



UNIVERSITÉ CATHOLIQUE DE LOUVAIN

TRAVAIL DE FIN D'ÉTUDE.

**Étude et flot de simulation
statistique de la variation d'énergie
dynamique d'un circuit logique en
régime sous-seuil**

Auteur :
Guillaume MARTIN

Superviseur :
David BOL

Jury :
Pr Jean-Didier LEGAT
François BOTMAN

Travail présenté en vue de l'obtention du diplôme d'ingénieur civil
électricien
19 août 2013

Abstract

The constant search for lowering the power consumption of digital circuits brings us into the world of sub-threshold circuits. In this world, delays of logic cells are exponentially dependent of the threshold voltage V_{th} . With the constant down-scaling of CMOS technology digital circuits, the variability of the V_{th} parameter increases dramatically. As a result, the change of the switching activity of the circuit [13]. Because this activity is directly linked to the dynamic energy consumption, this latest varies too.

This variability forces designers to take larger constraints on the choice of the voltage regulator and on the selection of the maximum power that can be delivered. Thus tools are needed to predict this variability into a circuit to avoid prohibitive guardband. This is what we intent to provide in this work.

First, analyzing a simple logic cell, we show that the relative variance $\frac{\sigma}{\mu}$ is influenced by two factors of the load. If the load is composed of logic cells, the first factor is the capacitive coupling between the output node and the output of the active load. The second factor is the slope of the input signal as faster cells will have shorter relative time when the input is at its maximum.

Then, to avoid prohibitively long SPICE simulation we manage to simulate the variability of the dynamic at gate level. To do so, we propose a new Monte-Carlo simulation flow at cell level which randomizes gate delays inside the netlist. For this analysis, we use a global coefficient $\frac{\sigma}{\mu}$. This is relevant as we demonstrate that just a global $\frac{\sigma}{\mu}$ underevaluates dynamic energy variations. We show that using a global coefficient in the case of sequential netlist lead to hold violation.

J'aimerais remercier mon promoteur David Bol et Gueric de Streeel pour le temps consacré à m'aider pour le présent travail.

Je remercie mes relecteurs Guy, David et Jean pour le temps passé sur mon orthographe et mon français.

Je tiens aussi à remercier EPLOL pour le soutien qu'ils m'ont apporté.

Enfin, je remercie Yoda sans qui je n'aurais eu la Force.

Table des matières

Introduction	1
1 Circuits logiques sous-seuil : fonctionnement, performances et variations.	5
1.1 Dépendance exponentielle du courant pour des transistors en régime sous seuil.	7
1.2 Impact du courant sur les délais de basculement des portes logiques.	10
1.3 Impact de la variation du courant sur le facteur d'activité du circuit logique : α_F	12
1.4 Lien entre le facteur d'activité et la consommation dynamique d'un circuit logique.	14
1.5 Couplage capacitif entre le nœud de la grille et celui du drain.	15
1.6 Propriétés de la distribution log-normale dans le cadre des délais.	17
2 Méthodologies statistiques de conception et de simulation des circuits logiques.	21
2.1 SSTA : Statistical static timing analysis	21
2.2 Simulation de la puissance consommée.	22
3 Analyse des variations d'une porte logique sous-seuil.	25
3.1 Dépendance sur la pente du signal d'entrée et le seuil de mesure.	27
3.2 Effet de la charge sur le rapport $\frac{\sigma}{\mu}$	34
3.2.1 Dans le cas d'une charge passive.	34
3.2.2 Dans le cas d'une charge active.	35
3.3 Influence de la taille et de la complexité d'une porte sur le rapport entre l'écart type et la moyenne.	43
3.4 Validation d'un modèle linéaire pour définir l'écart type du délai.	48
4 Mise en œuvre d'un nouveau flot de simulation statistique de l'énergie dynamique.	53
4.1 Flot proposé dans le cadre de ce mémoire.	53

4.2	Les outils développés :	54
4.3	Validation.	55
4.3.1	Vérification du générateur de nombres aléatoires	56
4.3.2	Simulation et comparaison avec des mesures d'un circuit combinatoire.	57
4.3.3	circuit séquentiel.	71
5	Conclusion	75
A	Pour appuyer le texte.	77
A.1	Vérification de la log-normalité des délais dans le cas de l'inverseur et de la porte XNOR.	77
A.2	Lexique	80
B	Les outils	81
B.1	Flot standard de développement.	81
B.2	Détails techniques des outils développés.	83
B.2.1	analysesaif.jar, soc_pwr_ana.jar, powerAnalysis.jar	86
B.3	Les différents formats et leurs utilités.	87

Introduction

Depuis quelques temps, nous voyons apparaître de plus en plus d'applications électroniques portables à faible consommation d'énergie. Les constructeurs diminuent la consommation dans le but d'augmenter l'autonomie sur batterie ou même de pouvoir fonctionner avec de la récupération d'énergie dans le milieu ambiant [21].

La tension d'alimentation est un levier efficace pour diminuer la consommation [21]. Au prix d'une augmentation des délais au sein des différentes cellules, elle fait diminuer l'énergie dynamique par opération de manière quadratique [19]. Si la vitesse de fonctionnement n'est pas critique, nous pouvons donc travailler avec une tension d'alimentation dite sous seuil c'est-à-dire inférieure à la tension de seuil V_{th} des transistors : "subthreshold voltage" [15]. C'est le cas par exemple pour le microcontroller SleepWalker [4] adapté pour l'"Internet-of-Things", d'une Sbox de Dina Kamel servant dans l'encryptage pour des tags RFID passifs [13] ou encore, pour un filtre adaptatif DLMS pour mal entendant [15].

Mais descendre en dessous de ce seuil a un impact sur le comportement du circuit logique. En effet, une fois en régime sous seuil, le courant passant dans le transistor ne se comporte plus de façon habituelle [15]. Celui-ci évolue exponentiellement vis à vis de la tension de grille (V_{gs}) et de la tension de seuil (V_{th}). Or, avec la miniaturisation des technologies, la dispersion de V_{th} augmentent fortement [17].

En dehors des variations qui surviennent à l'utilisation, les variations de V_{th} peuvent être regroupées en deux catégories : les variations globales, comme par exemple les gradients de fabrication, et les variations locales, par exemple les fluctuations des atomes dopants d'une porte logique à l'autre ([14]). Si la première catégorie peut être compensée typiquement avec un contrôle adaptatif de la tension (AVS) [4], la deuxième de par sa nature localisée ne peut être compensée.

La dépendance exponentielle du courant en régime sous seuil entraîne une grande sensibilité des délais des portes logiques aux variations de V_{th} . Ces variations de délais par rapport à leurs valeurs nominales vont à leur tour entraîner une variation du facteur d'activité du circuit digital [13], facteur d'activité qui influence directement l'énergie dynamique consommée par le circuit pour une opération.

Pour un même masque, nous n'aurons donc pas la même consommation d'un circuit digital fabriqué (die) à un autre [13]. Ceci entraîne des contraintes supplémentaires pour le régulateur de tension qui doit supporter une plus large plage de puissance. Il est donc important de pouvoir prédire ces variations de consommation sans pour autant faire exploser le temps de simulation comme le ferait le circuit simulé en SPICE.

Il n'existe actuellement pas d'outil qui prenne en compte la variation du facteur d'activité et son impact sur la variation d'énergie. Pour palier à ce manque, ce travail propose un flot de simulation permettant d'étudier l'impact qu'ont les variations locales de V_{th} sur la variation de consommation d'un die à l'autre.

Nous avons choisi d'utiliser le fichier contenant les différents timings des portes logiques et de faire une simulation de type Monte-Carlo au niveau des portes logiques. Nous générons des timings aléatoires dans la netlist sur base d'un $\frac{\sigma}{\mu}$ global faute d'avoir pu faire l'analyse automatique de la librairie. Ensuite, nous simulons les différents circuits générés et nous en déduisons l'énergie dynamique consommée dans chacun. De ces séries de mesures, nous déduisons l'écart type et la moyenne de l'énergie dynamique consommée que nous comparons avec les valeurs obtenues dans le cas de la Sbox en 65 nm CMOS de [13].

Bien que sous évaluée, la variation d'énergie calculée par cette méthode est présente et augmente bien avec la variation des délais. De plus, nous montrons que dans le cas d'un circuit purement combinatoire, faire l'analyse au niveau structurel (post synthèse) apporte la même information qu'au niveau *place and route* (post layout). L'analyse montre aussi que travailler avec un $\frac{\sigma}{\mu}$ global dans un circuit purement combinatoire amène aux mêmes variations qu'utiliser un $\frac{\sigma}{\mu}$ différent pour le temps de montée t_{rise} et le temps de descente t_{fall} de toutes les portes pour peu que nous prenons un $\frac{\sigma}{\mu}$ qui moyenne ces deux derniers.

Pour pouvoir faire cette analyse, nous étudions un modèle statistique représentant le comportement des différentes portes logiques. Nous avons utilisé un outil de *statistical static timing analysis* (SSTA). Bien que nous n'ayons pas réussi à automatiser le procédé, nous avons toutefois analysé en profondeur un inverseur, ce qui a mis en évidence que le rapport $\frac{\sigma}{\mu}$ des délais n'est pas constant avec la charge. Nous avons donc approfondi ce phénomène pour d'une part en chercher les sources, d'autre part évaluer si un modèle utilisant un $\frac{\sigma}{\mu}$ global reste viable.

Ce présent document est décomposé en quatre chapitres.

Le premier chapitre résume les connaissances permettant de comprendre le phénomène de variance d'énergie dynamique qui survient dans les circuits digitaux sous seuil.

Le deuxième chapitre rappelle le flot de simulation au niveau des portes logiques permettant d'extraire l'énergie dynamique sur base d'une netlist et des délais intrinsèques à celle-ci.

Nous analysons ensuite avec des simulations dans le chapitre 3 les variations intervenant lors de la commutation d'un inverseur. Nous y précisons les deux phénomènes qui entraînent que le rapport $\frac{\sigma}{\mu}$ n'est pas constant ; l'injection de charge et la forme du signal d'entrée. Ce chapitre se termine par la vérification qu'un modèle se basant sur un $\frac{\sigma}{\mu}$ constant est viable.

Dans le chapitre 4, nous validons le nouveau flot et les outils utilisés sur un circuit dont nous connaissons les variations [13]. Nous montrons que prendre un $\frac{\sigma}{\mu}$ global sous-évalue l'écart type relatif de l'énergie dynamique. Nous y montrons aussi que dans le cas d'un circuit purement combinatoire, faire l'analyse au niveau *place and route* n'améliore que peu la précision de l'estimation des variations de l'énergie dynamique.

Nous finissons le chapitre par l'analyse d'un circuit séquentiel, un processeur mips 8 bit en 65 nm CMOS. Et nous parcourons les problèmes qui surviennent lors de l'usage d'un rapport $\frac{\sigma}{\mu}$ global et non particulier à chacun des types de cellules.

Chapitre 1

Circuits logiques sous-seuil : fonctionnement, performances et variations.

La base de la plupart des circuits digitaux actuels est le transistor MOS-FET : "metal-oxide-semiconductor field-effect transistor".

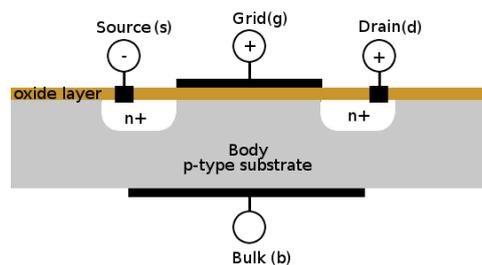


Figure 1.1 – Représentation en coupe d'un transistor MOSFET idéal.

Ce transistor comporte une source, un drain, une grille et le body connecté au substrat (bulk). Dans les circuits digitaux, il fonctionne en quelque sorte comme un interrupteur commandé par la grille. Quand la tension de grille dépasse la tension de seuil V_{th} ($V_{th} \approx 0.4$ sur la figure 1.2), le courant peut circuler entre la source et le drain mais avec différentes zones de fonctionnement (figure 1.2). Tout d'abord, il y a la zone linéaire où $\frac{\delta I_D}{\delta V_g} \neq \text{constante}$, puis celle en saturation avec $\frac{\delta I_D}{\delta V_g} \approx \text{constante}$.

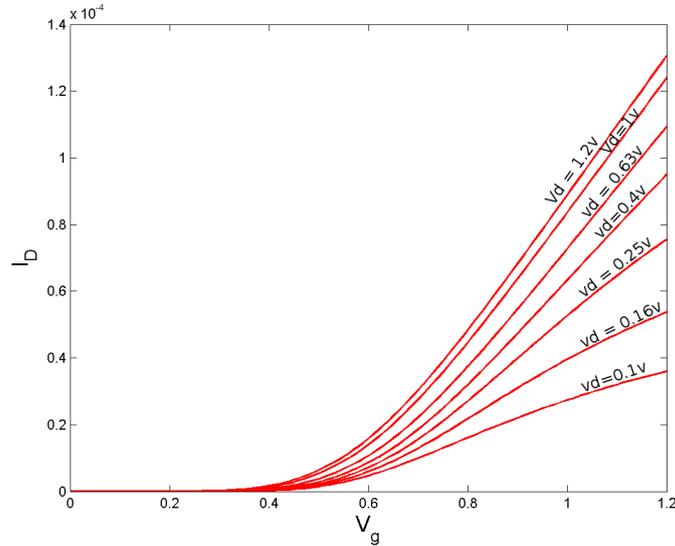


Figure 1.2 – Simulation ELDO de la caractéristique I_d/V_g d'un transistor NMOS en technologie cmos 65nm LP ($\frac{W}{L} = \frac{1}{0.3}$, Standard VT, 25° C, $V_{bulk} = 0$). V_D évolue en logspace de 0.1 à 1.2.

Agencés ensemble, ces transistors peuvent former des *portes logiques*. Chacune de ces portes a une fonction précise. Par exemple à la figure 1.3, nous pouvons voir : à gauche (a) un inverseur qui a pour but de transformer un 0 en 1 et un 1 en 0, à droite (b), une porte qui applique l'opération "not and" sur les deux signaux d'entrée.

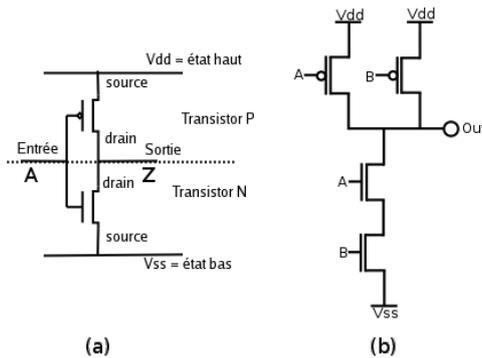


Figure 1.3 – Inverseur simple (a) et porte NAND (b). Source [22]

Ces portes logiques peuvent être agencées ensemble pour former n'importe quel type de circuit digital. La consommation de celui-ci se décompose en deux parties : la consommation statique et la consommation dynamique.

La première vient des courants de fuite (*leakage*) et est fonction d'état du processeur. La deuxième quant à elle est dépendante des opérations faites via le nombre de commutations des différentes portes logiques.

A chaque basculement de porte, il faut charger ou décharger la capacité de charge (C_Z). Le courant qui charge et décharge cette capacité induit une dissipation. L'énergie nécessaire est proportionnelle à d'une part la capacité de charge et d'autre part au nombre de commutations de la porte logique. L'énergie stockée dans une capacité peut s'écrire : $\frac{CV^2}{2}$.

Nous en déduisons que l'énergie dynamique moyenne consommée par opération peut s'écrire :

$$E_{dyn} \propto V_{DD}^2 \sum \alpha_f C_{Z,f} \quad (1.1)$$

Où α_f est le coefficient d'activité moyenne d'une porte et $C_{load,f}$ sa capacité de charge, V_{DD} est la tension d'alimentation et où nous supposons, $V_{ss} = 0V$.

Il est donc tentant de diminuer la tension d'alimentation du circuit digital pour en diminuer sa consommation. Une division par deux de la tension entraîne un gain de quatre sur la consommation d'énergie dynamique.

Ces derniers temps, nous avons vu la tension des circuits digitaux diminuer en raison de cette recherche d'économie d'énergie. Mais que ce passe-t-il lorsque cette diminution passe en dessous de la tension de seuil V_{TH} ? Dans cette condition, les transistors composants la porte logique ne sont jamais vraiment passant comme à la figure 1.2. Cela n'empêche pas de nombreuses réalisations de circuits digitaux de fonctionner correctement avec une tension d'alimentation proche ou sous seuil [15] [4] [13].

Dans ce chapitre, nous mettons en équations le courant passant dans un transistor quand V_{GS} est sous la tension de seuil. Ensuite, nous mesurons l'impact qu'ont ces variations sur la consommation dynamique. Nous finissons par décrire le phénomène d'injection de charges et des propriétés des distributions *log-normales* qui caractérisent les portes logiques fonctionnant en régime sous-seuil.

1.1 Dépendance exponentielle du courant pour des transistors en régime sous seuil.

Reprenons le transistor à la figure 1.1. Soit V_{gs} la différence de potentiel entre le point g et s. Nous pouvons voir sur la figure 1.2 que, une fois V_{gs} plus grand que la tension de threshold, le transistor devient passant. En passant le graphe figure 1.2 en échelle logarithmique nous pouvons étudier ce qu'il se passe en dessous du seuil (figure 1.4).

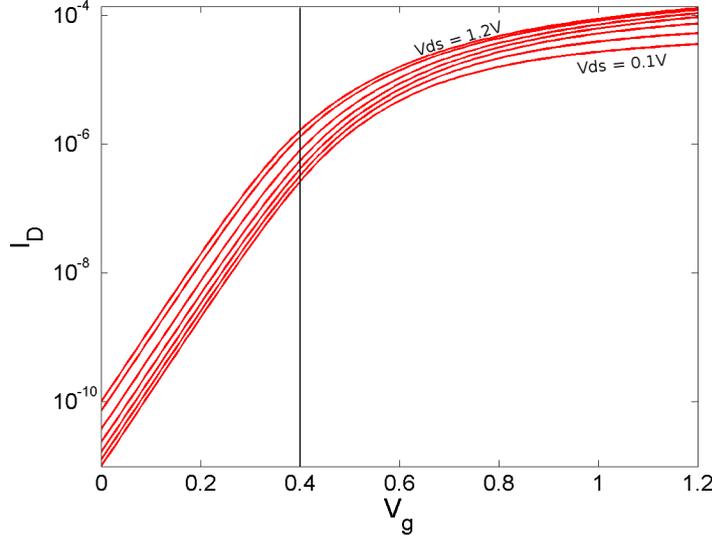


Figure 1.4 – Simulation ELDO de la caractéristique I_D/V_g d’un transistor NMOS en technologie cmos 65nm LP ($\frac{W}{L} = \frac{1}{0.3}$, Standard VT, 25° C, $V_{bulk} = 0$). V_D évolue en logspace de 0.1 à 1.2.

Dans ce cas, la tension est tellement petite que le transistor n’est jamais vraiment passant. V_{gs} est plus petit que la tension de threshold. Nous pouvons voir sur la figure 1.4 que nous avons clairement, en dessous de 0.4V, une dépendance exponentielle du courant avec V_{gs} . Reprenons l’équation utilisée par David Bol [5] :

$$I_{sub} = \mu_0 C_{ox} \frac{(n-1)}{L_{eff}} U_{th}^2 e^{\frac{V_{gs}-V_t}{nU_{th}}} (1 - e^{\frac{-V_{ds}}{U_{th}}})$$

Où μ_0 est la mobilité sans biais, C_{ox} la capacité d’oxyde de la grille, L_{eff} la longueur effective de canal, n le facteur d’effet du Body et U_{th} la tension thermique $\frac{kT}{q}$. Cette équation est réécrite par D. Bol sous la forme,

$$I_{sub} = I_0 \times 10^{\frac{V_{gs}+\eta V_{ds}}{S}} \times (1 - e^{\frac{-V_{ds}}{U_{TH}}})$$

Où I_0 est proportionnel à l’exponentielle de V_{th} et S est appelé ”subthreshold swing” et fait apparaître le facteur 10 : $S = \ln(10)nU_{th}$ [5].

Avec l’hypothèse que V_{th} suit une distribution normale [16] [17]. Comme $I_{sub} \propto e^{\frac{-V_{th}}{mV_T}}$ et qu’une normale négative reste une normale dont nous avons juste changé la valeur moyenne, nous observons effectivement des courants sous une forme log-normale (figure 1.5).

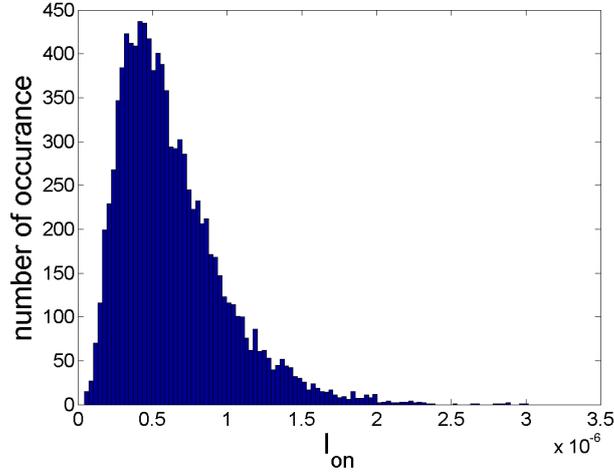


Figure 1.5 – Histogramme du courant d’une simulation Monte-Carlo réalisé avec un nmos de la technologie cmos 65nm LP ($\frac{W}{L} = \frac{1}{0.3}$, $L=0.06$, Standard VT, $25^\circ C$, $V_{bulk} = 0$) sur 10 000 itérations. Le drain et la grilles sont tous deux mis à 0.4V.

Plusieurs facteurs de type temporel ou spatial peuvent venir influencer V_{th} ([14] 3.2). Les facteurs temporels sont par exemple l’effet de la température ou celui de l’âge du circuit digital. Les facteurs spatiaux peuvent survenir de manière globale comme par exemple d’un die à l’autre (D2D), d’un wafer à un autre (W2W) ou encore d’un lot à l’autre (L2L) [14].

Pour les fluctuations internes au die, il en reste certaines qui se font par région (si le die est grand) et certaines qui se font localement. Les sources principales de ces variations locales sont la nature atomique des dopants, la netteté des lignes introduite par la lithographie ou encore les variations au niveau atomique de l’oxyde de grille [14].

Le phénomène de variation du V_{th} est amplifié par le fait que les composants deviennent de plus en plus petites. Pelgrom nous montre que les variations aléatoires locales diminuent quand la taille du transistor augmente car le paramètre aléatoire peut se moyennner sur une plus grande surface [17]. En diminuant la taille des transistors, nous augmentons donc les variances locales sur les composants du circuit digital.

Ces courants viennent charger la capacité de sortie de chacune des portes. Ce sont eux qui déterminent le temps d’inversion de la porte logique. Nous analyserons cela plus en détails dans la section suivante.

1.2 Impact du courant sur les délais de basculement des portes logiques.

Pour simplifier le travail des outils de synthèse, nous pouvons modéliser en deux phases le temps d'inversion d'une porte. La première partie couvre le délai de propagation interne à la porte logique t_{pro} et la deuxième le temps de montée ou de descente en sortie $t_{rise,fall}$ (figure 1.7). Pour calculer ces temps, plaçons des seuils et regardons quand le signal d'entrée A ou de sortie Z les franchissent (figure 1.7). Dans cette section, nous allons nous concentrer sur le temps de montée ou descente $t_{rise,fall}$.

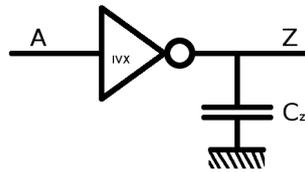


Figure 1.6 – schéma d'un inverseur simple avec sa charge capacitive en sortie.

