Low energy design of digital circuits

Mariem Slimani

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2013-ENST-0016

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Communication et Electronique »

présentée et soutenue publiquement par

Mariem SLIMANI

le 09 Avril 2013

Conception Basse Consommation de Circuits Numériques

Directeur de thèse : **Philippe Matherat**

Jury

M. Antoine DUPRÊT, HDR, Cea-leti	Rapporteur
M. Laurent FESQUET, HDR, TIMA	Rapporteur
M. Amara AMARA, Directeur de recherche, ISEP	Examinateur
M. Fernando SILVEIRA, Professeur, Universidad de la República	Examinateur
M. Habib MEHREZ, Professeur, LIP6, UPMC	Examinateur
M. Philippe MATHERAT, Chargé de recherche, CNRS, Télécom Paristech	Directeur de thèse

TELECOM ParisTech
école de l'Institut Mines-Télécom - membre de ParisTech
46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

T H È S E

À mes parents

Contents

Acknowledgments

Here, I would like to thank peoples without them this Ph.D project would be endless. First of all, I would like to thank Prof. Philippe Matherat for giving me the chance to do a PhD under his supervision. I would like to thank him too for his confidence in my choices and his continuous encouragement that helped me along the three years of my research work. I am very grateful to Prof. Fernando Silveira for his continuous guidance during the last two years. It was a pleasure to work with him and I learned a lot from his experience and his expertise. My gratitude also goes to Prof. Yves Mathieu for his help to master CAD tools. Next, I owe special thanks to Tarik Graba who took from his time to correct parts of this dissertation. Many thanks go to all the members of the Communications and Electronics Department at Telecom ParisTech, and to the administrative staff for their kindness and assistance. A thought goes also to all my friends in and outside Telecom ParisTech. My gratitude also goes to the professors that accepted to be part of my examination committee.

I want to express my deep gratitude to my parents for their endless love and support. They give me the motivation and the strength to advance in life. I owe the person I am to them. Special thanks to the love of my life, Montassar. His daily support and encouragement helps me overcoming the difficult moements during the three years of my research work. Finally, I would like to mention the new person in my life that keeps me smiling and sends me hope every day, my dear son Iyad.

Abstract

Over the last decade, power dissipation has been a significant concern as process technologies have scaled down to sub-nanometer. With the emergence of Ultra-Low-Power (ULP) applications such as wireless sensor nodes where primary concern is energy, techniques and methodologies for low energy design are more and more required.

This thesis focuses on different aspects of "Low Energy Design". First, reversible logic, as it is the first attempt for low energy computing, is briefly discussed. Then, we focus on dynamic energy saving in the combinational part of CMOS circuits. We propose a new method to reduce glitches based on dual threshold voltage technique. Simulation results report more than 16% average glitch reduction. We also show that combining dual-threshold to gate-sizing technique is very interesting for glitch filtering as it brings up to 27% energy savings. In the third part of this dissertation, we have been interested in sub-threshold operation where the minimum energy can be achieved using a reduced supply voltage. Sub-threshold operation has been an efficient solution for energy-constrained applications with low speed requirements. However, it is very sensitive to process variability which can impact the robustness and effective performance of the circuit. We propose a model valid in sub and near threshold regions in order to correctly estimate the circuit performance in a variability aware analysis. We provide an analytical solution for the optimum supply voltage that minimizes the total energy per operation while considering variability effects. Spice simulations matches the analytical result to within 6%.

Résumé détaillé de la Thèse

La dissipation d'énergie est devenue de nos jours l'une des contraintes majeures à prendre en compte lors de la conception des circuits électroniques et cela pour toutes les applications. Premièrement, les systèmes portables souffrent d'une autonomie restreinte et/ou de l'excès de poids des batteries. En fait, toute conception d'appareil mobile est un compromis entre les deux contraintes opposées que sont la grande autonomie et le faible poids. Malheureusement, l'amélioration de la technologie des batteries est relativement lente, ce qui rend essentielles les techniques visant la conception basse consommation. Deuxièmement, les circuits hautes performances, comme les microprocesseurs qui travaillent à des dizaines de GHz, émettent des quantités de chaleur proportionnelles à leur fréquence de fonctionnement. Et comme les fréquences de fonctionnement ne cessent d'augmenter dans la gamme des GHz, la dissipation est devenue un enjeu considérable dans la conception de ces circuits intégrés. En effet, au delà d'une certaine limite de température, la fiabilité du circuit pourra être sévèrement affectée. Pour évacuer cette chaleur, des systèmes de refroidissement peuvent être utilisés, mais ces systèmes sont trop encombrants et/ou trop chers. Pour les applications à Ultra basse consommation, l'énergie constitue même la première préoccupation bien avant le couple vitesse-densité. Ces systèmes sont aussi alimentés par des batteries, mais ces batteries ne peuvent pas être facilement chargées ni changées. Ils ont donc besoin d'une durée de vie très grande, qui peut atteindre plusieurs dizaines d'années. Par exemple, les stimulateurs cardiaques doivent avoir une telle durée de vie, afin d'éviter d'avoir à intervenir chirurgicalement pour le remplacer.

Ces multiples raisons ont motivé notre travail de recherche dont l'objectif est de proposer des nouveaux outils et méthodes qui facilitent le travail des concepteurs de circuits et systèmes visant la conception basse consommation.

Ce travail de thèse traite de différents aspects de la conception basse consommation des circuits électroniques numériques. Tout d'abord sont présentées les tentatives de calcul réversible, considérées comme essais de réalisation d'un calcul sans dissipation. Puis, je me suis intéressée aux dissipations des circuits CMOS puisque c'est la structure la plus couramment utilisée dans les circuits numériques. J'ai proposé deux approches pour réduire la consommation de ces circuits numériques. La première approche porte sur la réduction

de la dissipation due aux glitchs. J'ai proposé une nouvelle méthode qui consiste à adapter les tensions de seuil des transistors pour assurer un filtrage optimal de ces glitchs. Les résultats de simulation montrent que nous obtenons jusqu'à 16% de réduction des glitchs, ce qui représente une amélioration de 18% par rapport à l'état de l'art sur la base des circuits de référence ISCAS 85. La deuxième approche porte sur la réduction de la dissipation obtenue en faisant fonctionner les transistors MOS en régime d' inversion faible (sous-seuil). Les circuits fonctionnant dans ce régime représentent une solution idéale pour les applications ultra-basse-consommation. Par contre, l'une des préoccupations majeures est qu'ils sont plus sensibles aux dispersions des processus de fabrication, ce qui peut entraîner des problèmes de fiabilité. Je propose un modèle compact qui détermine le point d'énergie minimum de façon analytique, donc sans recourir à une simulation type SPICE, tout en étant suffisamment précis et robuste vis-à-vis de la variabilité (due à la dispersion). L'écart de résultat entre le modèle compact et un modèle SPICE complet est de 6%.

Ce synthèse détaillé est organisé comme suit. Tout d'abord, les sources de dissipation dans les circuits numériques sont présentées. Puis, je présente les deux principales contributions de ce travail de thèse à savoir la réduction des glitchs et la modélisation des circuits qui fonctionnent en régime d'inversion faible (sous-seuil).

