et (1.5), nous obtenons

$$I_D = WC_{ox}[(V_{GS} - V(x) - V_{TH})\mu_n \frac{dV(x)}{dx}] \quad (1.6)$$

Puisque I_D doit demeurer constant le long du canal, $V(x)$ et $dV(x)/dx$ doit varier tels que le produit $V_{GS}-V(x)-V_{TH}$ et dV/dx est indépendant de x . Tandis qu'il est possible de résoudre l'équation ci-dessus pour obtenir $V(x)$ en fonction de I_D , notre besoin immédiat doit trouver une expression pour I_D en fonction des tensions des bornes. À cet effet, nous écrivons :

$$\int_{x=0}^{x=L} I_D dx = \int_{V(x)=0}^{V(x)=V_{DS}} \mu_n C_{ox} W [V_{GS} - V(x) - V_{TH}] dV \quad (1.7)$$

C'est

$$I_D = \frac{1}{2} \mu_n C_{ox} \frac{W}{L} [2(V_{GS} - V_{TH})V_{DS} - V_{DS}^2] \quad (1.8)$$

Nous examinons maintenant cette équation importante de différentes perspectives pour gagner plus de perspicacité. D'abord, la dépendance linéaire de I_D sur le μ_n , le C_{ox} , et le W/L doit être prévue : une mobilité plus élevée rapporte a un plus grand courant pour une tension donnée de drain -source ; une capacité plus élevée Grille Oxyde mène à un plus grand densité d'électron dans le canal pour une tension donnée de grille -source ; et un plus grand W/L (appelé le dispositif à aspect de rapport) est équivalent à placer plus de transistors en parallèle.

En second lieu, pour un V_{GS} constant, I_D se varie paraboliquement avec V_{DS} figure. 1.10, et atteignant un maximum de :

$$I_{D,max} = \frac{1}{2} \mu_n C_{ox} \frac{W}{L} (V_{GS} - V_{TH})^2 \quad (1.9)$$

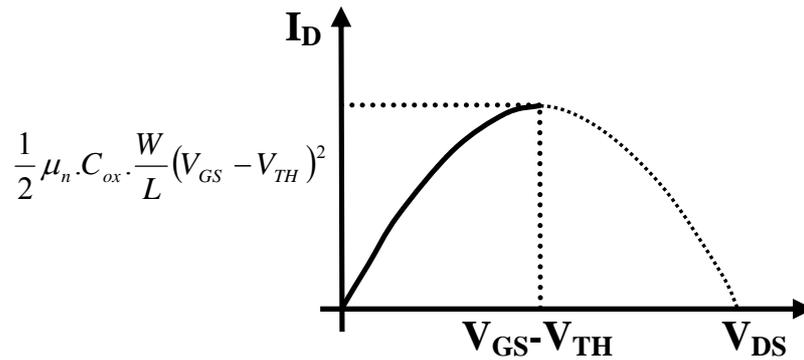


Figure 1.10 : Caractéristique parabolique de I_D - V_{DS} .

À $V_{DS}=V_{GS}-V_{TH}$. Il est commun à l'écriture W/L comme le rapport des deux valeurs par exemple, $5\mu\text{m} / 0.18\mu\text{m}$ (plutôt que 27.8) pour insister l'importance de choix de W et de L . Tandis que seulement le rapport apparaît dans beaucoup d'équations de MOS, la valeur individuelle de W et L devient également critique dans la plupart des cas. Par exemple, si les deux valeurs de W et le L sont doublées, le rapport demeure sans changement mais les augmentations de capacité de grille.

La relation non linéaire entre I_D et le V_{DS} indique que le transistor ne peut pas généralement être modélisé comme résistance linéaire simple.

Cependant, si $V_{DS} \ll 2(V_{GS}-V_{TH})$, l'Eq. (1.8) est réduite à :

$$I_D \approx \frac{1}{2} \mu_n C_{ox} \frac{W}{L} (V_{GS} - V_{TH}) V_{DS} \quad (1.10)$$

I_D - V_{DS} s'expose un comportement linéaire pour un V_{GS} donné. En effet, la résistance équivalente est donnée par V_{DS}/I_D :

$$R_{on} = \frac{1}{\frac{1}{2} \mu_n C_{ox} \frac{W}{L} (V_{GS} - V_{TH})} \quad (1.11)$$

1.4.3. Les régions de triode et de saturation

L'équation (1.8) exprime le courant de drain en fonction des tensions des bornes de dispositif, impliquant que le courant commence à chuter pour $V_{DS} > V_{GS} - V_{TH}$. Nous disons que le dispositif fonctionne dans "la région de triode" si $V_{DS} < V_{GS} - V_{TH}$ (la section qui augmente de la parabole). Nous employons également le terme "région de triode profonde" pour le $V_{DS} \ll 2(V_{GS} - V_{TH})$, où le transistor fonctionne comme une résistance [2].

En réalité, le courant de drain atteint, "la saturation " c'est-à-dire, devient constant pour $V_{DS} > V_{GS} - V_{TH}$ figure. 1.11.

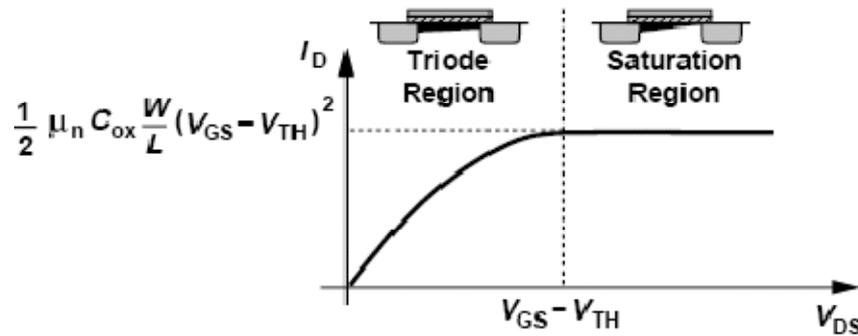


Figure 1.11 : Caractéristique globale du MOS.

Pour comprendre pourquoi, rappeler de figure 1.6 que le canal éprouve le pincement si $V_{DS} = V_{GS} - V_{TH}$. Ainsi, de plus un accroissement de V_{DS} décale simplement le point de pincement légèrement vers le drain. En outre, rappeler que les Eqs. (1.6) et (1.7) sont valides seulement où la charge du canal s'existe.

Elle confirme que l'intégration dans (1.7) doit entourer seulement le canal, c.-à-d., de $x=0$ à $x=L_1$ dans la figure 2.6 (b), et soit modifiée selon

$$\int_{x=0}^{x=L_1} I_D dx = \int_{V(x)=0}^{V(x)=V_{GS}-V_{TH}} \mu_n C_{ox} W [V_{GS} - V(x) - V_{TH}] dV \quad (1.12)$$

Noter que les limites supérieures correspondent au point de pincement du canal. En particulier, l'intégrale du côté droit est évaluée jusqu'à $V_{GS} - V_{TH}$ plutôt qu'à V_{DS} .

En conséquence,

$$I_D = \frac{1}{2} \mu_n C_{ox} \frac{W}{L} (V_{GS} - V_{TH})^2 \quad (1.13)$$

Un résultat indépendant de V_{DS} et identique à $I_{D, \max}$ dans (1.9) si nous assumons $L_1 \approx L$. la quantité $V_{GS} - V_{TH}$ appelé la "tension de overdrive," joue un rôle principal dans les circuits MOS. Des transistors MOSFET s'appellent parfois les dispositifs "loi carrée" pour souligner le rapport entre I_D et l'overdrive. Pour la brièveté, nous dénotons ci-après L_1 avec L .

1.4.4. La modulation de la longueur du Canal

Dans notre étude de l'effet de pincement, nous avons observé que le point auquel le canal disparaît quand on fait des mouvements vers la source à mesure que la tension de drain augmente. En d'autres termes, la valeur de L_1 dans la figure. 1.6 (b) change avec V_{DS} dans une certaine mesure. Ce phénomène appelée la "modulation de la longueur du canal" est illustré dans fig. 1.12, il rapporte un plus grand courant de drain à mesure que V_{DS} augmente parce que $I_D \propto 1/L_1$ dans Eq. (1.13). Semblable à l'effet Early dans les éléments bipolaires, la modulation de la longueur de canal a comme conséquence est une impédance finie de sortie donnée par l'inverse de la pente I_D - V_{DS} dans fig. 1.12.

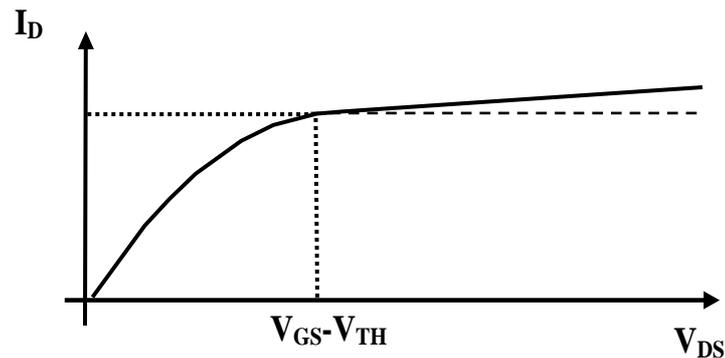


Figure 1.12 : La variation d' I_D dans la région de saturation.

Pour expliquer la modulation de la longueur du canal, nous supposons que L est constant, mais multiplier le membre droite de l'eq. (1.13) par une limite corrective:

$$I_D = \frac{1}{2} \mu_n C_{ox} \frac{W}{L} (V_{GS} - V_{TH})^2 (1 + \lambda V_{DS}) \quad (1.14)$$

Où λ s'appelle "coefficient de modulation de longueur de canal" c'est seulement une approximation, cette dépendance linéaire de I_D sur V_{DS} fournit toujours beaucoup de perspicacité sur l'implication de la modulation de longueur de canal dans la conception des circuits [1].

1.4.5. L'effet du body

Dans notre étude des transistors MOSFET, nous avons supposé que la source et le substrat (également appelé le "bulk" ou le "body") sont attachés à la masse. Cependant, cette condition n'a pas besoin de la tenir dans tous les circuits. Par exemple, si la borne de source est montée à une tension positive tandis que le substrat est toujours à zéro, donc la jonction de source substrat reste polarisée en inverse, et le dispositif encore fonctionne correctement [1].

Le schéma 1.13 illustre ce cas. La borne de source est attachée à un potentiel V_S par rapport à la masse tandis que le substrat est mettre à la masse, par un contact p^+ . La ligne tirée supplémentaire, ou le symbole de transistor indique la borne de substrat. Nous dénotons la différence de tension entre la source et le substrat (bulk) par V_{SB} .

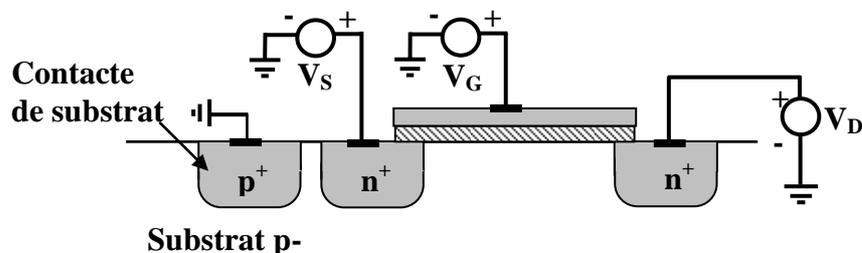


Figure 1.13 : L'effet de body

Un phénomène intéressant se produit pendant que la différence de potentielle de source substrat part de zéro : la tension de seuil du dispositif change. En particulier, comme la source

devient plus positive par rapport au substrat, V_{TH} augmente. Ce phénomène appelé "L'effet du corps," est formulé comme suite :

$$V_{TH} = V_{TH0} + \gamma(\sqrt{|2\phi_F + V_{SB}|} - \sqrt{|2\phi_F|}) \quad \text{I.15}$$

1.5. Le transistor PMOS

Comme dans les transistors bipolaires, il y a le NPN et le PNP, on peut se demander si un type p contre parti existe pour les transistors MOSFET. En effet, comme illustré dans la figure. 1.14 (a), le changement des polarités de dopage du substrat et des régions S/D avoir un résultat, d'un dispositif "PMOS". Le canal maintenant se compose des trous et est formé si la tension de la grille est au-dessous du potentiel de la source par une tension de seuil. C'est-à-dire, pour que le dispositif conduit, $V_{GS} < V_{TH}$, où V_{TH} lui-même est négatif. Nous dessinons le dispositif de PMOS comme dans figure. 1.14 (b), avec la borne de la source identifiée par la flèche et placé au dessus pour souligner son potentiel plus élevé. Le transistor fonctionne dans la région de triode si la tension de drain est près du potentielle de la source, il s'approche de la saturation quand V_D chute à $V_G - V_{TH} = V_G - |V_{TH}|$

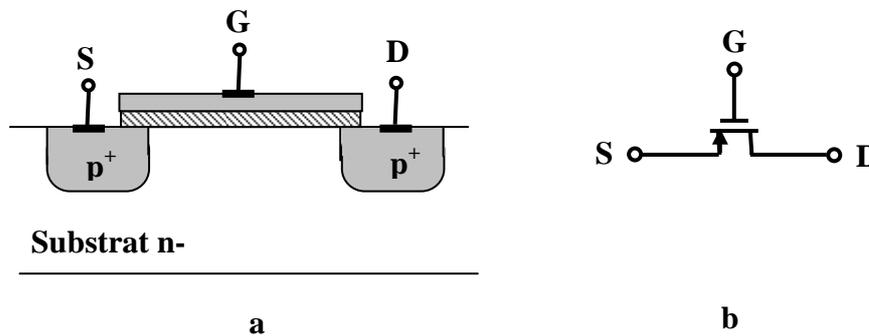


Figure 1.14 : (a) La structure de dispositif PMOS, (b) Le symbole de circuit

1-6. La technologie CMOS

La technologie CMOS est une technologie planaire destinée au développement des systèmes à très haute échelle d'intégration (VLSI). Grâce aux propriétés des transistors MOS complémentaires (notées CMOS pour Complementary Metal Oxyde Silicon), cette technologie permet de réaliser des circuits à faible coût et à basse consommation. Cet avantage à lui permis d'être reconnue comme la technologie de pointe la plus avancée et la plus maîtrisée dans le domaine de la micro-électronique.

Est-il possible d'établir des dispositifs NMOS et PMOS sur le même dispositif ?

Les figures 1.1 (a) et 1.14 (a) indiquent que les deux dispositifs exigent des différents types de substrat. Heureusement, un substrat local type n peut être créé dans un substrat type p, de ce fait adoptant des transistors PMOS, comme il est illustré dans la figure. 1.15, un "n well" enferme un dispositif PMOS tandis que le transistor NMOS réside dans le substrat p.

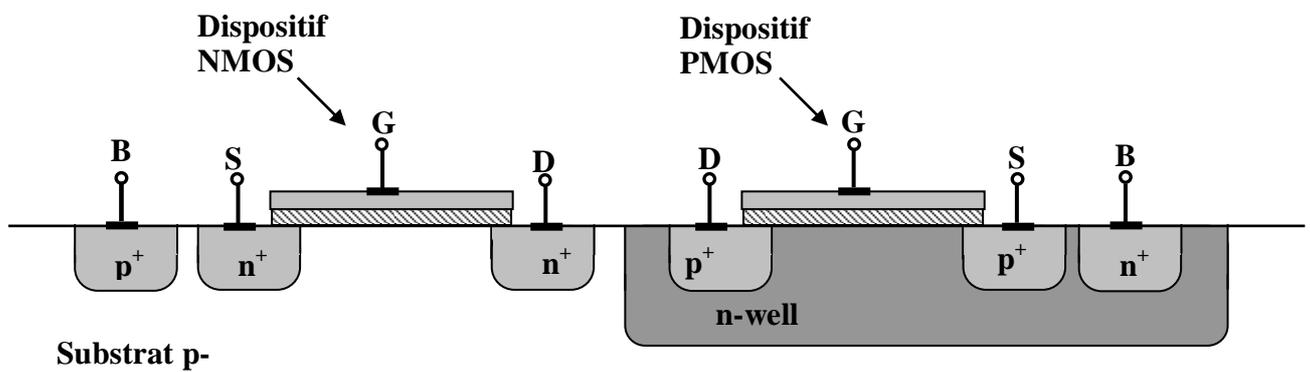


Figure 1.15 : technologie CMOS

La structure ci-dessus appelée "La technologie" CMOS " (complementary MOS), exige plus traitement complexe que les dispositifs simples NMOS ou PMOS. En effet, les premières générations de la technologie MOS a contenu seulement des transistors NMOS, et le coût plus élevé de processus CMOS a semblé prohibitif. Cependant, beaucoup d'avantages significatifs des dispositifs complémentaires par la suite rendre la technologie CMOS dominante et la technologie NMOS dépassé.

I.7. Conclusion

Une source du courant dépendante de la tension peut former un amplificateur avec une résistance de charge.

Les transistors MOSFETs sont des dispositifs électroniques qui peuvent fonctionner en tant que des sources du courant dépendantes de la tension.

Le transistor MOSFET se compose d'une plaque conductrice (la "grille ") placé sur un substrat semi-conducteur, et de deux jonctions ("source" et "drain") placés dans le substrat. La grille contrôle le flux du courant de la source au drain. La grille passe un courant presque zéro parce qu'une couche d'isolation la sépare du substrat.

Au delà d'une certaine tension de grille-source (la "tension de seuil"), des porteurs mobiles sont attirés à l'interface d'oxyde-silicium et à un canal est formé.

Si la tension de drain-source est petite, le dispositif fonctionne comme une résistance dépendante d'une tension.

Quand la tension de drain monte, la densité de charge près du drain chute. Si la tension de drain atteint un seuil supérieur à la tension de grille, le canal cesse d'exister près du drain, menant au "pincement ".

Le transistor MOSFET fonctionne dans la région "triode" si la tension de drain est supérieure à un seuil de la tension de grille. Dans cette région, le courant de drain est une fonction de V_{GS} et V_{DS} . le courant est également proportionnel au rapport hauteur/largeur de dispositif, W/L .

Le transistor MOSFET entre la "région de saturation" si le pincement du canal se produit, c.-à-d., la tension du drain est inférieur à un seuil du voltage de grille. Dans cette région, le courant de drain est proportionnel à $(V_{GS} - V_{TH})^2$.

Les transistors MOSFET fonctionnant dans la région de saturation se comportent comme des sources de courant et trouvent une large application dans les circuits microélectroniques.

Quand la tension de drain excède $V_{GS} - V_{TH}$ et le pincement se produit, l'extrémité de drain du canal commence à se déplacer vers la source, réduisant la longueur utile du dispositif. Appelé la "modulation de longueur du canal" cet effet mène à la variation du courant de drain dans la région de saturation. C'est-à-dire, le dispositif n'est pas une source de courant idéale.

Les transistors NMOS et PMOS sont fabriqués sur le même substrat pour créer la technologie CMOS.

Chapitre II : Les circuits logiques CMOS

II.1. Introduction

Il est pratiquement impossible de trouver des dispositifs électroniques dans nos vies quotidiennes qui ne contiennent pas des circuits numériques. Des montres et des appareils photo aux ordinateurs et aux téléphones cellulaires, les circuits numériques représentent plus de 80% du marché de semi-conducteur. Les exemples incluent des microprocesseurs, mémoires, et CI de traitement de signal numérique.

Ce chapitre sert d'introduction à l'analyse et à la conception des circuits numériques CMOS. L'objectif est de fournir une compréhension détaillée du niveau transistor des portes logiques, afin de préparer sur la conception des circuits numériques.

La conception logique de base des systèmes numériques utilise des modules tels que des portes, latches, et des bascules. Les portes logiques peuvent former un circuit "combinatoire" qui fonctionne comme un décodeur binaire Gray, de même, les portes et les bascules et peuvent comporter un circuit "séquentiel" qui sert comme compteur ou "machine d'état fini." Dans ce chapitre, nous étudions la conception interne de certains de ces modules et analyser leurs limitations.

II.2. Les portes logiques statiques

Le style logique le plus largement utilisé est le style statique complémentaire CMOS; il est vraiment une extension de l'inverseur statique CMOS à des entrées multiples.

Le style des circuits MOS complémentaire tombe sous une large classe des circuits logiques appelés circuits statiques dont les quels à chaque point à temps (excepté pendant la commutation transitoire), chaque sortie de porte est relié à V_{DD} ou à V_{SS} par l'intermédiaire d'un chemin à faible résistance. En outre, les sorties des portes assument à tout moment la valeur de la fonction booléenne implémentée par le circuit (ignorant, de nouveau, les effets transitoire pendant des périodes de commutation). C'est contrairement à la classe des circuits dynamiques, qui comptent sur la mémoire temporaire des valeurs du signal mémorisé dans la capacité des nœuds à grande impédance du circuit. La dernière approche a l'avantage que la porte résultante est plus simple et plus rapide. Mais leur conception et fonctionnement sont cependant plus impliqués et enclins à l'échec a cause une sensibilité accrue du bruit.

II.2.1. L'inverseur CMOS

L'inverseur est vraiment le noyau de toutes les conceptions numériques. Une fois que leur opération et leurs propriétés sont clairement comprises, on peut concevoir des structures plus complexes telles que des portes NAND et, des additionneurs, des multiplicateurs etc. L'analyse des inverseurs peut être prolongée pour expliquer le comportement des portes plus complexes telles que le NAND, et le NOR, ou les XOR, qui forment à leurs tours des modules qui construisent d'autres modules comme des multiplicateurs et des processeurs.

Considérer le circuit représenté sur la figure 2.1, la tension d'entrée V_{IN} commande les modes de conduction des deux transistors. Quand $V_{IN}=0$ le PMOS conduit la tension V_{DD} à la sortie. Quand $V_{IN}=V_{DD}$ le NMOS est passant, et transmet la masse GND (0V) à la sortie [3].

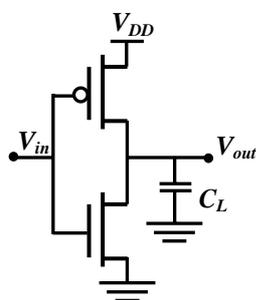


Figure 2.1 : L'inverseur CMOS

II.2.1.1. Les caractéristiques de transfert de l'inverseur en D.C.

Considérer l'inverseur montré dans figure 2.2 et la courbe caractéristique de transfert associée. Dans la région 1 des caractéristiques de transfert, la tension d'entrée est suffisamment basse (typiquement moins que la tension de seuil de M1), de sorte que M1 soit bloquer et M2 est fermé ($V_{GS} \gg V_{TH}$). À mesure que V_{in} est augmenté, M2 et M1 sont passant (région 2). V_{in} croissant cause plus loin le M2 d'être passant et M1 à entièrement est bloqué, comme montré dans la région 3.