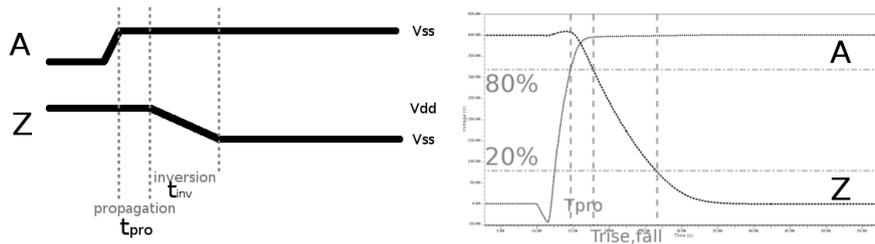


Figure 1.7 – Décomposition du temps de réaction en deux parties, la première, le temps de propagation, la deuxième, le temps de basculement. Application dans le cas d'une simulation ELDO avec les seuils placés à 80% et 20% de 0.4V. (inverseur X2 de la technologie UCLibs avec capacité de charge de 0.013pF).

Supposons que tout le courant I_{on} sert à charger la capacité de sortie C_Z . Nous savons que le courant représente des charges par seconde. La tension dans une capacité étant reliée directement au nombre de charges se trouvant dans cette capacité, nous pouvons dire que la vitesse de basculement d'une cellule ($t_{rise,fall}$) est alors inversement proportionnelle au courant I_{on} (eq 1.2).

$$I_{on} = C_Z \frac{\Delta V_Z}{\Delta t}$$

En supposant I_{on} constant pour simplifier le propos,

$$\Delta t = \frac{C_Z \Delta V_Z}{I_{on}} \quad (1.2)$$

Par l'équation 1.2, nous pouvons aussi dire que plus la capacité C_Z est grande, plus $t_{rise,fall}$ augmente.

Nous avons vu dans la section précédente que le courant de charge dépendait exponentiellement des fluctuations de dopage. En conséquence, le timing lui aussi dépend du dopage de manière exponentielle (eq 1.3).

$$t_{rise,fall} \propto \eta \frac{C_Z V_{dd}}{I_{on}} \quad (1.3)$$

En reprenant le modèle $I_{on} \propto I_{s0} e^{\frac{V_{dd}-V_{th}}{mV_T}}$

$$t_{inv} = \eta \frac{C_Z S V_{dd}}{2I_{s0}} e^{\frac{V_{th}-V_{dd}}{mV_T}} \quad (1.4)$$

Avec η un facteur de délai venant du fait que l'entrée ne suit pas une rampe parfaite [24]. En effet, I n'est pas constant sur toute la fourchette de tension. Enfin, C_Z est la capacité de sortie d'une cellule et V_{dd} la tension d'alimentation.

Nous voyons donc dans 1.4 que le délai d'inversion dépend lui aussi de l'exponentielle du dopage. Il suit donc lui aussi une distribution *log-normale*. Notons que le délai d'une chaîne de portes logiques suit lui aussi une distribution *log-normale* [13](figure 1.8). En effet, si nous dénommons par t_{di} le délai du chemin i :

$$\begin{aligned} t_{di} &= \sum t_{inv} \\ &= \eta \frac{C_s V_{dd}}{2I_{s0}} \sum e^{\frac{V_{th}-V_{dd}}{mV_T}} \\ &= \eta \frac{C_s V_{dd}}{2I_{s0}} e^{\frac{-V_{dd}}{mV_T}} \sum e^{\frac{V_{th}}{mV_T}} \end{aligned}$$

La somme de distributions *log-normale* reste une *log-normale* [24]. Nous avons donc des délais pour une suite de portes qui évoluent aussi en *log-normale*.

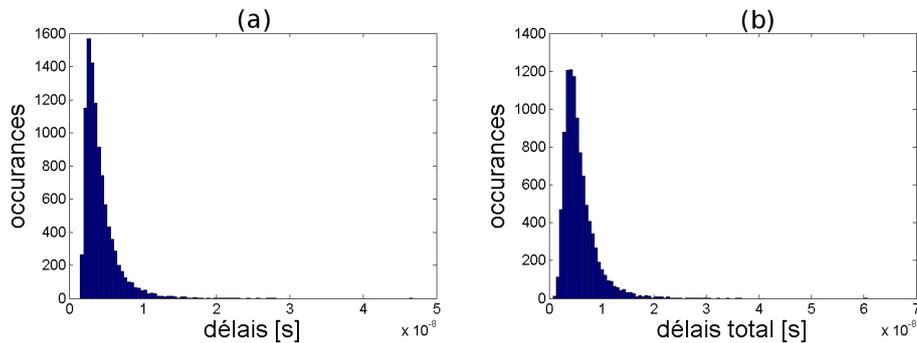


Figure 1.8 – Histogrammes des délais de descente t_{fall} des portes logiques (a) et délais total (b) $t_{fall} + t_{pro}$. Les deux sont la sortie d'un inverseur X2 de la librairie UCLibs. (65nm LP, 0.4V). Simulation réalisée avec une charge de 3.3fF pour des seuils de 80% à 20%. En (a) les deux seuils sont sur le signal de sortie. En (b), c'est 80% sur le signal d'entrée et 20% sur le signal de sortie. 10000 itérations Monte-Carlo en Eldo.

1.3 Impact de la variation du courant sur le facteur d'activité du circuit logique : α_F .

L'énergie dynamique consommée par notre circuit digital est proportionnelle à l'activité dans celui-ci. En effet, l'énergie dynamique consommée par opération peut s'exprimer comme (eq 1.1) :

$$E_{dyn} \propto V_{DD}^2 \sum \alpha_f C_{Z,f} \quad (1.5)$$

Où α_f représente le nombre de transitions d'une porte et $C_{Z,f}$ sa charge.

Sur base de résultats expérimentaux, Kamel montre dans [13] que pour un même circuit, nous observons une variation d'énergie non négligeable d'un die à l'autre. Pour chercher la source de cette variation, elle compare la consommation d'énergie de plusieurs circuits différents : deux chaînes d'inverseurs et une Sbox (fig 1.9).

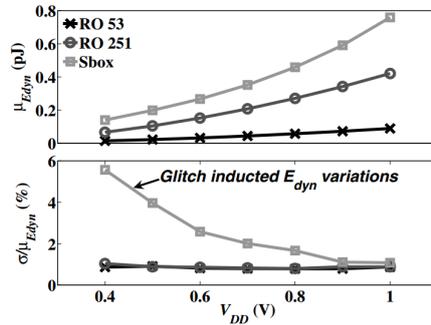


Figure 1.9 – Comparaison entre un circuit logique (Sbox) et deux chaînes d’inverseurs (RO53 et RO251). Résultat obtenu en technologie 65 nm LP CMOS à 25°C ref : [13]

Nous nous rendons directement compte que la variance des deux chaînes d’inverseurs est bien moindre que celle de la Sbox lorsque la tension diminue. Nous pouvons même voir que cette variance sur une chaîne d’inverseurs est presque indépendante de la tension d’alimentation (fig 1.9).

La différence entre les deux tient au fait que dans la chaîne d’inverseurs, le facteur d’activité α_F est fixé à 1 par construction. Seules les variations de la capacité de charge de chaque inverseur C_Z influe.

Comme les capacités de charge ne varient presque pas avec la tension, Kamel montre qu’en régime sous seuil, le facteur d’activité devient, tout comme le délai des portes logiques, lui aussi une variable aléatoire dont les paramètres dépendent de la tension d’alimentation [13].

Pour se convaincre intuitivement, nous pouvons nous baser sur la figure 1.10. Dans le cas où toutes les portes ont la même vitesse, le signal se propageant dans le chemin EFD (hachuré) et celui se propageant dans (A,B)CD (bleu) arrivent chacun en même temps à la porte de sortie. La dernière porte calcule donc une seule fois la sortie.

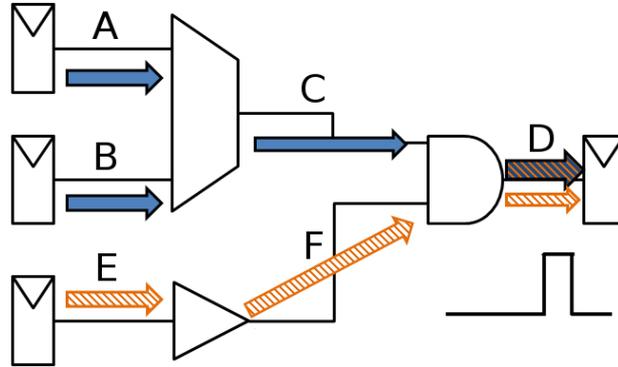


Figure 1.10 – Le coefficient d’activité varie car si le chemin plein ($\{A,B\}CD$) est plus rapide que le chemin hachuré (EFD), la sortie est recalculée deux fois.

Si maintenant, le chemin EFD est plus rapide que le chemin $(A,B)CD$, la sortie est d’abord calculée une première fois puis est recalculée lors de l’arrivée du signal passant par le chemin le plus lent. Nous avons donc fait une inversion en plus et donc augmenté le facteur d’activité. Cette apparition de glitch est induite par les variations dans les portes logiques dues au procédé de fabrication. Ces variations sont amplifiées par le fait que les portes sont de plus en plus petites [17].

Ce phénomène est encore amplifié parce qu’un glitch ainsi créé se propage dans la suite de la chaîne de portes logiques jusqu’à être arrêté par un registre. Or, dans un design en subthreshold, Zhai et les autres auteurs montrent qu’il faut, dans un pipeline, augmenter le nombre de portes logiques d’un chemin [24] ce pour relâcher les contraintes au niveau de la clock. Nous procédons à cette augmentation pour limiter l’écart-type relatif du délai sur un chemin. Le glitch se propagera donc plus longtemps et risquera à son tour de recréer de nouveaux glitches par un effet d’avalanche.

Kamel et les autres auteurs montrent que ces variations de consommation en 65nm peuvent aller jusqu’à 5.6% à 0.4 V pour la Sbox, alors que pour les chaînes d’inverseurs, cette variation reste de l’ordre de 1%.

1.4 Lien entre le facteur d’activité et la consommation dynamique d’un circuit logique.

Nous avons déjà donné plusieurs fois le lien entre l’énergie dynamique et le facteur d’activité par l’équation 1.1

$$E_{dyn} = \frac{V_{DD}^2}{2} \sum_{j=1}^n (\alpha_{F,j} C_{Z,j})$$

L'équation précédente n'est valide que dans le cas où les transitions sont complètes. Or, pour peu que le glitch généré par les deux entrées soit court comparé au temps de basculement, le nœud de sortie n'a pas le temps de s'inverser complètement. Ce résultat est montré par [13] dans le cas d'un nœud d'une Sbox (figure 1.11).

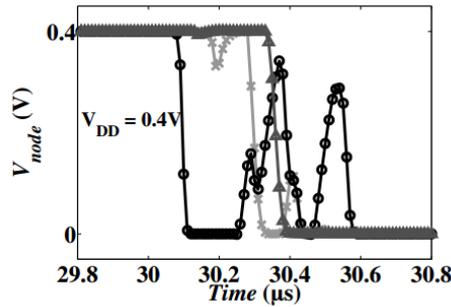


Figure 1.11 – Trois simulations SPICE Monte-Carlo réalisées par D. Kamel, sur un nœud interne à sa SBox [13]. Technologie nmos 65, $V_{dd} = 0.4V$. Source [13]

Cela illustre pourquoi que le facteur d'activité n'est pas forcément un nombre entier de transitions.

Les deux sections suivantes posent des bases pour le chapitre 2. D'un côté, nous verrons comment le délai de propagation augmente avec l'injection de charges et de l'autre nous placerons des bases mathématiques sur les distributions *log-normales*.

1.5 Couplage capacitif entre le nœud de la grille et celui du drain.

Prenons un résultat de simulation d'une chaîne de deux inverseurs avec une source en rampe (figure 1.12). Lorsque l'entrée d'une porte logique varie brusquement, nous voyons apparaître en sortie une surtension ou sous-tension en fonction du sens de la transition en entrée (fig 1.13). Ce phénomène est dû à un couplage capacitif entre la grille et le drain.

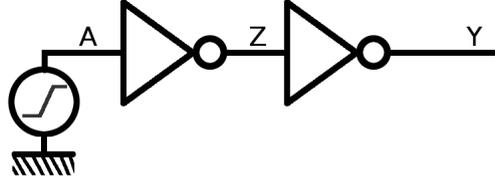


Figure 1.12 – Banc de test composé d’une chaîne de deux inverseurs, servant à mettre en valeur le phénomène d’overshoot.

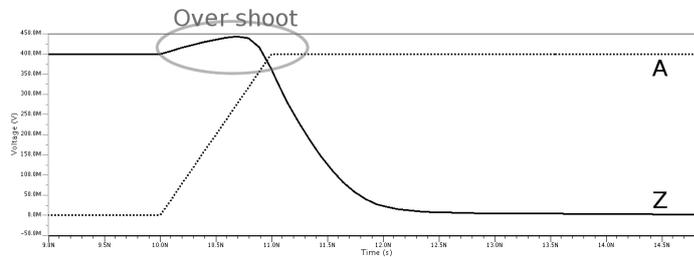


Figure 1.13 – Résultat de simulation pour le banc de test illustré à la figure 1.12. Nous pouvons y voir la tension des nœuds A et Z en fonction du temps. Sur la sortie Z apparaît une zone dans laquelle la tension dépasse la tension d’alimentation.

Ce phénomène influence le délai interne de la porte logique, plus particulièrement au niveau du temps de propagation et pas pour $t_{rise,fall}$. Lors de la mesure de ce délai, nous prenons le temps de propagation en nous basant sur le seuil par lequel, la tension passe (figure 1.7). Nous pouvons voir que si nous commençons par rehausser la tension via le couplage, il faut plus de temps pour passer ce seuil. Le temps de propagation t_{pro} a augmenté.

Une autre façon de le voir est que la variation du signal d’entrée injecte, via la capacité, des charges dans le nœud de sortie. C’est alors plus de charges que I_{on} doit retirer/ajouter dans la capacité de sortie. Comme le courant est fait de charges par seconde, à courant identique il faudra donc plus de temps pour décharger/charger la capacité.

Les capacités entre les différentes entrées d’un transistor peuvent être séparées en deux groupes. D’une part les capacités intrinsèques au canal et d’autre part les extrinsèques. Comme nous sommes sous seuil, l’importance des capacités parasites extrinsèques augmente relativement aux intrinsèques [3]. Nous pouvons voir sur la figure 1.14 un résumé des capacités sous seuil [3].

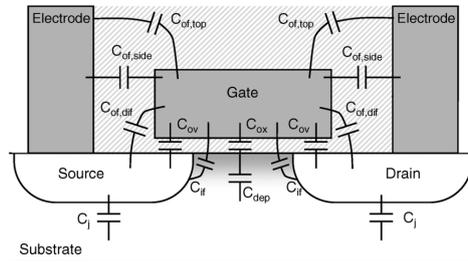


Figure 1.14 – Résumé des différentes capacités dans un transistor en régime sous seuil réalisé par David Bol. Source : [3]

La capacité qui nous intéresse plus particulièrement se trouve entre l'entrée (gate) et la sortie (drain). Elle est composée de la capacité de superposition entre le drain et la grille C_{ov} , de la capacité intrinsèque et extrinsèque entre la grille et le drain C_{if}, C_{of} et des capacités entre les contacts $C_{of,side}, C_{of,top}$.

Nous avons donc un phénomène de couplage via les capacités parasites des différents transistors liant l'entrée à la sortie. Ces injections dépendent de l'état dans lequel se trouve la porte. Elles peuvent être soit favorables soit défavorables à la vitesse de basculement de la porte logique.

1.6 Propriétés de la distribution log-normale dans le cadre des délais.

Nous avons vu que les courants et les délais se comportent de manière log-normale. Nous aurons donc besoin pour pouvoir faire une analyse Monte-Carlo, d'étudier un peu plus en détail la distribution log-normale. Nous voulons, à partir des moyennes et des écarts types mesurés, borner l'énergie par cycle des dies en conception.

Pour chacune des portes du design, les outils **cadence** actuels nous fournissent la valeur typique. C'est la valeur qui a le plus de chance d'apparaître pour cette porte.

Nos outils ont été développés en Java qui ne possède pas de librairie standard pour une génération random *log-normale*.

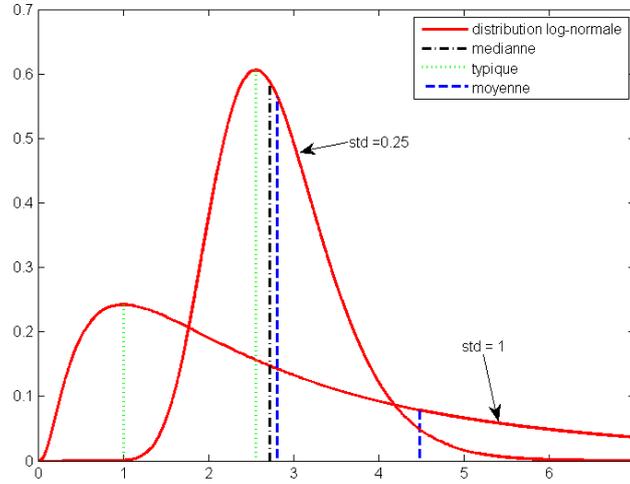


Figure 1.15 – Comparaison de deux log-normales $e^{(\mu+\sigma*Z)}$ où Z est une distribution normale normalisée centrée en zero et $\mu = 1$.

Cette section montre le lien entre la valeur typique du délai, la moyenne, et la valeur du μ et σ dans la distribution log-normale : $VA = e^{(\mu+\sigma*Z)}$ avec Z une VA normale, normalisée centrée en zero.

Nous pouvons voir sur la figure 1.15 que, contrairement à une normale, dans une log-normale la valeur typique diffère de la moyenne. Nous savons par [23] que la valeur typique est donnée par :

$$delaitypique : e^{\mu-\sigma^2} \quad (1.6)$$

$$delaimoyen : e^{\mu+\frac{\sigma^2}{2}} \quad (1.7)$$

$$mediane : e^{\mu} \quad (1.8)$$

$$variancedudelai : (e^{\sigma^2} - 1)e^{2\mu+\sigma^2} \quad (1.9)$$

posons K tel que :

$$\ln(delaitypique) = K = \mu - \sigma^2$$

Nous injectons $-K + K$ dans les équations 1.9

$$\begin{aligned} delaimoyen &= e^{\mu+\frac{\sigma^2}{2}-(K-K)} &&= e^{\frac{3}{2}\sigma^2+K} \\ variancedudelai &= (e^{\sigma^2} - 1)e^{2\mu+\sigma^2-(2K-2K)} &&= (e^{\sigma^2} - 1)e^{3\sigma^2+2K} \end{aligned}$$

Nous allons travailler en sigma/moyenne :

$$\frac{variancedudelai}{mean^2} = \frac{sigma^2}{mean^2} = e^{\sigma^2} - 1$$

ce qui nous donne :

$$\ln\left(\frac{\sigma^2}{\text{mean}^2} + 1\right) = \sigma^2 \quad (1.10)$$

$$\mu = K + \sigma^2 = \ln(\text{delaitypique}) + \sigma^2 \quad (1.11)$$

Nous pouvons désormais, à partir du générateur aléatoire normal de java, générer une log-normale sur base de la valeur typique et son sigma/mean.

Il faut maintenant pouvoir prédire une fourchette dans laquelle se trouvera la consommation de notre die. Nous avons choisi de prendre le percentile 3σ qui est un intervalle de confiance à 99,7%

$$(3\sigma)_{WC} = e^{\mu+3\sigma}$$

$$(3\sigma)_{BC} = e^{\mu-3\sigma}$$

Le *worst case* (WC) représente les portes les plus lentes tandis que le *best case* (BC) les plus rapides. μ et σ peuvent être ré-exprimés pour être applicables sur des données mesurées par la suite cadence :

$$\mu = \ln\left(\frac{\text{variance}}{\text{mean}^2} + 1\right) \quad (1.12)$$

$$\sigma^2 = \ln(\text{mean}) - \frac{\text{variance}}{2} \quad (1.13)$$

Nous avons maintenant en main les outils pour générer et vérifier un circuit digital et ses variations. Dans le chapitre suivant, nous verrons comment les données d'une netlist peuvent être extraites de manière standard.

Chapitre 2

Méthodologies statistiques de conception et de simulation des circuits logiques.

2.1 SSTA : Statistical static timing analysis

Pour introduire le SSTA, il faut commencer par introduire une technique plus traditionnelle, le STA (*static timing analysis*). Le STA est une technique déterministe et prend en terme de délai une valeur fixe pessimiste pour chaque porte logique. Ce choix surévalue la contrainte sur le timing [2] mais rend l'analyse "sûre". En relançant plusieurs fois l'analyse en changeant les conditions sur les portes logiques, nous vérifions que toutes rentrent dans les spécifications. Ces conditions sont appelées *corner* et représentent généralement des extrêmes de la technologie. Par exemple, le corner *SS* est le corner où tous les **PMOS** et **NMOS** sont lents (SS pour slow slow).

Le problème est que nous ne pouvons faire varier que des coins globaux. Il n'existe pas de méthode dans le STA pour simuler les variations locales des portes. Or, pour des circuits logiques de l'ordre du nanomètre, les variations locales d'une porte à l'autre sur un même die sont non négligeables [1] [2] [3]. Nous surévaluons donc la contrainte sur le timing pour que le circuit reste fonctionnel au détriment d'autres optimisations [2]. Par ailleurs, avec les augmentations de variabilité, il y a de plus en plus de coins et donc de simulations à tester [2].

Pour palier à ces problèmes, une nouvelle méthode fait son apparition, la SSTA : *Statistical static timing analysis*.

Contrairement à la STA, la SSTA remplace le délai de chaque porte ou bloc par une variable aléatoire équivalente. Pour gagner du temps de simulation, ce sont généralement des gaussiennes qui sont choisies [10]. La

méthode approxime ensuite le délai d'arrivée par chaque chemin par des *Cumulative Probability Distribution Functions* (CDFs) [10]. Nous obtenons donc pour chaque chemin une variable aléatoire équivalente pour laquelle nous pouvons vérifier que les contraintes de timing sont dans l'intervalle de confiance. En général, nous prenons comme dans le cas de la STA, le chemin le plus contraignant [2].

Cette technique, permet d'évaluer la distribution statique du délai des chemins critiques. Elle permet donc de s'assurer qu'un circuit digital reste fonctionnel malgré les variations locales. Mais elle ne mesure pas l'impact qu'ont les différentes variables aléatoires sur la variation de consommation d'un die à l'autre.

2.2 Simulation de la puissance consommée.

Lors de la création d'un circuit digital, nous disposons de plusieurs outils pour évaluer en cours de création la puissance consommée par le circuit. Le flot normal pour créer un circuit est expliqué à l'annexe B.1 [6]. Voici comment, à partir d'une netlist, nous pouvons simuler et récupérer la valeur de puissance consommée par le circuit.

Une fois la netlist créée, nous pouvons déjà avoir une idée de la consommation via les suites d'outils. Mais ces outils ne peuvent que considérer un facteur d'activité α_f standard pour chaque nœud. Il nous faut donc une manière plus précise pour déterminer la consommation en fonctionnement *réel*.

Le rapport d'activité : saif

Nous avons montré au chapitre 1.4 que, sur base du facteur d'activité de chaque porte logique et de la capacité que cette porte doit driver, nous pouvons estimer sa consommation. Nous possédons déjà via la netlist, la capacité de charge C_Z . Reste à trouver le facteur d'activité de chaque porte composant la netlist.

Pour ce faire, nous simulons la netlist en condition d'utilisation et nous enregistrons toutes les transitions qui se font [9]. Le fichier saif peut alors être généré à partir de cet enregistrement. Il contient un résumé des différentes transitions pour chaque nœud (code 2.1).