Sources de dissipation dans les circuits CMOS

L'énergie totale consommé par un circuit est la somme d'une énergie dynamique et d'une énergie statique. L'énergie statique est due aux courants de fuites qui circulent quand les transistors sont fermés car ces derniers sont loin d'être des interrupteurs idéales. L'énergie dynamique inclut l'énergie de court circuit due aux courants qui flottent entre les rails d'alimentation et l'énergie capacitive due aux charges et aux décharges des capacités parasites durant les transitions logiques.

Dans la littérature l'énergie dynamique désigne généralement l'énergie capacitive. Ceci est dû principalement à la faible contribution des courants de court circuit qui sont généralement inférieur à 7% de la consommation dynamique totale. Par ailleurs, pour des raisons de simplification, dans ce travail, l'énergie totale consommé par un circuit sera :

$$E_{Totale} = E_{capacitive} + E_{statique} \tag{1}$$

Dissipation capacitive

Pour illustrer la dissipation capacitive, on va considérer la structure générale d'une porte CMOS montrée dans la Figure 1.

Imaginons que la combinaison d'entrée est de sorte que la sortie fait une transition de

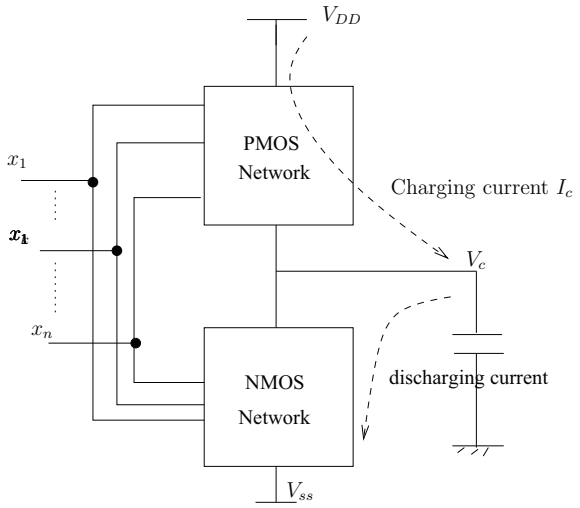


Figure 1: Structure générale d'une porte CMOS.

l'état logique 0 à l'état logique 1. Dans ce cas, le courant qui charge la capacité de sortie C_L peut s'écrire comme suit :

$$I_c(t) = C_L \frac{dV_c}{dt} \tag{2}$$

L'énergie donnée par la tension d'alimentation est donc :

$$E_{Tot} = \int_0^{t_1} V_{DD} I_c(t) dt \tag{3}$$

Où t1 est le temps nécessaire pour charger complètement la capacité de sortie initialement déchargé. En substituant l'équation 2 dans l'équation 3, l'énergie consommé suite à la transition de l'état logique 0 à l'état logique 1 est comme suit :

$$E_{Tot} = V_{DD} C_L \int_0^{t_1} \frac{dV_c}{dt} dt = V_{DD} C_L \int_0^{V_{DD}} dV = C_L V_{DD}^2 \tag{4}$$

La moitié de cette énergie est stockée dans la capacité C_L et l'autre moitié est dissipée sous forme d'effet joule dans les transistors PMOS.

Pendant la phase de déchargement, la tension dans la capacité de sortie C_L passe de V_{DD} à 0 et l'énergie stockée dans la capacité est dissipée dans les transistors NMOS. Pour conclure chaque transition logique dissipe $1/2 C_L V_{DD}^2$.

Donc si pour un calcul donné (multiplication, addition, FFT), on connaît le nombre de transitions nécessaires pour effectuer ce calcul « a », l'énergie capacitive consommée par ce calcul sera :

$$E_{capacitive} = \frac{1}{2}\, a C_L\, V_{DD}^2 \tag{5}$$

Dissipation statique

La consommation statique est due aux courants de fuites qui flottent dans les conditions statiques quand tous les nœuds sont à un état stable. Les différentes sources de fuite pour un transistor NMOS sont illustrées dans la Figure 2.

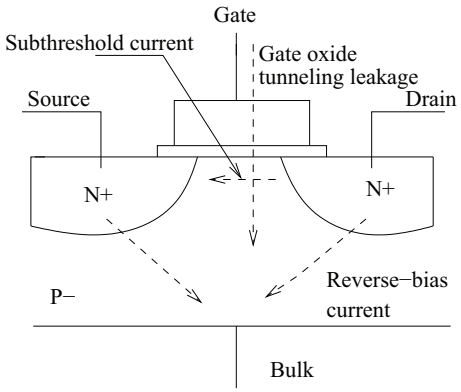


Figure 2: Différentes sources de fuite pour un transistor MOS.

Il y a principalement les courants de fuite sous-seuil qui circulent entre le drain et la source quand le transistor a une tension de grille inférieure à la tension de seuil du transistor. La puissance statique est comme suit :

$$P_{fuite} = V_{DD}\ I_{fuite} \tag{6}$$

Et si on aura besoin d'une durée de calcul T_d pour effectuer une opération, l'énergie de fuite s'écrit simplement :

$$E_{fuite} = P_{fuite}\ T_d \tag{7}$$

Discussion

Avec la miniaturisation des technologies, les courants de fuites augmentent considérablement. Selon les prévisions ITRS, jusqu'à 45nm, la consommation dynamique est plutôt dominante. Tandis qu'à partir du 32nm, la composante statique est devenue une partie non négligeable. Mais, les courants de fuites dépendent des paramètres technologiques comme l'épaisseur de l'oxyde. Donc, c'est au travail de technologue pour trouver une solution. D'ailleurs, il y a beaucoup de travaux de recherches qui ont été proposées récemment pour faire face aux problèmes liés aux courants de fuites.

Réduction des glitchs

Comme on a déjà mentionné, chaque transition logique dans le circuit consomme $1/2C_LV_{DD}^2$. Normalement, chaque nœud ne doit changer qu'une seule fois chaque période d'horloge. Mais, malheureusement, des transitions multiples peuvent se produire due à des états de calcul intermédiaires. Ils sont appelées glitchs et se génèrent à la sortie d'une porte lorsque ses entrées n'arrivent pas au même temps. Des études ont prouvé que la contribution des glitchs atteint 40% en moyenne de la consommation totale d'un circuit. Cette grande quantité d'énergie consommée par des transitions qui ne contribuent même pas au fonctionnement du circuit a motivé les travaux de recherche liée à la réduction des glitchs. Beaucoup de méthodes ont été proposé pour ce faire. Ces méthodes essaient de balancer la différence d'arrivée de temps pour garantir que les signaux d'entrée des portes logiques arrivent au même temps.

Les techniques proposées pour réduire les glitchs appartiennent à deux approches. La première (path-balancing) consiste à ajouter du délai aux signaux d'entrée qui arrivent tôt pour balancer le temps d'arrivée. La deuxième consiste à filtrer le glitch en augmentant le délai inertiel de la porte (Glitch filtering). La Figure 3 illustre ces deux techniques.