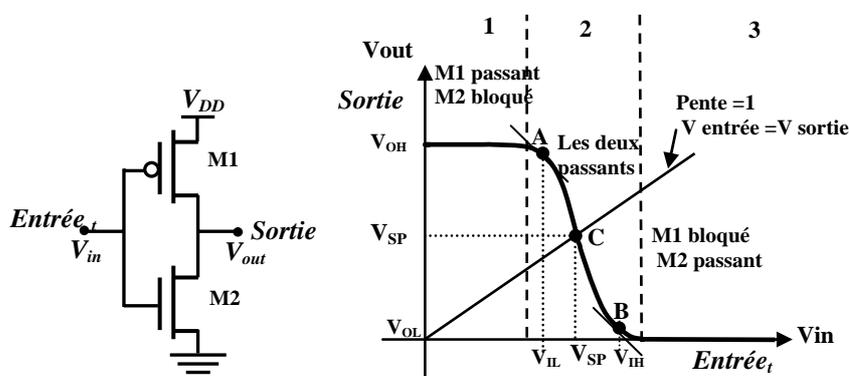


Figure 2.2 : Les caractéristiques de transfert de l'inverseur CMOS

La tension de sortie "élevée" maximum est marquée par V_{OH} , et la tension de sortie "basse" minimum, V_{OL} . Les points A et B sur cette courbe sont définis par la pente des courbes de transfert égalant -1. Les tensions d'entrée inférieures ou égales à la tension V_{IL} , sont définies par le point A, sont considérées comme un niveau logique bas sur l'entrée de l'inverseur. Des tensions d'entrée supérieures ou égales à la tension V_{IH} , définies par le point B, sont considérées comme un niveau logique haut sur l'entrée de l'inverseur. Les tensions d'entrée entre V_{IL} et V_{IH} , ne définissent pas une tension de niveau logique valide. Dans le meilleur des cas, la différence entre V_{IL} et V_{IH} , est zéro; cependant, ce n'est jamais le cas dans les circuits logiques réels.

II.2.1.2. Le point de commutation de l'inverseur

Considérer les caractéristiques de transfert de l'inverseur de base montré dans la figure. 2.2. Le point C correspond au point sur la courbe, où la tension d'entrée est égale à la tension de sortie. En ce moment, la tension d'entrée (ou de sortie) s'appelle la tension de point de commutation d'inverseur, V_{SP} , et les deux transistors MOSFET dans l'inverseur sont dans la région de saturation. Puisque le courant de drain dans chaque transistor MOSFET doit être égal [3] :

$$\frac{B_n}{2} (V_{SP} - V_{THN})^2 = \frac{B_p}{2} (V_{DD} - V_{SP} - V_{THP})^2 \quad (2.1)$$

La solution pour le V_{SP} donne

$$V_{SP} = \frac{\sqrt{\frac{B_n}{B_p}} V_{THN} + (V_{DD} - V_{THP})}{1 + \sqrt{\frac{B_n}{B_p}}} \quad (2.2)$$

II.2.2. Les portes complémentaires CMOS

II.2.2.1. Le concept

une porte statique CMOS est une combinaison des deux réseaux, Le réseau de dessus est construit à l'aide des transistors NMOS, alors que des transistors PMOS sont utilisés dans le réseau de dessous, La raison primaire de ce choix est que les transistors NMOS produisent "les zéros forts," et les transistors PMOS produisent "les uns forts". La figure 2.3 montre une porte logique générique de N entrées où toutes les entrées sont distribuées aux deux réseaux PMOS et NMOS. La fonction du réseau PMOS est fournir une connexion entre la sortie et le V_{DD} n'importe quand la sortie de la porte logique est censée pour être 1 (basé sur les entrées). De même, la fonction du réseau NMOS est relier la sortie au V_{SS} quand la sortie de la porte logique est censé pour être 0. Les deux réseaux PMOS et NMOS sont construits d'une

mode mutuellement exclusif tels qu'une et seulement un des deux réseaux conduit dans l'état d'équilibre. De cette façon, une fois que les coupures ont été arrangées, un chemin existe toujours entre V_{DD} et la sortie F , réalisant une sortie au niveau haut élevé ("un"), ou, alternativement, entre le V_{SS} et le F pour une sortie au niveau bas ("zéro"). C'est équivalent à énoncé que le nœud de sortie est toujours un nœud de basse impédance dans l'état d'équilibre.

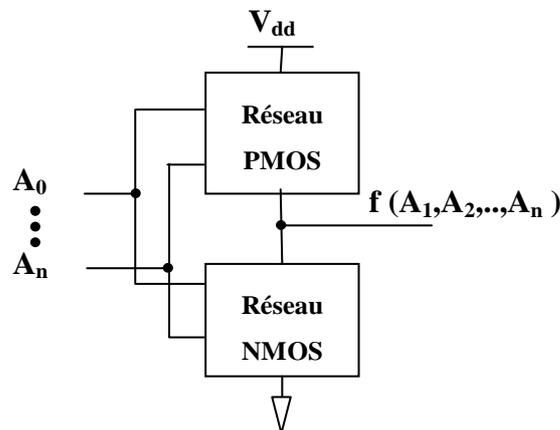


Figure 2.3 : La structure d'une porte complémentaire MOS

Un transistor peut être considéré comme un commutateur commandé par son signal de grille. Le commutateur NMOS est fermé quand le signal de commande est haut et est ouvert quand le signal de commande est bas. Un transistor PMOS agit en tant que commutateur inversé, qui est fermé quand le signal de commande est bas et ouvert quand le signal de commande est haut.

Un ensemble des règles de construction peut être dérivé pour construire des fonctions logiques (le schéma 2.3). Les dispositifs NMOS reliés en série correspondent à la fonction AND. Avec toutes les entrées au niveau haut, la combinaison série conduit et la valeur à une extrémité de la chaîne est transférée à l'autre extrémité. De même, les transistors NMOS se sont reliés en parallèle représentent une fonction OR. Un chemin de conduction existe entre la sortie et la borne d'entrée si au moins une des entrées est au niveau haut. Employer les arguments semblables, les règles de construction pour les réseaux PMOS peuvent être formulés. Une connexion des PMOS en série conduit si les deux entrées sont au niveau bas, représentant une fonction NOR ($\bar{A} \cdot \bar{B} = \overline{A + B}$), tandis que les transistors PMOS sont montés en parallèle pour implémenter une fonction NAND ($\overline{A + B} = \bar{A} \cdot \bar{B}$).

Utilisant les théorèmes de Morgan ($\overline{\overline{A + B}} = \bar{A} \cdot \bar{B}$ et $\overline{\bar{A} \cdot \bar{B}} = A + B$), on peut montrer que le réseau des PMOS et le réseau des NMOS déroulés d'une structure complémentaire CMOS ; se sont des réseaux duels. Ceci signifie qu'une connexion parallèle des transistors dans le réseau PMOS correspond à une connexion série des dispositifs NMOS.

II.2.2.2. La porte NAND à deux entrées

Le schéma 2.4 (a) montre une porte NAND à deux entrée ($F = \overline{A \cdot B}$). Le réseau de dessous se compose de deux transistors NMOS en série qui conduisent quand A et B sont hauts. Le réseau du dessus, et se compose de deux transistors PMOS en parallèles. Ceci signifie que F est 1 si au moins une des entrées est basse $A = 0$ ou $B = 0$, ce qu'est équivalent à $F = \overline{A \cdot B}$ La table de vérité pour la porte NAND à deux simples entrées est donnée dans le tableau 2.4 (b). on peut vérifier que la sortie F est toujours mise à V_{DD} ou à GND, mais jamais à tous les deux en même temps [4].

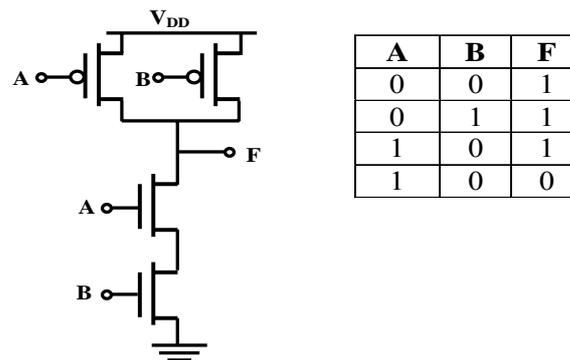


Figure 2.4 : La porte NAND

II.2.2.3. La porte NOR à deux entrées

L'implémentation CMOS de porte NOR ($F = \overline{A + B}$) est montré sur le schéma 2.5(a). Le réseau de dessus se compose de deux transistors NMOS en parallèles et le réseau du dessous, et se compose de deux transistors PMOS en série. La sortie est haut, si et seulement si les entrées A et B sont bas. Par contre la sortie F est bas quand seulement A ou B est haut, ce qu'est équivalent à $F = \overline{A + B}$. La table de vérité pour la porte NAND à deux simples entrées est donnée dans le tableau 2.5(b)

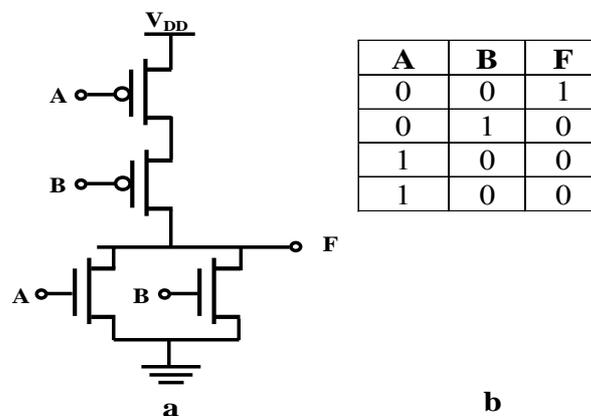


Figure 2.5 : La porte NOR

II.2.2.4. Le synthèse des portes complexes CMOS

On utilisant la logique complémentaire MOS, considérer la synthèse d'une porte complexe CMOS dont la fonction est $F = \overline{D + A \cdot (B + C)}$. La première étape dans la synthèse de la porte logique est de dériver un réseau de dessous comme représenté sur la figure 2.6. (a) en utilisant les transistors NMOS en série, pour implémenter la fonction AND et les transistors PMOS en parallèle pour implémenter la fonction OR. La prochaine étape est d'utiliser la dualité pour dériver le réseau dessus dans un mode hiérarchique. Le réseau NMOS est coupé en des plus petits sous réseau (sous-ensemble du réseau qui simplifient la dérivation du réseau [4]). Dans la figure 2.6 (b), les sous réseau (SN) pour le réseau dessous sont identifiés au plus haut niveau, SN1 et SN2 sont en parallèle ainsi dans le réseau complémentaire, ils seront en série. puisque SN1 se compose d'un transistor simple, il est tracé directement dans le réseau dessus. D'autre part, on a besoin d'appliquer de façon récursive la règle de dualité à SN2. A l'intérieur de SN2, nous avons SN3 et SN4 en série ainsi dans le réseau de dessous ils apparaîtront en parallèle. En conclusion, à l'intérieur de SN3, les dispositifs sont en parallèle ainsi ils apparaissent en série dans le réseau dessus. La porte complète est montrée dans la figure 2.6 (c).

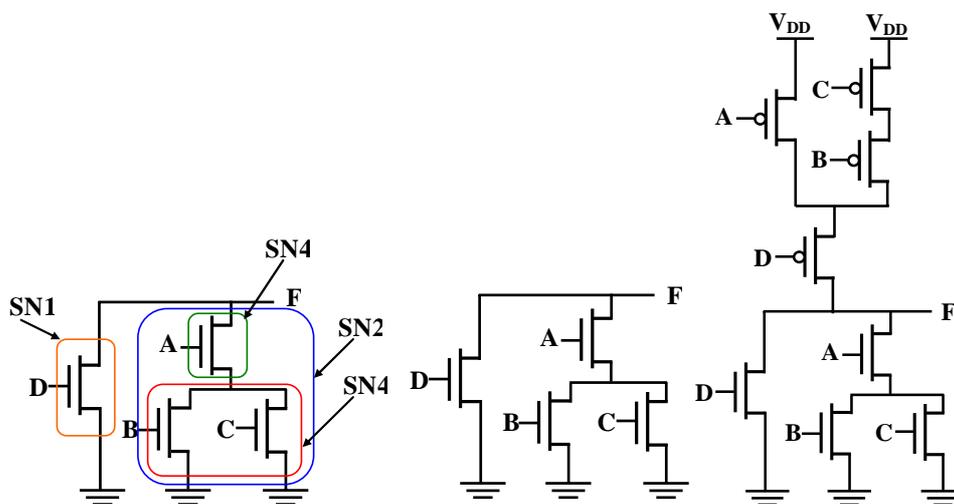


Figure 2.6 : Les portes complexes

II.2.3. La porte de transmission TG

La porte de transmission (TG) est utilisée dans la conception des circuits numériques CMOS pour passer ou ne pas passer un signal. Le schéma et le symbole logique de la porte de transmission (TG) sont montrés dans la figue.2.7. La porte se compose de la connexion en parallèle d'un transistor PMOS et d'un transistor NMOS. En référence à la figure 2.7, quand S

(pour sélectionner) est haut nous observons que la porte de transmission passe le signal d'entrée à la sortie. La résistance entre l'entrée et la sortie peut être estimée comme $R_n // R_p$ [3].

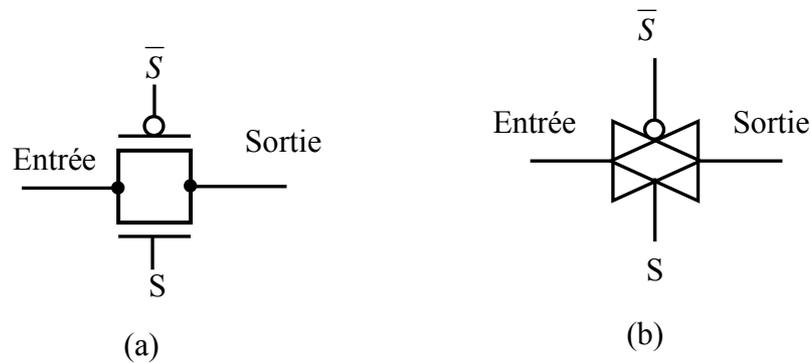


Figure 2.7 : la porte de transmission

Puisque le transistor NMOS passe le niveau logique bas bien et le transistor PMOS passe le niveau logique haut, mettant deux transistors MOSFET complémentaires en parallèle, comme a été montré dans fig. 2.7 il résulte que la porte TG passe les deux niveaux logiques bien. La porte TG CMOS exige deux signaux de commande, S et \bar{S} .

II.3. Les portes logiques dynamiques

Les portes logiques dynamiques ou synchronisées sont utilisées pour diminuer la complexité, augmenter la vitesse, et pour une dissipation de puissance inférieure. L'idée fondamentale derrière la logique dynamique est d'utiliser l'entrée capacitive du transistor MOSFET pour mémoriser une charge et pour se rappeler ainsi un niveau logique pour un usage ultérieur.

II.3.1. Le concept des circuits logiques dynamiques

Dans les circuits logiques statique discutés jusqu'à ce point, la sortie est valide du moment que les entrées soient bien définies. Un circuit logique dynamique, d'autre part, donne un résultat à la sortie qui est seulement valable pour une période courte. Si le résultat n'est pas employé immédiatement, la tension peut changer à temps et donner une valeur de sortie incorrecte. Un circuit logique dynamique emploie des nœuds capacitifs pour mémoriser la charge électrique.

II.3.2. Le principe fondamental de la logique dynamique

Considérant le transistor de passage NMOS montré dans fig. 2.7, conduisant un inverseur, si nous synchronisons la grille du transistor de passage par un niveau logique haut,

le niveau logique sur l'entrée, point A, sera passé à l'entrée de l'inverseur, point B. si ce niveau logique est un 0. L'entrée de l'inverseur sera forcée à la masse tandis que, un 1 logique forcera l'entrée de l'inverseur à $V_{DD} - V_{THN}$. Quand le signal d'horloge passe à 0, le transistor de passage est bloqué et l'entrée de l'inverseur se rappelle le niveau logique. En d'autres termes, quand le transistor est passant, la capacité d'entrée de l'inverseur est chargée à $V_{DD} - V_{THN}$ ou à la masse, à travers le transistor de passage. Aussi longtemps que cette charge est présente, la valeur logique est mémorisé, ce qui nous intéresse à ce point est le mécanisme de fuite qui peut fuir la charge stockée du nœud. Un nœud, tel que celui marqué B dans figure 2.8, s'appelle un nœud dynamique ou un nœud de stockage. Noter que ce nœud est un nœud à grande impédance et est facilement susceptible du bruit [3]

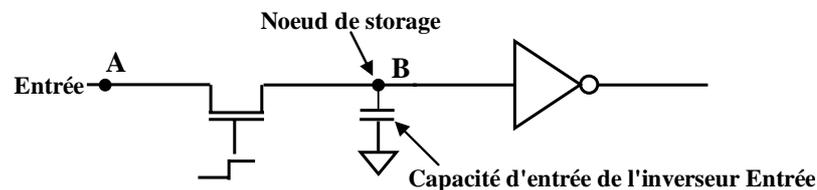


Figure 2.8 : Exemple d'un circuit dynamique et d'une capacité associée de stockage

II.3.3. Les familles des circuits logiques dynamiques :

Dans la porte logique dynamique la sortie est seulement valide pour une quantité de temps courte après que le résultat soit produit. Bien que ceci semble tout à fait restrictif, les réseaux dynamiques CMOS sont utiles pour la conception de système à grande vitesse. nous analyserons plusieurs familles logiques dynamiques CMOS.

II.3.3.1. La logique C2MOS

La logique Clocked CMOS (C^2 MOS), est une famille logique qui combine la conception logique statique avec la synchronisation réalisée en employant des signaux d'horloge. Le schéma 2.8 montre la structure générale d'une porte logique C^2 MOS. Les d'entrée A, B, et C sont connectées aux paires complémentaires de NMOS/PMOS comme dans la conception statique ordinaire où elles agissent comme les commutateurs ouverts ou fermés. La seule modification est la mise en place de deux FETS synchronisés entre les réseaux logique et la sortie. Le PMOS est contrôlé par signal \overline{CLK} et sépare le bloc logique PMOS et le C_{out} , alors que Mn est contrôlée par le CLK et il remplit la même fonction pour le bloc logique NMOS.

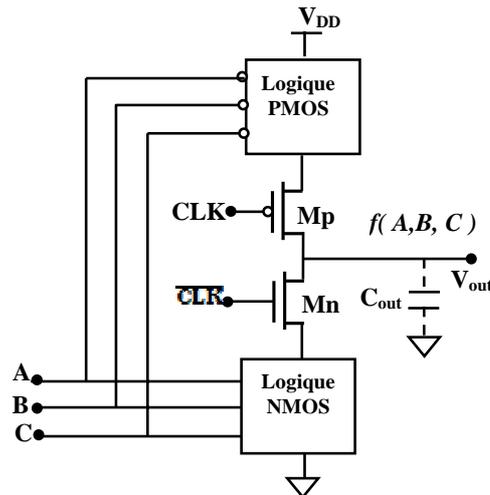


Figure 2.9 : La porte logique C²MOS

La logique C²MOS est employé pour réduire la dissipation de puissance et la taille de layout et pour augmenter la vitesse. La porte statique standard CMOS exige 2N transistors MOSFET pour une porte de n entrées. En général, une porte de n entrée C²MOS exige N+2 transistors MOSFET où deux transistors MOSFET sont utilisés dans le système de synchronisation. Des transistors MOSFET additionnels peuvent être utilisés pour le buffering ou pour aider la porte à apparaître dans son fonctionnement plus statique [3].

II.3.3.2. La logique PE

Cette section discute la logique PE où precharge evaluate logic. Considérer la porte NAND à trois entrées montrées dans la figue 2.9. Le fonctionnement de cette porte se fonde sur une entrée d'horloge simple. Quand CLK est bas, la capacité de nœud de sortie est chargée à V_{DD} par M5. Pendant la phase CLK d'évaluation, est haute, M1 est passant, et si A0, A 1, et A2 sont hauts, la sortie est tirée au niveau bas. La sortie logique est disponible seulement quand CLK est haut. La sortie est une logique 1 quand CLK est bas. Un inconvénient de la logique PE est que la sortie logique de la porte est une partie disponible du temps et pas de tout le temps comme dans les portes statiques.

Plusieurs caractéristiques importantes de la porte PE devraient être précisées. La capacité d'entrée de la porte PE est inférieure à cela de la porte statique. Chaque entrée est reliée à un seul transistor MOSFET où les entrées de porte statiques sont attachées à deux transistors MOSFET. Potentiellement la porte PE est alors plus rapide et dissipe moins de puissance [3].

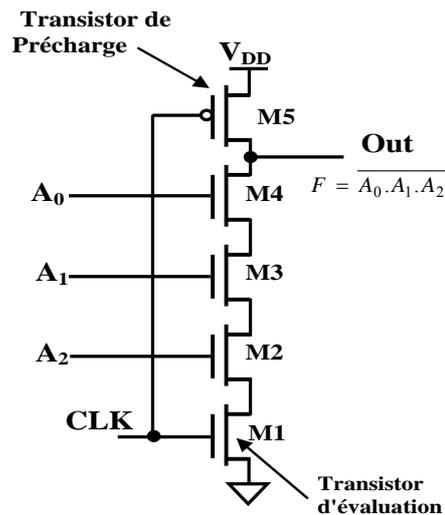


Figure 2.10 : La porte PE NAND a trois entrées

II.3.3.3. La logique Domino

La porte PE avec l'addition d'un inverseur s'appelle la logique domino (fig.2.11). Le nom domino vient du fait, qu'une porte logique dans une série des portes logiques dominos ne peut pas changer ses états de sortie jusqu'à ce que la porte précédente change ses états. Le changement de la sortie des portes se produit semblable à une série des pièces de domino en chute. L'inverseur utilisé dans la porte domino à l'avantage supplémentaire qu'il peut être dimensionné pour conduire de grandes charges capacitives [5].

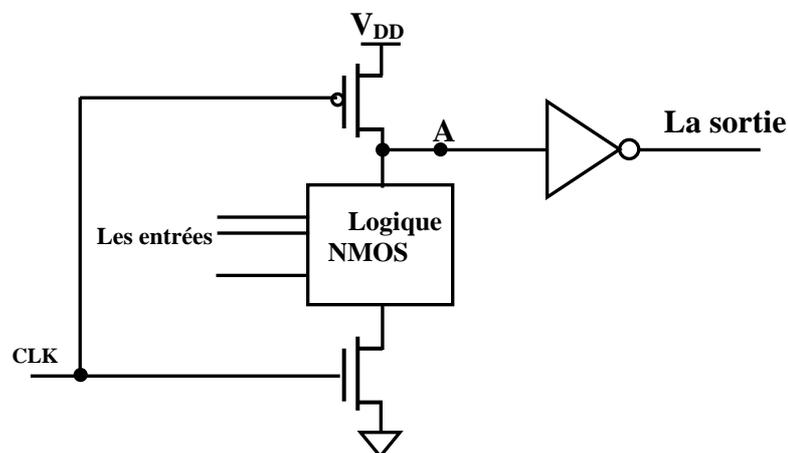


Figure 2.11 : La porte de logique domino

II.3.3.4. La logique NP

L'idée derrière l'implémentation d'une fonction logique utilisant la logique NP est montrée dans la figure 2.12. on arrange des étages de NMOS et de PMOS élimine le besoin de retard associé à l'inverseur utilisé dans la logique domino, rendant le fonctionnement possible à une vitesse plus élevée [5].