Nous pouvons voir que nous enregistrons le nombre de transitions de 0 vers 1 (ou l'inverse) mais aussi vers une indétermination. En connaissant le nombre d'opérations faites dans la simulation, nous pouvons déduire l'activité moyenne de chacune des portes par opération ou par cycle d'horloge.

```

1  (INSTANCE maSbox
2    (NET
3      (data_in \[0\]
4        (T0 12900000) (T1 17350387) (TX 0)
5        (TC 51) (IG 0)
6      )

```

Figure 2.1 – Exemple de contenu d'un fichier saif pour une noeud d'entrée.

Générer le rapport de puissance.

En fournissant aux outils de simulation le fichier saif ainsi produit, nous pouvons maintenant générer un rapport qui nous indique la consommation de notre circuit.

Ce rapport contient de nombreuses données telles que,

- La consommation statique (leakage).
- La consommation dynamique interne à chaque porte.
- La consommation dynamique pour switcher chaque nœud.
- La consommation totale.

(figure 2.2)

A nouveau, si nous connaissons le nombre d'opérations faites, nous pouvons intégrer cette puissance sur le temps de simulation et diviser par le nombre de cycles correspondant pour savoir combien le circuit consomme par opération.

Maintenant si il y a, comme expliqué au chapitre précédent, des glitches qui apparaissent, le nombre de transitions dans le fichier **saif** va augmenter. La consommation va donc varier aussi.

Dans ce travail, nous nous intéressons principalement à la partie dynamique de la consommation de puissance. Des solutions ont déjà été proposées pour calculer la consommation statique de manière statistique [18] [20].

Nous nous intéressons plus particulièrement sur le rapport $\frac{\sigma}{\mu}$ de celle-ci. En effet [13] montre que, d'un die à l'autre pour un même circuit, on a une grosse variation de consommation. C'est ce phénomène que nous allons quantifier dans la suite.

1		Internal	Switching	Leakage	Total		
2	Power Group	Power	Power	Power	Power (%)	Attrs	
3							
4	io_pad	0.0000	0.0000	0.0000	0.0000	(0.00%)	
5	memory	0.0000	0.0000	0.0000	0.0000	(0.00%)	
6	black_box	0.0000	0.0000	0.0000	0.0000	(0.00%)	
7	clock_network	0.0000	0.0000	0.0000	0.0000	(0.00%)	
8	register	0.0000	0.0000	0.0000	0.0000	(0.00%)	
9	sequential	0.0000	0.0000	0.0000	0.0000	(0.00%)	
10	combinational	3.2228e-04	3.2506e-04	3.8317e-06	6.5117e-04	(100.00%)	
11							
12	Total	3.2228e-04 mW	3.2506e-04 mW	3.8317e-06 mW	6.5117e-04 mW		
13	1						

Figure 2.2 – *Exemple de rapport de puissance, dans ce cas pour un circuit purement combinatoire.*

Chapitre 3

Analyse des variations d'une porte logique sous-seuil.

Nous l'avons vu dans le chapitre 1, la différence de timing d'une porte à l'autre entraîne l'apparition de glitches augmentant la consommation dynamique du circuit logique. Pour pouvoir simuler ce phénomène pour des circuits digitaux complexes, il faut atteindre un niveau d'abstraction plus élevé pour limiter le temps de simulation.

Nous avons besoin d'un modèle pour les différents délais des portes logiques et ce modèle doit si possible être simple pour rester dans des temps de simulations viables. Il servira à modéliser les délais dans une netlist de façon à se rapprocher le plus possible des variations que nous retrouvons dans les circuits digitaux finaux.

Puisque les outils standards nous donnent les délais typique des différentes portes dans le fichier **sdf**, il faut que le modèle se base sur ces délais. Nous essayons donc de déterminer l'écart type (σ) en nous basant sur ces valeurs typiques que nous supposerons directement liées aux délais moyens (μ) de chaque porte.

Nous l'avons vu dans la section 1.2, ces délais sont proportionnels à la charge en sortie de la porte logique. Si nous traçons σ et μ en fonction de cette charge (figure 3.1), nous pouvons voir que ni σ ni μ ne sont constants en fonction de la charge.

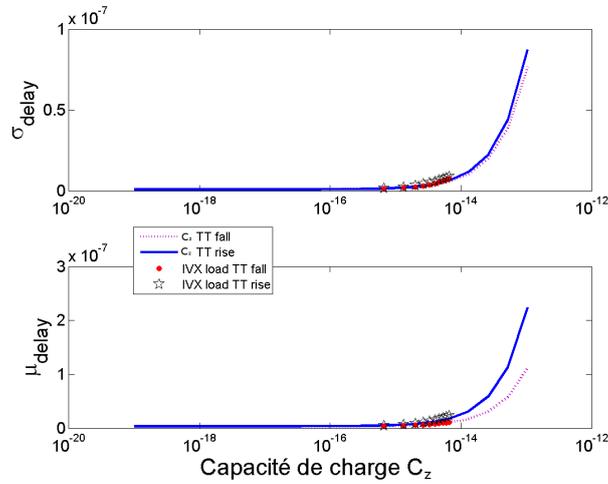


Figure 3.1 – Évolution du σ et du μ du délai d'un inverseur en fonction de la charge. Simulation réalisée sur un inverseur X2 en eldo 65nm LP, UCLibs. 0.4V. les seuils sont placé à 60% en entrée et 20% en sortie. Un inverseur X2 sans mismatch sert de source en entrée et une capacité variable sert de charge. Les points et les croix sont obtenus en remplaçant la charge par un nombre d'inverseurs X2 variable. Le tout est simulé en TT (typical typical).

Si nous regardons la littérature, la plupart des articles traitent le problème de variation du délai à charge fixe [11].

Nous allons dans un premier temps essayer de travailler avec le rapport $\frac{\sigma}{\mu}$ du délai en fonction de la charge, ce pour essayer d'isoler les différents phénomènes qui l'influencent. Ensuite, nous essayerons de déterminer des modèles en fonction du temps d'inversion moyen (μ).

Pour nos mesures, nous avons utilisé en entrée un seuil à 60% et en sortie un seuil de 20% pour tracer la figure 3.1. Ce sont les seuils utilisés dans la librairie pour déterminer le délai d'inversion d'une porte.

3.1 Dépendance sur la pente du signal d'entrée et le seuil de mesure.

Pour déterminer le rapport $\frac{\sigma}{\mu}$ en fonction de la charge, nous avons simulé l'inverseur le plus petit de la librairie UCLibs. Nous supposons que c'est lui qui a les plus fortes variations de sortie [17]. Comme source d'entrée, nous avons mis un inverseur sans *mismatch*. Nous avons aussi mis une capacité variable en sortie (figure 3.2). Le $\frac{\sigma}{\mu}$ du délai en fonction de la charge C_Z est donné à la figure 3.3. Ensuite, en remplaçant la capacité C_Z par une série de 1 à 10 inverseurs placés en sortie, nous obtenons une série de points sur la figure 3.3.

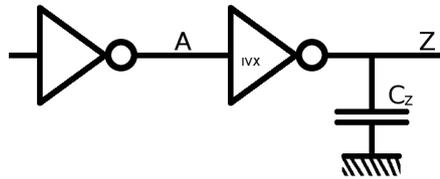


Figure 3.2 – Banc de test pour la simulation ELDO de la figure 3.3. Nous déterminons l'évolution du rapport $\frac{\sigma}{\mu}$ des délais avec la charge. Le signal d'entrée A est généré par un inverseur sans *mismatch*.

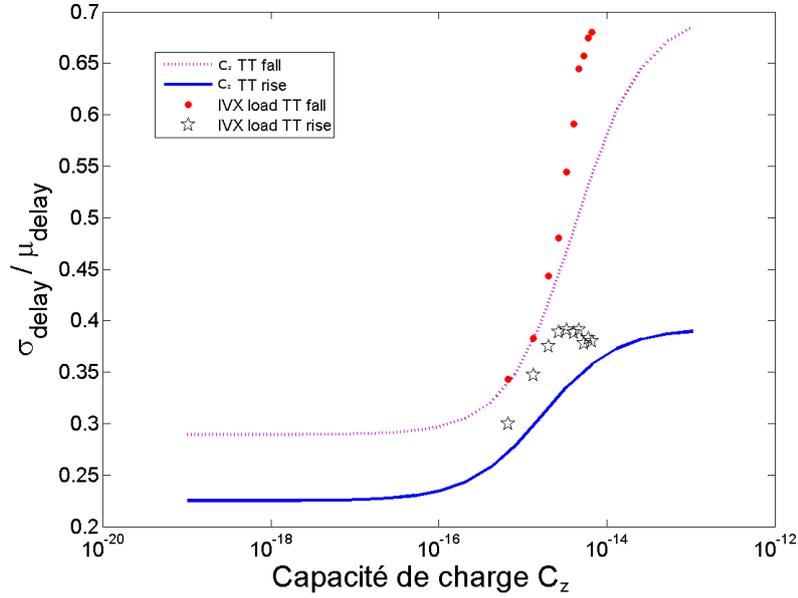


Figure 3.3 – Rapport $\frac{\sigma}{\mu}$ des délais d'un inverseur en fonction de la capacité de sortie. Un inverseur sans mismatch est placé à l'entrée de l'inverseur testé. Simulations réalisées sur 10000 mesures par capacité C_z . (UCLibs, 65nm, $V_{dd} = 0.4V$, IVX2.)

Si nous nous concentrons sur la courbe en fonction de la capacité C_z , nous voyons apparaître trois régions sur cette figure (figure 3.4). Dans la première, à faible capacité (a), nous pouvons voir que le $\frac{\sigma}{\mu}$ ne dépend pas de la valeur de la charge. Dans la deuxième (b), nous voyons apparaître une transitoire vers la troisième région dans laquelle le $\frac{\sigma}{\mu}$ est également indépendant de la charge(c).

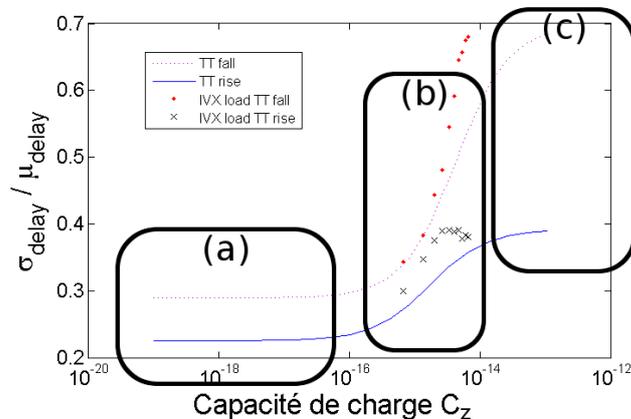


Figure 3.4 – la Figure 3.3 peut être décomposée en trois régions (a), (b) et (c).

Le premier plateau (a) correspond à la zone où la capacité de charge C_Z est négligeable par rapport à la capacité de sortie de la porte testée. Dans ce cas de figure, nous avons une capacité presque constante à charger dans cette première région. En effet, la capacité maximum sur la pin de sortie de la porte est de 0.035pF (valeur tirée de la librairie UCLibs : *SUB65LPSVT typ 04V 25C.lib*). Si nous considérons la capacité interne à la porte négligeable lorsque elle atteint un ordre de grandeur en dessous de celle de la charge, nous arrivons bien aux alentours de 10^{-3}pF soit du 10^{-15}F .

Retournons à la figure 3.3. Lorsque la charge est composée de plusieurs inverseurs mis en parallèle, nous voyons que les points simulés ne suivent pas la courbe lorsque la charge est une simple capacité variable. Notons que les abscisses de ces points sont plus proches de la capacité réelle à charger puisque composés de portes logiques comme un circuit réel. Or les capacités équivalentes de ces points se trouvent dans la région de transition (b). Ce qui veut dire que simuler un circuit demande un modèle valide dans la zone (b). Garder le $\frac{\sigma}{\mu}$ constant avec la charge n'est probablement pas le meilleur modèle possible.

Nous rajouterons en annexe A.1 les histogrammes de différents délais en différents points de la courbe qui montrent que nous gardons bien une distribution *log-normale* tout au long de la courbe $\frac{\sigma}{\mu}$ en fonction de la charge C_Z .

Dans un premier temps, nous allons analyser l'effet qu'a le signal d'entrée sur le $\frac{\sigma}{\mu}$. Nous remplaçons la source imparfaite qu'est un inverseur par une source de tension en rampe. Nous mettons une charge passive capacitive en sortie pour éviter un couplage rétro-actif.

Voyons ce qui se passe lorsque nous faisons varier la vitesse de basculement de la tension d'entrée et la capacité de charge (figure 3.5).

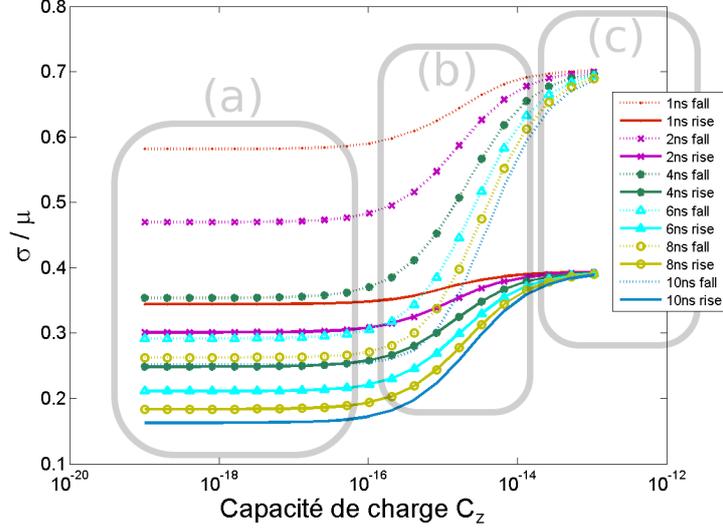


Figure 3.5 – Rapport $\frac{\sigma}{\mu}$ des délais de l'inverseur en fonction de la capacité de charge pour différentes rampes en entrée. La source de 1ns met 1ns pour basculer, celle de 10ns en met 10. Réalisé avec un inverseur X2 de la technologie cmos 65, librairie UCLibs à 0.4V.

Nous remarquons (fig 3.5) que plus le signal d'entrée bascule rapidement plus le premier plateau (a) monte au niveau du deuxième plateau (c) (def : (a) (c) sur la figure 3.4). La forme du signal d'entrée influence donc bien le rapport de $\frac{\sigma}{\mu}$.

La tension en entrée $V_A(t)$ n'est pas une constante et ne vaut pas V_{dd} durant toute la transition. Zhai ne prend pas en compte ce phénomène dans son équation 3.2.

$$t_{fall} = \frac{C_Z V_{dd}}{2I_{S0}} e^{\frac{V_{th,j} - V_A(t)}{mV_T}} \quad (3.1)$$

$$\approx \eta \frac{C_Z V_{dd}}{2I_{S0}} e^{\frac{V_{th,j} - V_{dd}}{mV_T}} \quad (3.2)$$

Il y a deux approximations dans la dernière équation (3.2), la première est que I_{on} n'est pas constant durant toute la transition [11]. La deuxième est que le η rajouté par Zhai ne permet pas d'expliquer la variation du $\frac{\sigma}{\mu}$. En effet, ce coefficient influence chaque valeur de la même manière et par conséquent, applique le même facteur d'échelle sur l'écart type (σ) et la moyenne (μ)(eq 3.3).

Dans un souci de simplification, nous ferons quand même l'hypothèse que $I_{on} \simeq cst$.

$$t_{fall} = \eta \frac{V_{dd}}{2I_{S0}} e^{-\frac{V_{dd}}{mV_T}} e^{\frac{V_{th,j}}{mV_T}} = \eta \cdot constante \cdot e^{\frac{V_{th,j}}{mV_T}} \quad (3.3)$$

A même charge, toutes les portes ne sont pas influencées de la même manière. Dans le graphe du haut de la figure 3.6, nous comparons deux rampes d'entrée, une de 1ns et une de 10ns. Nous pouvons voir que pour l'entrée à 10 ns et pour la porte la plus rapide, l'entièreté de la bascule se fait avec $V_A(t) \neq V_{dd}$. Pour les portes plus lentes, une partie de l'inversion se fait quand $V_A(t) = V_{dd}$. plus la porte est lente, plus cette partie prend un temps important relativement au temps ou $V_A(t) \neq V_{dd}$. Le recouvrement entre le signal d'entrée variable et la sortie en inversion est plus petit proportionnellement au temps total de basculement. La barre grise verticale marque le moment où $V_A(t) = V_{dd}$. En comparant ces deux signaux, nous pouvons dire que, contrairement à ce que présume Zhai, le facteur η devrait être différent pour chacun.

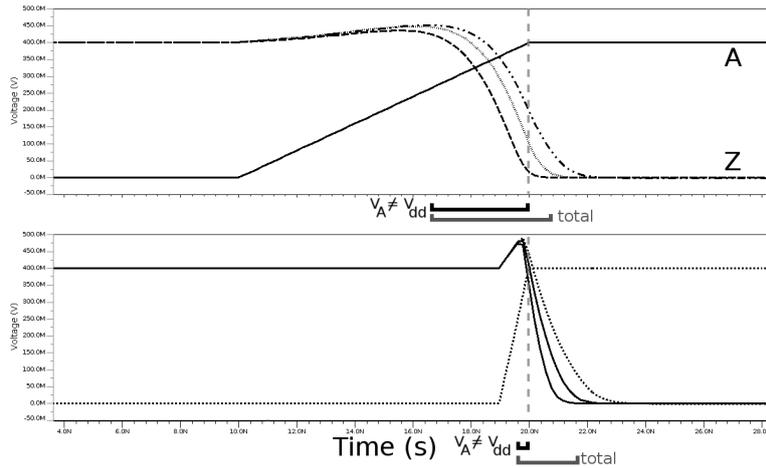


Figure 3.6 – Comparaison de l'effet de deux rampes d'entrées différentes lorsque la porte est soumise à une faible capacité de charge ($1e - 7 [pF]$). Nous prenons les rampes extrêmes testées, une de 1ns et une de 10 ns. En pointillé gris vertical, des lignes délimitant la fin de la rampe. Technologie UCLibs, 0.4 V, IVX2. Trois simulations Monte-Carlo dans les deux cas.

Nous savons que la vitesse de basculement de la porte limite ou augmente le temps relatif de superposition. Or changer la capacité de sortie change ce temps de superposition (figure 3.7). Si nous mettons une forte capacité C_Z en sortie, le temps de transition de l'entrée A devient négligeable par rapport à celui de sortie Z. La fonction de Zhai (eq : 3.2) s'applique bien et c'est la variance de I_{on} quand $V_A = V_{dd}$ qui domine. Nous sommes alors dans la région (c) de la figure 3.4.

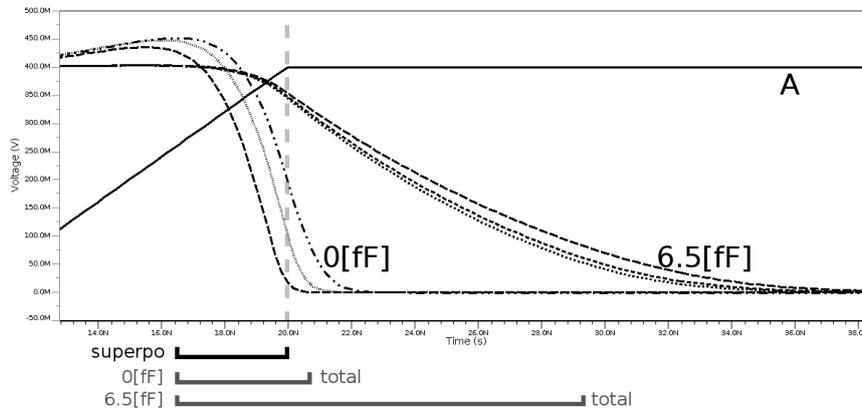


Figure 3.7 – Comparaison de l'effet de deux charges sur le même inverseur pour une rampe d'entrée fixée à un basculement en 10ns. Les charges sont $1e-7$ [pF] pour la plus faible et 0.0065 [pF] pour la deuxième.

Dans le cas où nous avons une très faible capacité C_Z , le temps de superposition où le nœud A et le nœud Z changent tous les deux est différent d'une porte rapide à une porte lente. Durant la superposition, la porte logique est plus lente que ce qu'elle ne serait si le signal d'entrée était bien V_{dd} .

La figure 3.8 appuie cette hypothèse. En changeant les seuils de détection pour ne plus prendre que le temps de bascule de Z (60% à 20%), la figure montre que le phénomène a moins d'influence sur $t_{rise,fall}$. Ce qui est logique puisque la partie superposée durant l'inversion est principalement composée du temps de propagation t_{prop} .

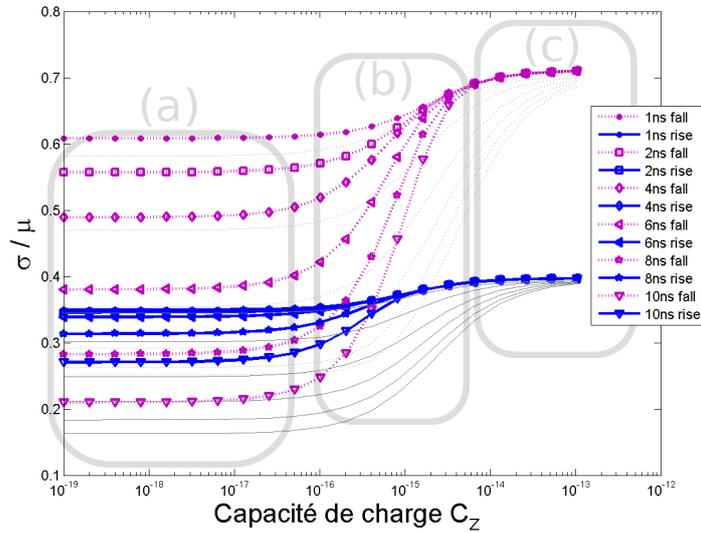


Figure 3.8 – Rapport $\frac{\sigma}{\mu}$ des délais de l'inverseur, cette fois sur le temps de bascule rise, fall de la porte (seuil de 60% à 20%). Nous pouvons voir qu'il faut une plus petite capacité de charge avant que l'effet du signal d'entrée ne se fasse sentir. De plus le rapport à faible capacité est plus proche de la variance de I_{on} . En gris, les mêmes courbes mais pour le délai total de la porte.

Voyons dans quelle mesure ce recouvrement influence le σ et le μ sur le temps total nécessaire à l'inversion de la porte. Sur la figure 3.9, nous pouvons voir que le phénomène de superposition fait augmenter σ et μ ($\sigma_{10ns} > \sigma_{1ns}$) et que l'effet sur μ est plus grand que celui sur le σ . Pour les plus faibles capacités, $\mu_{1ns} \approx \frac{\mu_{10ns}}{3}$ alors que $\sigma_{1ns} \approx \frac{4\sigma_{10ns}}{5}$. Ce qui confirme bien la figure 3.5. Nous pouvons donc dire que l'effet se fait principalement en augmentant le temps moyen de bascule qui varie plus vite avec le temps de bascule de la source que le σ .

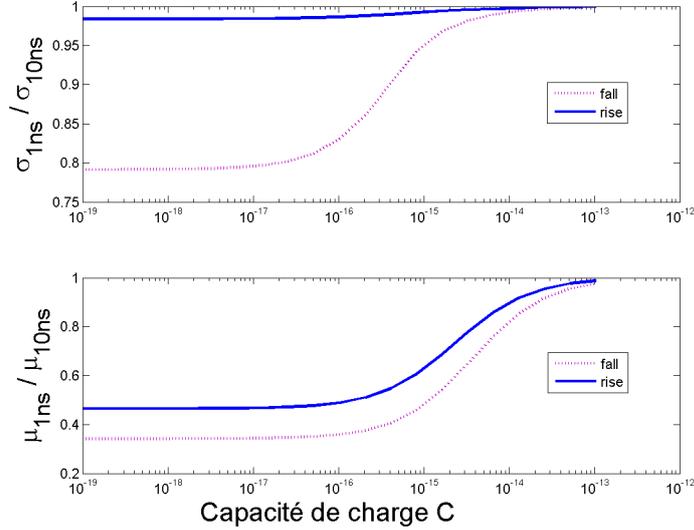


Figure 3.9 – Rapport entre les écarts types et les moyennes pour différentes valeurs de charge. Nous comparons le cas d’une rampe à 1ns avec une à 10ns.