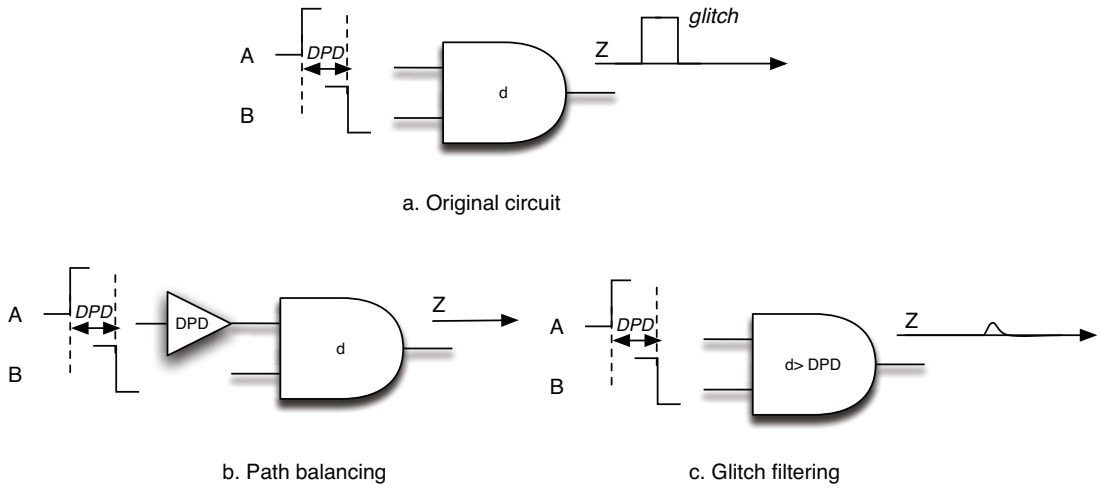


Figure 3: Différentes sources de fuite pour un transistor MOS.

Bien que ces méthodes permettent une réduction intéressante des transitions parasites, leur application dépend énormément des contraintes du circuit. Par exemple, le sous-dimensionnement des portes peut augmenter la durée de calcul ce qui limite son utilisation quand des contraintes de vitesse sont imposées. De même, insérer des extras portes pour balancer la différence des temps d'arrivée peut engendrer des problèmes de superficie. Donc, clairement, on peut dire qu'on a encore besoin d'autres techniques pour la réduction des glitchs afin de parvenir à une conception optimale.

Dans ce travail, je propose une nouvelle technique pour la minimisation des glitchs en jouant sur la tension de seuil. L'idée est d'utiliser la tension de seuil la plus élevé pour les portes susceptibles d'avoir un glitch afin d'augmenter leur délai inertiel. Un algorithme heuristique a été proposé pour résoudre ce problème. Les simulations effectuées sur des circuits tests de la famille ISCAS85 prouve une complémentarité entre la méthode proposée et la méthode du sous-dimensionnement des portes pour la réduction des glitchs.

Bien que plusieurs travaux se concentrent sur la réduction des glitchs, les outils de CAO pour l'analyse et l'optimisation de ces transitions parasites sont très limités. Ainsi, ce travail propose également une méthodologie pour détecter les glitchs et tester les techniques d'optimisation en utilisant les outils de Cadence. Cependant, les différentes étapes peuvent être reproduites en utilisant d'autres outils de CAO.

Détection des glitchs

Pour détecter un glicth à la sortie d'une porte logique, il faudra comparé son délai inertiel (d) à la différence d'arrivée de temps (DPD) de ces signaux d'entrée. Ainsi, le premier et le dernier temps d'arrivée doivent être connu à la sortie de chaque nœud dans le circuit. Pour ce faire, les travaux précédents utilisent généralement une procédure itérative où les portes sont parcourus dans un ordre topologique depuis les entrées primaires jusqu'aux sorties. Ainsi, le premier et le dernier temps d'arrivée à la sortie d'une porte logique sont respectivement l'accumulation du délai inertiel et le premier et le dernier temps d'arrivée des signaux d'entrée. La Figure 4 illustre cette procédure.

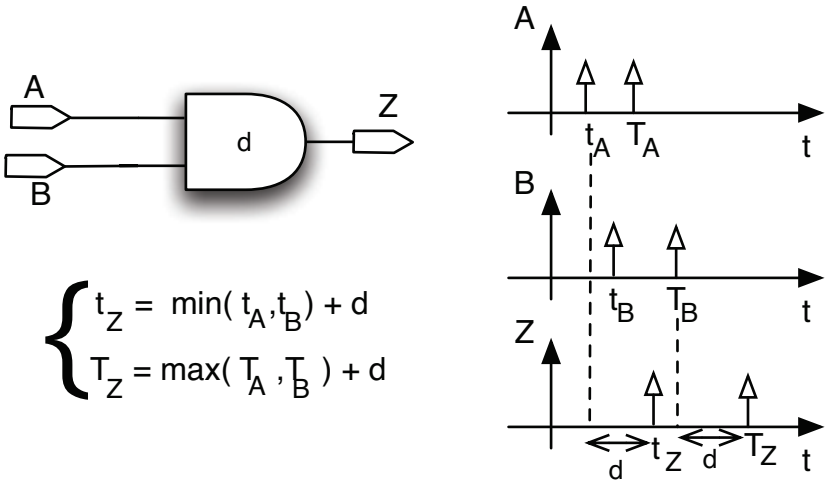


Figure 4: Détermination du premier (t_i) et du dernier (T_i) temps d'arrivée.

Généralement, le délai inertiel (d) est dérivé d'un modèle qui utilise des paramètres d'ajustement telles que la capacité de charge, le temps de transition d'entrée, température,

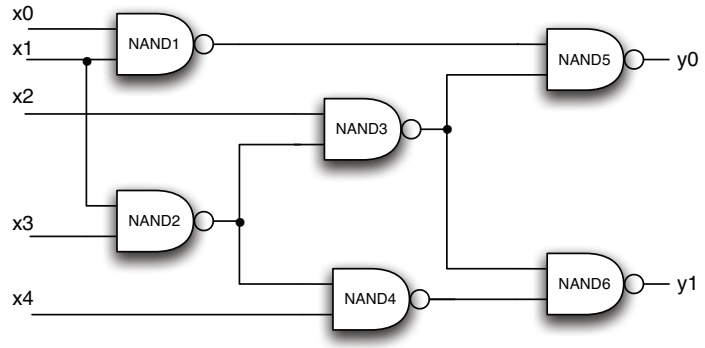
nombre d'entrées / sorties de la porte... etc. Cependant, une telle approche peut ne pas avoir une précision suffisante puisque les capacités de fils ne sont pas calculées dans ce cas.

Dans ce travail, pour analyser et détecter les glicths, on examine les rapports de timing générés après placement routage et extraction des parasites. Ici, on a utilisé l'outil de cadence (Cadence's Encounter toolset). J'ai développé un outil (principalement en Python) qui lit le rapport de timing et génère un rapport simplifié qui contient, pour chaque porte dans le circuit l'information suivante :

- Le délai inertiel (d);
- La différence d'arrivée de temps entre les signaux d'entrée (DPD).

Après ce rapport est remodifié pour inclure que les portes qui ne satisfait pas la condition de filtrage (d> DPD). Ce nouveau rapport contient alors toutes les portes susceptibles d'avoir un glitch et on l'appelle « rapport du glitch ».

La Figure 5 montre le rapport du glitch du circuit test c17 de la famille ISCAS85.



Gate	ID (ns)	DPD (ns)
NAND3	0.12	0.139
NAND4	0.09	0.139
NAND5	0.142	0.164

Figure 5: rapport du glitch du circuit c17.