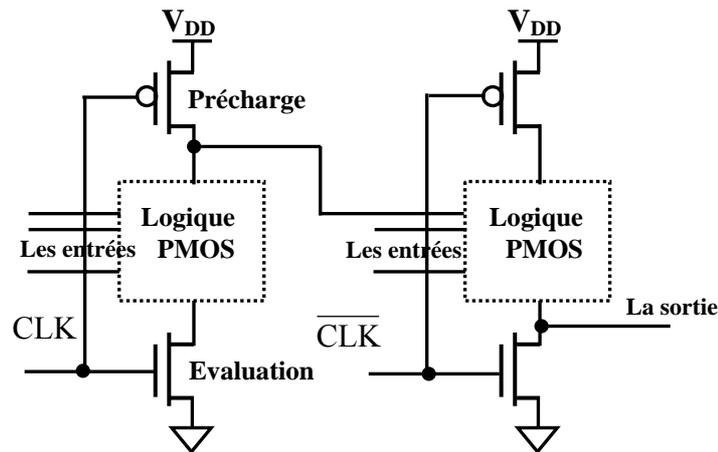


Figure 2.12 : La logique PN

II.4. Les circuits logiques séquentiels

Les circuits logiques combinatoires, décrits plus tôt, ont la propriété que la sortie d'un bloc logique est une fonction seulement des valeurs courantes de l'entrée. Pourtant pratiquement tous les systèmes utiles exigent le stockage d'information d'état, mener à une autre classe des circuits appelés les circuits logiques séquentiels. Dans ces circuits, la sortie dépend non seulement des valeurs courantes des entrées, mais également aux valeurs précédentes de l'entrée. En d'autres termes, un circuit séquentiel se mémorise rappelle une partie de l'histoire passée du système qu'il a dans sa mémoire.

II.4.1. Les bascules

Les latches et les bascules sont les éléments de base pour mémoriser l'information. Ils sont les modules fondamentaux pour tous les circuits séquentiels. Un verrou ou une bascule peut mémoriser un bit de donnée. La différence principale entre les latches et les bascules c'est que pour les latches, leurs sorties sont constamment affectées par leur entrée aussi longtemps que le signal de d'activation est affirmé. En d'autres termes, quand elles sont actives, leur contenu change immédiatement quand leur entrée change. D'autre part les bascules, avoir leur contenu change seulement au front montant ou descendant du signal d'activation. Après le front montant ou le front descendant du signal d'activation et pendant le moment où le signal est à un 1 ou à un 0 constant, le contenu de la bascule reste constant même si l'entrée change. Ce signal d'activation est habituellement le signal d'horloge.

II.4.1.1. La bascule RS

La bascule set reset (SR) montrée dans figure 2.13, est composée de deux portes NAND. Le symbole logique et la table de vérité sont également montrés dans cette figure.

Considérant le cas où S est haut et R est bas. Forçant R au niveau bas cause Q de passer au niveau haut. Puisque S est haut et Q est haut, la sortie \bar{Q} est bas. Considérer maintenant le cas où S et R sont bas. Dans ces circonstances, les sorties de bascule sont les deux au niveau haut. Cette bascule peut facilement être conçue et présentée avec les techniques des portes statiques logiques [3].

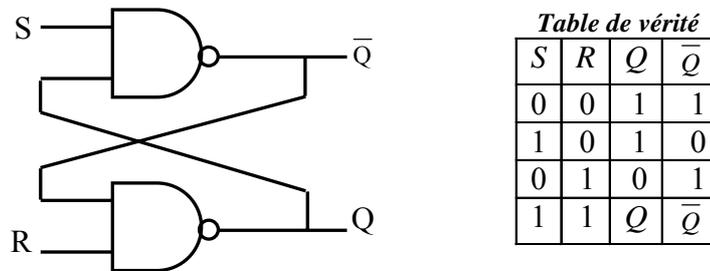


Figure 2.13 : La bascule set reset conçue à l'aide des portes NAND.

Une implémentation alternative de la bascule SR est montrée dans la figure 2.13 utilisant des portes NOR. Considérons le cas où S est haut et R est bas. Pour la porte NOR, une entrée au niveau haut force la sortie de la porte au niveau bas. Par conséquent, la sortie \bar{Q} est au niveau bas toutes les fois que l'entrée S au niveau haut. De même, toutes les fois que l'entrée R à niveau haut, la sortie Q doit être au niveau bas. Le cas où les deux entrées étant au niveau haut cause Q et \bar{Q} à passer à 0, ou en d'autres termes les sorties de la bascule ne sont plus des compléments. La figure 2.14 montre le symbole logique de la bascule de SR.

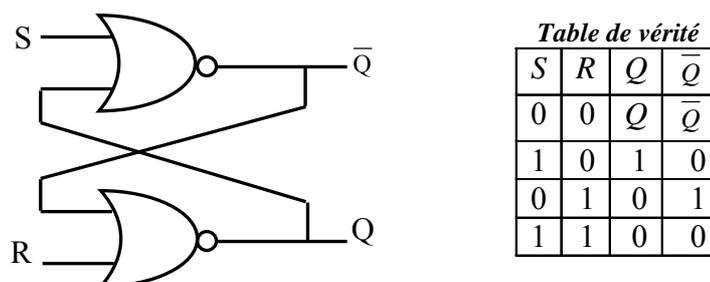


Figure 2.14 : La bascule set reset conçue à l'aide des portes NOR

II.4.1.2. Les bascules synchronisées

Des bascules synchronisées peuvent être divisées en trois catégories. La première catégorie se compose de ceux dans lesquels la largeur d'impulsion du signal d'horloge doit être courte comparée à la propagation du retard à travers la bascule. En d'autres termes, l'entrée d'horloge devrait être au niveau haut, et puis passer au niveau bas avant que la sortie de la bascule change l'état. La deuxième catégorie de bascule synchronisée se compose de ceux dans lesquels la sortie change, tandis que le signal d'horloge est au niveau haut. Ce type

de bascule s'appelle parfois une bascule à niveau sensible; La catégorie finale des bascules est le type déclenché par front d'impulsion. La sortie de la bascule change l'état sur le front montant ou descendant d'horloge.

1. La bascule JK (Impulsion d'horloge de largeur courte)

La bascule JK synchronisé est montrée dans figure. 2.15. La bascule JK est construite à l'aide de la bascule SR NAND et de deux portes NAND. Cependant, la différence de la bascule SR, J et K peuvent être au niveau haut en même temps sans les sorties devenant égales. Le fonctionnement de la bascule JK dépend de l'état précédent de la bascule. Avec le signal d'horloge a tenu le niveau bas, les entrées et les sorties de la bascule SR ne changent pas. Tenir le signal d'horloge à niveau haut fait osciller la sortie de la bascule entre une logique 0 et 1. Avec l'application d'une impulsion d'horloge courte et $J=K=0$, les sorties du bascule ne changent pas. Si $K=1$ et $J=0$, la sortie, Q est à 0 après l'application de l'impulsion d'horloge, tandis que si, $J=1$ et $K=0$, la sortie est au niveau haut. Si J et K sont au niveau haut, la sortie devient le complément de l'état précédent.

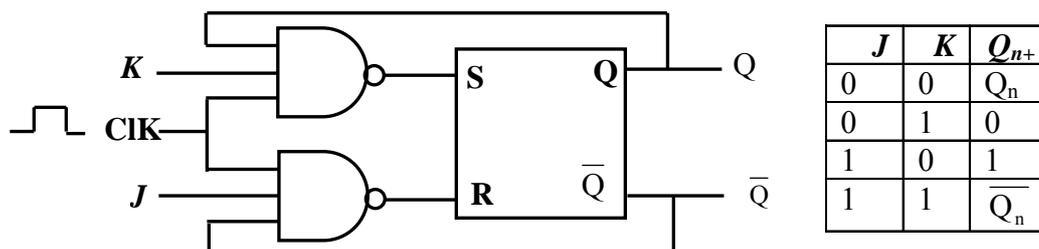


Figure 2.15 : La bascule JK synchronisé

Une bascule T peut être construit en utilisant la bascule JK par le réglage $J=K=1$, ou simplement en remplaçant les portes NAND à trois entrées de la figure 2.15 avec des portes de NAND à deux d'entrée. L'entrée d'horloge du bascule JK est employée comme entrée T de la bascule T. L'application d'une impulsion courte à l'entrée T de la bascule T cause le basculement des états de la sortie Q. Si la sortie est à niveau haut et l'entrée de T passe à 1, la sortie change au niveau bas. Le basculement de l'entrée T au niveau haut provoque encore la sortie de changer de nouveau à un niveau haut. La bascule T peut être employé pour diviser un signal d'horloge en demi, (en utilisant la bascule de fig.2.15).

2. La bascule D (sensibles de niveau)

Les bascules qui sont synchronisées avec un signal qui permet à la sortie de changer avec l'entrée, n'ont aucune condition spécifique de largeur d'impulsion pour l'horloge. Pour illustrer une bascule de niveau sensible considérer la bascule donnée ou D montré dans

fig.2.16 avec le symbole logique associé. Quand le signal d'horloge est haut, l'entrée de D peut passer directement à la bascule SR, si D est un 1 tandis que CLK est au niveau haut, la sortie, Q est un 1, alors que si D est au niveau bas la sortie est au niveau bas. Si D change à tout moment tandis que l'entrée de CLK est haute, la sortie suivra. Quand le signal CLK passe à 0, le niveau logique courant de D est verrouillé dans la bascule SR note que cette bascule n'est pas une bascule front sensible, parce que la sortie change indépendamment du temps de front de transition.

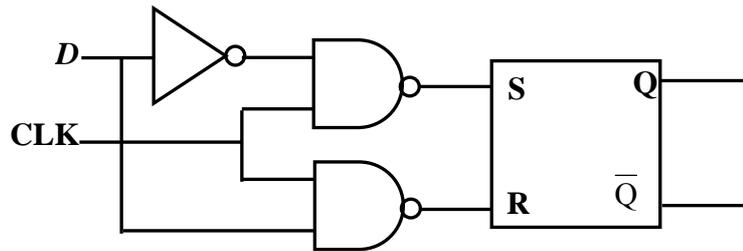


Figure 2.16 : Bascule D Sensible de niveau

3. La bascule JK maître esclave (déclenché par front d'impulsion)

La bascule JK maître- esclave, montré dans la figure 2.17 est un exemple d'une bascule déclenché par front d'impulsion. Quand le signal de CLK passe à 1, la bascule JK maître est activé .Puisque la bascule esclave ne peut pas changer des états quand CLK est haut, la largeur d'impulsion d'horloge ne doit pas être moins que le retard de propagation de la bascule. Quand CLK passe à 0, les données principales sont transférées à l'esclave. Si J et K sont au niveau bas, la sortie du maître demeure sans changement, et donc ainsi fait la sortie de l'esclave. Si $J = 1$ et $K = 0$, quand l'impulsion CLK passe à 0, la sortie principale, Q passe à 1. Quand CLK passe à 0, la sortie à niveau haut du maître est transférée à l'esclave. La bascule JK maître esclave se comporte juste comme la bascule JK de la section précédente excepté le fait que les données ne sont pas disponibles jusqu'à ce que CLK passe à 0, et il n'y a aucune restriction à la largeur d'impulsion de l'horloge (c.-à-d., la bascule est déclenché par front descendant), l'addition des possibilités reset ou set à la bascule peuvent être accomplies en ajoutant des portes logique entre NAND et la bascule SR.

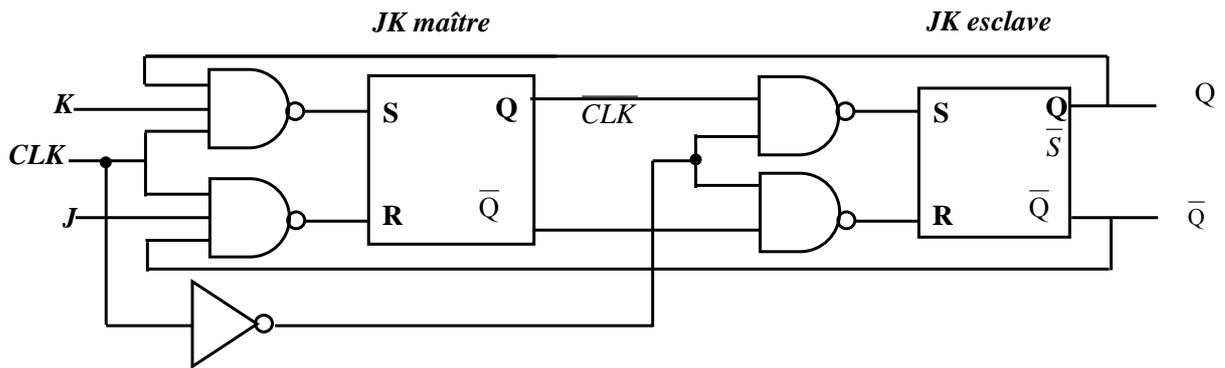


Figure 2.17: La bascule JK maître esclave déclenchées par front

II.4.2. Les latches et les registres

Un latch est un composant essentiel dans la construction d'un registre déclenché par front d'impulsion. C'est un circuit à niveau sensible qui passe l'entrée D à la sortie Q quand le signal d'horloge est haut. Ce latch serait en mode transparent. Quand l'horloge est au niveau bas, les données d'entrée prélevées sur le front descendant de l'horloge est maintenir stable a la sortie pour la phase entière, et le verrou est en mode de prise. Les entrées doivent être stables pendant une période courte autour du front descendant de l'horloge pour rencontrer les conditions setup et hold. Un latch fonctionnant dans les conditions ci-dessus est un verrou positif. De même, un verrou négatif passe l'entrée D a la sortie Q quand le signal d'horloge est bas [4].

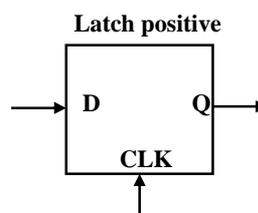


Figure2.18: Symbole du latch D

Contrairement aux latches a niveau sensibles, les registres déclenchés par front d'impulsion prélèvent seulement l'entrée pendant une transition d'horloge de 0 à 1 pour un registre déclenché par front d'impulsion positif, et de 1 à 0 pour un négatif registre déclenché par front d'impulsion .Ils sont typiquement construits en utilisant les primitifs de latch du schéma 2.18. Une configuration se reproduisant le plus souvent est la structure maître-esclave qui cascade un latch positif et un latch négatif.

II.4.2.1 Les registres dynamiques

Ceci a comme conséquence; une classe des circuits basés sur la mémoire temporaire de la charge des condensateurs parasites. Le principe est exactement identique à celui utilisé dans la logique dynamique la charge stockée sur un condensateur peut être employée pour représenter un signal logique. L'absence de la charge dénote un 0, alors que sa présence représente un 1 stocké. Aucun condensateur n'est idéal, malheureusement, et une certaine fuite de charge est toujours présente. Une valeur stockée peut par conséquent seulement être gardée pour une quantité de temps limitée, typiquement dans la gamme des millisecondes. Si on veut préserver l'intégrité de signal, une régénération périodique de sa valeur est nécessaire. Par conséquent nommé le stockage dynamique. La lecture de la valeur du signal stocké d'un condensateur sans perturber la charge exige la disponibilité d'un dispositif avec une impédance d'entrée élevée.

1. Le registre dynamique de porte de transmission

Un registre déclenché par front d'impulsion positif entièrement dynamique basé sur le concept maître esclave est montré sur le figure 2.19. Quand $CLK = 0$, la donnée d'entrée est prélevé sur le nœud 1 de stockage, qui a une capacité équivalente de $C1$ comprenant la capacité de la grille de $I1$, la capacité de jonction du $T1$, et la capacité de chevauchement de la grille du $T1$. Pendant cette période, l'étage esclave est en mode de prise, avec le nœud 2 dans un état (flottant) à grande impédance. Sur le front montant de l'horloge, le $T2$ de la porte de transmission s'active, et la valeur prélevée sur la droite du nœud 1 avant que le front montant propage à la sortie Q (à noter que le nœud 1 est stable pendant la phase élevée de l'horloge puisque la première porte de transmission est bloquée). Le nœud 2 stocke maintenant la version inversée du nœud 1. Cette implémentation d'un registre déclenché par front d'impulsion est très efficace car elle exige seulement 8 transistors. Les commutateurs de prélèvement peuvent être implémentés en utilisant seulement des transistors du passage NMOS, ayant pour résultat une implémentation quant même simple de 6 transistors. Le compte réduit de transistor est attrayant pour les systèmes à une performance élevée et de basse puissance [4].

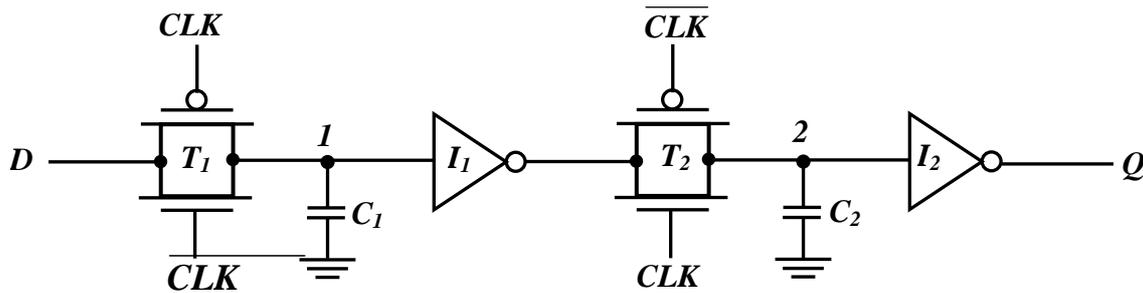


Figure 2.19: Registre dynamique de TG

2. Le registre dynamique C²MOS

Le schéma montre un registre dynamique ingénieux, basé sur le concept maître esclave peu sensible au chevauchement d'horloge. Ce circuit s'appelle le registre C²MOS (CMOS synchronisé) [4], et fonctionne en deux étapes [4].

1-Quand $CLK=0$ ($\overline{CLK} = 1$) : Le premier pilote à trois états est fonctionnel, et l'étage principal s'active comme un inverseur prélevant la version inversée de D sur le nœud interne X. L'étage est en mode d'évaluation. En attendant, la section esclave est dans un mode haute impédance, ou dans un mode de maintien. Les deux transistors M7 et M8 sont bloqués, découplant la sortie de l'entrée. La sortie Q maintient sa valeur précédente mémorisée sur le condensateur de sortie CL2.

2-Les rôles sont inversés quand $CLK=1$: la section de l'étage principale est en mode de maintien (M3-M4 bloqués), alors que la deuxième section évalue (M7-M8 passants). La valeur stockée sur CL1 propage au nœud de sortie par l'étage esclave, qui agit en tant qu'un inverseur.

Le circuit global fonctionne comme registre maître esclave déclenché par front d'impulsion positif très semblable au registre à base de porte de transmission présentée plus tôt. Cependant, il y a une différence importante :

Un registre C²MOS avec CLK , \overline{CLK} synchronisés est peu sensible au chevauchement, aussi longtemps que les temps de front montant et front descendant d'horloge sont suffisamment petits.

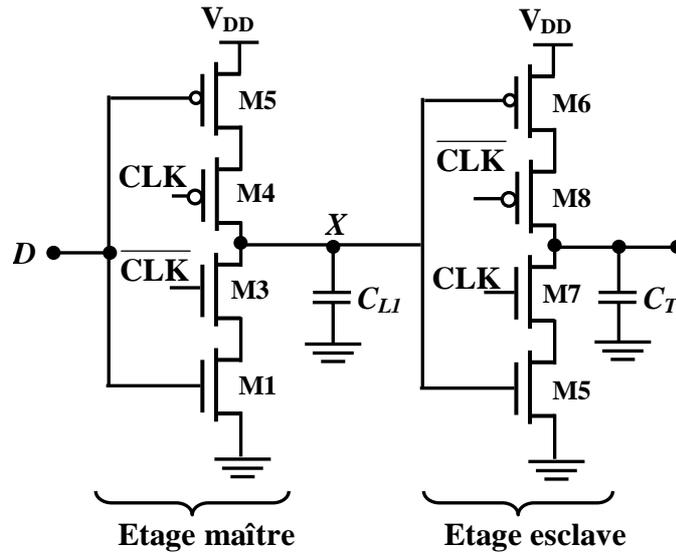


Figure 2.20: Registre dynamique C²MOS

II.5. La machine d'états finis

La figure 2.21 montre un schéma fonctionnel générique d'une machine d'états finis (FSM) qui consiste d'une logique combinatoire et des registres, qui tiennent l'état de système. Le système représenté ici appartient à la classe des systèmes séquentiels synchrones, dans lesquels tous les registres sont sous la commande d'une horloge globale simple. Les sorties du FSM sont une fonction des entrées présentes et de l'état présent. L'état futur est déterminé à base de l'état présent et les entrées présentes et retransmettre aux entrées des registres. Sur le front montant de l'horloge, le bit du prochain état est copié aux sorties des registres (après un certain retard de propagation), et un nouveau cycle commence. Le registre ignore alors les changements des signaux d'entrée jusque au prochain front montant. En général, les registres peuvent être déclenchés par front d'impulsion positif (où l'entrée des données sont copiées sur le front positif de l'horloge) ou de déclenché par front d'impulsion négatif (où des données d'entrée sont copiées sur le front négatif, comme est indiqué par un petit cercle sur l'entrée d'horloge) [4].