Nous avons ainsi montré que la forme du signal d’entrée influence le rapport $\frac{\sigma}{\mu}$ pour les faibles capacités.

3.2 Effet de la charge sur le rapport $\frac{\sigma}{\mu}$.

Maintenant que nous savons pourquoi le $\frac{\sigma}{\mu}$ évolue avec la charge et le signal d’entrée, regardons pourquoi les mesures ne sont pas sur la courbe d’une capacité simple dans le cas de plusieurs inverseurs.

Dans les deux sous-sections suivantes, nous expliquons la raison pour laquelle le couplage capacitif entre l’entrée et la sortie n’influe pas directement sur le $\frac{\sigma}{\mu}$. Ensuite, dans la sous section suivante, nous expliquons en quoi le couplage influe si la charge est active.

3.2.1 Dans le cas d’une charge passive.

Dans un premier temps, gardons une simple capacité comme charge C_Z et supposons la source du signal d’entrée parfaite. Comme vu en section 1.2, le délai pour une porte logique peut s’écrire :

$$t_{fall,rise} = \eta \frac{Q}{I_{on}} = \eta \frac{C_Z V_{dd}}{I_{on}} \quad (3.4)$$

Si nous voulons ajouter le couplage capacitif dans l’équation, nous pouvons décomposer Q en deux parties : en premier lieu, nous avons Q_C qui est

le nombre de charges nécessaires à injecter pour basculer le nœud de sortie. Et en deuxième lieu nous avons Q_{inj} qui représente le nombre de charges injectées par le signal d'entrée via la capacité entre le nœud d'entrée et le nœud de sortie C_{AZ} . Nous savons que les charges stockées dans la capacité sont :

$$Q_{inj} = C_{AZ}\Delta V$$

Supposant $V_{ss} = 0$. En remplaçant dans 3.4 Q par la charge totale, nous obtenons,

$$t_{inv} = \eta \frac{Q_{total}}{I_{on}} = \eta \frac{(C_Z + C_{AZ})\Delta V_{AZ}}{I_{on}} \quad (3.5)$$

Où ΔV_{AZ} est la différence de tension en A prise au moment où le signal A atteint le seuil de 60% et celle prise au moment où Z atteint la fin de son basculement.

$$\Delta V_{AZ} = V_{dd} * 40\% = V_{dd} \frac{2}{5}$$

Comme ΔV_{AZ} est déterminé et fixe, l'injection de charges ne peut être le facteur qui fait varier le $\frac{\sigma}{\mu}$. En effet, si nous gardons l'hypothèse que la variance des capacités est faible comparée à celle du courant I_{on} [13], le C_{AZ} n'amène qu'une variance négligeable comparée à celle des courants.

Les charges injectées par le couplage entre l'entrée et la sortie ne participent donc pas directement à la variation du $\frac{\sigma}{\mu}$.

3.2.2 Dans le cas d'une charge active.

Dans ce travail, nous qualifions de charge active une charge qui possède une source de tension. Par exemple, plaçons à la sortie de l'inverseur testé une série d'inverseurs servant de charge. Pour chacun de ces inverseurs j , nous ajoutons une capacité variable (C_{Y_j}) à leur sortie (Y_j) (figure 3.24).

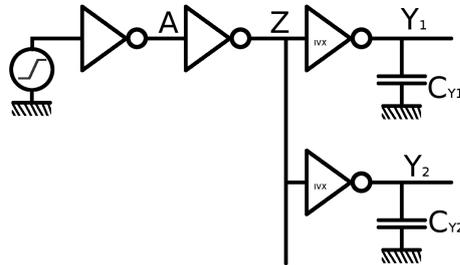


Figure 3.10 – Banc de test pour mettre en évidence le phénomène de rétro action de la charge active. Le nombre d'inverseurs de charge sur le nœud Z peut varier.

Dans ce cas la charge Q_{retro} rétro-injectée de Y à Z devient elle aussi une variable aléatoire. Par exemple, mettons un seul inverseur pour charge, inverseur j auquel nous ajoutons en sortie une capacité C_{Y_j} (figure 3.24). Nous savons qu'une fois un certain seuil passé, la charge active va s'inverser. Le couplage via la capacité C_{ZY} qu'il y a entre le nœud Z et le nœud Y_j induit une charge Q_{retro} rétro injectée dans le nœud Z .

Contrairement à Q_{inj} , Q_{retro} est une variable aléatoire de variance non négligeable. En effet, la charge Q_{retro} tout comme la charge Q_{inj} s'exprime :

$$Q_{retro} = C_{ZY} \Delta V_Y$$

Où C_{ZY} est la capacité entre Z et Y . Bien que nous supposons la variance de C_{ZY} négligeable, Q_{retro} reste une variable aléatoire. En effet, ΔV_Y est une variable aléatoire.

$$\Delta V_Y = \frac{Q_C}{C_{Y_j}} = \frac{I_j}{C_{Y_j} \Delta t}$$

Où I_j est le courant $I_{on}(t)$ de l'inverseur j servant de charge et Δt le temps qu'il faut entre le moment où l'inverseur de charge commence à basculer et le moment où le signal en Z atteint son seuil de mesure.

$$Q_{retro} = \frac{I_j C_{ZY}}{C_{Y_j} \Delta t}$$

Où I_j est une log-normale et Δt est elle aussi une log-normale dépendante du courant de l'inverseur testé.

$$\Delta t = \frac{\Delta V_{seuil} C_Z}{I_{on}}$$

Où I_{on} est le courant circulant dans l'inverseur dont nous voulons déterminer la variance et ΔV_{seuil} la différence de tension du nœud Z entre moment où l'inverseur de charge commence à réagir et le moment où la tension en Z V_Z atteint le seuil de mesure (20%) 3.12. Nous pouvons donc réinjecter dans l'équation de Q_{retro} :

$$Q_{retro} = \frac{I_j I_{on} C_{ZY}}{C_Z C_{Y_j} \Delta V_{seuil}} \quad (3.6)$$

Q_{retro} est donc bien une variable aléatoire de type log-normale. Nous avons $I_j I_{on}$ comme VA multiplié par un facteur constant.

Nous pouvons vérifier le phénomène : lorsque la capacité C_{Y_j} est à son minimum (=0), nous pouvons faire une analyse Monte-Carlo (figure 3.11). Nous observons, mise en évidence sur la figure 3.11, une région où la pente de la tension de sortie varie brusquement. Nous remarquons aussi que ce phénomène survient juste au moment où la charge commence à basculer (figure 3.12).

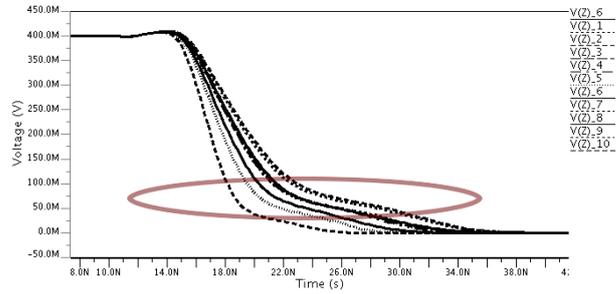


Figure 3.11 – Simulation d'un inverseur avec pour charge dix inverseurs à faible capacité C_{Y_j} . 65nm,LP,UCLibs. 10 itérations.

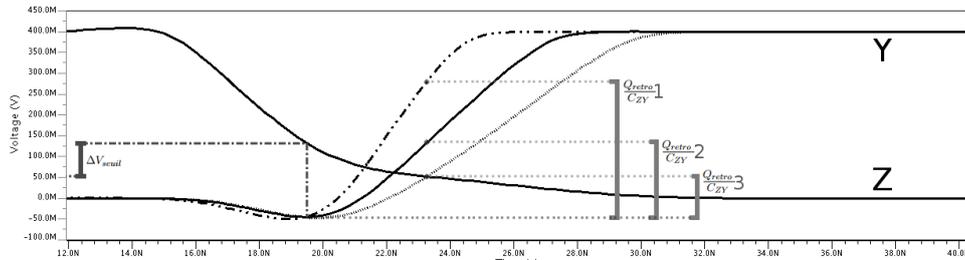


Figure 3.12 – Même simulation que 3.11 mais cette fois, nous n'avons gardé qu'une itération et nous avons superposé les signaux des inverseurs de charge. Nous pouvons voir que la charge rétro injectées Q_{retro} par chaque inverseur est bien différente d'un inverseur à l'autre. Le seuil de fin de mesure a été descendu sur le graphique de 0.08V à 0.05V pour amplifier visuellement le phénomène.

Si maintenant nous augmentons la capacité en sortie de la charge active pour ralentir son inversion, nous nous rendons compte que le phénomène de rétro action diminue (en gris clair sur la fig 3.13). Ce qui corrobore l'équation 3.6. En réitérant pour plusieurs valeurs de capacité mais toujours pour dix inverseurs, nous obtenons la figure 3.14. Cette figure nous montre que le rapport $\frac{\sigma}{\mu}$ diminue si nous augmentons la charge des inverseurs de sortie.

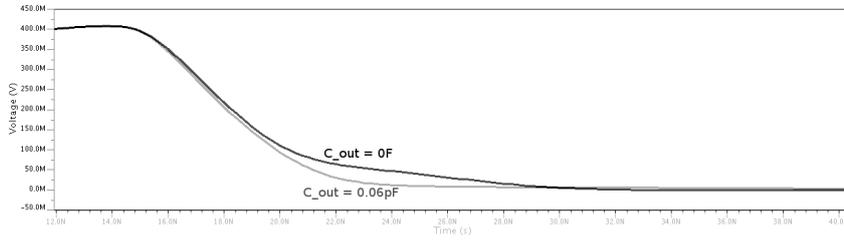


Figure 3.13 – Sortie Z de l'inverseur testé. En gris clair, la capacité C_{Y_j} est augmentée à 0.00666pF par inverseur, en foncé, la capacité C_{Y_j} est proche de 0 ($1e-7\text{pF}$).

L'équation 3.6 nous montre que la distribution du délai reste une *log-normale* puisqu'une multiplication d'exponentielle reste une exponentielle (figure 3.15).

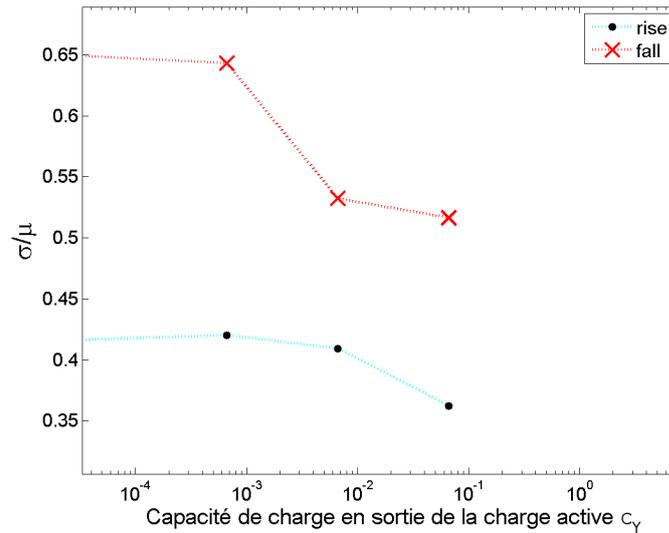


Figure 3.14 – Simulation Monte-Carlo du rapport $\frac{\sigma}{\mu}$ des délais de sortie de l'inverseur drivant Z en fonction de la capacité C_{Y_j} . Limitation du phénomène rétro actif du couplage en augmentant la capacité de charge de la charge active. La première charge testée hors du graphe est 10^{-7} [pF]. 10000 itérations avec 10 IVX2 en charge.

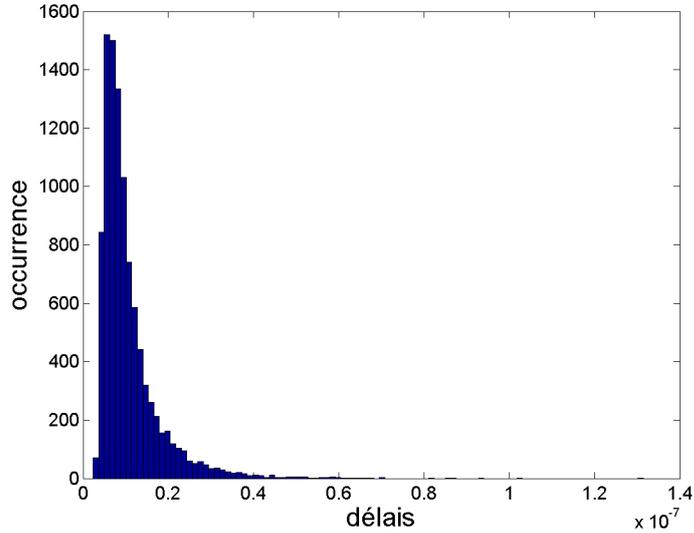


Figure 3.15 – Histogramme obtenu lorsque le phénomène de rétro injection a le plus d'effet ($C_{Y_j} \approx 0$). La distribution reste une log-normale. Réalisé avec 10 inverseurs en charge sur le nœud Z.

Nous avons montré que le rapport $\frac{\sigma}{\mu}$ d'un inverseur dépend non seulement de la capacité qu'il drive mais du type de la charge et de la capacité que cette charge drive. Et nous constatons que les simulations sont cohérentes avec le modèle mathématique. Nous voyons sur la figure 3.3 qu'à partir de 6 inverseurs, l'effet s'estompe dans le cas de t_{rise} . Nous supposons que c'est simplement l'effet de moyenne sur l'ensemble des $Q_{retro,j}$ renvoyés par les différents inverseurs de charge.

Nous pouvons approfondir l'analyse pour voir l'effet des différents paramètres en retirant le mismatch sur les différentes parties des portes logiques. Commençons par supprimer le mismatch de la charge active (figure 3.16). Nous voyons que Q_{retro} est bien dépendant de I_{on} .

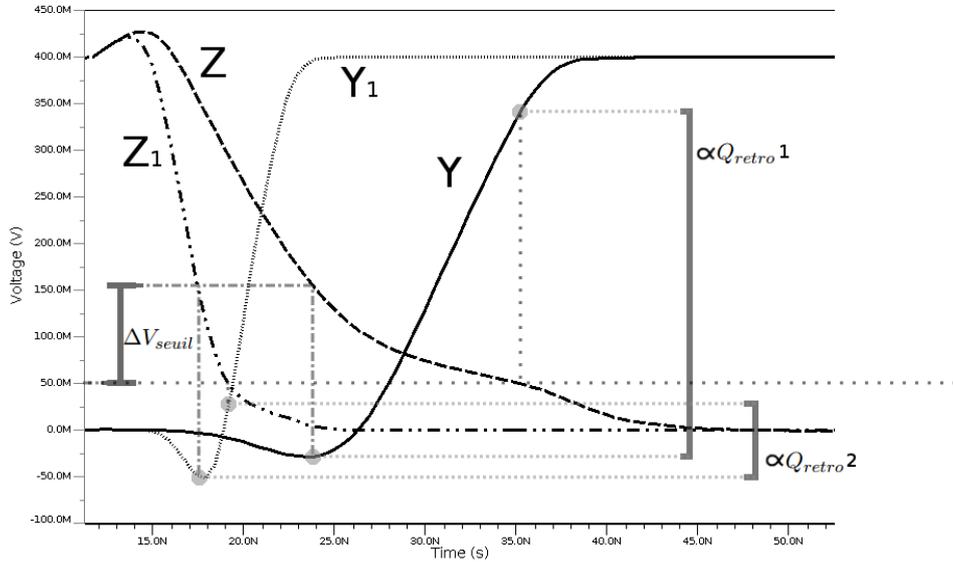


Figure 3.16 – Simulation Monte-Carlo de deux inverseurs avec en charge trois inverseurs sans mismatch. Nous n'avons gardé que deux itérations et nous avons superposé les signaux de sortie des inverseurs de charge (Y et Y_1). Nous pouvons voir que les deux charges rétro injectées Q_{retro} sont bien différentes d'un inverseur alors qu'il n'y a pas de mismatch sur les inverseurs en charge. Le seuil de fin de mesure à été descendu graphiquement de $0.08V$ à $0.05V$ pour amplifier visuellement le phénomène. De plus, nous avons sélectionné les extrêmes de la simulation Monte-Carlo c'est-à-dire le plus lent et le plus rapide.

Nous pouvons aussi annuler le mismatch sur la porte testée pour isoler l'effet du mismatch de la charge (figure 3.17). Nous voyons que la variation est bien plus grande dans le cas de la montée que dans le cas de la descente. En effet, si nous faisons la transition de 1 vers 0 en Z , l'inverseur en charge réagit en faisant la transition montante sur son nœud de sortie Y_j . Or un inverseur a le plus de variabilité de délai lors de cette transition montante.

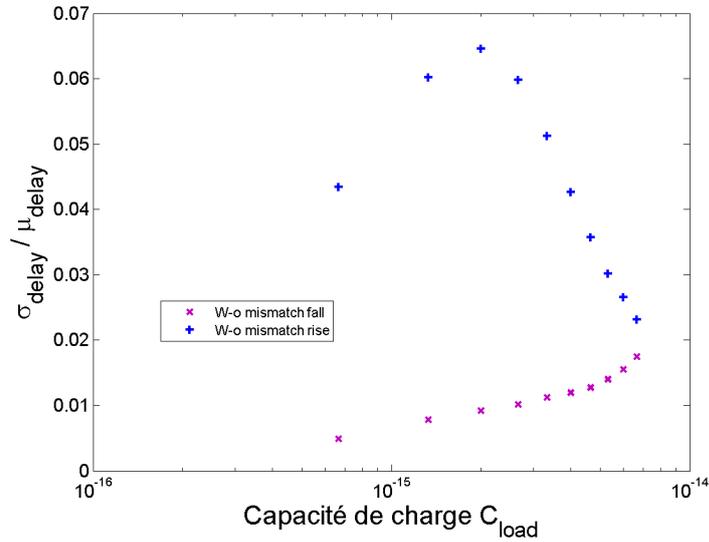


Figure 3.17 – *Effet isolé de Q_{retro} . Le mismatch a été retiré de l'inverseur testé. Il ne reste plus que le mismatch des inverseurs de charge. Simulations exécutées sur 10000 itérations à 0.4V 25°C. Nous remarquons que l'effet est plus marqué dans le cas où le signal Z fait la transition de 0 à 1. En effet, dans ces conditions, les charges s'inversent et chutent. Q_{retro} a donc une plus grande variance puisque les inverseurs en charge utilisent le **NMOS** pour s'inverser.*

La figure 3.18 résume les différents essais. Nous pouvons voir que si nous augmentons la capacité de couplage, nous augmentons la divergence vis-à-vis de la courbe des charges passive. Ce qui vient appuyer le raisonnement porté dans cette section.

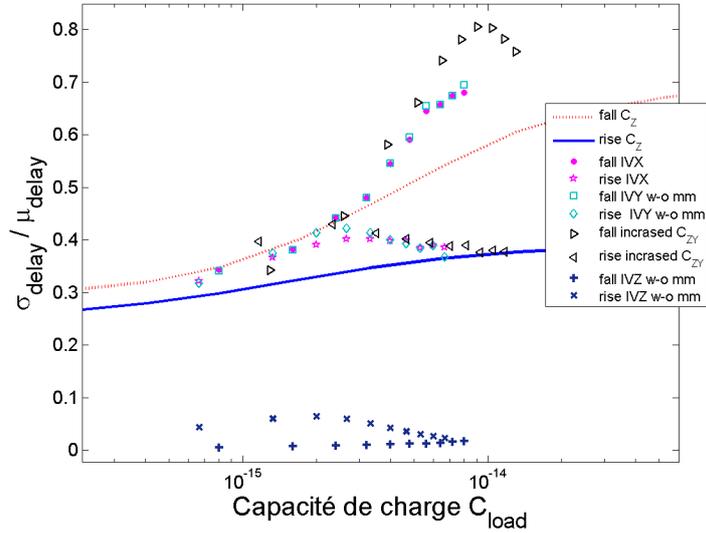


Figure 3.18 – Effet d’une charge active en faisant varier différents paramètres. Les cercles et les étoiles sont les simulations précédentes avec du mismatch partout sauf sur la source. Les carrés et les losanges sont des simulations où nous avons retiré le mismatch sur la charge. Les triangles sont obtenus en augmentant de 0.001pF la capacité de couplage. Enfin, les croix récapitulent la figure 3.17.

Notons que si une charge active influence notre inverseur, il est normal que notre inverseur influence à son tour l’inverseur qui lui fournit le signal d’entrée. Le phénomène de couplage capacitif influence donc indirectement le rapport $\frac{\sigma}{\mu}$ même dans le cas d’une charge passive. Et ce, en modifiant la forme du signal d’entrée en fonction de la vitesse d’inversion.

Tout au long de ce chapitre nous avons raisonné sur le temps de bascule quand les **NMOS** drivent la charge, t_{fall} . Le raisonnement reste valable dans le cas des **PMOS** (t_{rise}). Mais nous devons juste prendre en compte que comme les **PMOS** sont plus grands, leurs variances sont par nature plus faibles que celles des **NMOS** [17].

3.3 Influence de la taille et de la complexité d'une porte sur le rapport entre l'écart type et la moyenne.

Dans les sections précédentes, nous avons analysé en détail l'inverseur X2. Cette section a pour but de vérifier que des portes plus complexes subissent les mêmes effets.

Nous avons vu dans la section 1.1 que plus un transistor est petit plus il est sujet aux variations [17]. De plus, [24] montre qu'une suite de délais ayant une distribution *log-normal* reste un délai suivant une log-normale mais avec un $\frac{\sigma}{\mu}$ plus faible. Les deux hypothèses prises ensemble nous font penser que la porte logique avec le plus de variance est l'inverseur le plus petit. Nous allons comparer un inverseur (X2) avec un inverseur plus grand (X4) et une porte logique plus complexe, la porte XNOR.

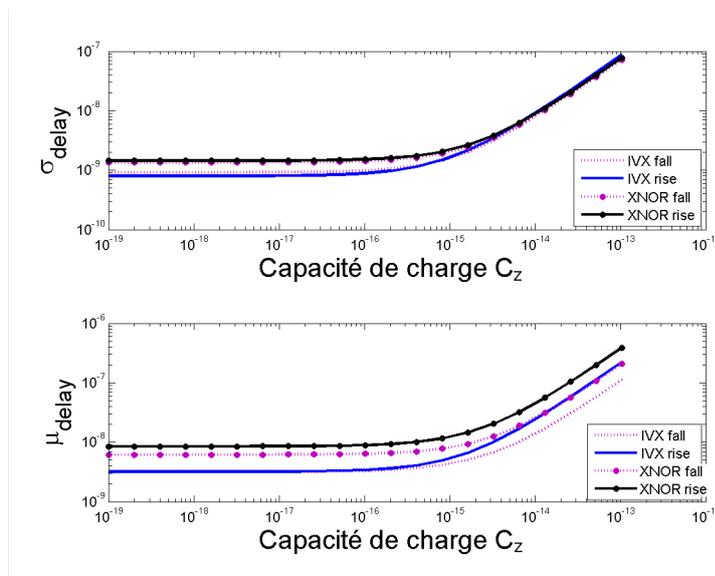


Figure 3.19 – σ_{delais} et μ_{delais} pour une porte XNOR X3 de la technologie 65nm LP. à 0.4V. 10000 itérations. Nous faisons varier l'entrée A du XNOR.