Dans ce travail et pour des raisons de simplification, le pourcentage de glitch est simplement le pourcentage des portes susceptibles d'avoir un glitch. Il est calculé en utilisant le nombre de portes dans le rapport du glitch. Cette métrique peut sur/sous estime l'activité liée aux glitchs qui dépend énormément des probabilités de transitions des signaux d'entrée. Cependant, il est évident que la diminution du nombre de glitchy portes diminue les transitions. Pour tester les algorithmes de minimisation de glitchs, je propose d'intégrer cet outil dans le flow de conception Top-down comme illustré dans la Figure 6.

Méthode proposée pour la réduction des glitchs (Dual-V_t)

La technique qu'on propose dans ce travail pour la minimisation des glitchs consiste à utiliser la tension de seuil la plus élevé pour les portes susceptibles d'avoir un glitch afin d'augmenter leur délai inertiel et assurer ainsi la condition de filtrage. Comme peut être déduit de l'équation 8, le délai de la porte augmente avec la tension de seuil augmente.

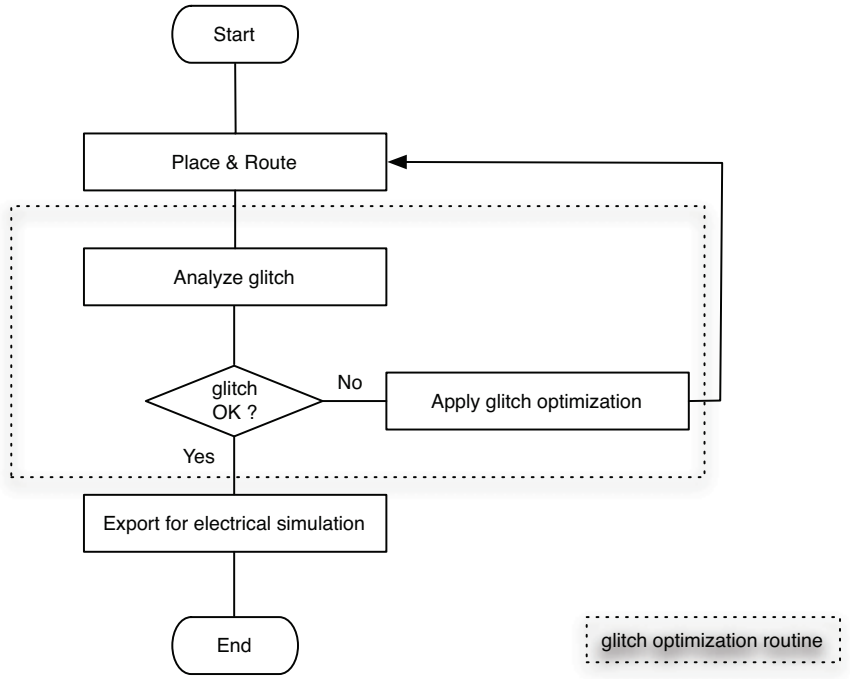


Figure 6: Flow de conception avec optimisation de glitch.

$$t_{pd} = \frac{C_L V_{DD}}{(V_{DD} - V_t)^{\alpha}} \tag{8}$$

Où C_L et α sont respectivement la capacité de charge et l'indice de saturation de vitesse.

Le tableau 1 illustre les délais inertiels de quelques portes appartenant à une bibliothèque industriel 65nm avec une tension de seuil standard (SVT) et une tension de seuil élevé (HVT). Ces résultats sont obtenus en utilisant un simulateur Spice avec une tension d'alimentation nominale $V_{DD} = 1.2$V.

Table 1: Délais inertiels de portes en SVT et HVT.

	Inertial delay (ps)		
Gates	SVT	HVT	% increase
AND	45.1	65.5	45.2
OR	55.2	80.1	45.1
NAND	20.4	28.1	37.7
NOR	25.4	36.1	42.1
		Average	42.5

Nous observons une augmentation moyenne de 42% dans le délai inertiel, passant d'une porte SVT à une porte HVT. Ainsi, l'utilisation des portes HVT peut filtrer complètement les glitchs si la différence d'arrivée de temps (DPD) n'est pas supérieure en moyenne à 1,4

le délai inertiel. Mais, même si le DPD est trop importante, l'utilisation des portes HVT peut diminuer l'amplitude des glitchs ce qui diminue aussi l'énergie dynamique.

Pour illustrer la technique proposée, considérons le circuit sur la Figure 7(a) qui montre un glitch à la sortie de la porte AND. Ce glitch apparaît due à l'arrivée en retard du signal d'entrée B par rapport au signal d'entrée A. Les résultats de simulation en utilisant une porte AND en SVT et après en HVT sont présentés dans la Figure 7(b).

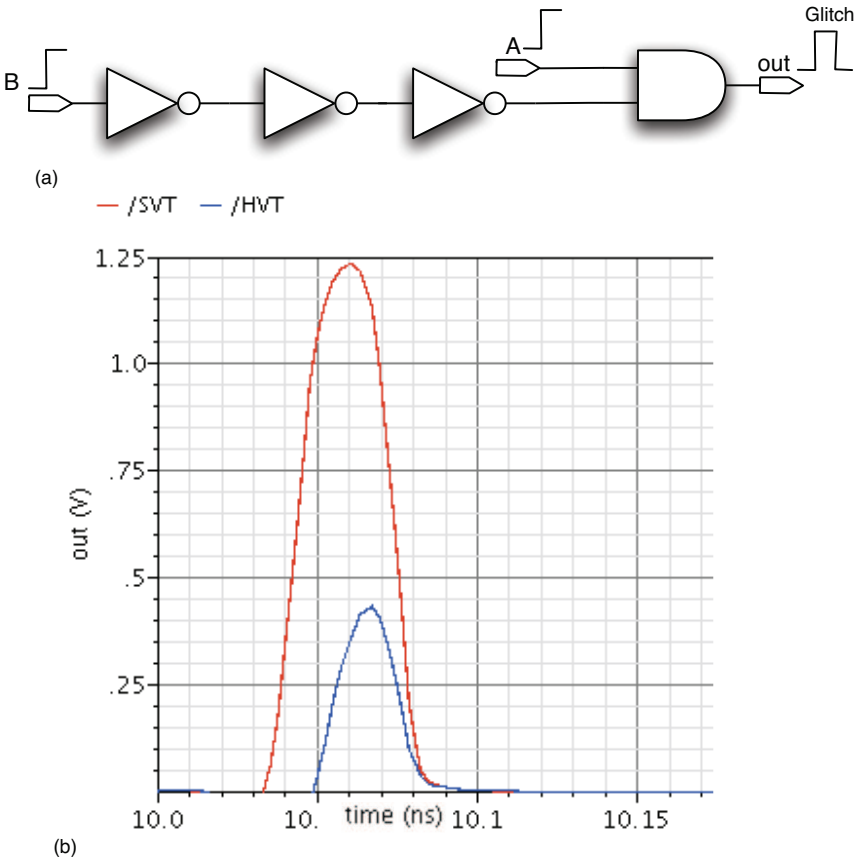


Figure 7: Tension du signal de sortie en utilisant différentes tensions de seuil.

Nous observons que le glitch est partiellement filtré lors de l'utilisation la porte HVT et cela est dû à l'augmentation du délai d'inertie.

On propose un algorithme d'optimisation heuristique (Algorithm 1) qui utilise la technique proposée.