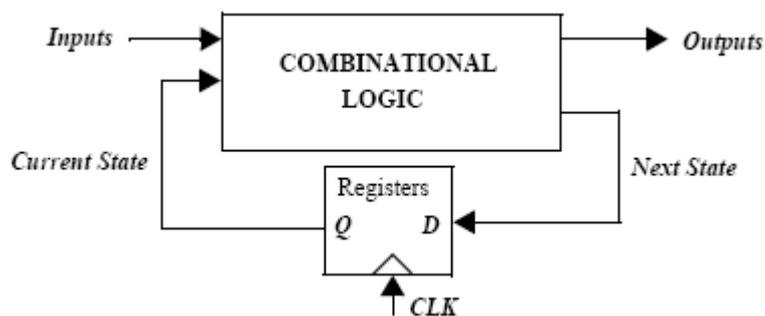
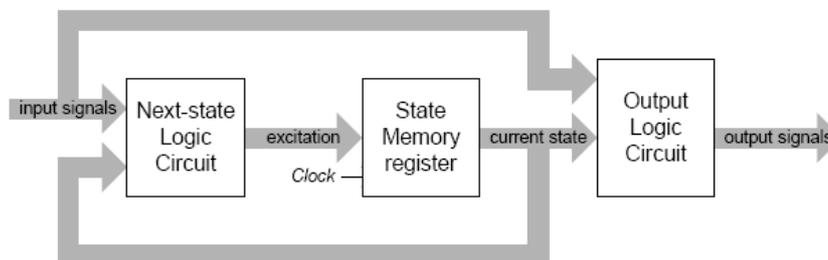
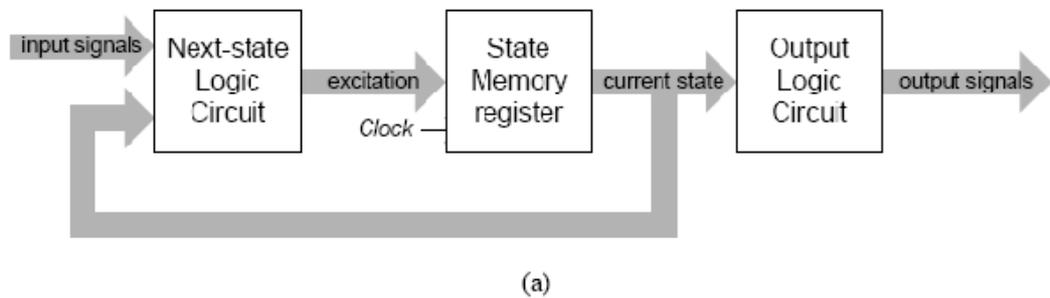


Figure2.21: diagramme bloc de la machine d'états finis utilisant un registre

II.5.1. Les modèles de la machine d'états finis (FSM) :

La figure 2.22(a) montre le schéma général pour le FSM de Moore où ses sorties dépendent seulement de son état actuel. La figure 2.22(b) montre le schéma général pour le FSM Mealy où ses sorties dépendent de l'état présent de la machine et également des entrées. Dans les deux figures, nous voyons que les d'entrées de l'état logique future sont les signaux d'entrée primaires et l'état présent de la machine. L'état logique futur génère des valeurs d'excitation pour changer la mémoire d'état. Une des différences dans les deux figures est celle pour le FSM de Moore, la logique de sortie a comme une entrée seulement l'état présent, tandis que, pour le FSM Mealy, la logique de sortie, a comme des entrées, l'état présent et les signaux d'entrée [6].



**Figure 2.22: Modèles de machine d'états Finis:
FSM de Moore(a); FSM de Mealy (b)**

II.5.2. L'Analyse des circuits séquentiels

Très souvent nous avons donné un circuit séquentiel et nous avons besoin de savoir son fonctionnement. L'analyse des circuits séquentiels est le processus dans lequel nous avons donné un circuit séquentiel et nous voulons obtenir une description précise du fonctionnement du circuit. La description d'un circuit séquentiel peut être sous forme des états futurs, d'une table de sortie, ou d'un diagramme d'état. Les étapes pour l'analyse des circuits séquentiels sont comme suit [6]:

1. dériver les équations d'excitation du circuit logique à partir des états futurs.
2. dériver les équations des états futurs en substituant les équations d'excitation dans les équations caractéristiques de la bascule.
3. dériver la table des états futurs à partir des équations des états futurs.
4. dériver les équations de sortie (si quel) du circuit logique de sortie.
5. dériver la table de sortie (si quel) à partir des équations de sortie.
6. tracer le diagramme d'état à partir de la table des états futurs et la table de sortie.

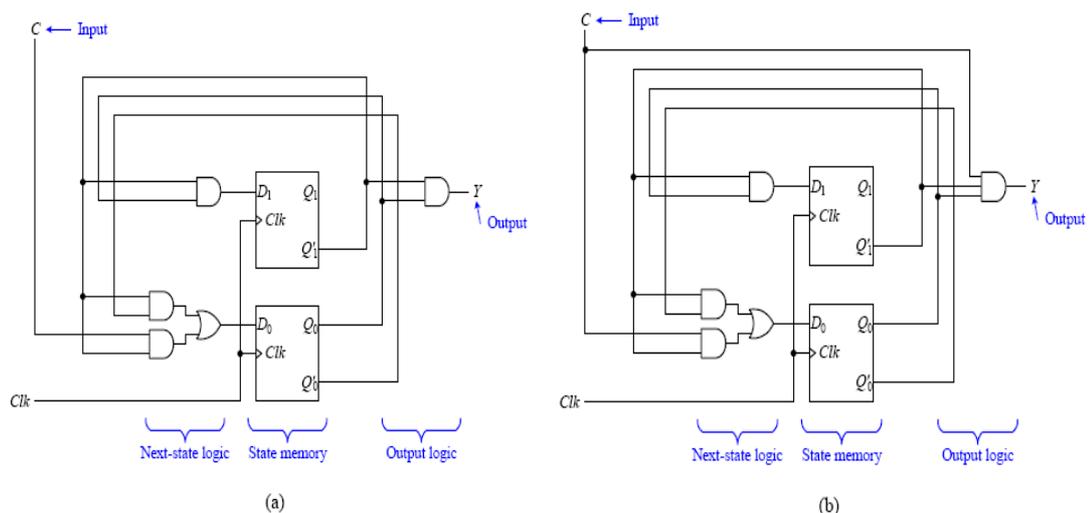
1-Les équations de l'excitation

Les équations L'excitation sont les équations de l'état futur logique du circuit, dans le FSM. Puisque l'état futur logique est un circuit combinatoire discuté dans la section précédente. L'état futur du circuit qui est dérivé de ces équations "excite" les bascules en les provoquant à changer les états. Ces équations fournissent les signaux aux entrées des bascules, et sont exprimées en fonction de l'état présent et les entrées de la FSM. L'état présent est déterminé par le contenu actuel des bascules, c'est-à-dire les "signaux de sortie de la bascule Q (et \bar{Q}). Il y a une équation pour chaque bascule.

Voici deux exemples des équations d'excitation pour les deux bascules D de la figure 2.23. La première équation fournit l'état futur du circuit pour l'entrée D de la bascule 1, et la deuxième équation fournit le circuit de l'entrée D de la bascule 0.

$$D_1 = \bar{Q}_1 Q_0 \quad (1)$$

$$D_0 = \bar{Q}_1 Q_0 + C\bar{Q}_1 \quad (2)$$



**Figure 2.23 : Exemple d'un circuit de machine d'états fini :
Moore(a) ; Mealy (b)**

2. Équation de l'état futur

Les équations de l'état futur spécifient ce qu'être l'état futur des bascules, elles sont dépendantes sur trois choses

- 1) l'état présent des bascules.
- 2) le comportement fonctionnel des bascules.
- 3) les entrées aux bascules.

L'état présent des bascules est juste les sorties des bascules Q. Le comportement fonctionnel d'une bascule, est décrit formellement par son équation caractéristique. L'équation caractéristique nous indique ce que doit qu'être Q_{next} , c.-à-d., ce qui doit être l'état futur. Les entrées aux bascules sont fournies par les équations d'excitation comme discuté dans la section 1 ci-dessus. Ainsi, pour dériver les équations de l'état futur, nous substituons les équations d'excitation dans les équations caractéristiques de la bascule correspondante. Par exemple, l'équation caractéristique pour la bascule D est

$$Q_{next} = D$$

Par conséquent, substituant les deux équations d'excitation (1) et (2) de la sous-section précédente dans l'équation caractéristique pour la bascule D nous donnera les deux équations suivantes de l'état futur.

$$Q_{1next} = D_1 = \overline{Q_1} Q_0 \quad (3)$$

$$Q_{0next} = D_0 = \overline{Q_1} \cdot \overline{Q_0} + C \cdot \overline{Q_1} \quad (4)$$

3. Table des états futurs

La table des états futurs est simplement la table de vérité dérivée des équations des états futurs. Elle énumère pour chaque combinaison de l'état présent les valeurs (Q) et les valeurs d'entrée, ce qui devrait être les valeurs des états futurs (Q_{next}). Ces valeurs des états futurs sont obtenues en substituant l'état présent et les valeurs des entrées dans les équations appropriées des états futurs.

La figure 2.24 montre un exemple de table des états futurs avec les états présents $Q_1 Q_0$ égalent à 00, 01, 10, et 11, et un signal d'entrée C. Les entrées dans la table sont les valeurs des états futurs $Q_{1next} Q_{0next}$. Ces valeurs des états futurs sont obtenues à partir de substituer les valeurs des états présents $Q_1 Q_0$ et la valeur d'entrée C dans les équations des états futurs (3) et (4) de la section 2 ci-dessus.

Current State Q_1Q_0	Next State	
	Q_{1next}	Q_{0next}
	$C=0$	$C=1$
00	01	01
01	10	11
10	00	00
11	00	00

Figure 2.24: Une table des états futurs avec quatre états et un signal d'entrée C

Par exemple, l'entrée gauche supérieure nous indique que si l'état présent est 00 et l'état d'entrée $C=0$; donc c'est vrai l'état futur du FSM ira 01. Puisque 01 est également l'état futur de l'état présent 00 et la condition $C = 1$ est vraie, ceci signifie que la transition de l'état 00 à 01 ne dépend pas de l'état C d'entrée, ainsi c'est une transition sans conditions. De l'état 01, il y a deux transitions conditionnelles : le FSM aura une transition à l'état 10 si la condition $C = 0$ est vraie, autrement si $C = 1$, il aura une transition à l'état 11. Les deux états 10 et 11 vont à 00 sans condition.

4. Équation de sortie

Les équations de sortie sont les équations dérivées de sortie du circuit logique combinatoire dans le FSM.

Selon le type de FSM (Moore ou Mealy), les équations de sortie peuvent dépendre juste de l'état présent ou de l'état présent et des entrées.

Pour le circuit de Moore de la figure 2.23 (a), l'équation de sortie est

$$Y = \overline{Q_1}Q_0 \quad (5)$$

Pour le circuit de Mealy de la figure 2.23 (b), l'équation de sortie est

$$Y = C\overline{Q_1}Q_0 \quad (6)$$

Un FSM typique aura beaucoup de signaux de sortie, et tellement il y aura une équation pour chaque signal de sortie.

5. Table de sortie

Comme la table des états futurs, la table de sortie est la table de vérité qui est dérivée des équations de sortie. Les tables des sorties pour le FSM Moore et Mealy sont légèrement différentes de l'un de l'autre. Pour le FSM de Moore, la table de sortie produit les listes pour chaque combinaison de l'état présent que devraient être la valeur de sortie. Mais pour le FSM Mealy, la table de sortie produit les listes pour chaque combinaison de l'état présent et les valeurs des entrées qui devraient être les valeurs de sortie. Ces valeurs de sortie sont obtenues en substituant l'état présent et les valeurs d'entrées dans les équations de sortie appropriées.

Les figures 2.25(a) et 2.25(b) montrent un modèle des tables de sortie pour le FSM Moore et Mealy, qui sont dérivés des équations de sortie (5) et (6) respectivement de la

section 4. Pour le FSM de Moore, le signal de sortie Y dépend seulement de la valeur d'état présent Q_1Q_0 , tandis que, pour le FSM Mealy, le signal de sortie Y dépend de l'état présent et de l'entrée C.

Current State Q_1Q_0	Output Y
00	0
01	1
10	0
11	0

(a)

Current State Q_1Q_0	Output Y	
	C = 0	C = 1
00	0	0
01	0	1
10	0	0
11	0	0

(b)

Figure 2.25: Table de sortie :(a) pour le FSM de Moore, (b) pour le FSM Mealy.

6 Diagramme d'état

Un diagramme d'état est un graphique avec des nœuds et des flèches dirigés reliant les nœuds. Le diagramme d'état décrit graphiquement le fonctionnement du FSM. Il y a un nœud pour chaque état du FSM et ces nœuds sont marqués avec l'état dans lequel ils représentent. Pour chaque transition d'état du FSM, il y a une flèche dirigée reliant deux nœuds. La flèche dirigée provient du nœud qui représente l'état présent lequel le FSM se transit, et va au nœud qui représente l'état futur vers lequel le FSM transit. Les flèches peuvent avoir ou ne peuvent pas avoir des étiquettes sur elles. Les flèches pour des transitions sans conditions d'un état à l'autre n'auront pas une étiquette. Dans ce cas-ci, seulement une flèche peut provenir de ce nœud. Les transitions conditionnelles d'un état auront deux flèches sortantes. Les deux flèches de cet état ont les conditions correspondantes de signal d'entrée marqué sur eux; une flèche avec légende quand l'état est vrai et une autre flèche avec légende quand l'état est faux.

La figure 2.26(a) montre un petit diagramme avec quatre états 00, 01, 10, et 11, et un signal d'entrée C. Ce diagramme d'état est dérivé de la table des états futurs représentée sur le schéma 2.24 et de la table de sortie du schéma 2.25(a). Il y a trois transitions inconditionnelles 00 à 01, 10 à 00, et 11 à 00, et une transitions conditionnelles, de 01 à 10 ou à 11. Pour la transition conditionnelle de 01, si la condition $C = 0$ est vraie, alors la transition de 01 à 10 est faite. Autrement, si la condition $C = 0$ est fausse, donc $C = 1$ est vrai, alors la transition de 01 à 11 est faite.

Le signal de sortie Y sur le schéma 2.26(a) est marqué à l'intérieur de chaque nœud dénotant que le sortie dépend seulement de l'état présent. Par exemple, quand le FSM est dans

l'état 01, la sortie Y est 1, tandis que, dans l'état 11, Y est 0. Par conséquent, ce diagramme d'état est pour le FSM de Moore.

Sur la figure 2.26 (b), le signal de sortie Y est marqué sur les flèches dénotant que la sortie dépend de l'état présent et du signal d'entrée C. Par exemple, quand le FSM est dans l'état 01, si le FSM prend la flèche gauche pour C=0 à l'état 10, alors il produira un 0 pour Y. Cependant, si le FSM prend la flèche droite pour C=1 à l'état 11, puis lui produira un 1 pour Y. Par conséquent, ce diagramme d'état est pour le FSM Mealy.

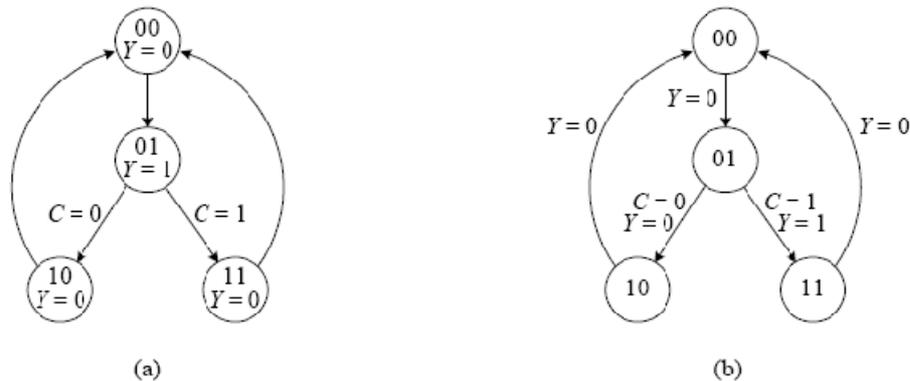


Figure 2.26 : Un diagramme d'état avec quatre états et un signal d'entrée, FSM de Moore(a), FSM Mealy (b)

II.5.3. Le synthèse des circuits séquentiels

La synthèse des circuits séquentiels est juste l'inverse de l'analyse des circuits séquentiels. Dans la synthèse, nous commençons par ce qui est habituellement une description fonctionnelle ambiguë du circuit que nous voulons. De cette description, nous devons proposer le fonctionnement précis du circuit en utilisant un diagramme d'état. Le diagramme d'état nous permet de construire les tables des états futurs et de sortie. Le circuit peut alors être dérivé des tables des états futurs et de sortie. Pendant le processus de synthèse, il y a beaucoup d'optimisations possibles de circuit en termes de taille de circuit, vitesse, et puissance d'énergie qui peut être effectuée. Dans cette section, nous nous concentrerons seulement sur synthétiser un circuit séquentiel fonctionnellement correct [6].

Les étapes pour la synthèse des circuits séquentiels sont comme suit :

1. produire un diagramme d'état de la description fonctionnelle du circuit.
2. dériver la table des états futurs du diagramme d'état.
3. dériver la table de sortie du diagramme d'état.
4. convertir la table des états futurs à la table d'implémentation.

5. dériver les équations d'excitation pour chaque entrée de la bascule de la table d'exécution.
6. dériver les équations de sortie de la table de sortie.
7. dessiner le schéma de circuit basé sur les équations d'excitation et de sortie.

II.6. Conclusion

Il est bien connu dans l'ingénierie que la conception d'une construction complexe sans compréhension antérieure des modules fondamentaux est une route sûre à l'échec. Ceci se tient sûrement pour la conception des circuits numériques aussi bien.

Les blocs fonctionnels de base dans les circuits numériques d'aujourd'hui sont les dispositifs de semi-conducteur de silicium, plus spécifiquement les portes logiques CMOS combinatoires et séquentielles.

La description du fonctionnement des portes logiques combinatoires, c'est pour mettre en valeur les propriétés et les paramètres qui sont particulièrement importants dans la conception des portes numériques.

Le rôle des portes logiques dynamiques C²MOS a été apprécié pendant longtemps dans le monde des circuits intégrés numériques.

Enfin la connaissance et la compréhension nécessaires d'une autre classe des circuits, connue sous le nom de séquentielle ou régénératrice est une motivation principale pour la conception de la machine d'états, qui inclut une partie de logique combinatoire et un module séquentiel qui tient l'état.

Chapitre III : Conception assistée par ordinateur des circuits numériques

III.1.Introduction

L'apparition des circuits ASIC (Application Specific Integrated Circuits) et des circuits logiques programmables de type PLD (Programmable Logic Device), CPLD ou FPGA (Field Programmable Gate Array), a permis à l'utilisateur de créer, dans ces circuits, toutes les fonctions logiques qu'il souhaite.

Les outils de développement mis à la disposition des utilisateurs doivent donc permettre de passer de la description du comportement d'une fonction logique à son câblage dans le circuit et cela de la manière la plus simple possible.

Le flot de conception des circuits intégrés VLSI (intégration à très grande échelle) est un terme utilisé à décrire le processus de la conception de chip. Le groupe concepteur/conception du circuit emploie une multitude d'outils de conception assistée par ordinateur dans tout le flot de conception pour concevoir (tape out) un circuit intégré CI. Il y a de nombreux outils de la conception assistée par ordinateur (CAO) disponible commercialement pour aider les concepteurs dans la conception. Chaque étape de flot de conception implique un outil de CAO. Ainsi, le choix des outils de CAO dicte la facilité et la convenance du processus de conception. Chaque outil a quelques caractéristiques essentielles qui le rendent avantageux pendant une certaine phase de conception. Les outils de CAO devraient faciliter la conception de chip dans une période courte le plus possible. En même temps la conception doit être fiable et robuste. Selon le choix de l'outil, et en tirant profit de ses caractéristiques essentielles, une optimale solution à travers le flot de conception numérique peut être atteinte.

III.2. Les outils CAO de la microélectronique

La nécessité pour que le CAO soit disponible est en grande partie due au volume des données impliqués plutôt que les difficultés de la conception, plus la nécessité d'assurer que la conception est aussi loin que possible sans erreur avant d'encourir n'importe quelles dépenses irrémédiables de la fabrication. Le temps et l'argent sont donc le principe fondamental conduisant des facteurs derrière le besoin des bonnes ressources de CAO, ceci être particulièrement nécessaire pour les activités de conception faite sur commande où le coût de la conception et le délai d'arrivée au marché sont critiques.

Les outils de Conception Assistée par Ordinateur (CAO) dans le domaine de la microélectronique sont très divers. Mais on focalise sur les outils principaux implémentés dans notre travail concernant la conception d'un circuit logique (machine d'états).

III.2.1. Le langage VHDL

L'abréviation VHDL signifie VHSIC Hardware Description Language (VHSIC: Very High Speed Integrated Circuit). Ce langage a été écrit dans les années 70 pour réaliser la simulation de circuits électroniques. Ensuite on l'a étendu en lui rajoutant des extensions pour permettre la conception (synthèse) des circuits logiques programmables (P.L.D. Programmable Logic Device) et pour décrire des systèmes numérisés dessinés a une implantation dans des circuits programmable (FPGA, ASIC).il remplace la saisie du schéma traditionnelle .la plupart des outils de développement autorise les deux types de saisie .le langage VHDL est en principe standardisé. Chaque outil de développement dispose cependant de bibliothèques spécifiques, aujourd'hui Le VHDL devient un langage de spécification, de simulation et également de conception.

Le VHDL est un langage portable qui va trouver place dans le cycle de conception du niveau spécification au niveau porte mais aussi lors de la génération des vecteurs de test. La description VHDL est inséparable de la simulation de type événementielle.

- **les descriptions comportementales** : Elles spécifient le comportement au moyen de formules ou de programmes d'actions.
- **les descriptions structurelles** : Elles énoncent une interconnexion de composants. Une description structurelle est la transcription directe d'un schéma.

Structure d'une description VHDL simple.

Une description VHDL est composée de 2 parties indissociables à savoir :

- L'entité (ENTITY), elle définit les entrées et sorties.

- L'architecture (ARCHITECTURE), elle contient les instructions VHDL permettant de réaliser le fonctionnement attendu [7].

```

entity dlatch is
  port (EN, PR, CL: in bit; -- commandes
        D: in bit; -- entrée
        Q: out bit); -- sortie
end dlatch;
architecture bhv of dlatch is
  begin
    process (EN, PR, CL, D)
      begin
        if PR = '1' then
          Q <= '1';
        elsif CL = '1' then
          Q <= '0';
        elsif EN = '1' then
          Q <= D;
        end if;
      end process;
    end bhv;

```

Code 1 : Modèle d'une latche D avec preset et clear asynchrones

III.2.2. PSpice

PSpice est l'un des plus célèbres logiciels de simulation et d'analyse des circuits électroniques. Il permet de réaliser le schéma d'un montage quelconque, dont les composants sont décrits de manière à reproduire aussi fidèlement que possible des composants réels, puis de simuler le fonctionnement de ce montage pour l'analyser sous toutes les niveaux, à l'aide d'outils aussi variés que sophistiqués.

Sa puissance et son extraordinaire richesse ont fait de PSpice une référence mondiale dans le domaine de la conception et de la simulation de circuits. PSpice est un logiciel de simulation mixte (analogique ou logique).

La simulation effectue les analyses principales suivantes :

- Une analyse temporelle (Transient ou TRAN) correspond à une simulation en fonction du temps.
- Une analyse fréquentielle (AC Sweep/Noise ou AC) permet d'effectuer une simulation en fonction de la fréquence.
- Une analyse continue (DC Sweep ou DC) permet d'effectuer une simulation en fonction d'une grandeur pouvant être : Une tension, un courant, un paramètre global, un paramètre de modèle de simulation, la température.

➤ Une analyse paramétrique permet d'effectuer une simulation temporelle ou fréquentielle ou continue avec en plus la variation d'une grandeur. Cette option permet de faire varier certains paramètres soit sur des générateurs (tension ou courants), soit sur des composants

III.2.3. Cadence

Cadence est un environnement électronique de la conception automatisée (EDA) qui intègre dans un cadre simple des différentes applications et des outils (classe des propriétaires et d'autres fournisseurs), permettant à supporter toutes les étapes de conception et de vérification des IC a travers un environnement simple. Ces outils sont complètement générales supportent différentes technologies de fabrication. Quand une technologie particulière est choisie, un ensemble de configuration et dossiers de technologie connexes sont utilisés pour adapter l'environnement de cadence. Cet ensemble de dossiers est généralement référé comme kit de conception.