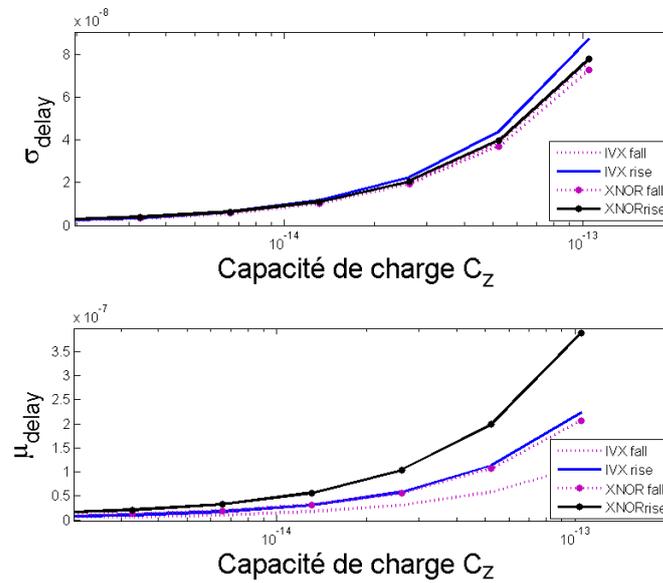


Figure 3.20 – Zoom sur les hautes capacités C_Z de σ_{delais} et μ_{delais} pour une porte XNOR X3 de la technologie 65nm LP. à 0.4V. 10000 itérations. Nous faisons varier l'entrée A du XNOR.

Nous pouvons voir sur les figures 3.19 et 3.20 la différence entre la moyenne et l'écart type d'une part de l'inverseur X2 et d'autre part de la porte XNOR X3. Nous remarquons que le temps moyen est toujours plus grand que dans le cas d'un inverseur X2 et que le σ équivalent à celui de l'inverseur simple est légèrement plus grand. Nous avons donc un $\frac{\sigma}{\mu}$ plus petit (figure 3.21).

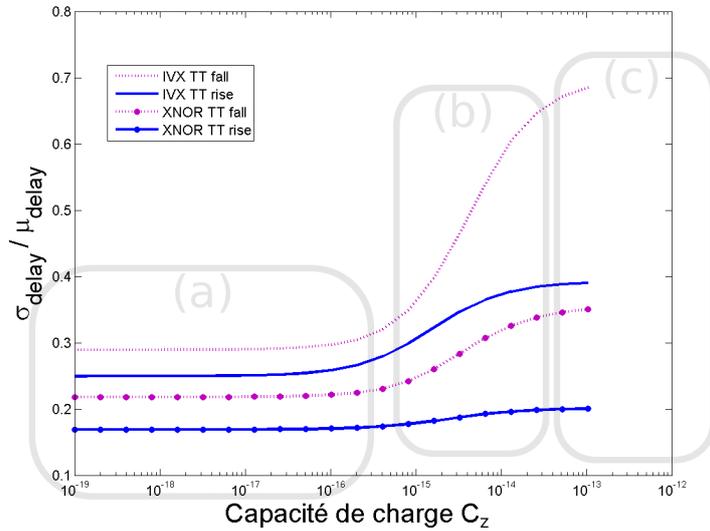


Figure 3.21 – Comparaison entre les $\frac{\sigma}{\mu}$ d'un inverseur X2 et d'un XNOR pour différentes charges C_Z . Technologie 65nm LP. à 0.4V, 10000 itérations.

Le plateau (a) commence à des valeurs de capacité plus hautes que dans le cas de l'inverseur. Ce qui est logique puisque la capacité de sortie d'une porte XNOR est plus grande que celle d'un inverseur pour des transistors de taille équivalente. De plus, le $\frac{\sigma}{\mu}$ de l'inverseur X2 est plus élevé que celui du XNOR.

Nous vérifierons encore une fois à l'annexe A.1 que nous gardons aussi une *log-normale* sur toute la plage des capacités testées.

Le deuxième phénomène à analyser est l'effet de la taille de l'inverseur. Une cellule plus grande moyenne les paramètres aléatoires sur une plus grande région [17]. Nous vérifions sur la figure 3.22 que c'est bien le cas :

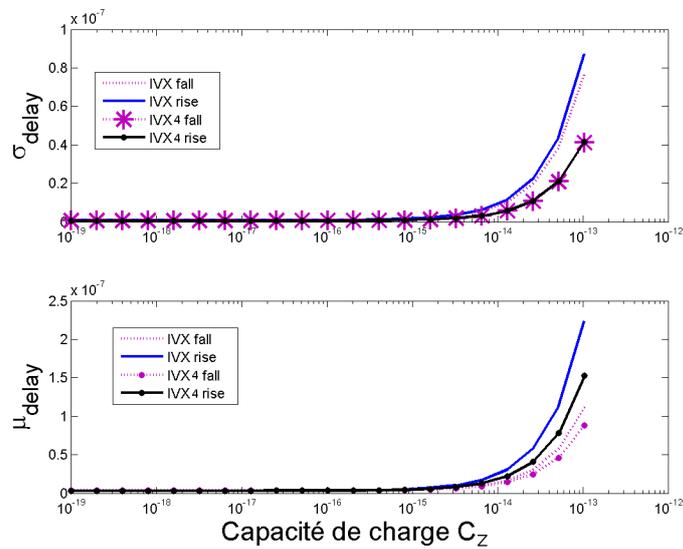


Figure 3.22 – Comparaison entre deux inverseurs de tailles différentes (un X2 et un X4). Technologie 65nm LP, à 0.4V 25°C. Remarquons que dans le cas de l'inverseur X4, $\sigma_{t_{fall}}$ se superpose avec $\sigma_{t_{rise}}$.

Nous remarquons sur la figure 3.22, que l'écart type est plus faible dans le cas de l'inverseur X4. De plus, nous ne constatons plus de différence entre l'écart type de t_{rise} et t_{fall} . Le temps moyen est aussi plus petit. Au final, nous notons (figure 3.23) que le rapport $\frac{\sigma}{\mu}$ est effectivement plus petit que dans le cas de l'inverseur X2. Les régions (a) (b) (c) (figure 3.4) sont encore distinctes mais sont plus proches les unes des autres. Comme la capacité de sortie du X4 est plus grande, la régions (a) s'étend sur une plus grande plage de capacité.

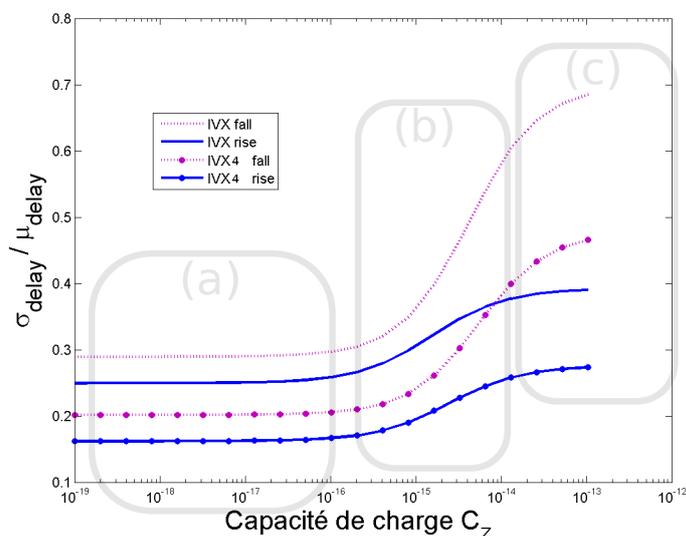


Figure 3.23 – Comparaison du rapport $\frac{\sigma}{\mu}$ des délais entre deux inverseurs de tailles différentes, un X2 et un X4. Technologie 65nm LP, à 0.4V.

Nous constatons que les $\frac{\sigma}{\mu}$ ne sont pas constants pour les trois portes logiques. Plus une porte est grande plus cet effet est atténué. La complexité influence aussi le rapport entre l'écart type et la moyenne. La porte XNOR X3 a un rapport plus petit que l'inverseur X4 alors que ses transistors sont plus petits.

Nous avons donc isolé les différents paramètres qui influent sur le $\frac{\sigma}{\mu}$. Nous avons aussi vu que ces phénomènes influencent aussi les autres portes que l'inverseur.

En pratique, nous ne connaissons pas de manière triviale la charge en sortie ni la vitesse de transition de la porte d'entrée. Nous nous contenterons d'un modèle simple qui considère le rapport $\frac{\sigma}{\mu}$ constant. La section suivante évalue la validité d'un tel modèle.

3.4 Validation d'un modèle linéaire pour définir l'écart type du délai.

Dans la pratique, nous ne disposons ni de la charge à driver ni de la vitesse d'inversion du signal d'entrée. Comme nous ne disposons que des outputs de **cadence**, nous ne disposons que des délais typiques des différentes portes logiques dans le circuit digital. Faute de mieux, nous allons vérifier la validité d'un modèle qui prend un $\frac{\sigma}{\mu}$ constant pour modéliser σ en fonction de μ . Ce modèle est applicable puisque en fixant le rapport $\frac{\sigma}{\mu}$, nous savons déterminer le μ (section 1.6) à partir de la valeur typique du délai fourni dans le sdf.

Pour avoir une référence, nous nous penchons sur la validité d'un modèle d'ordre 1 en plus de celui d'un $\frac{\sigma}{\mu}$ constant.

Reprenons le banc de test à la figure 3.24. Mais plutôt que de tracer en fonction de la charge, nous allons maintenant mettre en abscisse le délai moyen (figure 3.25).

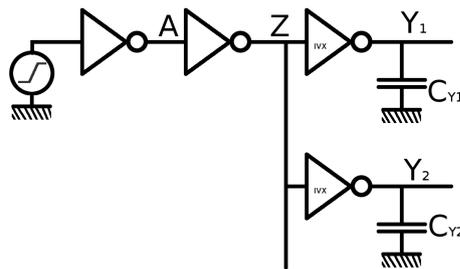


Figure 3.24 – Banc de test pour la simulation ELDO pour faire ressortir le délai en fonction d'une charge active. Le nombre d'inverseurs en charge sur le nœud Z peut varier.

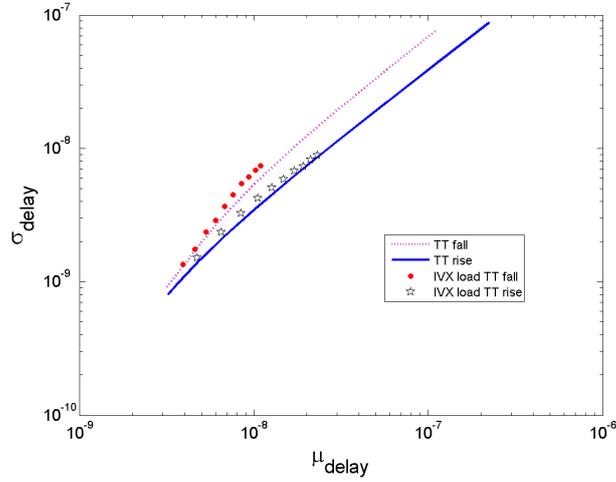


Figure 3.25 – σ en fonction de μ ($\sigma(\mu)$). Mêmes conditions que la figure 3.1 : 1000 itérations, un IVX2 UCLibs, 0.4V.

Nous nous rendons compte que σ semble presque lié linéairement avec μ . Nous espérons donc qu'un modèle linéaire est suffisant pour modéliser le $\sigma(\mu)$ (figure 3.26). Comme pour un circuit réel, le modèle est entièrement basé sur les mesures avec des inverseurs en charge.

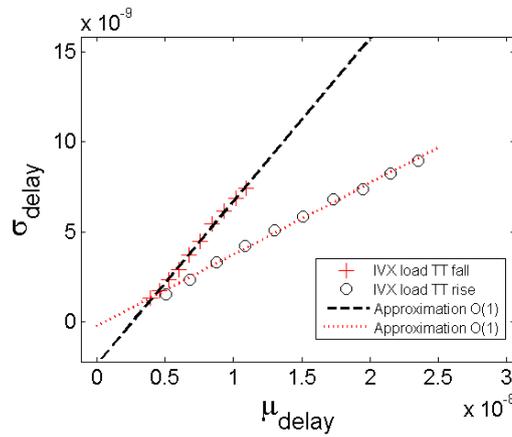


Figure 3.26 – Superposition du modèle Taylor d'ordre 1 et des mesures faites avec de 1 à 10 inverseurs en charge sur le noeud Z. la capacité de sortie de ces inverseurs est nulle ($C_Y = 0$). 1000 itérations, un IVX2 UCLibs, 0.4V.

Nous l'avons vu, le facteur $\frac{\sigma}{\mu}$ est aussi influencé par la charge et la vitesse de basculement de notre *test bench*. Nous allons donc faire varier trois paramètres : le nombre d'inverseurs que la source doit driver, le nombre

d'inverseurs que la porte testée doit driver et la capacité que chacun des inverseurs servant de charge doit driver (figure 3.27 et 3.28).

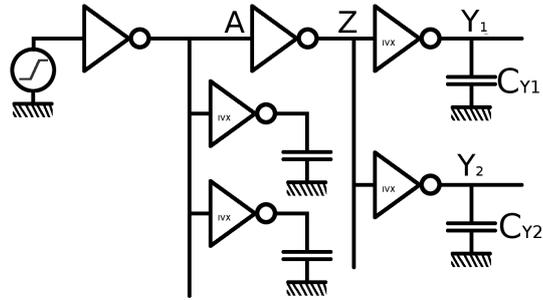


Figure 3.27 – Banc de test ajoutant l'effet de la source qui doit driver plusieurs inverseurs IVX en charge sur le noeud A. Le nombre d'IVX en charge au noeud A et en Z est variable ainsi que leurs capacités.

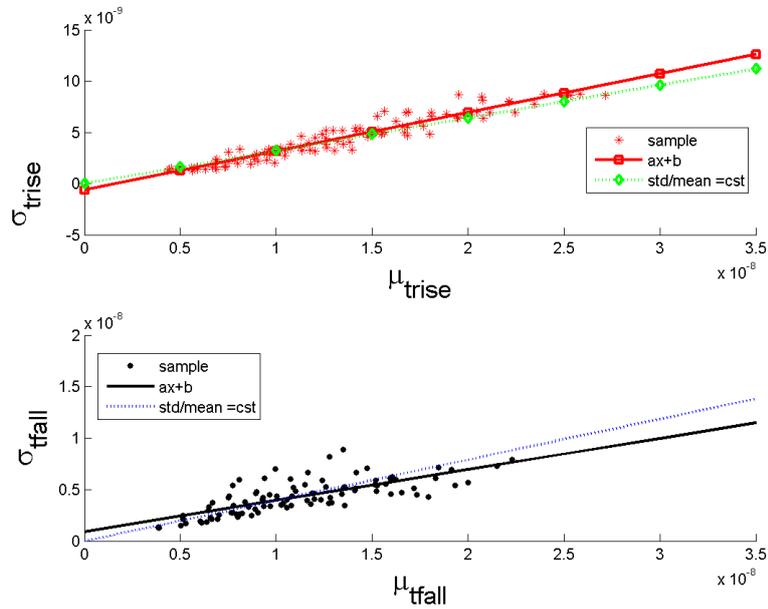


Figure 3.28 – Choucroute de points obtenue en faisant varier le nombre d'inverseurs à driver par la source (en A) et en charge (en Z). La source doit driver 1, 3, 5, 7 ou 9 inverseur(s). Pareil pour le nombre d'inverseurs en charge sur l'inverseur testé. Dans chaque cas, nous mettons quatre capacités différentes en sortie ($0,6.6e-4, 6.6e-3$ ou $6.6e-2$)[pF]. Pour chacune des capacités, nous faisons 1000 itérations Monte-Carlo. Nous ajoutons sur le graphe les deux modèles linéaires, Taylor d'ordre 1 et $\frac{\sigma}{\mu}$ constant.

Nous constatons que la tendance linéaire reste présente, surtout pour les transitions à plus faibles écarts-types. Un modèle linéaire garde donc tout son sens nous n'obtenons pas pour autant d'aussi bons résultats qu'en prenant en compte le délai de la porte précédente et la rétro-action Q_{retro} des portes suivantes.

Si nous allons plus loin et que nous prenons un modèle où le $\frac{\sigma}{\mu}$ est constant, nous observons à la figure 3.28 que le modèle reste proche de celui d'ordre 1.

Nous considérons donc un modèle avec le rapport entre l'écart type et la moyenne constant, crédible et fonctionnel bien qu'imparfait. C'est ce modèle que nous allons utiliser dans la suite pour nos simulations Monte-Carlo.

Chapitre 4

Mise en œuvre d'un nouveau flot de simulation statistique de l'énergie dynamique.

4.1 Flot proposé dans le cadre de ce mémoire.

Comme dit précédemment, les outils actuels s'assurent de la fonctionnalité du circuit. Le but ici est d'utiliser les outils existants pour en tirer des informations sur la variation d'activité et d'énergie dynamique que nous obtiendrions si nous fabriquions ce circuit digital.

Ce travail propose une approche où, comme en SSTA, toutes les portes logiques sont remplacées par des variables aléatoires à la distribution prédéfinie. Contrairement à la SSTA, nous faisons un Monte-Carlo respectant ces distributions sur l'ensemble du circuit digital.

Un résumé de ce flot est visible à la figure 4.1. Dans un premier temps, nous récupérons du flot normal la netlist qui contient les portes et leurs inter-connexions (a). Nous récupérons aussi le fichier **sdf** qui contient les différents timings des portes (a).

A partir de ce **sdf**, nous générons un fichier de configuration vierge **cfg** qui contient les différents templates pour les différents type de portes. Ces templates sont complétés par le modèle généré via la *cell Characterisation*.

Ensuite, à partir de ce nouveau fichier de configuration qui inclut un modèle de distribution pour les différentes portes, nous générons une série de **sdf** (b). Chacun de ces **sdf** contient désormais des timings aléatoires qui suivent le modèle prédéterminé. Dans ce chapitre, nous utilisons le $\frac{\sigma}{\mu}$ global vu dans le chapitre précédent.

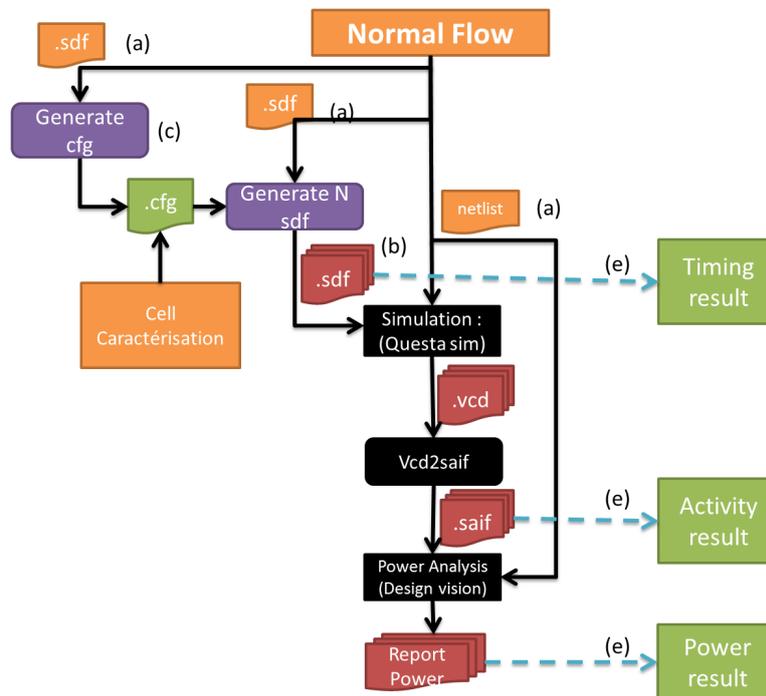


Figure 4.1 – Résumé du flot développé dans le cadre de ce mémoire : sur fond noir, les différents outils *cadence*. En mauve les outils développés, et en orange les sources nécessaires au fonctionnement du flot.

Nous appliquons à chacun de ces sdf la procédure vue au chapitre 2.2. Durant cette procédure, nous utilisons les fichiers intermédiaires pour générer des rapports sur le timing des portes, le facteur d’activité, la puissance consommée par chacun des circuits.

Puisqu’il est nécessaire de disposer d’une **netlist** et un **sdf**, nous pouvons nous insérer à deux moments dans le flot standard.

- Après la **synthèse**.
- Après le **”place an route”**.

Nous insérer après la synthèse sera moins lourd en calculs mais aussi moins précis que de se placer après le **”place an route”** (post layout). En effet, nous n’y retrouvons pas encore le *clock tree* ou les délais des interconnexions dûs au lignes R-C entre les portes.

4.2 Les outils développés :

Pour pouvoir réaliser le flot d’analyses, nous avons codé une série de scripts et d’outils pour parser et modifier les fichiers **sdf** et lire les rapports des différents outils *cadence*.

La suite est capable de :

Créer un fichier template cfg à partir d'un sdf. (c)

Le fichier **cfg** contient les modèles de variation pour les différentes portes standards ($\sigma = f(\mu)$). À partir d'un fichier **sdf**, nous générons le fichier **cfg** qui contient un template pour chacun des différents types de portes logiques se trouvant dans le fichier sdf. Nous pouvons aussi préciser un $\frac{\sigma}{\mu}$ global qui sera utilisé par défaut lors de la génération de sdf. Nous pouvons encore préciser un $\frac{\sigma}{\mu}$ différent pour les t_{rise} et t_{fall} .

Lire un fichier cfg et un sdf pour générer un ou plusieurs sdf. (b)

La deuxième fonction est de générer une série de fichiers **sdf** avec des timings variables. A partir d'un fichier **cfg**, nous pouvons varier les différentes distributions des portes logiques. Ces fichiers peuvent être générés à partir d'un seed. Pour un seed donné, les **sdf** générés seront identiques.

Récapituler les informations des différents outils cadence dans des fichiers. (e)

La dernière fonctionnalité de la suite logicielle est de rassembler et résumer les différents rapports générés par les différents outils. Ces butineurs d'informations ne fonctionnent qu'avec la suite cadence utilisée lors de la simulation. Ils lisent de multiples rapports de puissance, de multiples **sdf** ou encore de multiples **saif** et en font un résumé utilisable en matlab.

Avec ces outils et les outils traditionnels, nous pouvons faire une étude de la variabilité de l'énergie dynamique. La section suivante va d'une part vérifier que le générateur random est correct et d'autre part montrer l'analyse de l'effet du $\frac{\sigma}{\mu}$ sur l'énergie consommée pour la Sbox de [13].

4.3 Validation.

Dans cette section, nous allons vérifier notre flot à l'aide de la netlist d'une Sbox proposé par Kamel. Une Sbox (Substitution-box) est un simple circuit combinatoire servant dans un algorithme de chiffrement symétrique. Elle effectue une substitution non linéaire sur les 256 entrées possibles (8 bit).

Dans un premier temps, nous vérifions que notre générateur de nombres est correct en comparant ses résultats avec la théorie. Nous regardons aussi si utiliser la netlist post layout influence fortement ou non le $\sigma_{E_{dyn}}$ ou si se contenter d'une analyse de synthèse est suffisante. Ensuite, nous essayons

deux modèles pour la variance des portes. Un où toutes les transitions ont le même $\frac{\sigma}{\mu}$, l'autre où les transitions montante ont un $\frac{\sigma}{\mu}$ moins élevée que les descendantes.

Nous finirons par essayer le flot sur un circuit séquentiel.

4.3.1 Vérification du générateur de nombres aléatoires

Comme java ne possède pas de librairie standard pour faire des statistiques, nous avons codé un générateur de nombres aléatoires. Il se base sur la valeur typique trouvée dans le `sdf` ainsi que sur le rapport $\frac{\sigma}{\mu}$ suggéré par le modèle. Dans un premier temps, nous pouvons vérifier que le résultat obtenu est bien une *log-normale* à la figure 4.2.