Pour assurer un meilleur filtrage de glitch, nous avons combiné la technique proposée avec le downsizing technique. Pour ce faire, on applique premièrement l'algorithme heuristique Dual–V_t proposé. On utilise la méthode de Hill climbing pour parvenir à une solution optimale. Puis, on utilise le downsizing pour augmenter le délai inertiel des portes qui restent encore dans le rapport du glitch. Et à chaque fois, on vérifie que les contraintes de timing sont respectées.

Algorithm 1 Algorithme heuristique Dual-V_t

```
GlitchOptimization(G,S,L) begin
Criticalpathdelay(G, S)
if C ← glitchreport(G, S) ≠ 0 then
    D ← Sort(C)
    S' ← S
    for all ν_j ⊆ D do
        s_j^svt ← search(ν_j, S)
        s'_j ← s_j^hvt
        Criticalpathdelay(G, S')
        if chektiming(G, S') = failed then
            s'_j ← s_j^svt
        end if
    end for
end if
end
```

Résultats

Les algorithmes proposés (Dual-V_t et Dual-V_t/down-sizing) ont été implémentés en Python. Ils ont été testé sur 6 circuits test de la famille ISCAS85 synthétisés avec une bibliothèque industriel 65nm avec une tension de seuil standard (SVTLP). Le tableau2 montre le pourcentage de réduction de glitch en utilisant Dual-V_t, le down-sizing et l'algorithme unifié Dual-V_t/down-sizing.

Table 2: Réduction de glitch avec 5% de relaxation de contraintes de timing.

Circuits		% of glitch	% of glitch reduction		
Name	Gates		Dual-V_t	down–sizing	Dual-V_t/down–sizing
c432	160	28.5	41.6	22.9	41.6
c499	202	42.8	12.2	12.2	21.1
c880	383	44.9	8.7	12.2	18.2
c1908	880	30.6	8.0	15.4	17.4
c2670	1193	23.9	21.7	25.7	38.6
c3540	1669	33.6	5.5	2.1	5.6
	Average:	34.0	16.2	13.0	23.7

Les résultats montrent que l'algorithme Dual–V_t permet d'atteindre 16% de réduction de glitch en moyenne. Nous constatons aussi que la technique de Dual-V_t permet un meilleur filtrage de glitch que la technique down-sizing. On remarque que le circuit c432 ne bénéficie pas de la combinaison Dual–V_t / down-sizing. Ceci peut s'expliquer par le fait que la plupart des glitchs dans ce circuit sont causés par un DPD relativement faible qui peut être supprimé en appliquant simplement le Dual–V_t technique. Tandis que les

autres portent qui figurent dans le rapport du glitch ont un DPD trop large qui ne peut même pas être balancée en utilisant le down-sizing. Pour d'autres circuits comme le c2670, le down-sizing semble être plus efficace que le Dual–V_t. En effet, Les glitchs causés par des larges DPD sont mieux filtrés en utilisant le down-sizing comme il augmente le délai inertiel beaucoup plus que le Dual–V_t. A partir de là, nous concluons que les techniques Dual–V_t et down-sizing sont complémentaires, puisque la première est plus adaptée aux portes avec des faibles DPD alors que la deuxième est plus efficace en cas de larges DPD.

Le tableau3 présente le pourcentage de réduction de l'énergie totale pour les circuits c432 et c3540 appartenant à la famille ISCAS85. Ces résultats sont obtenus via les simulations Spice avec des stimuli aléatoires.

Table 3: Réduction d'énergie totale en appliquant dual–V_t, down–sizing and Dual–V_t/down–sizing.

Circuits		% of total energy reduction		
Name	Gates	Dual-V_t	down–sizing	Dual-V_t/down–sizing
c432	160	27.3	33.4	48.9
c3540	1669	9.2	16.8	27.1

A l'inverse de ce qui est attendu, down-sizing présente une réduction de l'énergie totale mieux que Dual–V_t. Ceci peut être expliqué par la quantité d'énergie économisée grâce à l'utilisation de petits transistors. Néanmoins, cela ne contredit pas, en aucun cas, les résultats du tableau où la technique Dual–V_t permet un meilleur filtrage des glitchs. Nous remarquons que la combinaison Dual–V_t/down-sizing permet une amélioration de 15% de l'énergie totale consommée comparée à la technique down-sizing appliquée toute seule. Cela montre encore une fois la complémentarité de la technique proposée Dual–V_t et de la technique down-sizing connue pour la minimisation des glitchs.

Modélisation des circuits sous seuil

Une autre stratégie pour réduire l'énergie est d'utiliser la tension d'alimentation la plus basse celle qui satisfait la performance requise. Récemment il a été démontré que le point d'énergie minimale peut être atteinte en utilisant une tension d'alimentation inférieure à la tension de seuil des transistors. Cette opération sous-seuil a été une solution idéale pour les applications ultra basse consommation qui ont une vitesse relativement faible. L'émergence de ces applications a motivé la recherche liée à la logique sous-seuil. Toutefois, ces circuits sont plus sensibles aux variations des procédés de fabrication. Ce qui peut impacter la robustesse et la performance du circuit.

Dans ce travail, l'impact de la variabilité sur les performances des circuits travaillant

sous-seuil est analysé et modélisé. Les simulations sur un circuit test synthétisé avec une bibliothèque industriel 65nm valide les modèles développés. Nous montrons que la variabilité déplace le point d'énergie minimale vers la région d'inversion modérée. Ainsi, quand on tient en compte de la variabilité, il faudra utiliser un modèle complet qui inclut à la fois la région d'inversion faible et celle modérée est nécessaire afin de modéliser correctement les performances du circuit autour du point d'énergie minimale. Nous remarquons, cependant, que l'énergie reste parfaitement estimée en utilisant le modèle en inversion faible. Ainsi, en utilisant les équations simples du modèle WI, nous avons développé une expression analytique qui détermine la tension d'alimentation optimale celle qui minimise l'énergie totale et ceci en tenant compte des variations des procédés de fabrication.

L'opération sous-seuil : Concept

L'opération sous-seuil consiste à réduire la tension d'alimentation en dessous de la tension de seuil V_t afin d'atteindre le point d'énergie minimale. Le concept est simple : comme la puissance dynamique P_{dyn} est proportionnelle au carré de la tension d'alimentation V_{DD}, une petite réduction de la tension d'alimentation provoque une réduction quadratique de la consommation de puissance dynamique, mais aussi une importante augmentation du délai de calcul. Ce qui engendre une augmentation de l'énergie de fuite. Les tendances opposées des deux formes d'énergie (dynamique et fuite) conduisent à un point d'énergie minimal atteint à une tension d'alimentation optimale. Ce point se produit souvent dans la région d'inversion faible (WI), où les courants de fuite sous-seuil sont utilisés comme le courant de drain actif.

Figure 8 trace le courant de drain I_D en fonction de V_{GS} pour un transistor NMOS appartenant à une technologie 65nm.