III.2.4. Les simulateurs numériques

Les simulateurs numériques sont des éléments incontournables d'un flot de conception, ils sont devenus le cœur des environnements complets de vérification. Ils sont désormais capables pour les plus puissants d'entre eux de gérer plusieurs langages simultanément, de supporter les assertions ou de faire tourner de manière native un test bench.

Cette richesse des solutions de simulation fait qu'aujourd'hui on peut distinguer trois groupes de sociétés impliquées dans ce domaine. Ces trois leaders de CAO proposent chacun une offre complète de vérification basée sur leur moteur de simulation: Synopsys avec son simulateur VCS, Cadence dont le simulateur NC SIM est désormais intégré au sein de la plate-forme de vérification Incisive et enfin Mentor avec ModelSim.

III.2.5. Le simulateur ModelSim

ModelSim est le simulateur VHDL/Verilog/System C mixte le plus répandu au monde. La popularité des produits ModelSim est le reflet de l'engagement de Model Technology à fournir la meilleure technologie de simulation, la performance, le support et le prix. Les produits ModelSim ont une architecture unique utilisant des technologies comme la compilation directe optimisée pour réduire les temps de compilation et de simulation. Ces innovations permettent de grandes performances en compilation/simulation, l'entière liberté d'utiliser le VHDL et/ou le Verilog, ainsi qu'une capacité inégalée de personnaliser le

simulateur. Enfin les utilisateurs apprécient la facilité d'utilisation et la qualité du support technique.

III.3. Le flot de conception générique des circuits intégrés

Le flot de conception est un terme employé pour décrire les diverses phases de conception d'un circuit intégré. La première chose requise pour commencer une conception, est les spécifications de ce qui a besoin être fait (des `specs '). Donné des spécifications, l'approche la plus générale adoptée est représentée sur un schéma 1.1. [8]

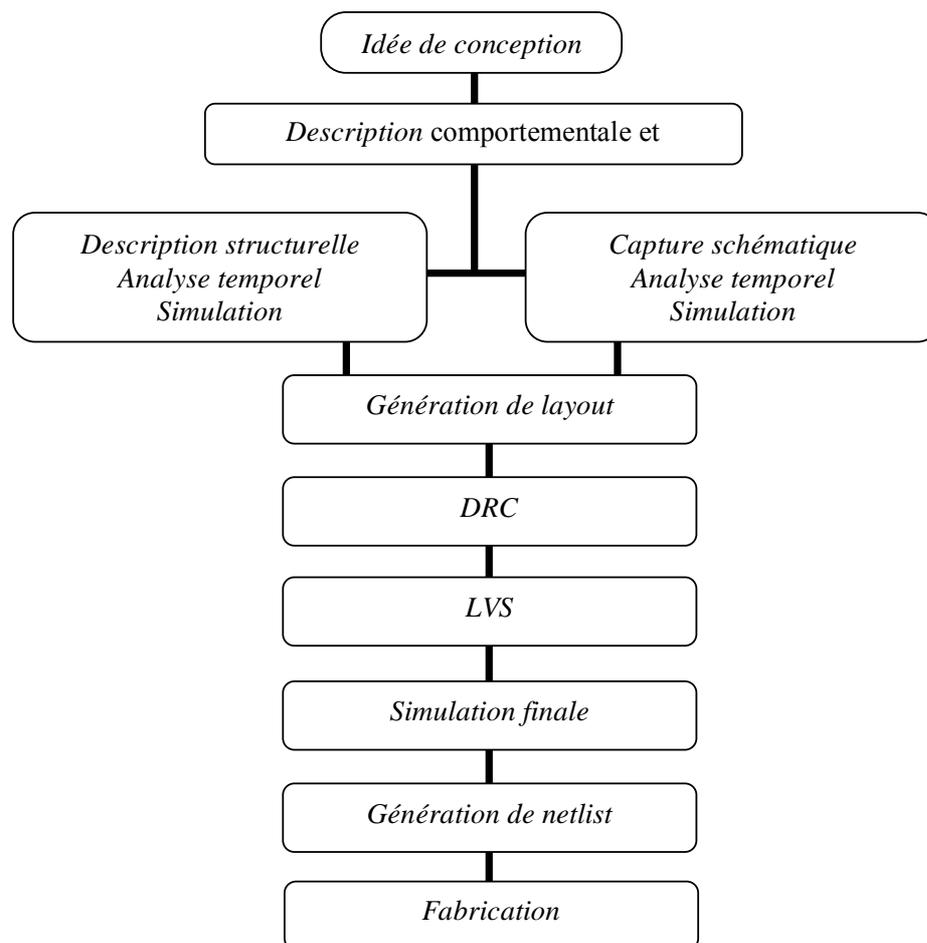


Figure 3.1: Le flot de conception generique des circuits integrés

III.3.1. Idée de conception

Basé sur les spécifications données, l'équipe de conception forme une idée générale au sujet de la solution au problème. Des décisions au niveau de système sont prises concernant la conception et un consensus général est atteint concernant les blocs fonctionnels principaux

qui entrent dans la fabrication du chip. À la fin de cette étape, un bloc général de diagramme de la conception est convenu. Les outils de CAO ne sont généralement pas nécessaires à cette étape.

III.3.2. Description comportementale et simulation

III.3.2.1. Description comportementale

Des langages de description de matériel (HDLs) sont employés pour modeler l'idée de conception (schéma fonctionnel). Les détails du circuit et les composants électriques ne sont pas indiqués. Au lieu de cela, le comportement de chaque bloc au niveau le plus élevé de l'abstraction est modelé.

Des simulations sont alors exécutées pour voir si les blocs fonctionnent en effet comme prévu et le système entier exécute dans l'ensemble. Les descriptions comportementales sont importantes en tant qu'elles corroborent l'intégrité de l'idée de conception.

III.3.2.2. Description structurale

Des programmes HDL sont employés pour décrire la construction de la conception au prochain niveau de l'abstraction. Dans cette étape, divers composants électroniques et détails du circuit qui disparaissent dans la fabrication des divers blocs du système sont modelés. Ceux-ci peuvent inclure les primitifs comme les portes logiques et les très grands blocs préconçus appelés cellules standard. Des simulations sont alors exécutées sur chaque bloc séparément pour tester son exécution. Une fois que ceci est terminé, tous les blocs sont interconnectés et une simulation au niveau système est exécutée. Une réussite s'assure qu'un circuit électronique peut en effet être construit pour apparier la description comportementale.

III.3.2.3. Description schématique

Une description schématique est semblable à une description structurale. Dans un contexte rigoureux, le schéma peut être considéré comme le niveau plus bas d'abstraction, doit être considérée en tant que la topologie des circuits. Dans la plupart des cas, les outils graphiques sont utilisés pour construire les circuits dans une planche virtuelle comme l'environnement. Juste comme dans l'étape précédente, de divers composants électroniques comme les portes de logique et les cellules standards sont placés et interconnectés pour faire les blocs de circuit. C'est une forme alternative de la description structurale avec plus

d'attention étant payée à la topologie. Une simulation réussie assure qu'un circuit électronique qui apparie la description comportementale peut être construite.

III.3.2.4. Simulation

La simulation est l'une des étapes clés d'un flot de conception. C'est à ce niveau que le design d'un circuit au niveau RTL est vérifié d'un point de vue fonctionnel, avant la synthèse. Cette opération, qui a toujours existé en conception électronique, est censée vérifier que les fonctions d'un circuit prévues dans les spécifications sont bien réalisées par la conception écrite dans un langage spécifique (VHDL, Verilog) au niveau registres. Pour y parvenir, les ingénieurs de vérification écrivent des suites de stimulus, des scénarios de test, qu'ils appliquent au fichier de conception. L'objectif étant ensuite de comparer les réponses obtenues en sortie du simulateur aux réponses attendues selon la fonction testée.

III.3.3. Description Layout

C'est le niveau le plus bas possible de l'abstraction dans la description du circuit. Dans cette étape, le circuit est décrit en respectant le silicium et d'autres matériaux qui entrent dans la fabrication du circuit intégré (CI). Une attention extrême est prêtée à la géométrie des composants. En d'autres termes, c'est à quoi le CI ressemblerait, si nous jetions un coup d'œil plus étroit à lui à travers le microscope. Des éléments parasites doivent être extraits à chaque étape. Le placement et l'interconnecte de chaque entité individuelle de la précédente étape doit d'être indiqué. Des simulations étendues de chaque bloc sont exécutées dans cette étape. Un plan de la puce est fait et les blocs sont routés ensemble. Après la phase de routage, une simulation de la conception entière est exécutée.

III.3.3.1. Les cellules standards

Au niveau de la conception des standards cellules; les portes de base ou les blocs, Par exemple multiplexeurs, additionneurs, flip-flops, et les fonctions logiques de base, sont fournis par un vendeur puce. Donc les cellules sont déjà conçues et peuvent être placées dans notre disposition.

III.3.3.2. Le compactage (le layout compact)

Dans la conception full-custom tous les circuits sont créés à la main par un concepteur; qui donne la possibilité d'utiliser des styles de circuits spéciaux et de dimensionnement arbitraire des transistors. Tanque les temps de conception est beaucoup plus longue que dans

la de conception basés sur les cellules standards. La conception full custom est principalement utilisée dans les parties critiques.

Le compactage est un outil très utile quand des layout doivent être ajustées pour la minimum superficie sans transgresser les règles de la conception. Le compactage ne change pas la géométrie entière de layout. L'outil de compactage serre le layout à l'espacement minimal des règles de conception. Il fait ceci en alternance des directions de compactage horizontales et verticales jusqu'à ce qu'aucun autre espace ne puisse être trouvé sans transgresser les règles de conception.

III.3.4. DRC

La conception bascule entre le layout, le LVS et le DRC. La vérification des règles de conception (Design Rule Check) dépend souvent de la technologie de processus finale la laquelle est utilisé pour fabriquer la puce. Le contrôle des règles de conception s'assure que les règles établies par la technologie de processus de fabrication ne sont pas violées. Un bon exemple serait, certains les processus ont besoin des transistors, des fils et de polysilicium pour être d'une certaine largeur minimum. Le layout dessiné devrait être basé sur telles contraintes.

III.3.5. LVS

Après l'accomplissement de la vérification du DRC, Le contrôle du layout contre schématique (Layout Versus Schematic) est exécuté. Ceci s'assure que le layout est en conformité avec le schéma. Le processus de conception déplace en arrière et avant entre layout, LVS et DRC.

III.3.6. Les fichiers CIF, GDS II

C'est l'étape finale avant la fabrication. La plupart des installations de fabrication reçoivent soumissions dans le format de CIF ou de GDS II. Ce sont des fichiers générés par l'ordinateur qui décrivent le Layout de CI. Ils incorporent tous les détails nécessaires pour fabriquer la puce. Un travail d'équipe de conception de puce vient à une pause momentanée jusqu'au premier prototype des puces arrivent de l'installation de fabrication. Après que le silicium arrive, la phase de test commence. Les premiers essais de base incluent une vérification fonctionnelle de puce dans des conditions idéales. Une fois que la fonctionnalité est vérifiée, les essais étendus sont exécutés par de divers paramètres électriques s'étendant

comme la fréquence de fonctionnement et la température aux extrémités. Le comportement de la puce est alors documenté dans manuel d'utilisateur de la puce.

III.4. Les ASIC et les circuits logiques programmables

Le circuit intégré (CI), aussi appelé puce électronique, est un composant électronique reproduisant une ou plusieurs fonctions électroniques plus ou moins complexes, intégrant souvent plusieurs types de composants électroniques de base dans un volume réduit, rendant le circuit facile à mettre en œuvre. Un circuit intégré comprend sous des formes miniaturisées principalement des transistors, des diodes, des résistances, des condensateurs, plus rarement des inductances car elles sont plus difficilement miniaturisables.

Les circuits intégrés numériques les plus simples sont des portes logiques (et, ou, non), les plus complexes sont les microprocesseurs et les plus denses sont les mémoires. Il existe une très grande variété de ces composants divisés en deux grandes catégories: PLD et ASIC. Une famille importante des circuits intégrés est celle des circuits logiques programmables (PLD pour programmable logic circuit), on parle plus juste des circuits à architecture programmable ou encore de circuits à réseaux logiques programmables. (PLD, CPLD et FPGA). Parallèlement à ces circuits, on trouvera les ASIC (Application Specific Integrated Circuits) qui sont des composants où le concepteur intervient au niveau du dessin de la pastille de silicium en fournissant des masques à un fondeur.

III.4.1. Les ASIC

Le Circuit Intégré Spécifique à une Application (ASIC) est une des manières de réaliser un système de calcul matériel. Pour chaque application, un circuit différent est créé, soit en le construisant entièrement, soit en configurant une grille d'éléments préconstruits. L'avantage des ASICs sur les circuits reconfigurables, pour une même technologie, est évidemment leur rapidité, puisque les connexions sont créées physiquement plutôt que programmées. Toutefois, leur force est également leur faiblesse, puisque leur conception nécessite plus de temps, étant donné qu'en tout cas une couche de métal doit être apposée. De plus, ils ne sont en aucun cas programmables, et une erreur dans le design implique un changement complet du circuit. Quatre grandes classes d'ASICs sont présentées ici, le graphique 2.1 les exposant avec un niveau de complexité croissant de gauche à droite. Nous allons les détailler dans l'ordre chronologique de leur apparition.

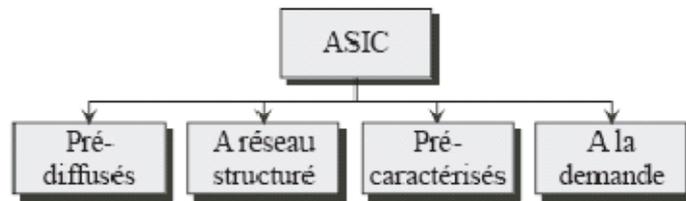


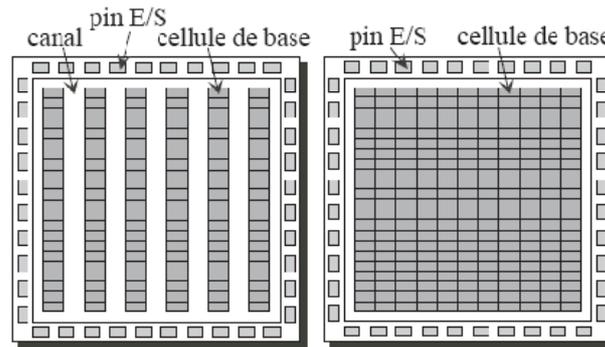
Figure 3.2 : Les différents types de circuits ASIC.

1. A la demande

Pour la création d'un ASIC à la demande, communément appelé Full-Custom, les ingénieurs contrôlent chaque élément du circuit fabriqué. Ils dessinent les transistors, les placent et les connectent, sans qu'aucune contrainte prédéfinie n'existe concernant le placement des éléments. La taille de chaque transistor peut être changée afin d'optimiser le fonctionnement du système. L'avantage de cette approche est l'efficacité du circuit final en termes de quantité de silicium et de vitesse. En effet, il est possible de placer les éléments de manière optimale, sans aucune perte de place, et de le faire en tenant compte des contraintes de temps d'exécution. Par contre, si une erreur s'est subrepticement glissée dans la conception, le dessin du circuit entier doit être revu, dans un long et coûteux processus de correction. Ce type d'ASIC n'est donc pas du tout adapté au prototypage, étant donné le temps et l'investissement nécessaire à chaque prototype. [8].

2. Prédifusés

Dans le milieu des années 60, Fairchild Semiconductor créa le précurseur des tableaux de portes, sous la forme d'un circuit appelé Micromatrix, composé d'une centaine de transistors. Les ingénieurs devaient dessiner à la main les interconnexions sur des feuilles de plastique, pour réaliser un circuit spécifique. Ensuite, en 1967, un nouveau circuit, appelé Micromosaic, et contenant quelques centaines de transistors, fut proposé par la même société, avec cette fois-ci un programme capable de générer automatiquement les interconnexions en fonction des spécifications des ingénieurs. Ce sont toutefois les compagnies telles qu'IBM et Fujitsu qui développèrent le concept d'ASIC prédifusé (ou à tableau de portes). Dans le cas des mers de portes les vendeurs fabriquent un circuit composé d'un tableau de cellules de base telle que des transistors [9]



(a) Un tableau de portes à canal

(b) Une mer de portes

Figure 3.3 : Deux types de circuits prédifusés.

3. Pré-caractérisés

Les ASIC pré-caractérisés (en anglais Standard Cellont vu le jour au début des années 80, dans le but de combler les lacunes des circuits prédifusés. Contrairement à ces derniers, le silicium n'est pas prédifusé avec des éléments de base. Le vendeur propose ici une librairie de cellules de base, ainsi que des modules plus complexes tels que des RAMs, des microprocesseurs, et bien d'autres. Le développeur utilise alors ces divers éléments pour créer une liste de modules et de leurs interconnexions décrivant son système, et le vendeur se charge ensuite de créer le circuit intégré.

L'avantage de cette approche réside dans l'optimisation des modules fournis par le fabricant. Le placement de ces modules est réalisé de manière à également optimiser les performances, tout en minimisant la place nécessaire sur le silicium. Dès lors, la solution pré-caractérisée est très proche de l'optimal obtenu grâce à l'approche fullcustom, et nécessite un temps de développement nettement moindre.

4. A réseau structuré

Comme nous le verrons à la section suivante, les circuits reconfigurables ont l'avantage d'être d'excellents candidats pour les applications industrielles de petit volume ainsi que pour le prototypage, un design pouvant être immédiatement testé. Toutefois, ils ne peuvent, en termes de vitesse, rivaliser avec un circuit ASIC spécialement conçu pour une application. C'est pourquoi en 2003 le dernier-né des circuits programmable a fait son apparition³ : l'ASIC à réseau structuré.

Un ASIC à réseau structuré consiste en un circuit composé de différents éléments semblables à ceux que l'on peut trouver dans un FPGA standard, tels que look-up tables 4,

inverseurs, flip-flops, et bien d'autres. Toutefois, à l'inverse du FPGA, ces différents blocs ne sont pas connectés entre eux. Par analogie avec les mers de portes, on pourrait les appeler mers de macros, des modules de complexité supérieure étant prédiffusés. Le circuit de base créé, il reste donc des couches de métal disponibles qui permettent de réaliser la connectique nécessaire à la réalisation d'un design particulier [8].

III.4.2. Les circuits logiques programmables

Les circuits programmables sont apparus en 1970, et depuis se sont complexifiés de manière spectaculaire. Il existe dans la littérature plusieurs manières de les classer, certains considérant que les CPLDs (Complex Programmable Logic Device) ne sont pas un sous-ensemble des PLDs (Programmable Logic Device). Nous choisirons toutefois l'approche illustrée à la figure 3.4 consistant à séparer les PLDs en deux sous-classes, à savoir les SPLDs (Simple Programmable Logic Device) et les CPLDs.

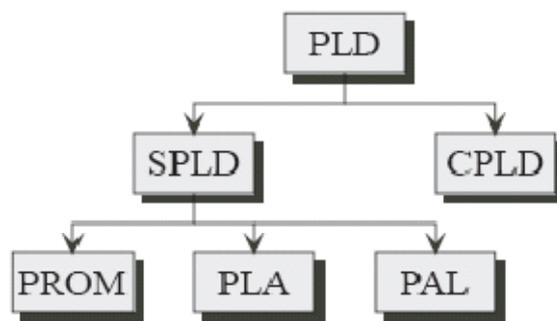


Figure 3.4 : Une classification des circuits logiques programmables.

1. SPLD

Les SPLDs (Simple Programmable Logic Device), dans une description haut niveau, sont composés d'une grille de portes ET et d'une grille de portes OU, les deux étant reliées. Les entrées du système peuvent être connectées aux portes ET, et le résultat des portes OU correspond à la sortie du système (Figure 3.5). Dans ces circuits, les connexions sont préexistantes, les différentes lignes étant reliées par des fusibles, des transistors EPROM, ou des transistors EEPROM. En brûlant certains de ces fusibles, ou en programmant les transistors, il est alors possible de réaliser différentes fonctions logiques.

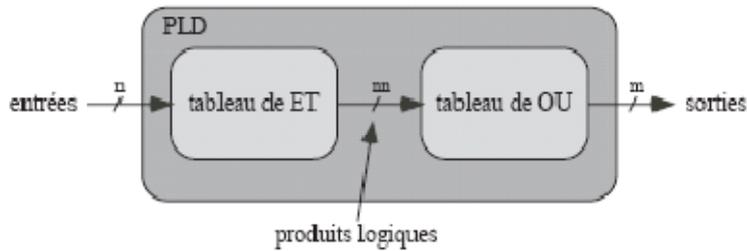


Figure3.5: L'architecture d'un SPLD.

a. PROM

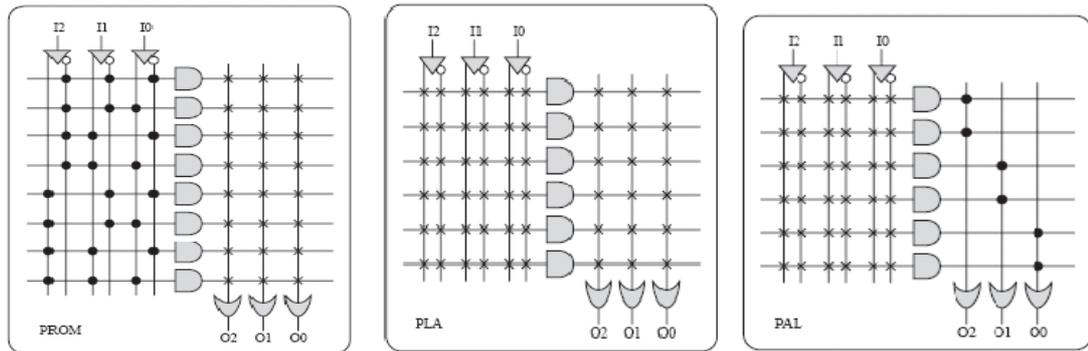
Les premiers circuits programmables à avoir vu le jour sont les PROMs (Programmable Read-Only Memory) (Figure 3.6), le terme PROM étant introduit en 1970. Ces circuits sont des mémoires accessibles en lecture, qui contrairement aux ROMs, sont programmables. En effet, une ROM est livrée déjà configurée, et ne peut être accédée qu'en écriture, alors qu'une PROM est une mémoire vide, qui peut être écrite, une seule fois, par l'utilisateur. Après cette unique écriture, une PROM se comporte exactement comme une ROM.

b. PLA

Les PLA (Programmable Logic Array) apparurent aux environs de l'année 1975 dans le but de pallier les limitations des Promis. En effet, dans un PLA (Figure 2.14(a)), contrairement à une PROM, toutes les interconnexions peuvent être programmées, ce qui en fait le PLD le plus général. Il est alors possible de définir les produits et les sommes, les rendant particulièrement efficaces lorsque plusieurs sorties utilisent les mêmes produits. Une amélioration s'accompagnant souvent de désagréments, citons un des désavantages du PLA comparé à une PROM : étant donné que les deux tableaux (ET/OU) sont programmés, le temps de propagation d'un signal de l'entrée à la sortie est nettement plus important que lorsqu'un seul tableau l'est.

c. PAL

Vers la fin des années 1970, les PALs (Programmable Array Logic) furent introduites afin de contrer le problème de vitesse de propagation des PLAs. Dans un PAL (Figure 3.6 (b)), les connexions entre les portes ET et OU sont fixes, et les connexions entre les entrées et les portes ET peuvent être programmées. L'avantage des PALs sur les PLAs est donc leur rapidité, mais elles présentent l'inconvénient de n'avoir qu'un nombre limité de produits pour chaque porte OU.