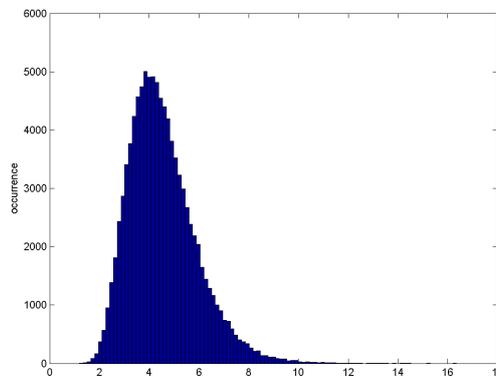


Figure 4.2 – Valeur typique choisie : 4, $\frac{\sigma}{\mu}$: 0.3. 100000 valeurs aléatoires prises avec le générateur du logiciel développé. Nous pouvons aussi voir que la valeur typique est bien 4.

Dans un second temps, nous comparons les valeurs obtenues pour différentes contraintes avec celles prédites par la théorie (table 4.1).

Typique imposé	1	4	10	4	4
$\frac{\sigma}{\mu_{impose}}$	0.75	0.75	0.75	0.3	1
moyenne théorique	1.9531	7.8125	19.5313	4.552	11.3137
moyenne	1.9563	7.7839	19.5592	4.5394	11.3385
$\frac{\sigma}{\mu}$	0.7545	0.7493	0.7513	0.2999	1.0027

Table 4.1 – La partie supérieure du tableau indique les conditions imposées sur le générateur. Pour chaque condition, nous avons calculé la valeur moyenne théorique. 100000 valeurs aléatoires par simulation.

Les valeurs moyennes théoriques sont calculées via la théorie vue en 1.6. Nous constatons que le générateur est imparfait mais pertinent. Nous

passons donc maintenant à une analyse du circuit global.

4.3.2 Simulation et comparaison avec des mesures d'un circuit combinatoire.

Pour valider notre flot, nous reprenons la netlist d'une Sbox utilisée par Dina pour sa thèse [13]. Elle utilise la même technologie que nous : 65nm LP, à 0.4V. Cependant son design est entièrement *full custom* alors que le nôtre est réalisé avec des outils automatiques.

Les statistiques de Kamel ont été faites sur vingt dies différents [13]. Sur chacun des dies, dix patterns sont testés (voir le CD joint). Il s'agit d'une suite de 256 transitions aléatoires dans laquelle tous les nombres composables en 8bit sont présents une et une seule fois [14]. Avec ces patterns, elle a obtenu une variabilité de 5,6% de l'énergie dynamique sur son circuit [13].

Dans les mêmes conditions (0.4V et 25°C), nous reprenons ces dix patterns, que nous exécutons à tour de rôle. Dans un premier temps, nous prenons un modèle où le $\frac{\sigma}{\mu}$ est constant et identique pour toutes les transitions des portes logiques et nous augmentons progressivement ce rapport pour se rapprocher des 5,6% sur l'écart-type relatif de l'énergie dynamique.

Nous pouvons voir à la figure 4.3 que l'écart-type relatif du coefficient d'activité moyen ($\frac{\sigma_\alpha}{\mu_\alpha}$) augmente bien avec l'écart-type relatif des délais des portes logiques. Le coefficient α est obtenu en faisant pour chaque itération Monte-Carlo la moyenne des activités des différentes portes logiques. Nous calculons ensuite sa variance et sa moyenne sur les différentes itérations. Avec 500 itérations, nous obtenons un point de la figure 4.3.

La figure 4.3 montre que plus nous avons de variabilité dans les délais plus nous avons de variabilité dans l'activité du circuit digital. Cela est visible en figure 4.4 où μ_α est presque constant (évolution <1%) alors que σ_α évolue fortement sur la fourchette des $\frac{\sigma}{\mu}$ testés.

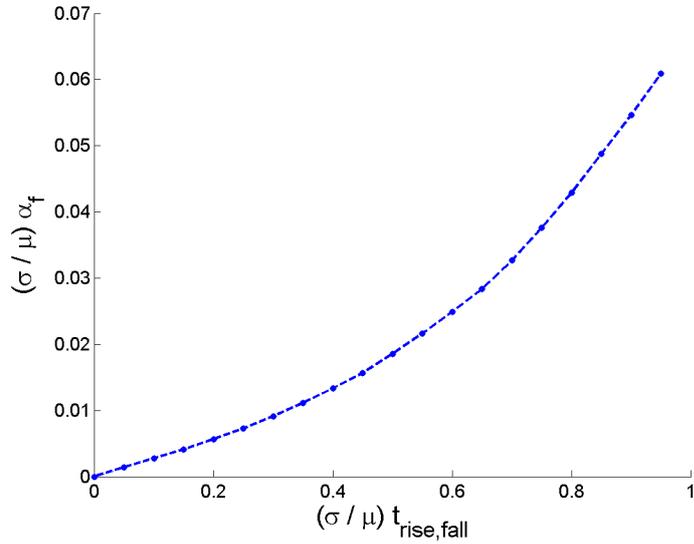


Figure 4.3 – Évolution du rapport $\frac{\sigma}{\mu} \alpha_f$ en fonction du $\frac{\sigma}{\mu}$ du délai des portes logiques. Simulations effectuées en post synthèse. 500 itérations Monte-Carlo avec une charge de 0.005 pF sur tous les outputs.

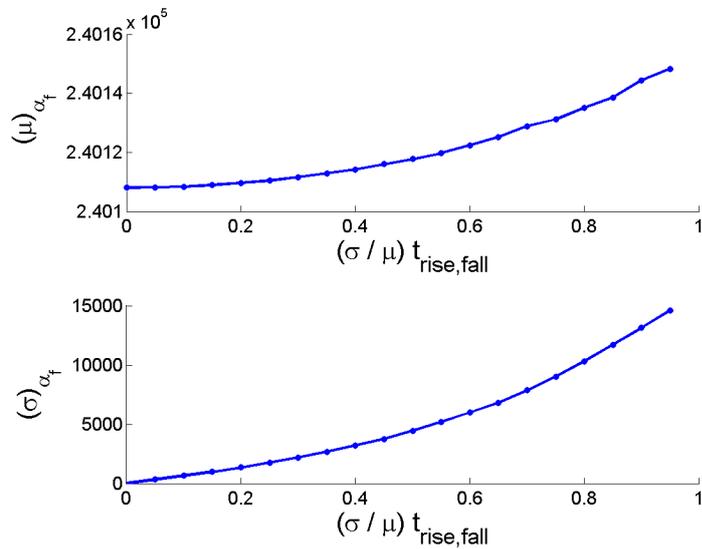


Figure 4.4 – Évolution du facteur d'activité moyen et de son écart-type en fonction du $\frac{\sigma}{\mu}$ choisi. Simulations effectuées en post synthèse. 500 itérations Monte-Carlo avec une charge de 0.005 pF sur tous les outputs.

Si la variance de l'activité augmente avec celle des délais, nous nous

attendons à ce que la variance de la puissance dynamique consommée augmente aussi (figure 4.5 et 4.6). Nous avons donc bien un flot de design dont l'énergie dynamique mesurée dépend de la variance des délais.

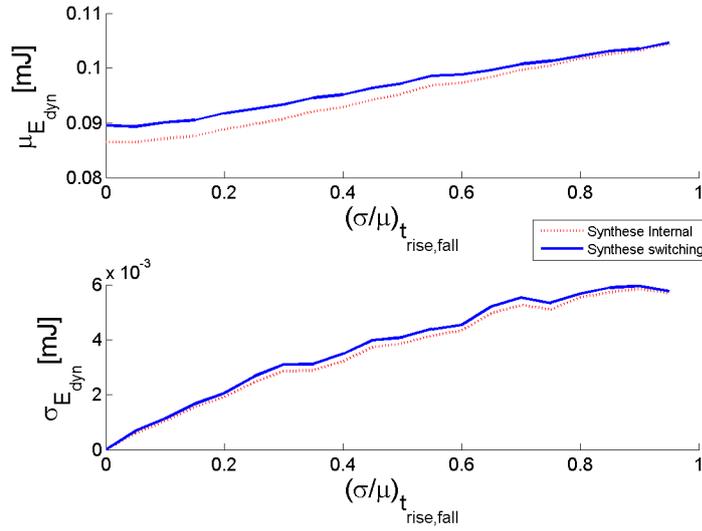


Figure 4.5 – Évolution de la moyenne et de l'écart-type de l'énergie dynamique en fonction du rapport $\frac{\sigma}{\mu}$ des délais des portes. Simulations effectuées en post synthèse avec un temps de 1200ns entre les variations à l'entrée. 500 itérations Monte-Carlo avec une charge de 0.005 pF sur tous les outputs. 10 patterns effectués.

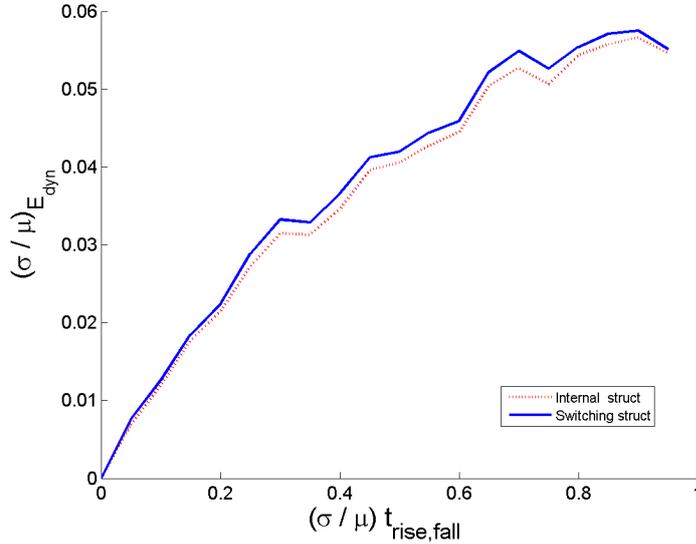


Figure 4.6 – Évolution du rapport $\frac{\sigma E_{dyn}}{\mu E_{dyn}}$ en fonction du rapport $\frac{\sigma}{\mu}$ des délais des portes. Simulations effectuées en post synthèse avec un temps de 1200ns entre les variations à l’entrée. 500 itérations Monte-Carlo avec une charge de 0.005 pF sur tous les outputs. 10 patterns effectués.

Nous remarquons que nous n’atteignons pas les 5,6% de Kamel [13]. Et ce même en sur évaluant largement le rapport, $(\frac{\sigma}{\mu})_{t_{rise,fall}}$. En effet, si on se base sur l’analyse du chapitre précédent 3, le $\frac{\sigma}{\mu}$ max dans le cas du plus petit inverseur est 0.75. Or si nous faisons une moyenne des $\frac{\sigma}{\mu}$ sur tout le design, nous ne pouvons qu’être plus petit que 0.75. Dans le cas de la sbox par exemple, il n’y a que 9 inverseurs X2 sur 259 cellules en tout. Nous sommes donc largement en dessous des $\frac{\sigma}{\mu} = 0.95$ des résultats de simulation.

La différence peut venir de plusieurs choses. Dans la simulation digitale, chaque transition consomme la totalité de l’énergie prévue alors que dans le cas d’un vrai circuit, les transitions ne sont pas forcément franches (ch 1.4). Ça peut aussi être dû au fait que Kamel a fait un design en *full custom* alors que nous l’avons fait en placement automatique. Mais le *full custom* entraînant des capacités plus faibles, devrait jouer en notre faveur. En effet, si les capacités sont plus faibles, dans le design de Kamel les $\frac{\sigma}{\mu}$ sont plus faibles. Ça peut finalement être dû aux variations de procédé de fabrication des dies qu’utilise Kamel. Si par manque de chance tous ses wafers sont en corner slow **PMOS** slow **NMOS** par exemple.

Nous montrons aux figures 4.7 et 4.8 que le coefficient d’activité et l’énergie suivent encore une *log-normale*. En utilisant les outils *matlab* développés par **Mike Sheppard** pour détecter la distribution fittant le mieux ces datas, nous pouvons voir que dans le cas de l’énergie, la distribution *normale* et

log-normale sont presque confondues (figure 4.8). Mais c'est la *log-normale* qui correspond le mieux à la distribution.

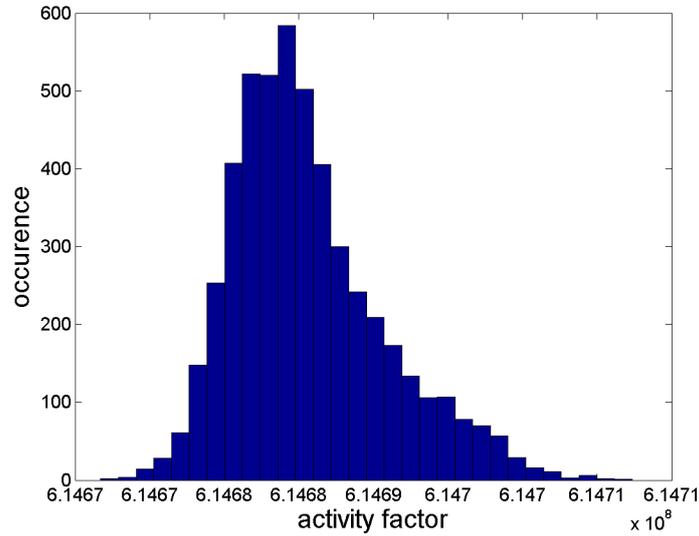


Figure 4.7 – Histogramme du facteur d'activité dans le cas où $\frac{\sigma}{\mu} = 0.3$. Simulations réalisées sur 4995 points (5*999) avec une charge de 0.005 pF sur tous les outputs du circuit combinatoire.

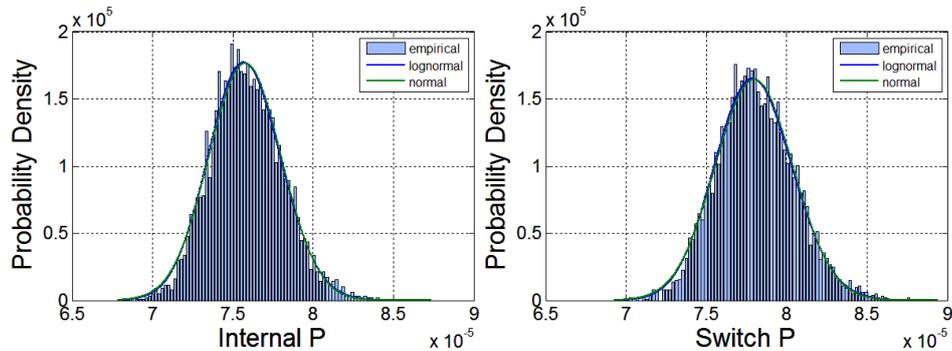


Figure 4.8 – Histogramme de l'énergie dynamique interne (gauche) et de switching (droite) pour un $\frac{\sigma}{\mu} = 0.3$. Simulations réalisées sur 4995 points avec une charge de 0.005 pF sur tous les outputs. Analyse réalisée avec l'outil "allfitdist" développé et distribué gratuitement par **Mike Sheppard**.

Dans les sections suivantes, nous voyons le gain de précision que nous obtenons en passant par la netlist *post layout* ou en utilisant un $\frac{\sigma}{\mu}$ différents pour t_{rise} ou t_{fall} . Le test utilisant des $\frac{\sigma}{\mu}$ différents pour chaque type de

porte n'a pas pu être mené à bien, l'outil "variety" de cadence supposé nous donner les modèles des différentes portes n'ayant pu être mis en œuvre.

$\frac{\sigma}{\mu}$ global, circuit purement combinatoire.

Faire les simulations *post layout* nous rapproche de la réalité. Il peut donc être intéressant de comparer les résultats obtenus avec la netlist *post synthèse* à ceux obtenus avec la netlist *post layout*.

Pour cette sous-section nous avons réalisé le *place and route* de la Sbox de Kamel (figure 4.9). Nous avons utilisé la technologie 65nm et forcé la densité maximum tout en gardant les cellules sur des rails séparés.

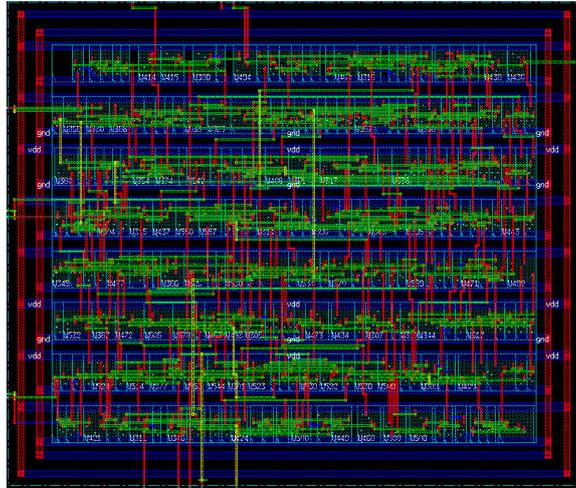


Figure 4.9 – Vue du layout de la Sbox. 65nm LP. Les rails de cellules sont séparés : la technologie n’acceptait pas les rails joints deux à deux.

Nous gardons les mêmes bornes que dans le cas *post synthèse*. c’est-à-dire une simulation avec un $\frac{\sigma}{\mu}$ entre 0 et 0.95. Nous constatons (figure 4.10) que le $\frac{\sigma_{\alpha}}{\mu_{\alpha}}$ est légèrement plus grand dans le cas *post layout* que dans le cas *post synthèse*. Remarquons qu’il y a moins de cellules dans le premier (*Layout*) avec 252 cellules contre 259 dans le cas *post synthèse*. Malgré ça, c’est bien le cas *post layout* qui a son écart-type qui augmente le plus avec le facteur $\frac{\sigma}{\mu}$ choisi (figure 4.11). La différence venant des capacités à charger plus élevées, à $\frac{\sigma}{\mu}$ identique, les μ de la netlist *post synthèse* seront plus petits que ceux de la netlist *post layout*. En conséquence, les σ seront plus élevés dans le cas *post layout*. Cette variance plus grande génère plus d’activité même si il y a moins de portes.

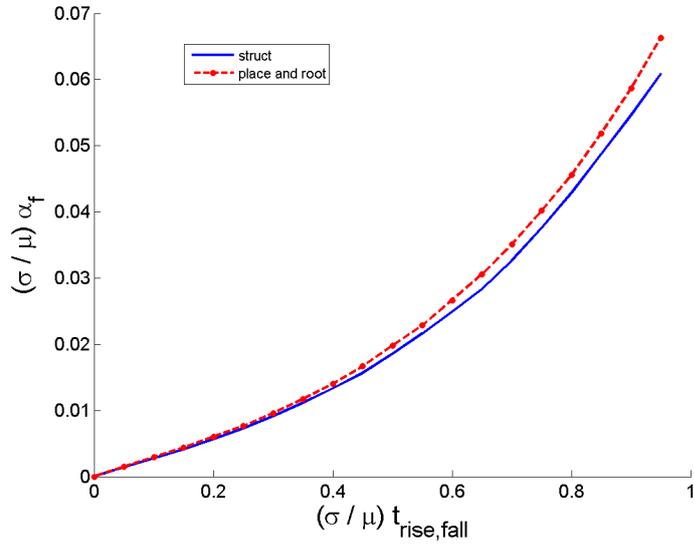


Figure 4.10 – Comparaison du facteur d'activité moyen entre la netlist en sortie post layout et de la netlist post synthèse. 500 itérations Monte-Carlo avec une contrainte de charge de 0.005 pF sur tous les outputs.

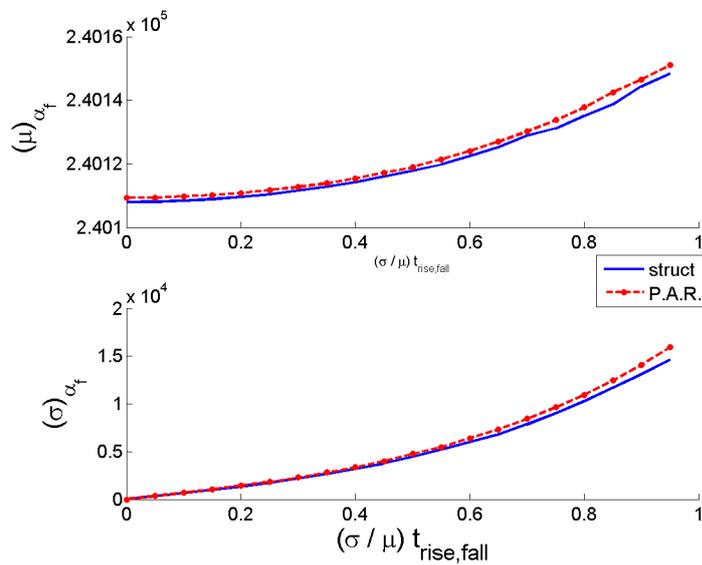


Figure 4.11 – Comparaison de la moyenne et l'écart type du facteur d'activité moyen entre la netlist en sortie post layout et de la netlist post synthèse. 500 itérations Monte-Carlo avec une contrainte de charge de 0.005 pF sur tous les outputs.

En comparant, nous voyons (figure 4.13) que les $\frac{\sigma_{E_{dyn}}}{\mu_{E_{dyn}}}$ évalués sur la netlist *post layout* ou sur la netlist *post synthèse* sont presque équivalents. Pour des faibles $\frac{\sigma}{\mu}$ le *place and route* donne une valeur de $\frac{\sigma_{E_{dyn}}}{\mu_{E_{dyn}}}$ légèrement plus grande.

Nous remarquons que l'énergie interne dynamique diminue alors que l'énergie de switching augmente (figure 4.12). Ce qui est cohérent : les capacités de charges sont plus grandes dans le cas *post layout* mais il y a moins de cellules.

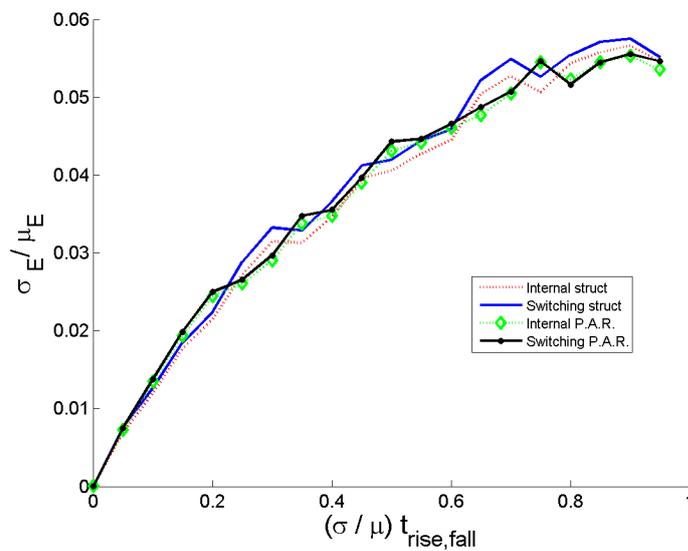


Figure 4.12 – Comparaison entre $\frac{\sigma_{E_{dyn}}}{\mu_{E_{dyn}}}$ en *post synthèse* (*struct*) et *post layout* (*place and route*, *P.A.R.*). Simulations réalisées avec 10 patterns, à 25°C et à 0.4V. Technologie UCLibs 65nm.