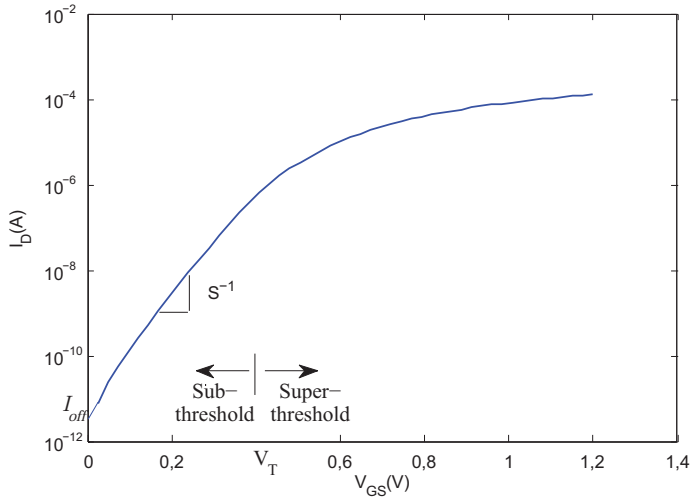


Figure 8: I_D en fonction de V_{GS} pour un transistor NMOS.

Cette courbe souligne deux caractéristiques principales de l'opération sous le seuil. Tout d'abord, dans la région WI, le courant de drain dépend exponentiellement de V_{GS}. D'autre part, la baisse de la tension d'alimentation provoque la dégradation du rapport courant active/courant de fuite(I_{on}/I_{off}), du facteur S défini comme suit: $S = nV_t ln10$, où n est le facteur de pente sous le seuil.

Notez que, la réduction exponentielle du courant active Ion dans la région d'inversion faible conduit à une augmentation exponentielle du délai de calcul comme le montre la figure . Cela limite l'utilisation de l'opération sous-seuil aux circuits qui ne nécessitant pas de hautes fréquences de fonctionnement. Par exemple, les réseaux de capteurs sans fils sont une de ces applications.

La Figure 9 montre l'évolution de l'énergie totale, active et de fuite d'une chaîne d'inverseurs de profondeur logique 23. Comme prévu, l'énergie dynamique (E_{dyn}) diminue quadratique avec la tension d'alimentation tandis que l'énergie de fuite (E_{leak}) augmente de manière exponentielle en raison de l'augmentation exponentielle du temps de calcul. Ces tendances opposées conduisent à un point d'énergie minimal atteint à la tension d'alimentation optimale $V_{DDopt} = 0.2V$.

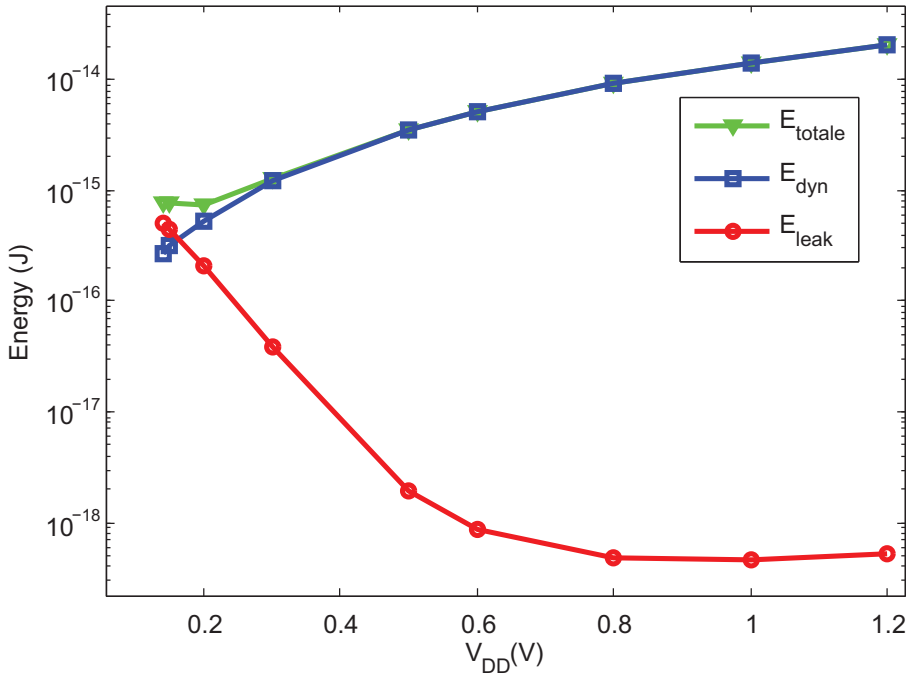


Figure 9: Evolution de l'énergie totale, active et de fuite pour une chaîne de 23 inverseurs.

L'opération sous seuil : impact de la variabilité

Les variations des procédés de fabrication (variabilité) sont des fluctuations des paramètres du transistor autour de la valeur prévue et qui sont due à la non idéalité de la fabrication

des procédés. Ils sont généralement classés en variations globales (intra-die) variations locales (inter-die). Comme les courants sous seuil dépendent exponentiellement de la tension de seuil, la variation de Vt induit une variabilité du courant considérablement importante dans la région d'inversion faible. Les deux types de variations (globales et locales) peuvent induire des variations de la tension de seuil Vt. Cependant, il a été démontré que les variations globales peuvent être compensée en utilisant une polarisant le substrat (Adaptive Body Biasing). Ainsi, dans notre travail on considère que les variations locales.

La Figure 10 trace la distribution du délai d'un additionneur 32 bits synthétisé avec une bibliothèque industriel 65nm avec une tension d'alimentation nominale $V_{DD} = 1.2V$ et sous-seuil $V_{DD} = 0.2V$. Les résultats sont obtenus à partir d'une simulation Monte Carlo avec 1000 points.

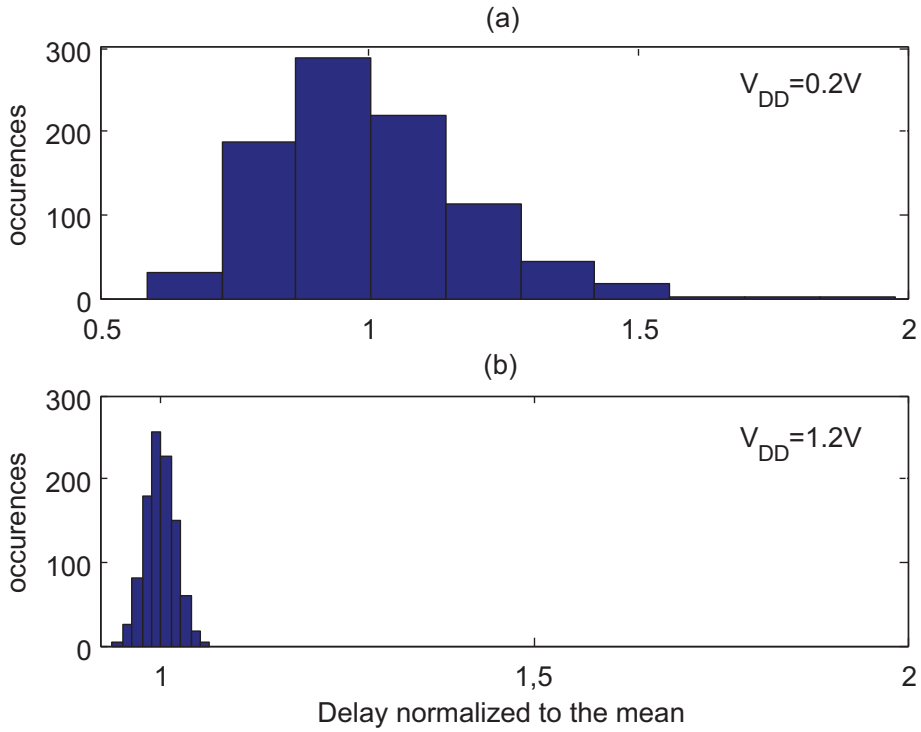


Figure 10: Distribution du délai d'un additionneur 32 bits (a) $V_{DD} = 0.2V$ (b) $V_{DD} = 1.2V$.