(a) L'architecture fonctionnelle d'une PROM (b) L'architecture d'un PLA (c) L'architecture d'un PAL

Figure 3.6 : Architectures d'une PROM , d'un PLA et d'un PAL.

2. CPLD

Les CPLDs (Complex Programmable Logic Device), apparus au début des années 80, sont donc le résultat de l'évolution des PLDs. Ils permettent l'implémentation de systèmes nettement plus complexes, et sont composés d'éléments de base programmables, connectés entre eux par un réseau d'interconnexions relativement simple. Ces éléments de base sont du type SPLD, et peuvent, comme dans le cas de la famille MAX3000 d'Altera , être composés d'un tableau de portes ET programmables et de portes OU fixes (une sorte de PAL), ainsi que d'un registre. La technologie de programmation des CPLDs dépend évidemment du constructeur, et peut être de type EPROM, EEPROM, FLASH ou SRAM.

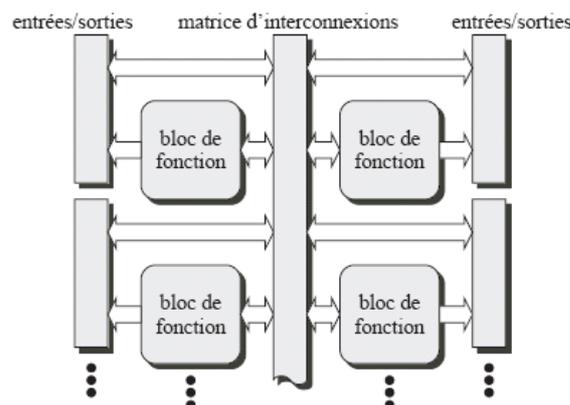


Figure 3.7 : L'architecture d'un CPLD

2.1.FPGAs

Les circuits FPGAs permettent d'implémenter des systèmes numériques aussi complexes que ceux réalisés jusqu'alors grâce aux ASICs, tout en ayant le grand avantage de pouvoir être programmés électriquement. Ils sont principalement composés d'un tableau

d'éléments plus ou moins complexes pouvant être configurés. Ainsi que d'un réseau complexe de connexions également configurables [10] (Figure 3.8).

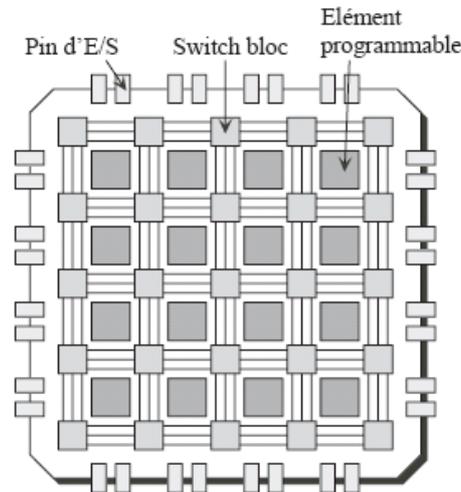


Figure 3.8 : L'architecture générale du FPGA.

III.5. Conclusion

Les outils de CAO sont une brique fondamentale dans la conception des circuits numériques car le choix des outils CAO devient critique. Un bon outil CAO peut rigoureusement ramener le temps requis à concevoir un grand bloc numérique

Les outils de CAO devraient faciliter la conception de chip dans une période courte le plus possible. En même temps la conception doit être fiable et robuste. Selon le choix de l'outil, et en tirant profit de ses caractéristiques essentielles. Une solution optimale peut être atteinte, par la recherche d'un flot de conception numérique optimal dont le but est d'employer un ensemble d'outils CAO suivant un processus de conception, qui doit être rapide, bon marché, fiable et acceptable. Un outil particulier est le meilleur adapté à chaque étape de conception, puisque chaque outil a quelques caractéristiques essentielles qui le rendent avantageux pendant une certaine phase de conception.

De nombreuses familles de circuits sont apparues depuis les années 70 avec des noms très divers suivant les constructeurs. Il est difficile de comparer les ASIC et les CPLD/FPGA. Les méthodes et temps de développement ne sont pas du tout les mêmes. L'utilisation des ASIC va surtout être justifiée par la production en grande série du circuit à réaliser.

Chapitre IV : Conception d'une machine d'états à base des latches C²MOS

IV.1. Introduction

La complexité des projets électroniques atteint un niveau inégalé et les équipes de design doivent aujourd'hui faire face à un nombre très important de phénomènes physiques.

Aujourd'hui il y a plusieurs centaines de suites d'outil de CAO sur le marché, qui peut nous aider dans la conception de circuit intégré (IC). Basé sur le besoin, les conditions, une équipe de conception à besoin de choisir convenablement le bon ensemble d'outils de CAO pour le développement.

Nous avons utilisé l'outil de CAO Cadence Virtuoso pour concevoir notre circuit. La plateforme Cadence Virtuoso permet de concevoir dans des délais optimaux tous les circuits intégrés, les projets analogiques/digitaux/mixtes et RF.

IV.2. La plateforme Cadence Virtuoso

La plateforme Cadence Virtuoso est un environnement intégré permettant la conception de circuits ASIC en respectant un cahier des charges de très haut niveau et des délais exigeants. Virtuoso respecte les contraintes spécifiques, il permet la simulation en mode multiple, le routage accéléré et l'analyse avancée du circuit intégré. Virtuoso autorise la méthodologie d'optimisation "meet-in-the-middle" qui combine la vitesse de conception en mode "top-down" et la précision en mode "bottom-up".

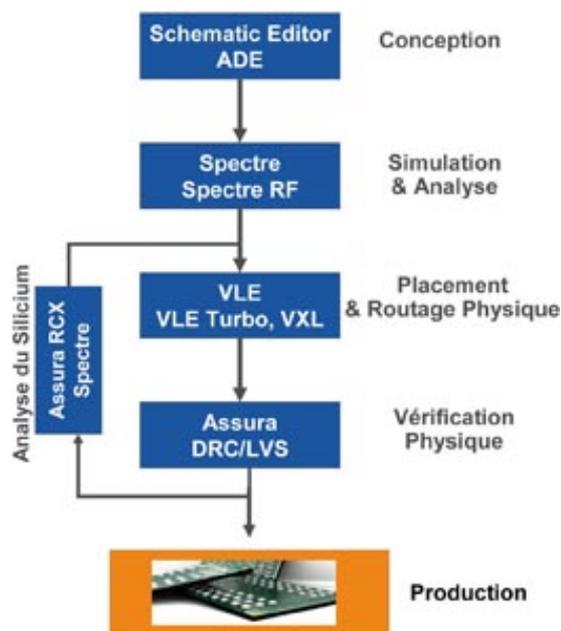


Figure 4.1 : La plate forme Cadance Virtuoso

IV.2.1. Virtuoso Schematic Editor

L'éditeur Virtuoso Schematic Editor, anciennement Virtuoso Composer, constitue l'environnement de conception et d'édition pour la plateforme Virtuoso. Depuis la définition architecturale à partir des langages standards de représentation Verilog, VHDL et C jusqu'à l'implémentation des composants au niveau transistor, Virtuoso Schematic Editor nous accompagne à chaque étape de conception de notre projet électronique.

IV.2.2. Virtuoso Spectre Circuit Simulator

Virtuoso Spectre Circuit Simulator propose un environnement de simulation complet, rapide et efficace pour optimiser les circuits analogiques et mixtes. L'analyse est conduite de façon détaillée au niveau transistor. Le concepteur électronique doit vérifier le bon fonctionnement de tous les composants du circuit suivant les spécifications du cahier des charges, tout en respectant les tolérances de fabrication les plus exigeantes. Avec les modèles SPICE courants, la vérification peut s'avérer mission impossible compte tenu de la complexité des projets et du délai imposé. C'est la raison pour laquelle l'utilisation d'une plateforme intégrée et performante de conception et de simulation devient un avantage décisif.

Le simulateur Virtuoso Spectre est un outil de conception analogique indispensable au quotidien dans nos projets électroniques. La prise en main de l'outil est intuitive. Le très haut niveau de précision facilite le débogage des siliciums. Pour réussir une simulation, une bonne compréhension des principes électroniques fondamentaux est un avantage évident : Virtuoso Spectre nous accompagne précieusement tout au long de notre projet en rappelant les éléments théoriques de fonctionnement des circuits.

IV.2.3. Virtuoso Layout Editor

Virtuoso Layout Editor est l'outil de routage physique qui s'intègre dans l'environnement de conception Virtuoso. Il supporte l'implémentation physique de tous les éléments digitaux, analogiques et mixtes dans le circuit, au niveau des composants, des cellules et des sous-ensembles.

IV.3. La conception du circuit de machine d'états

La machine d'états est un bloc de la partie numérique d'un circuit ASIC, ce circuit ASIC est une sonde ultrasonique (**UltraSonic Low Voltage SAP for General Electric Medical**). Ce projet était réalisé pendant un stage pratique en Allemagne dans l'Université

Technique de Berlin. La technologie AMS 0.35µm est utilisée pour implémenter la machine d'états.

IV.3.1. La spécification

La conception d'un circuit machine d'états commence par la spécification, le fonctionnement du FSM doit être identique à une description VHDL, autrement dit le circuit est décrit par un programme VHDL, Ce programme a été simulé dans le simulateur **Modelsim** (annexe B)

IV.3.2. Le diagramme des états.

Le diagramme des états est une présentation graphique de fonctionnement de la machine d'états .il est extrait du programme VHDL.

La machine d'états comporte 7 états, chaque état présent fait une transition vers un état futur si seulement s'il y a une condition qui s'active (signal de transition) pour cet état futur, sinon la machine reste dans son état présent. Initialement la machine est dans un état présent Q0, si aucune transition n'est active. Mais si les signaux (de transition) **ncreset Q6** et **Q5** s'activent ensemble, la machine d'états passe de l'état présent Q0 à l'état futur Q1, et ainsi de suite, si les conditions de transition pour un état futur sont actives la machine fait une transition de l'état présent vers cet état futur.

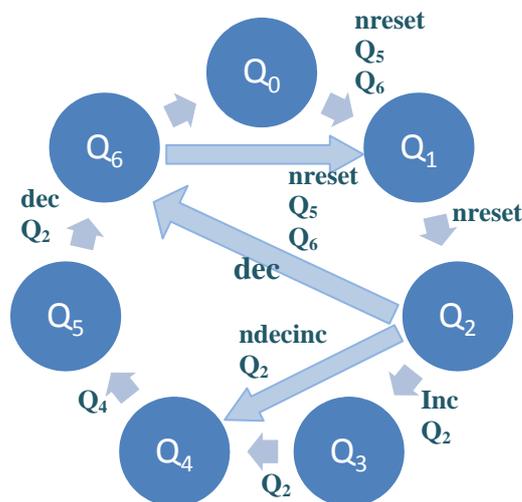


Figure 4.2 : Graphe des états de la machine d'états

IV.3.3. La structure de la machine d'état

IV.3.3.1. Le bloc de la machine d'état

Le bloc de la machine d'états est basé sur le circuit principal de la machine d'états, il englobe aussi des circuits périphériques qui délivrent les signaux de transitions et d'horloge.

- **-7 inverseurs**: dont leur fonction est d'inverser des signaux exigés.
- **-2 portes NAND** : la première qui délivre le signal **incint** et la deuxième délivre **decint**.
- **-1 porte NOR** a deux entrées **incint** et **decint**, et sa sortie est le signal **incdecint**.
- **-3 bascules RS**. la première bascule génère le signal de transition **inresetint** à partir des deux entrées **row-nreset** et $\overline{Q_1}$, la deuxième bascule à deux entrées **row-inc** et $\overline{Q_3}$ et sa sortie est l'une des entrées de la première NAND et la troisième bascule à deux entrées **row-dec** et $\overline{Q_6}$ et sa sortie sera l'une des deux entrées de la deuxième NAND.
- **3 multiplexeurs** à deux entrées et un signal de commande.

La figure 4.3 présente le symbole du circuit et la figure 4.4. La structure interne de bloc qui comporte la machine d'états et les sous bloc périphériques.

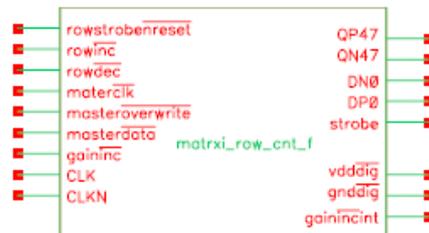


Figure 4.3 : Symbole du bloc Matrix -row.

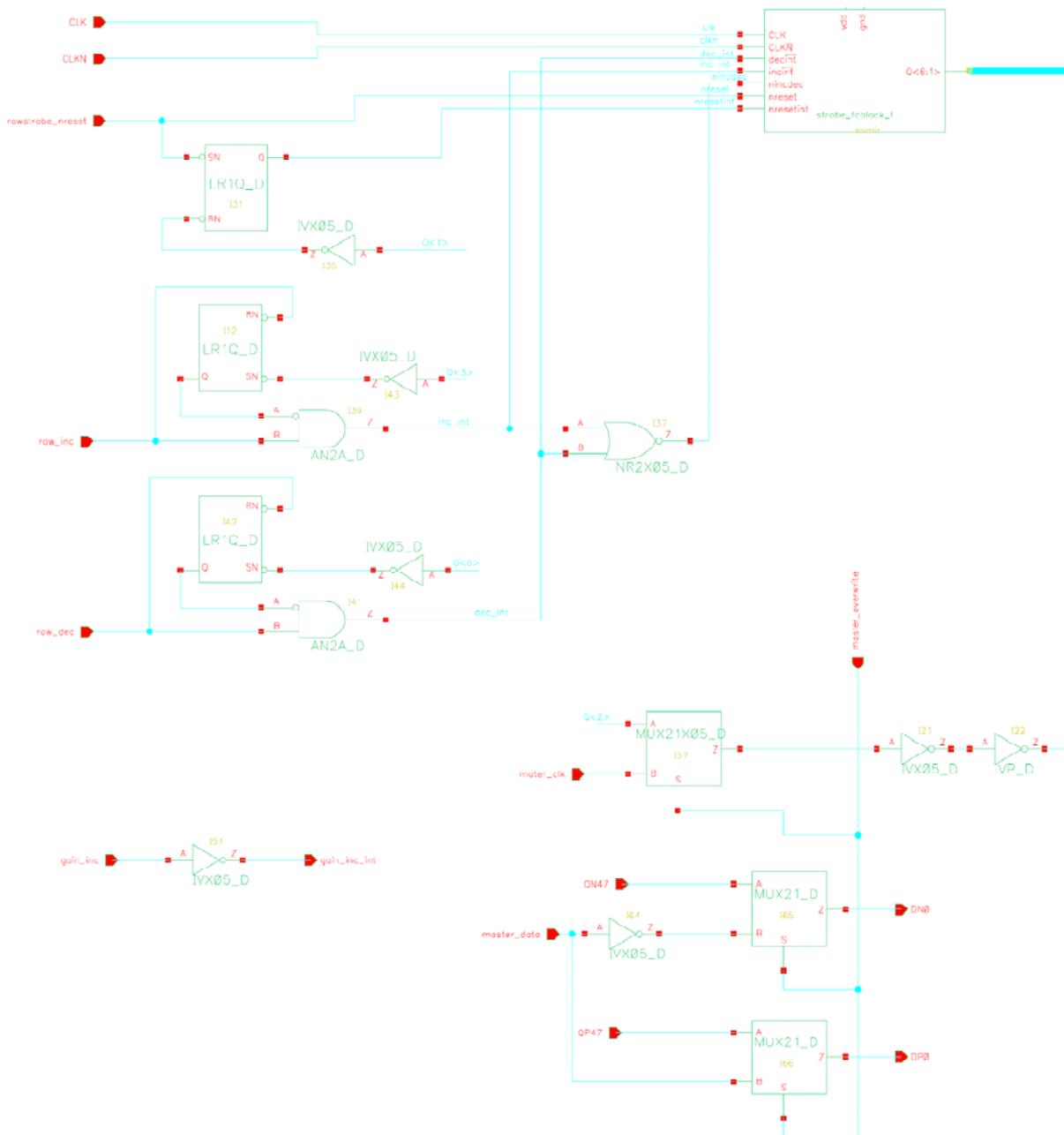


Figure 4.4: Le bloc qui contient la machine d'états et sous blocs périphériques.

IV.3.3.2. La machine d'état

Le symbole de circuit machine d'états est présenté dans la figure suivant :

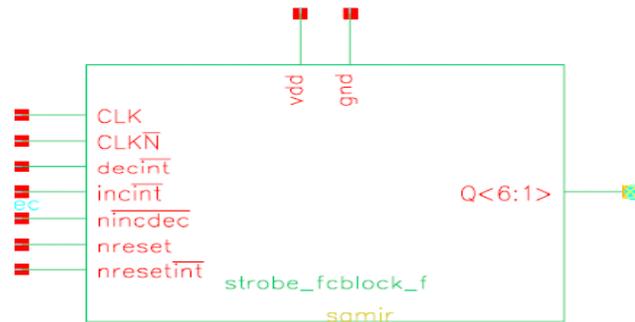


Figure 4.5: circuit de machine d'états

Le schéma interne de circuit de la machine d'états est montré dans la figure 4.6. dont tous les paramètres des transistors W et L sont identique et minimales avec $w=0.35\mu\text{m}$, $L=0.5\mu\text{m}$. Ce choix est fait pour éviter l'effet de bruit.

Le circuit synchronisé par deux signaux d'horloge CLK et CLKN qui exploite la technique de C²MOS. Il est constitué de 6 registres et chaque registre dynamique est implémenté à l'aide de deux latches C²MOS. Le fonctionnement de registre sera expliqué plus tard.

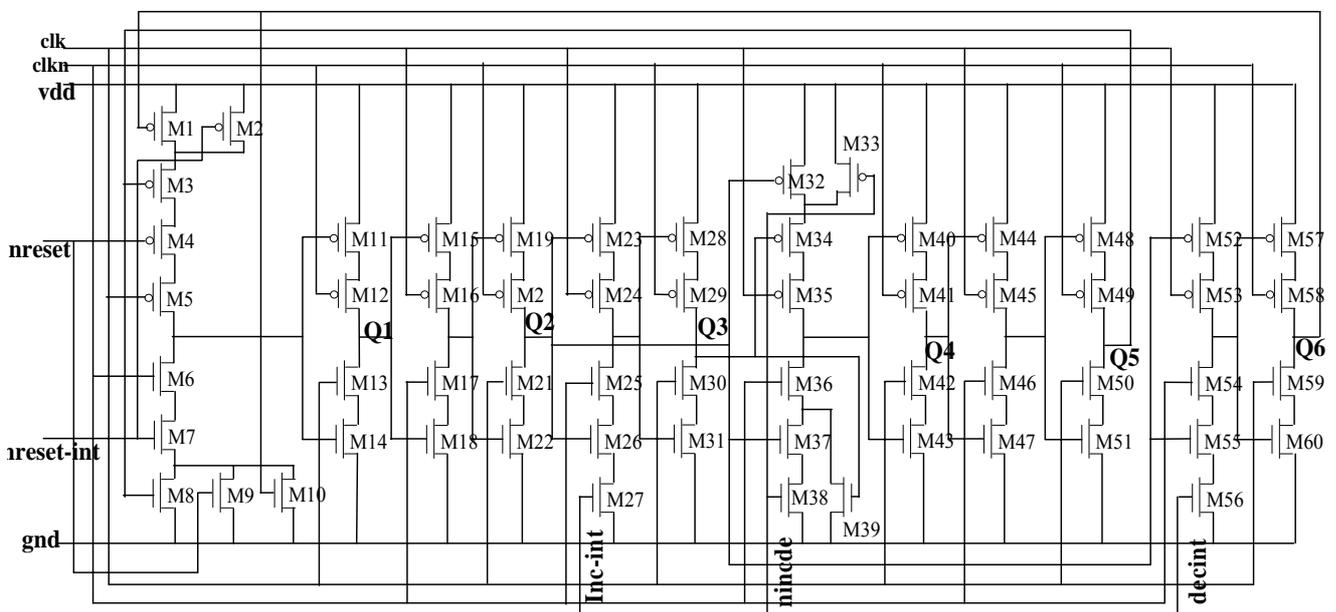


Figure 4.6: Schéma interne de circuit de machine d'états

IV.3.4. Le fonctionnement du circuit de la machine d'états

Pour comprendre le fonctionnement du circuit de la machine d'états, il faut comprendre le fonctionnement de lache CMOS qui compose le registre. Le lache CMOS est implémenté par un simple inverseur CMOS, ou par une porte CMOS complexe (inverseur CMOS complexe)

IV.3.4.1. Le lache C²MOS (inverseur simple synchronisé)

Un inverseur C²MOS de base est montré sur la figure 4.7. Ceci est construit en utilisant une seule paire complémentaire Mp1 et Mn1 de la façon habituelle.

Quand CLK=1, les deux FETS synchronisés sont en activité et l'entrée logique commandes les FETS Mp1 et Mn1. si $V_{IN}=0V$, puis $V_{OUT}=V_D$ tandis que l'entrée $V_{IN}=V_{DD}$ à comme conséquence $V_{OUT}=0v$. Quand l'horloge fait une transition au CLK=0 ? La sortie est pilotée dans un état Hi-Z.

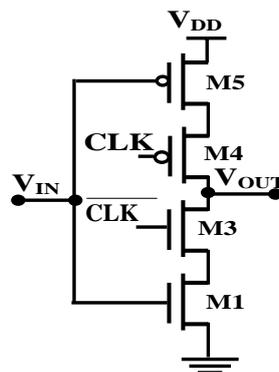


Figure 4.7: L'inverseur C²MOS

IV.3.4.2. La porte logique complexe C²MOS

Des portes logiques complexes sont construites en utilisant l'inverseur CMOS comme une base. Le NMOS et le PMOS sont placées pour agir en tant que des transistors du passage. Dans l'ordre de construire une porte logique complexe, on remplace le NMOS de l'inverseur simple par un réseau de NMOSs qui sont connectés pour fonctionner comme un grand commutateur. De même, nous substituerons un réseau PMOSs au simple PMOS utilisé dans l'inverseur, et visualisons le réseau PMOS comme commutateur "géant". Afin d'assurer le fonctionnement électrique approprié, cependant, nous devons prendre le soin de sorte que le fonctionnement de réseau NMOS complète le fonctionnement de réseau PMOS. Ceci signifie que si un réseau est un commutateur fermé, l'autre doit être ouvert.