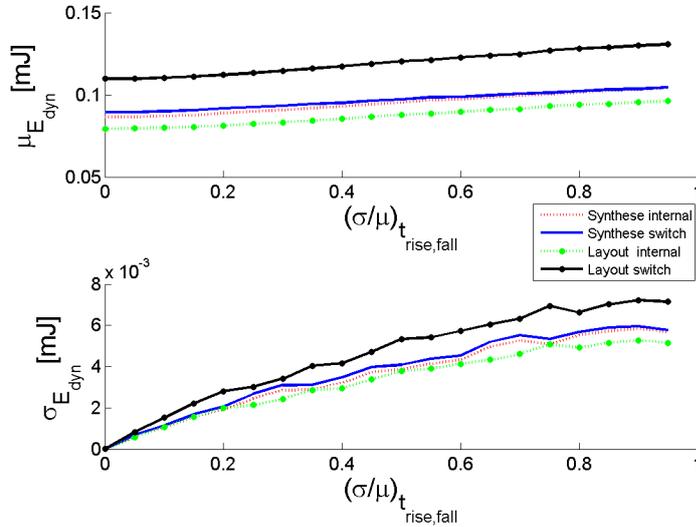


Figure 4.13 – Comparaison entre σ et μ en post synthèse (struct) et post layout (place and route, P.A.R.). Simulations réalisées avec 10 patterns, à 25° C et à 0.4 V. Technologie UCLibs 65nm.

En conclusion, faire des simulations *post layout* dans le cas d'un circuit combinatoire n'apporte pas grand chose. Les rapports $\frac{\sigma E_{dyn}}{\mu E_{dyn}}$ sont fort proches. Nous pouvons donc évaluer le rapport $\frac{\sigma E_{dyn}}{\mu E_{dyn}}$ en *post synthèse* et l'utiliser en *post layout*. Nous pouvons ainsi obtenir la variance du circuit *post layout* sans devoir faire une simulation Monte-Carlo.

$\frac{\sigma}{\mu}$ différents pour t_{rise} t_{fall} .

Nous avons vu dans le chapitre précédent que la loi de Pelgrom stipulait que la variation des délais de montée t_{rise} est plus faible que pour celle de descente t_{fall} . Dans cette section, nous allons voir ce que nous gagnons à prendre en compte ce phénomène du point de vue global.

Dans le cas *post synthèse* et *post layout*, nous avons choisi un rapport de 3/2 entre $(\frac{\sigma}{\mu})_{t_{fall}}$ et $(\frac{\sigma}{\mu})_{t_{rise}}$. Nous prenons cette valeur car le facteur varie sur toute la plage de 4/5 pour les plus petites capacités et de 1/2 pour les plus grandes capacités. Nous avons donc une valeur au milieu de la fourchette. C'est le rapport approximatif obtenu pour une charge de 4 inverseurs.

Dans un premier temps, nous gardons en abscisse le $\frac{\sigma}{\mu}$ de descente, nous voyons bien qu'utiliser un $(\frac{\sigma}{\mu})_{t_{rise}}$ fait diminuer les variations du facteur d'activité moyen (figure 4.14) avec au final, un rapport $\frac{\sigma_{\alpha}}{\mu_{\alpha}}$ plus petit (figure 4.15).

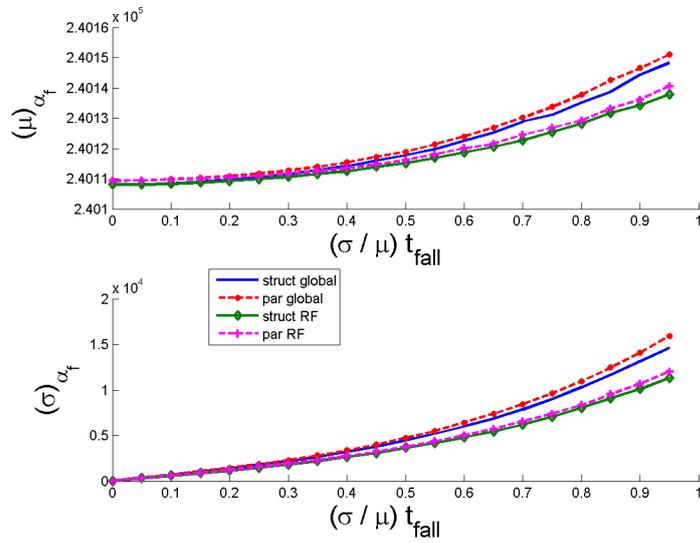


Figure 4.14 – Évolution de l'écart type et de la moyenne du facteur d'activité moyen en fonction du $\frac{\sigma}{\mu}$ du délai de descente t_{fall} . 500 itérations par point.

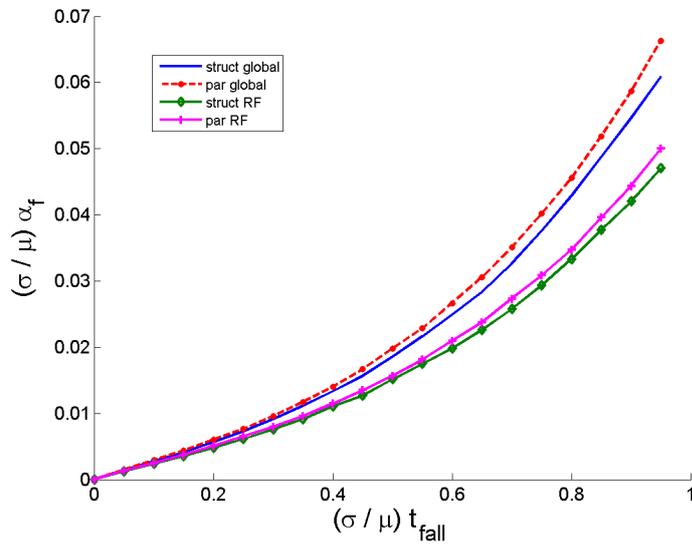


Figure 4.15 – Évolution du rapport $\frac{\sigma_{\alpha}}{\mu_{\alpha}}$ du facteur d'activité moyen en fonction du $\frac{\sigma}{\mu}$ du délai de descente t_{fall} . 500 itérations par point.

Dans les deux derniers graphes, nous comparons les σ_{α} et μ_{α} obtenus en fonction du $\frac{\sigma}{\mu}$ de la transition de descente $(\frac{\sigma}{\mu})t_{fall}$. Pour pouvoir comparer les

deux courbes, il faut ramener la courbe définie par le $(\frac{\sigma}{\mu})_{fall}$ à un $\frac{\sigma}{\mu}$ global.
 Nous pouvons faire cette transformation grâce à une simple moyenne.

$$\left(\frac{\sigma}{\mu}\right)_{rise,fall} = \frac{\left(\frac{\sigma}{\mu}\right)_{rise} + \left(\frac{\sigma}{\mu}\right)_{fall}}{2}$$

Nous pouvons tracer à nouveau les graphes mais cette fois en $\frac{\sigma}{\mu}$ global. Nous remarquons que à coefficient global donné l'activité moyenne augmente si nous avons des coefficients $\frac{\sigma}{\mu}_{t_{rise}}$ et $\frac{\sigma}{\mu}_{t_{fall}}$ différents (figure 4.16). Par contre pour ce qui est de l'énergie dynamique, les courbes se superposent (figure 4.17 et 4.18). Nous avons ajouté sur le graphe le cas où le $\sigma_{t_{rise}}$ est deux fois plus petit que le cas $\sigma_{t_{fall}}$ (RF 2).

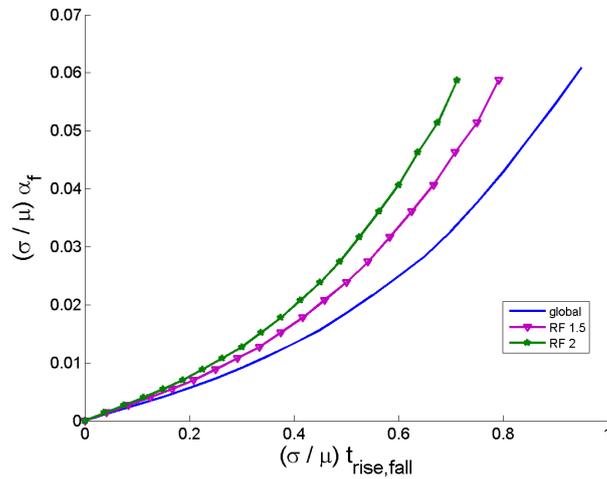


Figure 4.16 – Évolution du rapport $\frac{\sigma_{\alpha}}{\mu_{\alpha}}$ de l'énergie interne aux portes en fonction du $\frac{\sigma}{\mu}$ global du délai moyen pour trois rapports entre $\sigma_{t_{rise}}$ et $\sigma_{t_{fall}}$. 1, 1.5 et 2. Simulation faite en post synthèse avec 500 itérations par point.

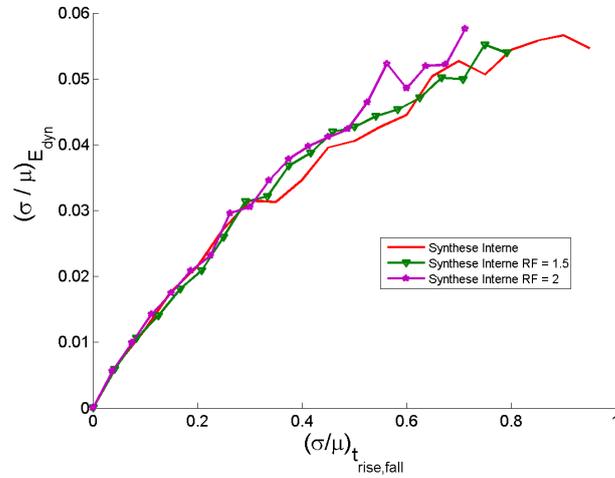


Figure 4.17 – Évolution du rapport $\frac{\sigma_{E_{dyn}}}{\mu_{E_{dyn}}}$ de l'énergie interne aux portes en fonction du $\frac{\sigma}{\mu}$ du délai moyen pour trois rapports entre $\sigma_{t_{rise}}$ et $\sigma_{t_{fall}}$. 1, 1.5 et 2. Simulation faite en post synthèse avec 500 itérations par point.

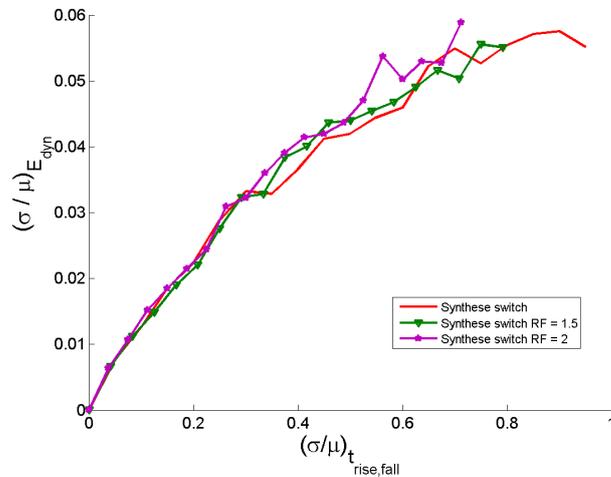


Figure 4.18 – Évolution du rapport $\frac{\sigma_{E_{dyn}}}{\mu_{E_{dyn}}}$ de l'énergie pour basculer les sorties des portes en fonction du $\frac{\sigma}{\mu}$ du délai moyen pour trois rapports entre $\sigma_{t_{rise}}$ et $\sigma_{t_{fall}}$. 1, 1.5 et 2. Simulation faite en post synthèse avec 500 itérations par point.

Nous en concluons que dans le cas d'un circuit purement combinatoire, pour simplifier l'analyse, nous pouvons travailler avec un $\frac{\sigma}{\mu}$ global identique pour toutes les portes à condition que celui-ci soit égal à la moyenne des $\frac{\sigma}{\mu}$ du design. Faire pareille hypothèse influencera le facteur d'activité moyen

mais donnera des variations d'énergie dynamique de même ordre d'amplitude. Nous terminons cette section par la figure 4.19 qui montre que ce raisonnement reste valable dans le cas *post Layout*.

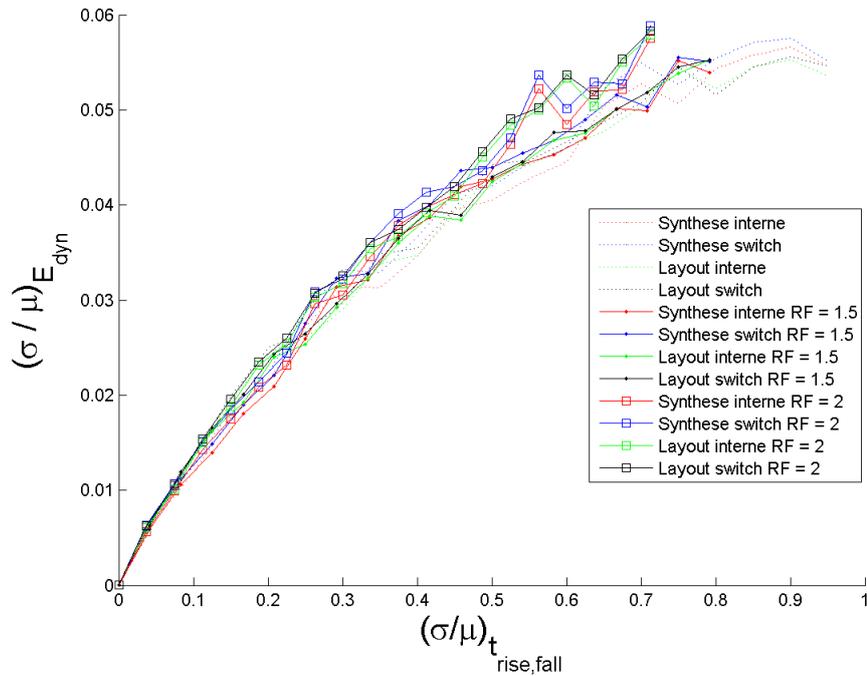


Figure 4.19 – Évolution du rapport $\frac{\sigma E_{dyn}}{\mu E_{dyn}}$ de l'énergie dynamique des portes en fonction du $\frac{\sigma}{\mu}$ global du délai moyen pour trois rapports entre $\sigma_{t_{rise}}$ et $\sigma_{t_{fall}}$. 1, 1.5 et 2. Simulation faite en post synthèse et en post layout avec 500 itérations par point.

4.3.3 circuit séquentiel.

Dans le cas d'un circuit séquentiel, utiliser un $\frac{\sigma}{\mu}$ global est plus délicat. En effet, l'outil a dimensionné le circuit pour des valeurs pessimistes ou dans le cas d'un outil SSTA, pour une distribution donnée. En mettant un $\frac{\sigma}{\mu}$ global, nous imposons pour certaines portes une variance plus grande que celle qu'elles auraient naturellement. Ceci entraîne des violations au niveau des registres. En effet, pour pouvoir enregistrer l'information dans le registre, il faut d'une part que, au moment où nous décidons d'enregistrer, l'information soit présente en entrée de la porte (T_{SETUP}), d'autre part, il faut que cette donnée reste stable suffisamment longtemps que pour être enregistrée (T_{HOLD}).

Si il suffit de ralentir l'horloge pour résoudre le premier problème, T_{HOLD} lui est uniquement dépendant du délai du chemin. Et lorsque nous augmentons trop la variance des portes, nous voyons apparaître des T_{HOLD} violations qui compromettent la simulation.

Dans cette section, nous simulons un Mips [6]. Pour ce qui est du $\frac{\sigma}{\mu}$ global, nous l'augmentons progressivement jusqu'à obtenir ces violations. Tant que le nombre de violations est en dessous de 10% des sample, nous les traitons comme des outlier et les supprimons du fichier de résultats.

Nous simulons ce Mips en utilisant un "dummy code" qui calcule des nombres premiers. L'horloge est réglée sur 1200ns pour être sûr de ne pas introduire de T_{SETUP} . Nous simulons pour une durée de fonctionnement de 15s.

Contrairement au cas d'un simple circuit combinatoire, nous voyons cette fois que le design présente peu de variabilité pour des faibles valeurs de $\frac{\sigma}{\mu}$. Nous supposons que c'est à cause du fait que le design est plus grand et par conséquent, la variabilité de tous les circuits se moyenne (figure 4.20 jusque 4.23). La variation est donc plus faible sur l'ensemble du circuit que dans le cas purement combinatoire et pour le coefficient d'activité moyen α et pour l'énergie dynamique.

Nous voyons qu'à partir d'un certain seuil, la variation de consommation explose et se met à croître. Ce seuil correspond à l'apparition des premiers outliers. Bien que ces derniers aient été retirés, mieux vaut ne pas accorder trop d'importance aux résultats obtenus au dessus de ce seuil.

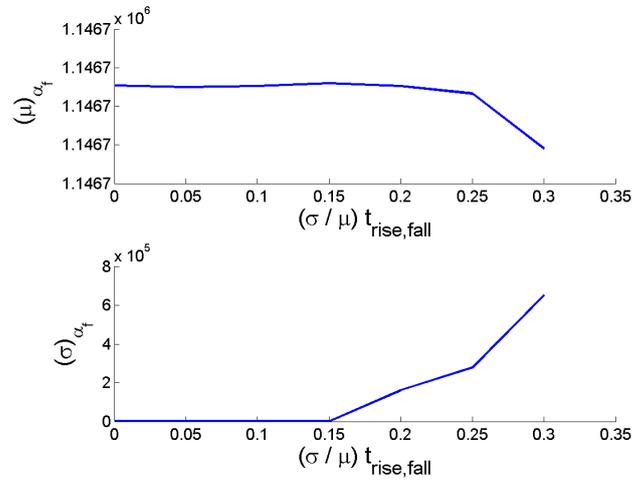


Figure 4.20 – Évolution de l'écart type et de la moyenne du facteur d'activité du MIPS dans le cas d'un $\frac{\sigma}{\mu}$ global. Il y a 500 itérations Monte-Carlo par point mais sont retirés les outliers qui surviennent lors des T_{HOLD} violations. Dans le pire cas, nous avons 45 outliers dans le cas d'un $\frac{\sigma}{\mu} = 0.3$.

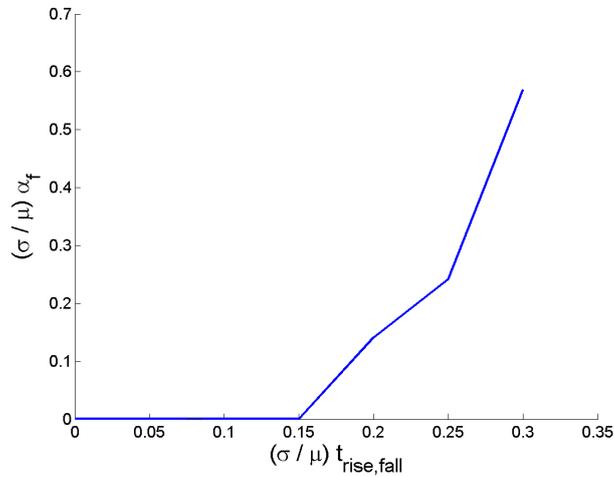


Figure 4.21 – Évolution du $\frac{\sigma}{\mu}$ du MIPS dans le cas d'un $\frac{\sigma}{\mu}$ global. Il y a 500 itérations Monte-Carlo par point mais sont retirés les outliers qui surviennent lors des T_{HOLD} violations. Dans le pire cas, nous avons 45 outliers dans le cas d'un $\frac{\sigma}{\mu} = 0.3$.

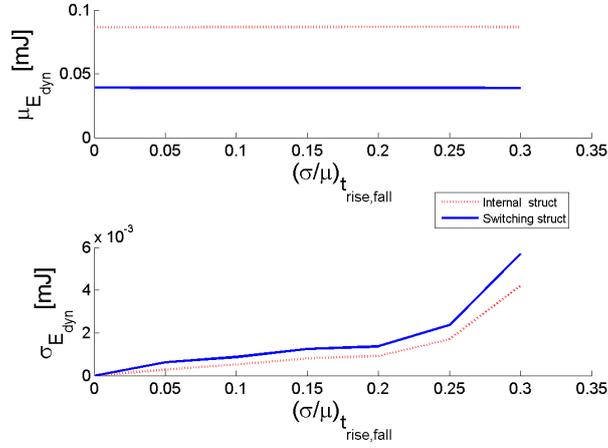


Figure 4.22 – Évolution de l'écart type et de la moyenne de l'énergie dynamique pour le MIPS dans le cas d'un $\frac{\sigma}{\mu}$ global. Il y a 500 itérations Monte-Carlo par point mais sont retirés les outliers qui surviennent lors des T_{HOLD} violations. Dans le pire cas nous avons 45 outliers dans le cas d'un $\frac{\sigma}{\mu} = 0.3$.

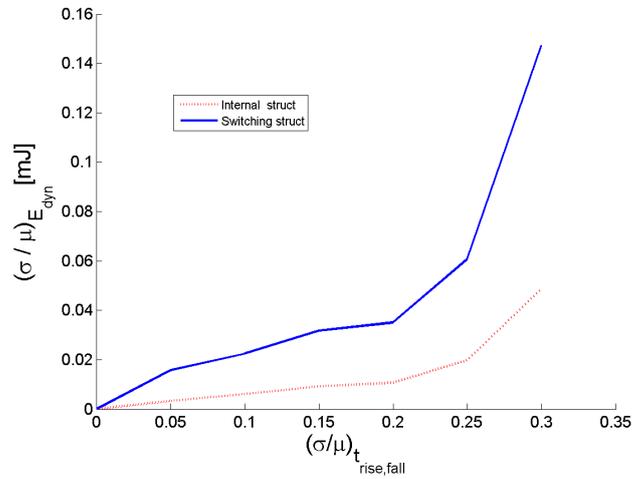


Figure 4.23 – Évolution du $\frac{\sigma_{E_{dyn}}}{\mu_{E_{dyn}}}$ pour le MIPS dans le cas d'un $\frac{\sigma}{\mu}$ global. Il y'a 500 itérations Monte-Carlo par point mais sont retirés les outliers qui surviennent lors des T_{HOLD} violations. Dans le pire cas nous avons 45 outliers dans le cas d'un $\frac{\sigma}{\mu} = 0.3$.

En conclusion, si dans le cas séquentiel nous voyons apparaître des variations d'énergie, elles sont plus faibles que dans le cas de la Sbox de par

la taille du circuit total bien plus grand. Nous constatons aussi que l'étude par un $\frac{\sigma}{\mu}$ global n'est pas adaptée car elle amène des problèmes de timing. L'étude à été réalisée au niveau synthèse où aucun effort n'a été fait par l'outil de compilation pour réduire les risques de T_{HOLD} . Une piste est de réessayer la même simulation dans le cas *post layout*.

Chapitre 5

Conclusion

Nous avons fourni un flot qui permet effectivement de faire varier le coefficient du facteur d'activité et donc l'énergie dynamique du circuit digital. Dans le cas d'un circuit combinatoire, ce flot peut être tout aussi bien évalué en *post synthèse* ou en *post layout*, les résultats obtenus sont du même ordre. De même, à $\frac{\sigma}{\mu}$ global identique, utiliser des $\frac{\sigma}{\mu}$ différents pour t_{rise} et t_{fall} n'influence pas le résultat.

Cependant, ce flot sous-évalue la variation d'énergie qu'auront les différents dies produits. De plus, si nous utilisons un $\frac{\sigma}{\mu}$ global, nous introduisons des T_{HOLD} violations rendant le design non fonctionnel.

Pour palier à ce deuxième problème, il faudrait étudier plus en détails le $\frac{\sigma}{\mu}$ pour chaque type de portes et en fonction de la charge que celle-ci drive. En effet, nous montrons dans ce document que le rapport $\frac{\sigma}{\mu}$ du délai est dépendant de la charge et du type de celle-ci.

Si la charge est active, un phénomène de couplage capacitif peut venir modifier les délais de la porte étudiée. De plus, au plus la charge en sortie est grande, au plus le temps de transition du signal d'entrée devient négligeable sur le temps total de basculement de la porte. À faible charge, c'est donc la vitesse de transition du signal d'entrée qui impose le $\frac{\sigma}{\mu}$ tandis qu'à forte charge, c'est le courant I_{on} .