Nous observons que la déviation standard avec une tension d'alimentation sous seuil est deux fois plus importante que celle avec une tension nominale. Cela implique une variabilité plus élevée dans la région d'inversion faible qui doit être pris en compte lors de la conception des circuits sous seuil. En effet, cette variabilité peut engendrer des problèmes de fonctionnement. Par exemple, si les variations de procédés de fabrication favorisent le transistor NMOS par rapport au PMOS, alors les courants de fuites dans le transistor NMOS seront plus importants et le courant active au niveau de PMOS sera plus faible. Avec ce scénario, le réseau PMOS peut échouer à charger la capacité de la sortie à

une tension considérée comme un 1 logique.

La variabilité a aussi un impact sur le point d'énergie minimale. La Figure 11 trace l'énergie en fonction de la tension d'alimentation pour le circuit test additionneur 32-bit avec et sans l'impact de la variabilité. On observe que la variabilité augmente l'énergie de fuite ce qui augmente l'énergie totale et déplace le point d'énergie minimale dans la région d'inversion modéré.

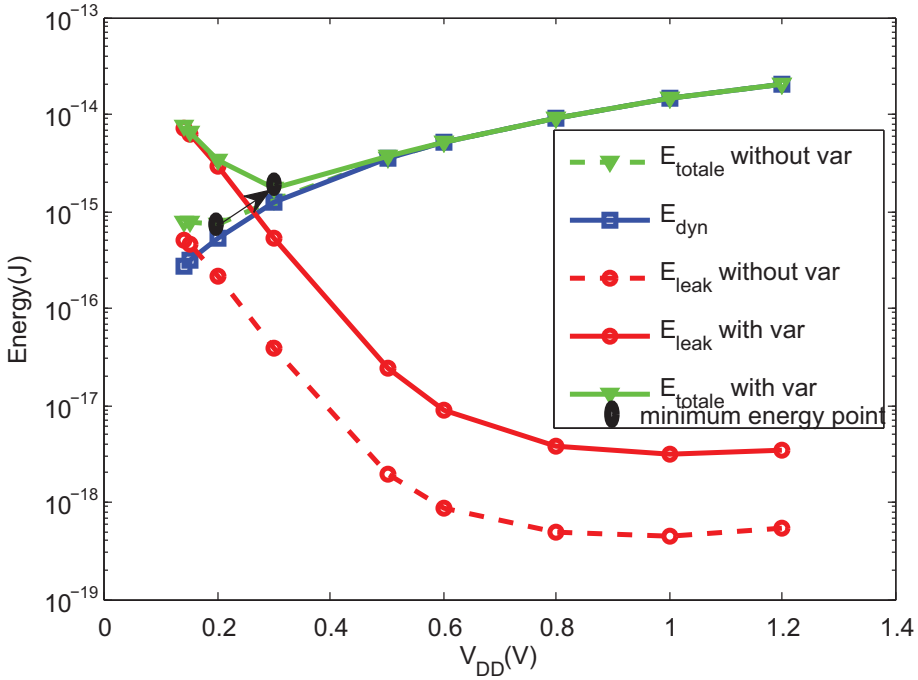


Figure 11: Evolution de l'énegie sans et avec l'impact de la variabilité.

Modélisation des circuits sous seuil en tenant compte de la variabilité

La caractérisation du point d'énergie minimale (PEM) a été adressée dans les travaux précédents en utilisant le simple modèle d'inversion faible. Cependant, la variabilité affecte considérablement le PEM qui se déplace vers la région d'inversion modérée. La modélisation dans cette région en appliquant un modèle complet est également envisagée dans les travaux précédents, mais l'impact de la variabilité n'est pas été analysé dans ce cas.

Dans cette section, nous présentons un modèle qui s'étend sur les régions d'inversion faible et modérée et qui en plus tiens en compte de la variabilité. Ce modèle est basé sur les expressions EKV. Nous commençons avec un modèle simple basé sur les caractéristiques d'un inverseur, puis nous montrons que ce modèle permet de prédire le comportement des circuits plus complexes. Nous montrons à travers les simulations Spice et Matlab que le modèle d'inversion faible (WI) n'est plus suffisant pour modéliser les performances d'un système exposé aux variations des procédés de fabrication. Le modèle proposé est conçu

comme un outil simple pour évaluer les compromis entre énergie, vitesse et variabilité confrontés lors de la conception des circuits sous seuil.

Modélisation des circuits sous seuil en tenant compte de la variabilité

Dans la région d'inversion faible et celle modéré, le courant du drain peut être exprimé par :

$$\begin{aligned} I_{DS} =& I_S(ln^2[1+exp\frac{V_{GS}-V_t}{2nU_T}]- \\ & ln^2[1+exp\frac{V_{GS}-V_t}{2nU_T}exp\frac{-V_{DS}}{2U_T}]) \end{aligned} \tag{9}$$

Où n est le facteur de pente sous seuil, V_{GS} et V_{DS} sont respectivement les tensions entre la grille et la source et le drain et la source, V_t est la tension de seuil et I_S le courant de saturation. La Figure 12 montre le courant de drain I_D en fonction de la tension V_{GS} pour un transistor NMOS.

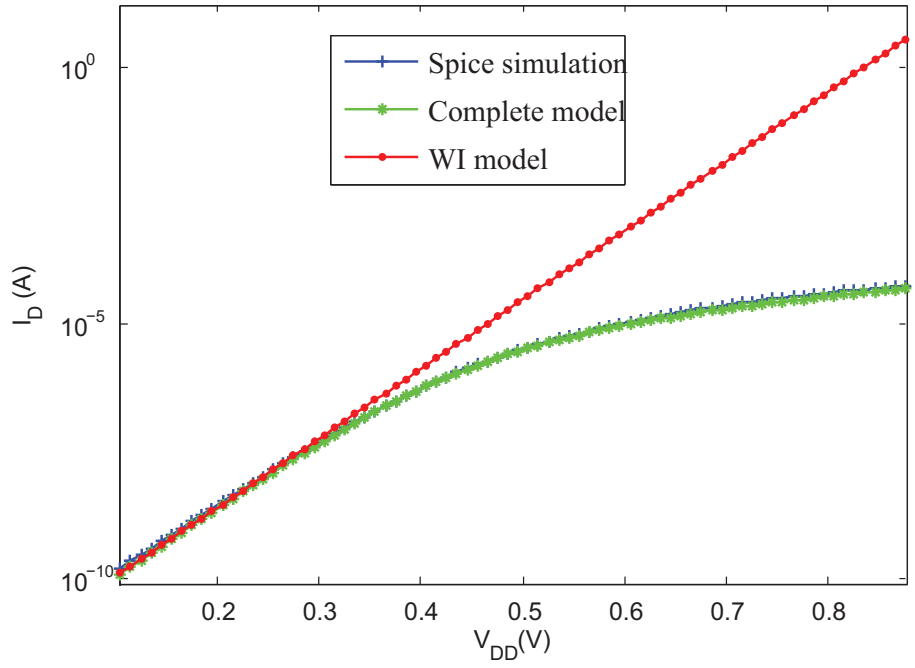


Figure 12: I_D en fonction de V_{GS} pour un transistor NMOS.