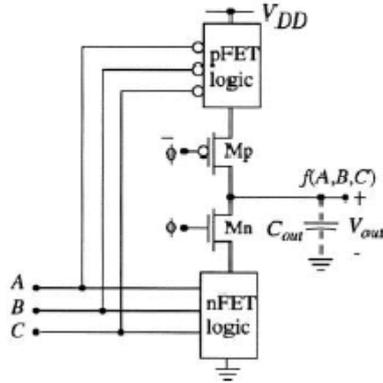
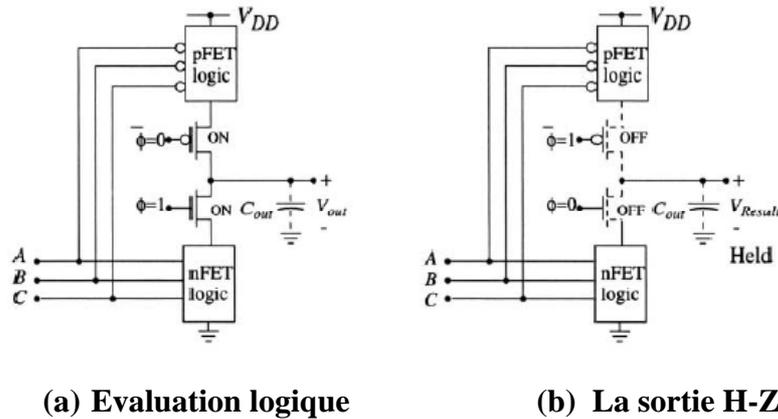


Figure 4.8: Fonction logique complexe C²MOS

Le fonctionnement de la porte peut être compris en étudiant les effets de l'horloge $\phi(t)$ comme récapitulé sur la figure 4.9.

Quand l'horloge est à un niveau $\phi=1$ comme dans la figure 4.9 (a), les deux transistors Mn et Mp sont en activité. Ceci connecte les deux réseaux logiques au nœud de sortie, et la porte se dégrade à son circuit équivalent statique; la différence principale est les temps plus longs de commutation dus aux parasites supplémentaires. Après que les coupures transitoire (transients) se soient délabrées, le condensateur C_{out} de sortie sera chargé à une tension $V_{out}=0V$ ou $V_{out} =V_{DD}$. La figure 4.9(b) montre le circuit, quand $\phi=0$ et les deux transistors Mn et MP sont ouvert. Ceci isole le nœud de sortie des deux réseaux logique, et la valeur de $V_{out}=V_{result}$ est retenue sur C_{out} .



(a) Evaluation logique

(b) La sortie H-Z

Figure 4.9 : fonctionnement de la porte logique C²MOS

IV.3.4.3. Le registre C²MOS

Considérer le circuit montré dans la figure 4.10. Ce circuit est basé sur le concept du latch C²MOS précédent. Le registre C²MOS a deux parties maître/esclave. Si la partie maître fonctionne, il stocke le niveau logique dans une capacité de nœud de sortie par rapport à la partie esclave et en H-Z. et vice versa. Quand la partie maître est en H-Z et la partie esclave fonctionne, elle préleve le niveau logique stocké dans le nœud de sortie du maître.

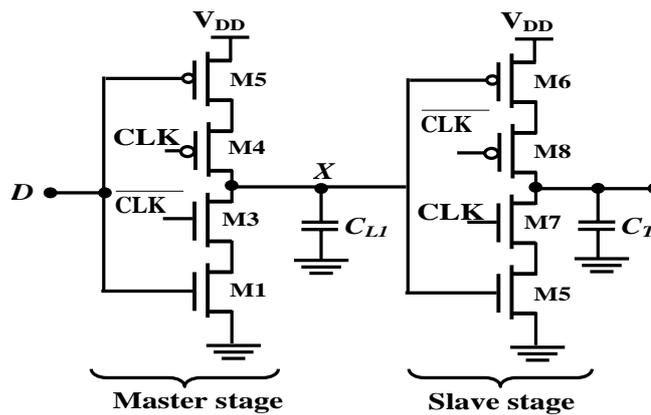


Figure 4.10: registre C²MOS.

Le premier et Le deuxième registres sont montrés dans la figure 4.11 (a) et 4.11 (b). Chaque registre de la machine d'états est semblable au registre 1 (registre 4) ou bien il est comme le registre 2 (registre 3, 5 et 6).

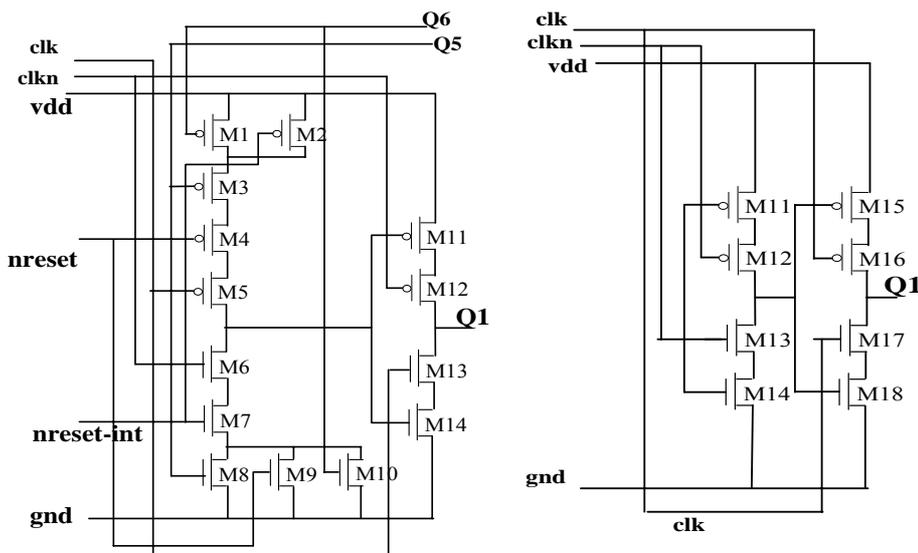


Figure 4.11: Registre 1 (a), Registre 2 (b)

IV.4. La simulation

Après que la description au niveau transistor d'un circuit soit accomplie en utilisant Schematic Editor, l'exécution électrique et la fonctionnalité du circuit doivent être vérifiées utilisant un outil de simulation. La simulation détaillée au niveau transistor de notre conception sera la première validation détaillée de son fonctionnement, par conséquent, il est extrêmement important d'accomplir cette étape avant de procéder aux étapes suivantes d'optimisation de conception. Basé sur des résultats de simulation, le concepteur modifie

habituellement certaines des propriétés de dispositif (telles que le rapport de largeur à longueur de transistor) afin d'optimiser l'exécution.

La phase initiale de simulation sert également à détecter certaines erreurs de conception qui ont été créées pendant l'étape d'entrée schématique. Elle est tout à fait commune pour découvrir des erreurs telles qu'une connexion absente ou un croisement non voulu de deux signaux dans le schéma. Certaines de ces erreurs (par exemple, nœuds flottants) peuvent être détectées même avant la simulation, en serrant l'icône le contrôle et sauver dans la fenêtre schématique.

On a fait des simulations temporelles sur le circuit à l'aide d'un circuit de test présenté dans la figure 4.14. Les simulations examinent le fonctionnement de la machine d'états, et l'influx des signaux de transitions *inc*, *dec*, *incdec*, *reset*, *nreset*, et les signaux d'horloge *CLK* et *CLKN* sur les états de sortie de la machine Q_1 Q_2 Q_3 Q_4 Q_5 Q_6 .

IV.4.1. Le circuit de test bench

Le circuit de Test bench est construit autour du circuit FSM. Les signaux d'entrées sont générés à l'aide des sources de tension de signal carré (V_1 , V_2 , V_3 , V_4) avec la possibilité de varier la période de signal. Les signaux générés sont les deux signaux d'horloge *CLK* et *CLKN* et aussi les signaux de transition *reset*, *dec*, *inc*. Les deux signaux *nreset* et *nincdec* sont générés indirectement à l'aide des trois signaux des transitions précédentes. Le signal *nreset* est la sortie *Q* d'une bascule RS, dont les entrées *R* et *S* de cette bascule sont l'état Q_1 et le signal de transition *nreset*. Le signal *nincdec* est une fonction NOR à deux entrées *dec* et *inc*.

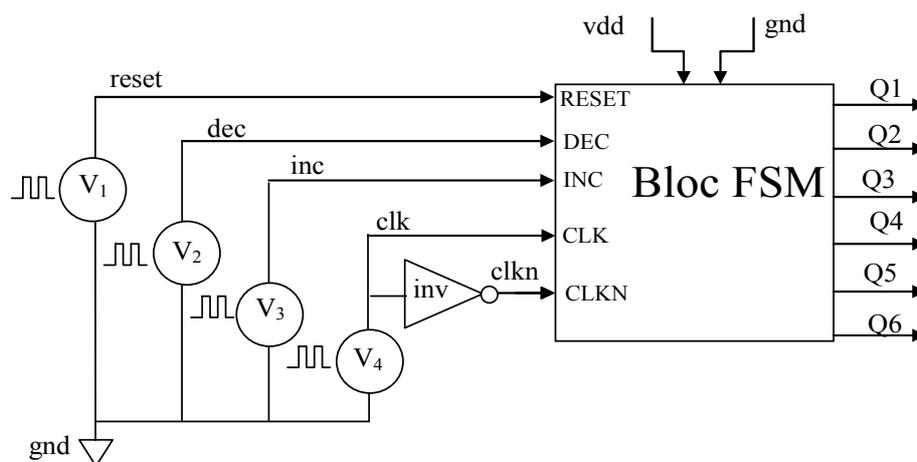


Figure 4.12 : Le circuit de test bench

V.4.2. Les résultats de la simulation des signaux de transition

La figure 4.13 montre les résultats de simulation des 4 signaux des transitions (conditions de transitions) *reset*, *inc dec*, *nincdec*, respectivement. Le signal *nreset* est le signal inverse de signal *reset*. Le signal, *nincdec* est une combinaison des deux signaux decint, incint (fonction NOR).

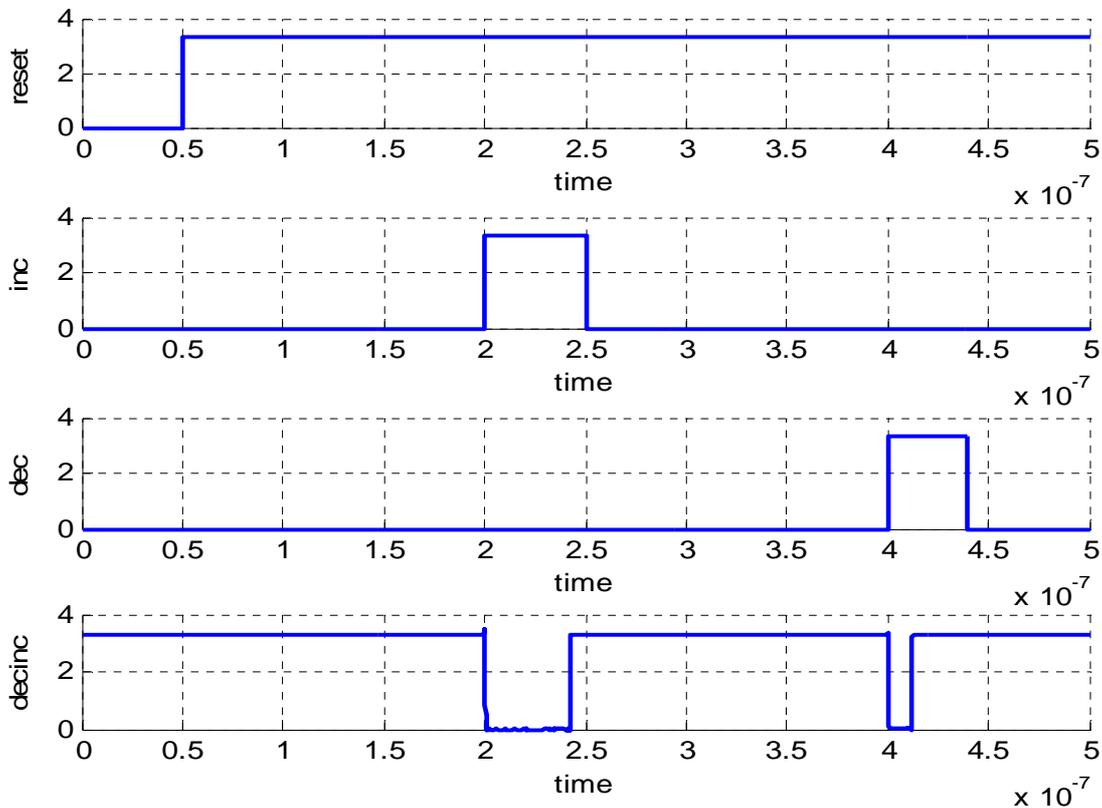


Figure 4.13: Simulation des signaux de transition

Les deux signaux d'horloge CLK et CLKN sont complémentaires et non chevauchés puisque les circuits logiques C²MOS sont insensibles au chevauchement des signaux d'horloge.

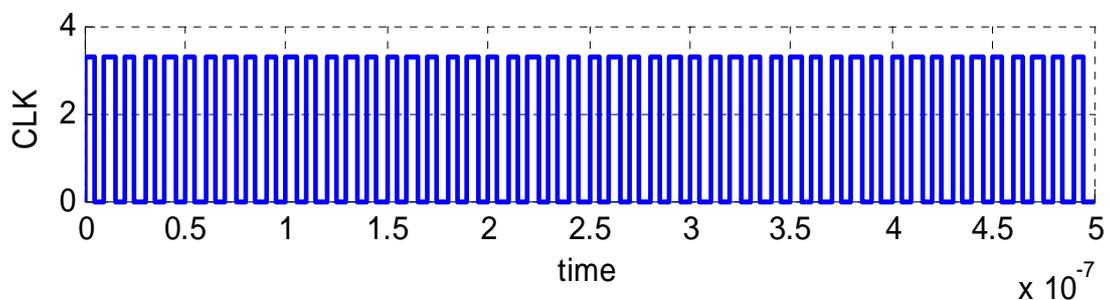


Figure 4.14 : Simulation des signaux d'horloge

IV.4.3. Les résultats de la simulation des états de la machines d'états

La figure 4.14, montre 6 signaux présentant les 6 états de la machine d'états .un état présent se transit vers un état futur en fonction de la présence et ses signaux des transitions (chaque état a ses propre signaux de transition).

On observe que le fonctionnement de la machine d'états et identiques au diagramme d'états de la figure c.à.d. au cahier de charge (programme VHDL).

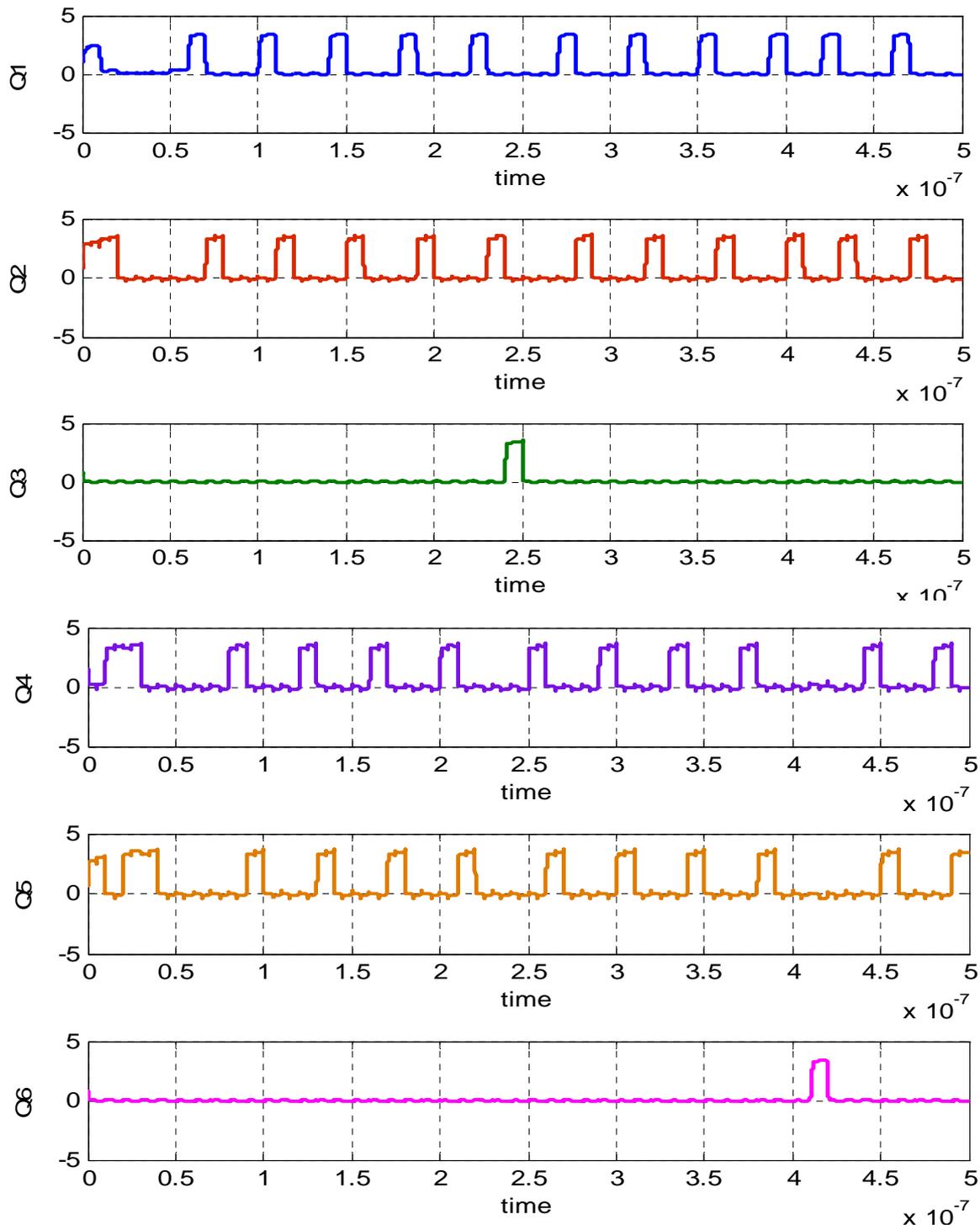


Figure 4.15: Simulation des états de la machine d'états

IV.4.4. Les résultats de la simulation du courant de fuite

Une autre simulation concerne la dissipation de puissance et le courant de fuite. Ces deux paramètres sont très importants, la connaissance du courant de fuite permet de déterminer la puissance dissipée Il y un courant de fuite, lorsque les deux transistors complémentaires connectés à l'horloge CLK et CLKN sont saturés simultanément dans une durée de temps courte.

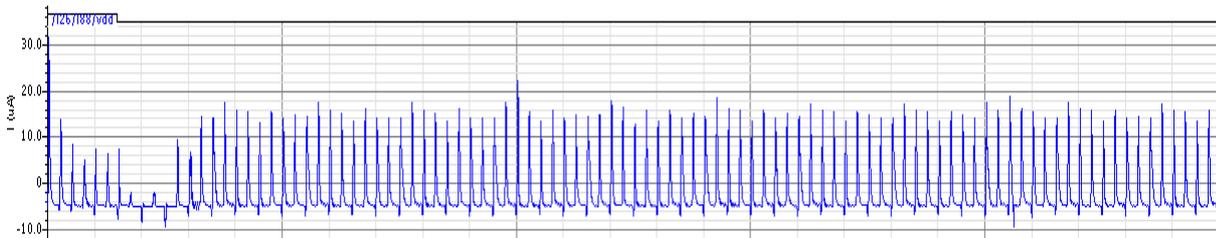


Figure IV.16: Simulation du courant de fuite

IV.5. Le layout

Le layout est le processus de la création d'une représentation physique précise d'un schéma de technologie (netlist), qui se conforme aux contraintes a imposé par le processus de fabrication, l'écoulement de conception, et les conditions d'exécution avérées faisable par la simulation [20].

Un exemple du layout d'un inverseur MOS [4].

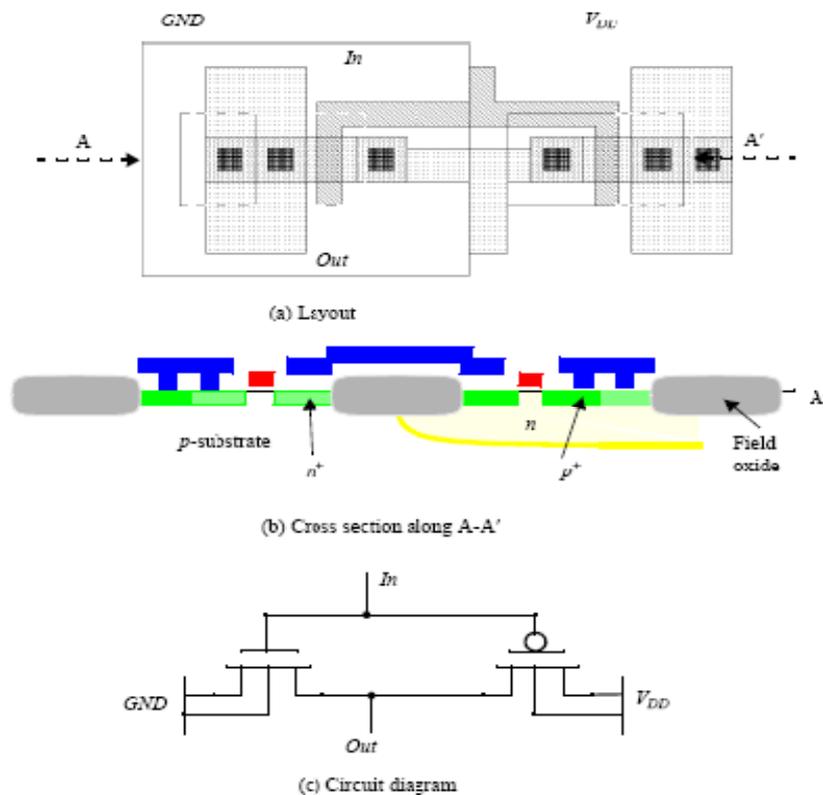


Figure 4.17 : Un exemple du layout d'un inverseur

IV.5.1. La conception physique des masques layout

La conception de chip concentre sur deux tâches principales :

- 1-Traduire la fonction logique nécessaire en circuit électronique équivalent.
- 2-Créer les réseaux de commutation rapides.

Les opérations de synthétisation logique sont accomplies par le placement approprié et raccordement des transistors MOSFET, c.-à-d., la topologie de circuit. D'autre part il est plus difficile de contrôler, l'exécution de commutation, car elle dépend des tailles des transistors, les caractéristiques des connexions de circuit, et la résistance et la capacité parasites dans le circuit.

Au niveau de la conception physique on arrange avec spécification de la taille exacte et l'endroit exacts de chaque forme géométrique sur chaque couche matérielle du chip, ceci est accompli en concevant chaque masque qui est nécessaire pour fabriquer la structure à trois dimensions. Ceci est fait en utilisant un outil de dessin de CAO connu sous le nom de l'éditeur layout, qui nous permet d'indiquer le modèle de chaque étape lithographique dans l'écoulement de processus. L'étape physique de conception donne des caractéristiques importantes telles que la densité de packaging de transistor et les propriétés électriques de transmission de l'interconnexion "câble" qui est possibles dans une ligne donnée de processus de fabrication. Chaque étape modelée dans le flot de processus exige un masque séparé. Les étapes des masques requises dans le processus de base du n-well CMOS sont comme suit.