Au final, se baser uniquement sur les timings ne semble pas la meilleure approche. Il faudrait utiliser les outils SSTA déjà présents pour générer nos **sdf** randomisés. Cette technique aurait deux avantages. Elle prendrait en compte les corrélations entre les différentes portes qui sont déjà implémentées dans le cadre de l'analyse par block des chemins critiques en SSTA. Elle ne générerait pas de T_{HOLD} violation, si le circuit est bien dimensionné, contrairement à l'utilisation d'un $\frac{\sigma}{\mu}$ global.

Annexe A

Pour appuyer le texte.

A.1 Vérification de la log-normalité des délais dans le cas de l'inverseur et de la porte XNOR.

Dans le chapitre 3, nous analysons un inverseur et une porte XNOR de la technologie 65nm LP, à 0.4 v. Nous leur faisons driver une charge de plus en plus grande. Cette annexe a pour but de vérifier que l'on garde bien une distribution *log-normale* tout le long de l'évolution de la charge de sortie C_Z .

Dans un premier temps, nous pouvons voir que c'est le cas pour l'inverseur aux figures : A.1 et A.2.

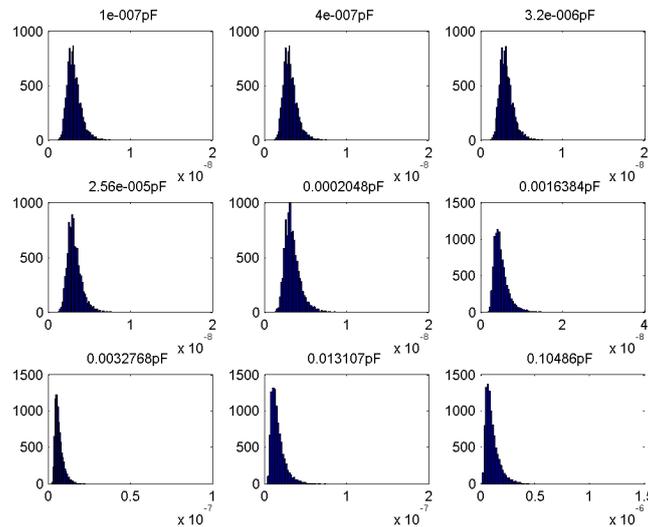


Figure A.1 – Histogrammes pour différentes charges, des délais t_{fall} . Réalisé avec un inverseur X2 65nm LP, 0.4V 25° C.

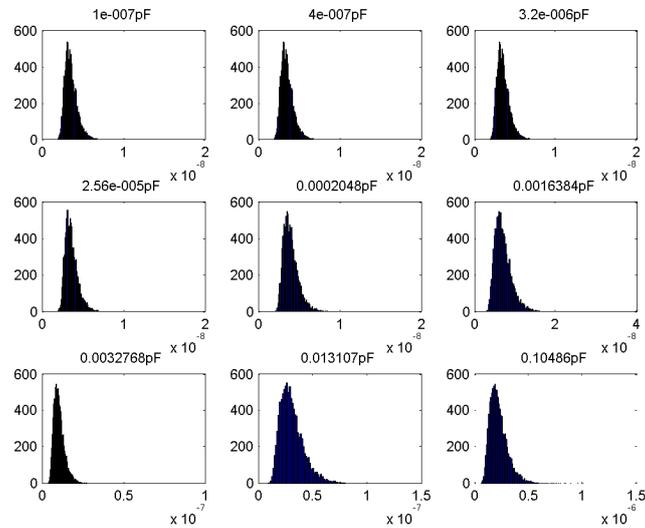


Figure A.2 – Histogrammes pour différentes charges, des délais t_{rise} . Réalisé avec un inverseur X2 65nm LP, 0.4V 25° C.

Nous remarquons que les distributions suivent bien une *log-normale*.

Dans un deuxième temps, nous pouvons voir que cela reste le cas pour le XNOR au figures : A.3 et A.4.

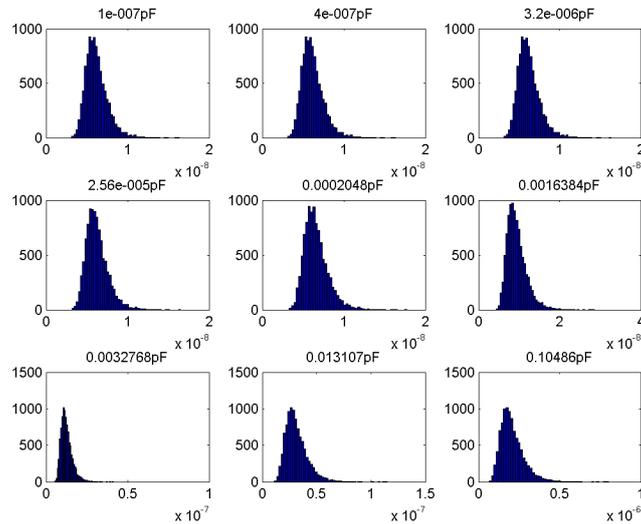


Figure A.3 – Histogrammes pour différentes charges, des délais t_{fall} . Réalisé avec une porte XNOR3 65nm LP, 0.4V 25° C.

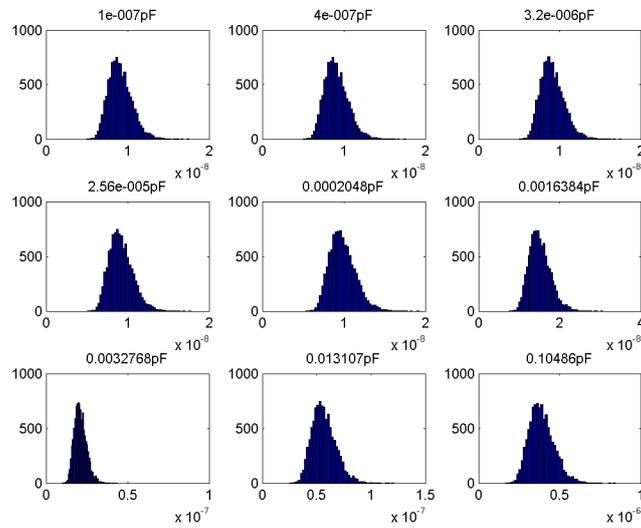


Figure A.4 – Histogrammes pour différentes charges, des délais t_{rise} . Réalisé avec une porte XNORX3 65nm LP, 0.4V 25° C.

Nous remarquons que les distributions se rapprochent plus d'une distribution *normale* que dans le cas des inverseurs mais elles restent *log-normales*.

L'hypothèse de distribution lognormale reste donc valide sur toute la plage des capacités de charge. Nous n'avons pas à nous en préoccuper lors de la partie simulation.

A.2 Lexique

Terme	Explication
V_{dd}	tension d'alimentation supérieure. (0.4 V dans ce document sauf explication contraire)
V_{ss}	tension d'alimentation inférieure.(0 V dans ce document sauf explication contraire)
S	<i>subthreshold swing</i> : $\ln(10)nU_{th}$
V_{th}	Threshold voltage.
U_{th}	Tension thermique : $\frac{kt}{q}$.
V_{gs}	Potentiel entre la grille et la source.
I_{on}	Courant de sortie d'un transistor allumé.
$t_{fall,rise}$	temps de bascule d'une porte logique en montée ou en descente.
t_{rise}	temps de bascule d'une porte logique en montée.
t_{fall}	temps de bascule d'une porte logique en descente.
t_{pro}	temps de propagation.
A	nœud d'entrée de la porte logique testée.
Z	nœud de sortie de la porte logique testée.
Y	nœud de sortie de la charge active.
C_{ZY}	capacité de couplage entre Z et Y.
C_{AZ}	capacité de couplage entre A et Y.
C_Z	Capacité en sortie de la porte logique testée
C_{Y_j}	Capacité en sortie de la charge active
Q_{inj}	charges injectées de A vers Z.
Q_{retro}	charges injectées de Y vers Z.
α_f	coefficient d'activité moyen.
$\alpha_{f,j}$	coefficient d'activité de la porte j.
σ	écart type des délais
μ	moyenne des délais
σ_α	écart type du facteur d'activité moyen
μ_α	moyenne du facteur d'activité moyen
$\sigma_{E_{dyn}}$	écart type de l'énergie dynamique
$\mu_{E_{dyn}}$	moyenne de l'énergie dynamique
$\frac{\sigma}{\mu}$	Écart type relatif des délais d'inversion d'une porte.
$\frac{\sigma_\alpha}{\mu_\alpha}$	Écart type relatif du facteur d'activité moyen.
$\frac{\sigma_{E_{dyn}}}{\mu_{E_{dyn}}}$	Écart type relatif de l'énergie dynamique.
SSTA	Statistical static timing analysis
sdf	fichier <i>Standard Delay Format</i> . voir annexe suivante.
saif	fichier <i>Switching Activity Interchange format</i> . voir annexe suivante.
vcd	fichier <i>Value change dump</i> . voir annexe suivante.

Annexe B

Les outils

B.1 Flot standard de développement.

Cette annexe a pour but de vous présenter un flux standard de conception d'un circuit digital. Nous reprendrons le flot vu tel que au cours **ELEC 2560** : Conception des circuits digitaux par David Bol (figure B.1).

Le flot que nous allons expliquer commence du concept et s'arrête à la première étape de fabrication. Il est divisé en trois parties, la partie conceptuelle qui regarde si le circuit fonctionne dans des conditions idéales, la partie structurelle, ou on transforme le circuit en une série de cellules standards et enfin la dernière partie, ou on relie physiquement ces cellules ensembles (figure B.1).

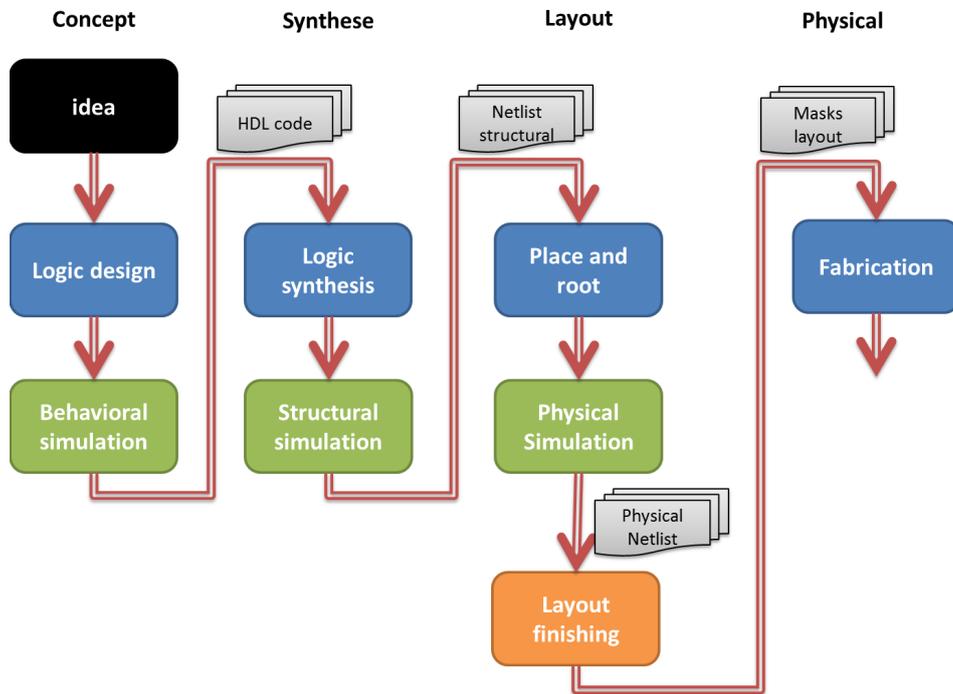


Figure B.1 – Flot de conception, de l'idée jusqu'à la fabrication du premier prototype. Le flot continue par des tests physiques (non représenté).

Nous partons d'une idée, que nous codons sous forme d'un langage précis tel que le VHDL ou le verilog. C'est le **logic design**. Les fichiers ainsi codés sont ensuite testés dans des conditions idéales.

Si les tests sont concluants, nous pouvons reprendre ce code pour ensuite faire une synthèse (**Logic Synthesis**) : nous le transformons en une série de portes logiques contenues dans la librairie utilisée. Désormais, chacune de ces portes a une latence, le délai qu'il faut entre le moment où l'entrée change et le moment où la sortie se stabilise à sa nouvelle valeur logique. Ce procédé nous produit d'une part une netlist, et d'autre part un fichier **sdf** (Annexe B.3). La netlist explicite comment les portes sont reliées entre elles, le fichier sdf, lui contient les timings des différentes portes.

Nous pouvons simuler cette netlist, et si elle reste fonctionnelle, fournir les deux fichiers à l'étape du **place and route**. À cette étape, nous venons placer les cellules sur le floor plans. Il en ressort une nouvelle netlist, plus complète que la précédente et des nouveaux délais dus aux inter-connexions nouvellement créées.

À nouveau, si la simulation passe, nous pouvons continuer le flot de conception en complétant les différents masques et en les envoyant dans le processus de fabrication.

B.2 Détails techniques des outils développés.

Les outils développés sont des outils sans interface graphique. Cette annexe a pour but d'en expliquer l'utilisation. Le but de ces outils est expliqué dans le chapitre 4.2.

randomSDF.jar

randomSDF est le logiciel principal développé, il permet de générer un fichier de configuration (**.cfg**) depuis un sdf, de générer un fichiers sdf suivant les modèles du fichier de configuration et lire une série de fichier sdf pour en résumer le contenu dans un seul fichier.

Les paramètres qu'il accepte sont les suivants :

<code>-help</code>		: display the help.
<code>(-i ou -input)</code>	<code><file></code>	lien vers le fichier d'entrée.
<code>(-o ou -output)</code>	<code><file></code>	lien vers le fichier d'entrée.
<code>(-c ou -config)</code>	<code><file></code>	finchier config.
<code>(-r ou -report)</code>	<code><path></code>	passse en mode analyse de sdf. Lit tous les sdf dans le répertoire et les résume dans le fichier <code>-output</code> .
<code>(-s ou -seed)</code>	<code><seed></code>	Pour générer selon un seed.
<code>(-x)</code>	<code><N></code>	Pour générer N sdf.
<code>(-n ou -gencfg)</code>		mettre pour générer un .cfg .
<code>(-l ou -log)</code>		Pour lognormal.
<code>(-sigma)</code>		pour créer le fichier <code>.conf</code> avec un sigma global ou imposer le sigma global.

`-input` permet de lire le fichier `<file>` comme source. ce fichier est soit le sdf de départ, soit lorsque `-report` est précisé, le début du nom des fichiers *sdf* à analyser.

Par exemple : `randomSDF.jar -i autoGen -r /dumypath/ -o report.power` va d'une part lire tous les fichiers commençant par *autoGen* se trouvant dans `/dumypath/` et écrira le résultat sous forme de tableau dans le fichier `report.power`.

`-output` le fichier dans lequel nous souhaitons écrire. Le programme écrase systématiquement le fichier précédent.

–**config** lien vers le fichier qui contient les modèles pour la génération des différents sdf.

–**report** passe le programme en mode analyse de sdf. et donne le path dans lequel sont les sdf à analyser. le programme va lister toutes les portes logiques et leur timing et résumer l'ensemble dans un fichier texte sous forme de tableau séparé par des espace.

–**seed** Toutes les opérations aléatoire ce font celons ce seed. A seed donné, les N fichiers générés seront toujours identiques au N autres fichiers (pour $j : 0 < j < N$ le fichier j sera différent du fichier j+1).

–**x** indique le nombre de fichiers que l'on veut générer. Le –output devient alors le début du nom de chaque fichier. Les fichiers générés auront le nom <outpu>j.delay ou $j : 0 < j < N+1$.

–**gencfg** permet de lire un sdf pour générer un template de fichier de configuration. Ce fichier template contiendra un template par cellule de type différent présente dans le sdf –input.

–**log** passe le générateur en mode *log-normal* par default, *normal*.

–**sigma** lors de la génération du *.cfg* si un sigma est précisé, le fichier *.cfg* contiendra ce fichier global.

Le fichier .cfg le fichier *.cfg* est un fichier respectant la syntaxe sdf [12]. Chaque valeur contenue dans ce fichier représente le $\frac{\sigma}{\mu}$ de ce paramètre. Cependant, un commentaire **# DEFAULT_SIGMA :<global> :<rise> :<fall>** dans le header permet de déterminer le sigma global, le sigma globale des t_{rise} et le sigma global des t_{fall} .

B.2.1 **analysesaif.jar, soc_pwr_ana.jar, powerAnalysis.jar**

Sont tous trois des fichiers d'analyse des outils de la suite cadence. Ils prennent tous un répertoire et un début de nom de fichier et résument tous les rapports dans un fichiers sous forme de tableau.

```
program.jar <path> <Name> <output>.
```

par exemple, *analysesaif.jar ./saif/ Autogen ../Result/report_000.rpt* vas venir analyser tous les fichiers commençant par *Autogen* dans le dossier *./saif/* et le résultat sera contenu dans le fichier *../Result/report_000.rpt*

analysesaif.jar analyse des fichiers saif et les résumes sous forme d'un tableau contenant l'activité de chaque cellule.

soc_pwr_ana.jar Analyse les rapports de puissance du logiciel *en-counter* de cadence. résume dans un tableau de quatres colonnes. puissance dynamique interne , de switching, puissance statique et total.

powerAnalysis.jar identique mais lit les rapport de *design_vision* de cadence.

B.3 Les différents formats et leurs utilités.

Dans cette section, Nous allons présenter les différents formats utilisés tout au long du texte.

sdf : Standard Delay Format

Le fichier sdf, est utilisé dans le cadre de ce mémoire pour indiquer les délais internes à la netlist analysée. Il respecte la syntaxe précisé par [12].

cfg : Configuration

Le fichier configuration respecte les spécifications des **sdf**. Sauf qu'ils n'indique pas des délais mais bien des $\frac{\sigma}{\mu}$ dans les différents paramètres. Il y a toutes fois une ligne qui ne respecte pas le format, il sagit de celle qui détermine le $\frac{\sigma}{\mu}$ global qui est juste un commentaire respectant une syntaxe particulière expliquée dans l'annexe précédente.

vcd : value change dump

Fichier format ASCII généré par Questa_Sim [9]. Il contient l'entièreté des signaux du design si nous lui demandons de tout enregistrer. Inutilisable en temps que tel pour la simulation synthèse. Par contre, convient dans le cas de la simulation post layout car l'outil *encounter* de cadence supporte le format. Ce fichier peut soit être transformé en saif, pour ne garder que le nombre de transitions de chaque porte soit transformé en **wlf** pour visionner les nœuds en fonction du temps.

saif : Switching Activity Interchange format

Version moins complète que le **vcd**, il contient le nombre de transitions faites par chaque porte durant la simulation. Il contient aussi le temps que chaque nœud à passé à une valeur particulière. il sert donc de base pour le calcul de l'énergie dynamique mais aussi statique consommée par le circuit digital.

Bibliographie

- [1] A. Agarwal, K. Chopra, D Blaauw, and V. Zolotov. Circuit optimization using statistical static timing analysis. In *Design Automation Conference, 2005. Proceedings. 42nd*, pages 321–324, 2005.
- [2] D Blaauw, K. Chopra, A. Srivastava, and L. Scheffer. Statistical timing analysis : From basic principles to state of the art. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 27(4) :589–607, 2008.
- [3] D. Bol, R. Ambroise, D. Flandre, and J. Legat. Interests and limitations of technology scaling for subthreshold logic. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 17(10) :1508–1519, 2009.
- [4] D. Bol, J. De Vos, C. Hocquet, F. Botman, F. Durvaux, S. Boyd, D. Flandre, and J. Legat. Sleepwalker : A 25-mhz 0.4-v sub- mm^2 – $\mu\text{W}/\text{MHz}$ microcontroller in 65-nm lp/gp cmos for low-carbon wireless sensor nodes. *Solid-State Circuits, IEEE Journal of*, 48(1) :20–32, 2013.
- [5] David Bol. *Pushing ultra-low-power digital circuits into the nanometer era*. PhD thesis, Université catholique de Louvain. Ecole Polytechnique de Louvain. Département d’Electricité, 2008.
- [6] David Bol. Lelec 2570 - synthèse des circuits intégrés digitaux (cid), 2013.
- [7] Inc Cadence Design Systems. Variety reference manual, product version 3.2p3, Release Date 8June2012.
- [8] B.H. Calhoun, A. Wang, and A. Chandrakasan. Modeling and sizing for minimum energy operation in subthreshold circuits. *Solid-State Circuits, IEEE Journal of*, 40(9) :1778–1786, 2005.
- [9] Mentor Graphics Corporation. Questa® sim user’s manual. including support for questa sv/afv software version 10.0d, 2011.
- [10] Anirudh Devgan and Chandramouli Kashyap. Block-based static timing analysis with uncertainty. In *Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design, ICCAD ’03*, pages 607–, Washington, DC, USA, 2003. IEEE Computer Society.

- [11] F. Frustaci, P. Corsonello, and S. Perri. Analytical delay model considering variability effects in subthreshold domain. *Circuits and Systems II : Express Briefs, IEEE Transactions on*, 59(3) :168–172, 2012.
- [12] Open Verilog International. Standard delay format specification version 3.0, May 1995.
- [13] D. Kamel, C. Hocquet, O.-X. Standaert, D. Flandre, and D. Bol. Glitch-induced within-die variations of dynamic energy in voltage-scaled nanocmos circuits. In *ESSCIRC, 2010 Proceedings of the*, pages 518–521, 2010.
- [14] Dina Kamel. *TRANSISTOR-LEVEL DESIGN OF LOW-POWER NANOSCALE DIGITAL CIRCUITS FOR SECURE APPLICATIONS*. PhD thesis, Ecole Polytechnique de Louvain.
- [15] C.H. Kim, H. Soeleman, and K. Roy. Ultra-low-power dlms adaptive filter for hearing aid applications. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 11(6) :1058–1067, 2003.
- [16] Samuel B. Luckenbill. Building bayesian networks with analog sub-threshold cmos circuits. *Department of Electrical Engineering : Yale University 15 Prospect Street, New Haven, CT 06511*, 2002.
- [17] M.J.M. Pelgrom, Aad C J Duinmaijer, and A.P.G. Welbers. Matching properties of mos transistors. *Solid-State Circuits, IEEE Journal of*, 24(5) :1433–1439, 1989.
- [18] Rajeev R. Rao, Anirudh Devgan, David Blaauw, and Dennis Sylvester. Parametric yield estimation considering leakage variability. In *Proceedings of the 41st annual Design Automation Conference, DAC '04*, pages 442–447, New York, NY, USA, 2004. ACM.
- [19] Adel S. Sedra and Kenneth C. Smith. *Microelectronic Circuits Revised Edition*. Oxford University Press, Inc., New York, NY, USA, 5th edition, 2007.
- [20] Ruijing Shen, Sheldon X.-D. Tan, and Jinjun Xiong. A linear algorithm for full-chip statistical leakage power analysis considering weak spatial correlation. In *Proceedings of the 47th Design Automation Conference, DAC '10*, pages 481–486, New York, NY, USA, 2010. ACM.
- [21] N. Verma, J. Kwong, and A.P. Chandrakasan. Nanometer mosfet variation in minimum energy subthreshold circuits. *Electron Devices, IEEE Transactions on*, 55(1) :163–174, 2008.
- [22] Wikipédia. Complementary metal oxide semi-conductor — wikipédia, l’encyclopédie libre, 2013. [En ligne ; Page disponible le 17-juillet-2013].
- [23] Wikipédia. Loi log-normale — wikipédia, l’encyclopédie libre, 2013. [En ligne ; Page disponible le 8-avril-2013].
- [24] Bo Zhai, Scott Hanson, David Blaauw, and Dennis Sylvester. Analysis and mitigation of variability in subthreshold design. In *Proceedings of*

the 2005 international symposium on Low power electronics and design,
ISLPED '05, pages 20–25, New York, NY, USA, 2005. ACM.