Nwell : le masque de n-well

Active : régions où des FETS seront placés

Poly : le modèle de porte de polysilicium

Pselect : régions où les ions type -p implanté formera des régions

Nselect : régions où le type n- implanté formera des régions

Poly contact : coupe dans l'oxyde qui fournit les contacts Metall au poly

Active Contact : coupe dans l'oxyde au lequel fournir les connexions de Metall au n+ou p+

Metall : modèle pour la première couche de métal

Via:coupe d'oxyde coupé pour la connexion de Metall au Metall2

Metall2 : modèle pour la deuxième couche de métal.

La structure de la technologie CMOS et son layout est présenté dans les figures 4.18 et 4.19 respectivement.

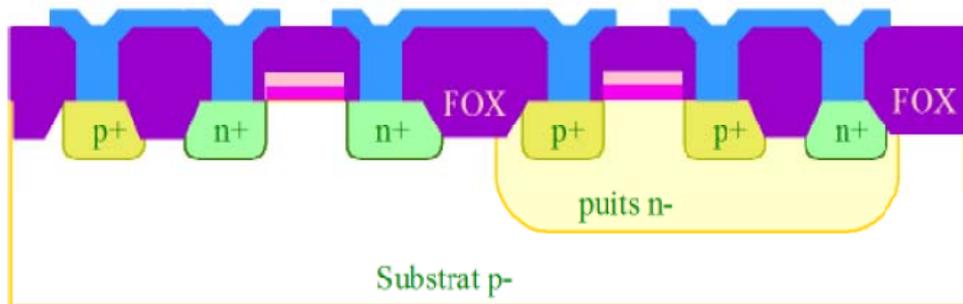


Figure 4.18: Structure CMOS

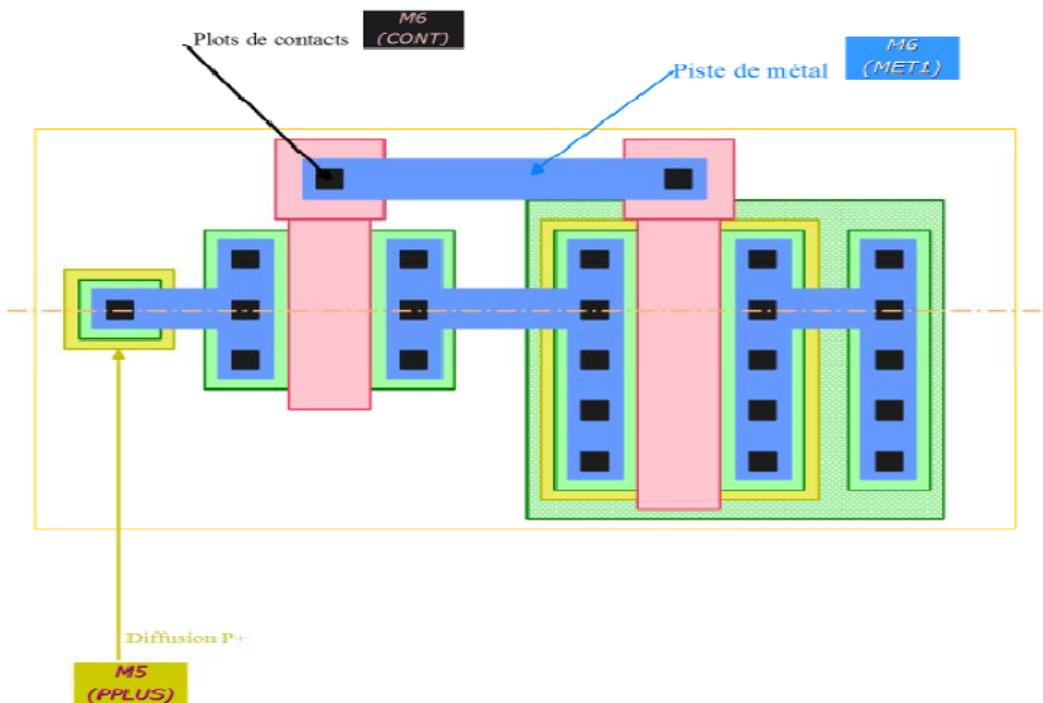


Figure 4.19 : Layout d'une structure CMOS

IV.5.2. Le layout compact et les cellules standards

Le style de dessin de layout est la façon d'implanter la fonctionnalité d'un circuit dans le silicium. Il définit le coût de fabrication du circuit car il détermine le nombre d'étapes technologiques nécessaires à son intégration et le temps de développement.

Les cellules standards (Standard Cell), sont des circuits regroupant des fonctions logiques standard disponibles sous forme de cellules précaractérisées dans une bibliothèque.

Lors de la conception de circuits complexes on utilise souvent plusieurs fois les mêmes circuits de base. Il est donc intéressant de ne dessiner qu'une fois le layout ce circuit.

Le layout compact (Full Custom), est entièrement dessiné à la main; la présence de fonctions analogiques peut conduire à la mise en œuvre de cette approche qui ne se justifie que pour des volumes de production élevés.

IV.5.3. Le dessin du layout Full-Custom utilisant Virtuoso-XL

Virtuoso XL est l'outil d'édition et de connectivité de base qui automatise chaque étape de conception de layout, de génération composante par un routage automatique/interactive. Il a été utilisé en tant qu'élément d'une méthodologie de conception physique de la machine d'états, le virtuoso XL a produit le layout d'après le schéma et a édité les layouts qui ont défini la connectivité. Le virtuoso XL continuellement a permis de contrôler les connexions des composants dans le layout et les compare avec des connexions dans le schéma. On a employé le virtuoso XL pour voir les nets incomplètes, courts, et les invalides connexions, et les chevauchements pour nous aider à câbler notre conception.

Le compactage est un outil très utile puisque le layout du circuit FSM doivent être ajustés dans un minimum superficie de $(12 \times 60 \mu\text{m}^2)$, ce qui implique l'implémentation de la méthode du full custom, pour dessiner le layout du circuit. À la différence des cellules standards, les contraintes de taille et de largeur du layout compact placées sont contextuelles. En d'autres termes, le rapport hauteur/largeur du circuit dépend de celui de ses voisins. Dans ce cas-ci, la largeur du circuit a dépendu de la maximale largeur permise dans le chip ASIC.

Cette technique a permis de dessiner le layout des transistors MOSFETs à la main, en respectant les règles de la conception et ensuite faire le routage des connexions figure 4.20.

Il y a également des avantages techniques à être gagnés quand on a un layout plus petit que possible (un layout compact) : l'interconnexion des fils peut être courte que possible, réduisant de ce fait les effets de charge parasites et d'interférence. Le temps nécessaire pour produire une cellule standard est loin d'être moins que dans le dessin du full custom. Ainsi une différence principale affectant la densité est le câblage d'interconnexion.

IV.5.4. Design Rule Check (DRC)

Le layout de masque créé doit se conformer à un ensemble complexe de règles de conception, afin d'assurer une inférieure probabilité des défauts de fabrication. Un outil construit dans l'éditeur layout, appelé Design Rule Checker, est employé pour détecter toutes les

violations de règle de conception pendant et après que la conception de layout de masque du circuit.

IV.5.5. Layout Versus Schematic (LVS)

Une fois que le layout accomplit toutes les règles de conception, la prochaine étape de vérification suit. Le netlist derrière la vue de layout est extraite et comparée à celle de la vue schématique. C'est le contrôle de Layout Versus Schematic (LVS).

Après la vérification DRC et LVS, le dessin des masques de layout est terminé et on obtient un layout compacte de la machine d'états, qui montré dans la figure 4.20.

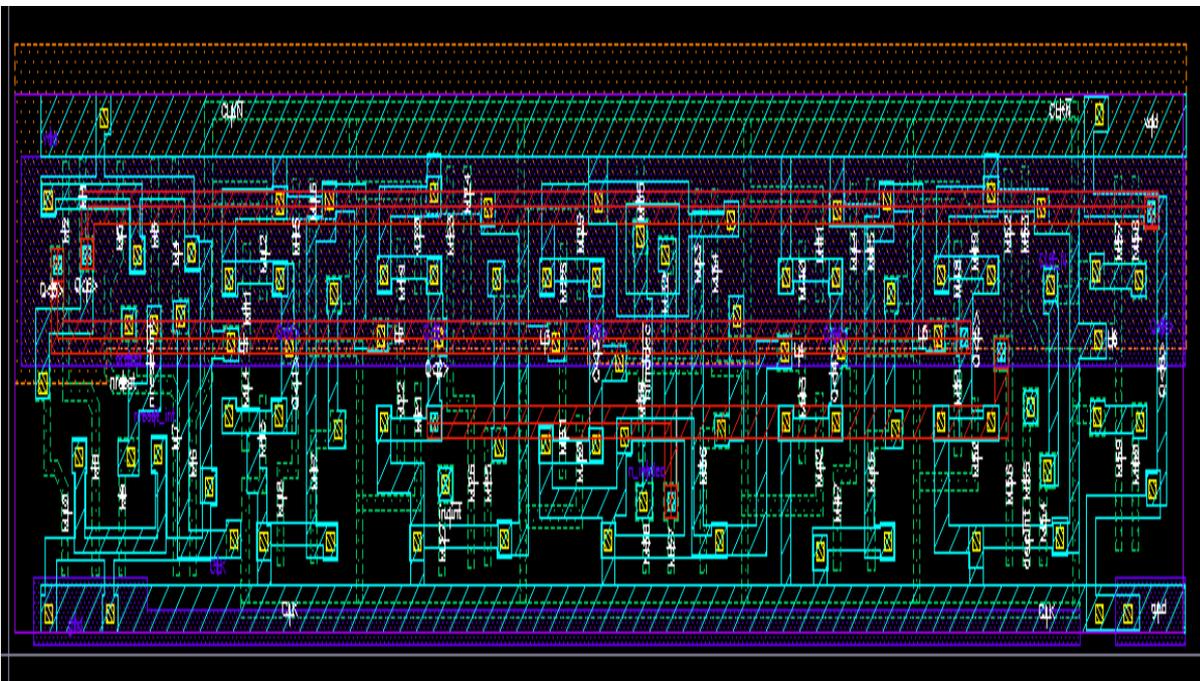


Figure 4.20 : Layout compact de la machine d'états

IV.6.Conclusion

Une chaîne de conception assistée par ordinateur offrant une interface multifenêtres et graphiques sur laquelle s'appuie une collection d'outils logiciels, s'adapte à différentes tâches de conception. Dans le cadre ce projet, nous avons utilisé l'outil de CAO Cadence.

L'environnement de conception Cadence Virtuoso, constitue de plusieurs outils logiciels et un ensemble de conception (design kit), qui nous permet de concevoir, de simuler, et de dessiner les masques layout d'un circuit (machine d'états) en technologie 0.35 μm . Nous avons utilisé Virtuoso spectre pour la simulation du circuit, ensuite l'outil Virtuoso XL pour le dessin des masques layout et. La dernière étape est la vérification des règles de dessin (DRC), la comparaison de graphes électriques entre schématique et masques(LVS), utilisant L'outil Assura.

Conclusion générale

Conclusion générale

Ce travail vise à aboutir une connaissance approfondie de la conception du circuit logique CMOS. Le travail bibliographique réalisé nous a permis notamment de mieux cerner les difficultés liées à la conception du circuit intégré sur silicium. Cette étude est un premier pas dans la conception d'un circuit complet utilisant des techniques innovantes de C²MOS.

Le premier chapitre a permis de poser un cadre théorique de la technologie CMOS. Avant de pouvoir développer des méthodes de conception pour machine d'états, il été nécessaire d'étudier le transistor MOS et sa structure, ces caractéristiques.

Après ces études proches du composant MOS, L'analyse des circuits logiques CMOS a montré que le circuit dynamique CMOS mieux adaptées à la conception de la machine d'états. Après avoir illustré l'apport des styles logiques et surtout la logique C²MOS à la conception des latches.

Ensuite, une brique clef de la conception des circuits logiques qu'est les outils CAO. une vue d'ensemble de ces outils et méthodologies de conception nous permet de conclure que sans ces CAO la complexité courante de la conception n'aurait pas été réalisable

Une fois cette étape franchie, il a été possible de formaliser certaines étapes cruciales de la conception d'une machine d'états telle que la spécification et la simulation et le dessin de layout. Ainsi, l'étude s'est focalisée sur le fonctionnement de FSM et ces circuits (latche - registres).

Les premiers résultats de ces travaux de recherche concernent le niveau d'abstraction de la conception circuit numériques, qui commence au niveau du transistor MOS. On considère les équations de physique de semi-conducteurs régissant le comportement du dispositif en concevant une porte numérique.

La philosophie de conception des circuits numérique est basée sur le bon choix de l'outil de conception CAO. et les méthodologies de conception suivant un flot qui décrit un processus de conception.

L'application des techniques de conception des portes logique dynamique C²MOS est essentiellement dont le but de diminuer la complexité, augmenter la vitesse, et pour une dissipation inférieure de puissance de consommation.

Enfin la conception du circuit machine d'états, nous donne une occasion pratique pour maîtriser l'outil Cadence Virtuoso et comprendre la méthode et les étapes de conception des circuits logiques CMOS, telle que la spécification, la simulation et le dessin du layout.

Annexe A

Description VHDL de la machine d'états

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity strobe_fcblock1 is
  port(
    clk           : in std_logic;
    clk_n         : in std_logic;
    nreset        : in std_logic;
    nreset_int    : in std_logic;
    inc_int       : in std_logic;
    dec_int       : in std_logic;
    n_incdec      : in std_logic;
    Q0            : out std_logic_vector(6 downto 1)
  );
end strobe_fcblock1;

architecture simmodel of strobe_fcblock1 is
  signal Q,L : std_logic_vector(6 downto 1);

begin
  Q0 <= Q;

  -- Register 1 --
  R1: process(clk)
    variable result : std_logic_vector(1 downto 0);
  begin
    if (clk='0' and clk'event) then
      if (nreset = '1' and (Q(5) = '1' or Q(6) = '1' or nreset_int = '1'))
then result(0) := '0'; else result(0) := 'Z'; end if;
      if ((nreset_int = '0' and (Q(5) = '0' and Q(6) = '0')) or nreset =
'0') then result(1) := '1'; else result(1) := 'Z'; end if;
      if resolved(result) /= 'Z' then L(1) <= resolved(result); end if;
    end if;
    if (clk='1' and clk'event) then
      if (L(1) = '1') then result(0) := '0'; else result(0) := 'Z'; end if;
      if (L(1) = '0') then result(1) := '1'; else result(1) := 'Z'; end if;
      Q(1) <= resolved(result);
    end if;
  end process;

  -- Register 2 --
  R2: process(clk)
    variable result : std_logic_vector(1 downto 0);
  begin
    if (clk='0' and clk'event) then
      if (Q(1) = '1') then result(0) := '0'; else result(0) := 'Z'; end if;
      if (Q(1) = '0') then result(1) := '1'; else result(1) := 'Z'; end if;
      if resolved(result) /= 'Z' then L(2) <= resolved(result); end if;
    end if;
    if (clk='1' and clk'event) then
      if (L(2) = '1') then result(0) := '0'; else result(0) := 'Z'; end if;
      if (L(2) = '0') then result(1) := '1'; else result(1) := 'Z'; end if;
      Q(2) <= resolved(result);
    end if;
  end process;
end strobe_fcblock1;

```

```

-- Register 3 --
R3: process(clk)
    variable result : std_ulogic_vector(1 downto 0);
    begin
        if (clk='0' and clk'event) then
            if (Q(2) = '1' and inc_int = '1') then result(0) := '0'; else
result(0) := 'Z'; end if;
            if (Q(2) = '0') then result(1) := '1'; else result(1) := 'Z'; end if;
            if resolved(result) /= 'Z' then L(3) <= resolved(result); end if;
        end if;
        if (clk='1' and clk'event) then
            if (L(3) = '1') then result(0) := '0'; else result(0) := 'Z'; end if;
            if (L(3) = '0') then result(1) := '1'; else result(1) := 'Z'; end if;
            Q(3) <= resolved(result);
        end if;
    end process;

-- Register 4 --
R4: process(clk)
    variable result : std_ulogic_vector(1 downto 0);
    begin
        if (clk='0' and clk'event) then
            if (Q(3) = '1' or (Q(2) = '1' and n_incdec = '1')) then result(0) :=
'0'; else result(0) := 'Z'; end if;
            if ((Q(2) = '0' or n_incdec = '0') and Q(3) = '0') then result(1) :=
'1'; else result(1) := 'Z'; end if;
            if resolved(result) /= 'Z' then L(4) <= resolved(result); end if;
        end if;
        if (clk='1' and clk'event) then
            if (L(4) = '1') then result(0) := '0'; else result(0) := 'Z'; end if;
            if (L(4) = '0') then result(1) := '1'; else result(1) := 'Z'; end if;
            Q(4) <= resolved(result);
        end if;
    end process;

-- Register 5 --
R5: process(clk)
    variable result : std_ulogic_vector(1 downto 0);
    begin
        if (clk='0' and clk'event) then
            if (Q(4) = '1') then result(0) := '0'; else result(0) := 'Z'; end if;
            if (Q(4) = '0') then result(1) := '1'; else result(1) := 'Z'; end if;
            if resolved(result) /= 'Z' then L(5) <= resolved(result); end if;
        end if;
        if (clk='1' and clk'event) then
            if (L(5) = '1') then result(0) := '0'; else result(0) := 'Z'; end if;
            if (L(5) = '0') then result(1) := '1'; else result(1) := 'Z'; end if;
            Q(5) <= resolved(result);
        end if;
    end process;

-- Register 6 --
R6: process(clk)
    variable result : std_ulogic_vector(1 downto 0);
    begin
        if (clk='0' and clk'event) then
            if (Q(2) = '1' and dec_int = '1') then result(0) := '0'; else
result(0) := 'Z'; end if;
            if (Q(2) = '0') then result(1) := '1'; else result(1) := 'Z'; end if;
            if resolved(result) /= 'Z' then L(6) <= resolved(result); end if;

```

```

end if;
if (clk='1' and clk'event) then
  if (L(6) = '1') then result(0) := '0'; else result(0) := 'Z'; end if;
  if (L(6) = '0') then result(1) := '1'; else result(1) := 'Z'; end if;
  Q(6) <= resolved(result);
end if;
end process;

end architecture;

```

Programme VHDL de simulation de la machine d'états 'testbench'

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

```

```

library work;
use work.recon_cntl3;

```

```

-----
----

```

```

entity recon3_tb is

```

```

end recon3_tb;

```

```

architecture tb of recon3_tb is

```

```

  component recon_cntl3

```

```

    port (

```

```

      seg_active   : out std_logic_vector(6 downto 0);
      gnd_active   : out std_logic_vector(6 downto 0);
      ref_active   : out std_logic_vector(6 downto 0);
      outh_active  : out std_logic_vector(27 downto 0);
      outl_active  : out std_logic_vector(27 downto 0);
      clk          : in  std_logic;
      inv_en       : in  std_logic;
      nreset       : in  std_logic;
      gnd          : in  std_logic;
      vdd          : in  std_logic;

```

```

    );

```

```

  end component;

```

```

  signal seg_active   : std_logic_vector(6 downto 0);
  signal gnd_active   : std_logic_vector(6 downto 0);
  signal ref_active   : std_logic_vector(6 downto 0);
  signal outh_active  : std_logic_vector(27 downto 0);
  signal outl_active  : std_logic_vector(27 downto 0);
  signal clk          : std_logic := '0';
  signal inv_en       : std_logic;
  signal nreset       : std_logic := '0';

```

```

  signal simulation   : std_logic := '1';

```

```

begin

```

```

  DUT: recon_cntl3

```

```
port map (  
    seg_active    => seg_active,  
    gnd_active    => gnd_active,  
    ref_active    => ref_active,  
    outh_active   => outh_active,  
    outl_active   => outl_active,  
    clk           => clk,  
    inv_en        => inv_en,  
    nreset        => nreset,  
    gnd           => '0',  
    vdd           => '1'  
);  
  
clk_proc : process begin  
    wait for 5 ns;  
    clk <= not clk;  
    wait for 5 ns;  
    if (simulation = '0') then  
        wait;  
    end if;  
end process;  
  
work : process begin  
    inv_en <= '0';  
    wait for 20 ns;  
    nreset <= '1';  
    wait for 1 us;  
    nreset <= '0';  
    inv_en <= '1';  
    wait for 20 ns;  
    nreset <= '1';  
    wait for 1 us;  
    simulation <= '0';  
    wait;  
end process;  
end architecture;
```

Bibliographie

Bibliographie

- [1] B.Razavi, *Fundamentals of Microelectronics*, Wiley, 2006.
- [2] B.Razavi, *Design Of Analog CMOS Integrated Circuits*, Mc Graw Hill, New york, 2001.
- [3] R. Jacob Baker, Harry W.Li and David E.Boyce, '*CMOS Circuit Design, Layout and Simulation.*' The Institute of Electrical and Electronics Engineers, Inc., New York. 1998.
- [4] Jan M Rabaey, '*Digital Integrated Circuits,A Design Perspective*', 2002.
- [5] Kurt Hoffman ,'*System integration from transistor to large scale integration circuits*' ,Wiley, 2004.
- [6] Hwang E.'*Principles of digital logic design*' ,.2003.
- [7] Douglas L Perry, '*Vhdl Programming By Example*' , 5^{eme} edition, Mc Graw Hill , 2002.
- [8] Sagar Vidya Reddy ,'*Digital design flow options*' , Thèse de Master de science ,The Ohio State University,2001.
- [9] Addison Wesley '*Application Specific Integrated Circuits*' , SP , 2004.
- [10] Clive Max Maxfield '*The Design Warriors Guide to FPGA*' , Newnes ,2004.
- [11] Adel S.Sedra , Kenneth C.Smith '*A Microelectronic circuits*' ,university of Toronto ,5th edition, oxford university press, 2004
- [12] Stanley L. Hurst, '*VLSI,' custom microelectronics digital, analog, and mixed signal*' ,Marcel Dekker INC, New York 1999.
- [13] Kevin F.Brennan, April S. Brown '*Theory of Modern Electronic Semiconductor Devices*' , Georgia Institute of Technology, Wiley 2002.
- [14] B. Van Zeghbroeck '*Principles of Semiconductor Devices*' prentice Hall, 2001.
- [15] M. Rafiquzzaman '*Fundamentals of Digital Logic and Microcomputer Design*' , 5th edition, Wiley, Hoboken, New Jersey, 2005.
- [16] Mark Balch '*Complete Digital Design*' , Mc Graw Hill,-2003
- [17] Naveed A. Sherwani '*Algorithms for VLSI physical design automation*' 3ed edition, 1999.
- [18] John P. Uyemura '*CMOS logic circuit design*' Kluwer Academic Publishers, New York, 2002.
- [19] John F. Wakerly '*Digital Design Principles and Practices*' , 1999.
- [20] Dan Clein '*CMOS IC layout Concepts, Methodologies, and Tools*' , Newnes ,2000.