Comme prévu, le modèle d'inversion faible n'est pas suffisant pour modéliser le courant dans la région proche du seuil. Nous observons aussi que le modèle complet n'est pas si précis dans la région d'inversion forte due à l'absence de modélisation de la réduction de la mobilité et de la saturation de vitesse.

L'expression du délai de l'inverseur peut être estimée comme suit:

$$T_d = \frac{C_L V_{DD}}{I_{on}} \tag{10}$$

Où C_L est la capacité de sortie, V_{DD} est la tension d'alimentation et I_{on} est le courant saturé.

Pour extraire les paramètres du modèle, nous avons appliqué la méthode gm / ID. Les valeurs suivantes ont été obtenues pour le transistor nMOS d'un inverseur basique appartenant à la bibliothèque industrielle 65nm considérée :

- n = 1.22;
- $V_t = 0.38$(V);
- $\beta = 4.83e^{-4}(A/V^2)$.

Pour l'analyse de la variabilité, nous allons considérer que les variations de V_t et de β, modélisés comme des distributions normales avec des paramètres μ_{V_t}, σ_{V_t}, μ_β, σ_β), extraites à partir des simulations Monte Carlo. Le modèle en tenant compte des variations des procédés est dérivé de l'équation en substituant les valeurs de β et V_t par les échantillons β et V_t qui suivent la distribution normale. En appliquant l'expression du délai, on aura les paramétres de la distribution du délai de l'inverseur.

Pour un circuit complexe avec une profondeur logique L_D correspondant au nombre d'inverseurs qui composent le chemin critique du circuit, le délai s'écrit:

$$T_{circuit} = L_D.T_d \tag{11}$$

En tenant compte de la variabilité, le délai d'un circuit sera la somme de L_D distribution de délai de l'inverseur. Cette somme aura différente expression dépendant de la nature de T_d. En effet, dans la région d'inversion faible, le courant dépend exponentiellement de V_t. Ainsi, avec une distribution normale de V_t, le courant aura une distribution lognormale. Dans la région d'inversion forte, le courant dépend quadratiquement de V_t. Ainsi, il aura une distribution normale. Mais la question qui se pose, c'est quelle distribution aura dans la région d'inversion modéré ?.

Pour savoir quelle distribution suit au mieux dans cette région, j'ai utilisé une méthode graphique qui consiste à superposer les données issues de la simulation Monte Carlo à des données synthétiques. J'ai constaté que jusqu'à $0.7V$ la distribution du délai est plutôt lognormale. J'ai développé 2 expressions analytiques qui estiment la dispersion du délai d'un circuit complexe dans le cas où la distribution est lognormale et dans le cas ou elle est normale.

La Figure 13 compare les résultats des expressions analytiques aux résultats obtenues par simulations Spice.

Nous constatons que le modèle suit parfaitement la simulation SPICE pour les deux

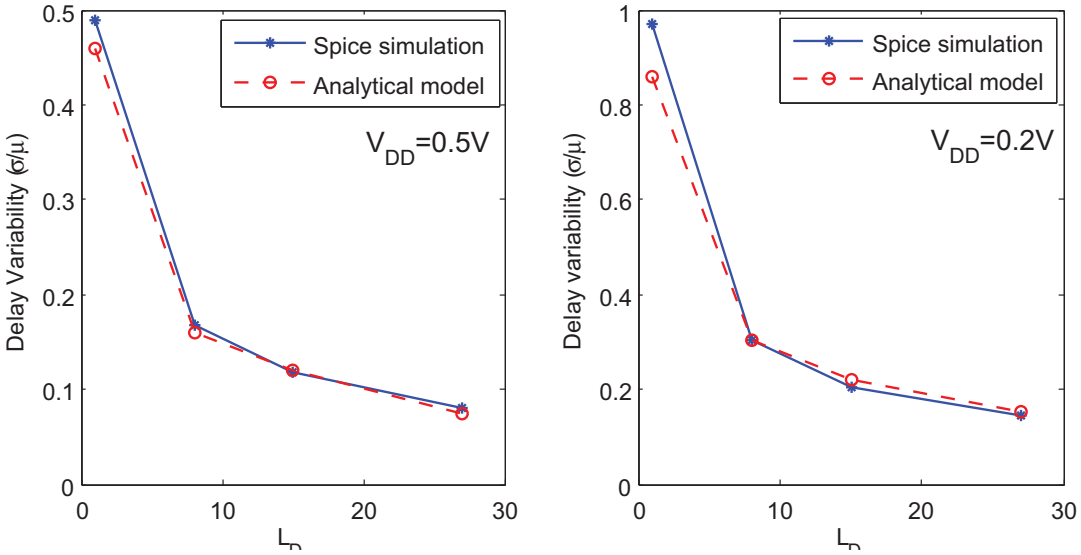


Figure 13: Dispersion du délai pour différents profondeur logique avec $V_{DD} = 0.5V$ et $V_{DD} = 0.2V$.

tensions d'alimentation.

J'ai aussi cherché à trouver le délai pire cas dans le cas des deux distributions. La Figure 14 trace le délai pire cas du circuit test additionneur 32-bit obtenu avec le modèle d'inversion faible, le modèle complet et la simulation Spice.

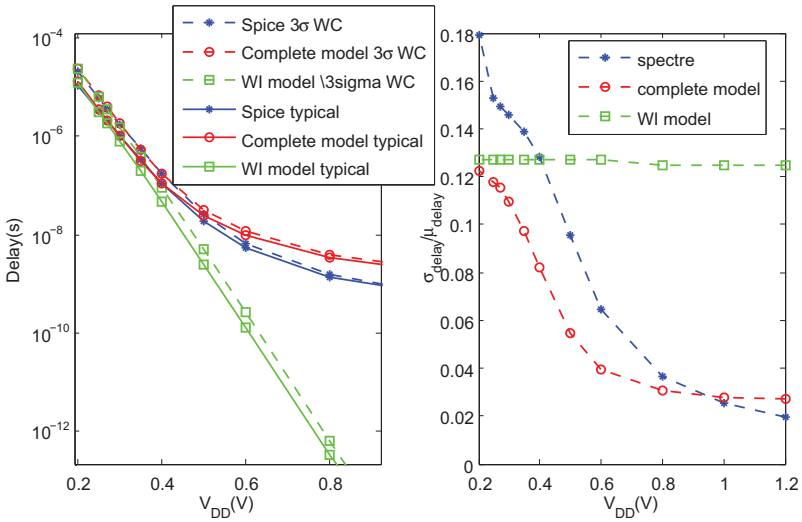


Figure 14: Evolution du délai nominal et du délai pire cas d'un additionneur 32-bit obtenu avec le modèle d'inversion faible, le modèle complet et la simulation Spice.

On observe que le modèle complet suit parfaitement la simulation Spice alors que le modèle d'inversion faible dévie à partir de 0.3 V.

Modélisation du point d'énergie minimale avec variabilité

L'expression de l'énergie en tenant compte de la variabilité est issue de l'expression habituelle en remplaçant le délai par le pire délai. La Figure 15 montre l'énergie consommée par le circuit test additionneur 32-bit en considérant les variations des procédés de fabrication.

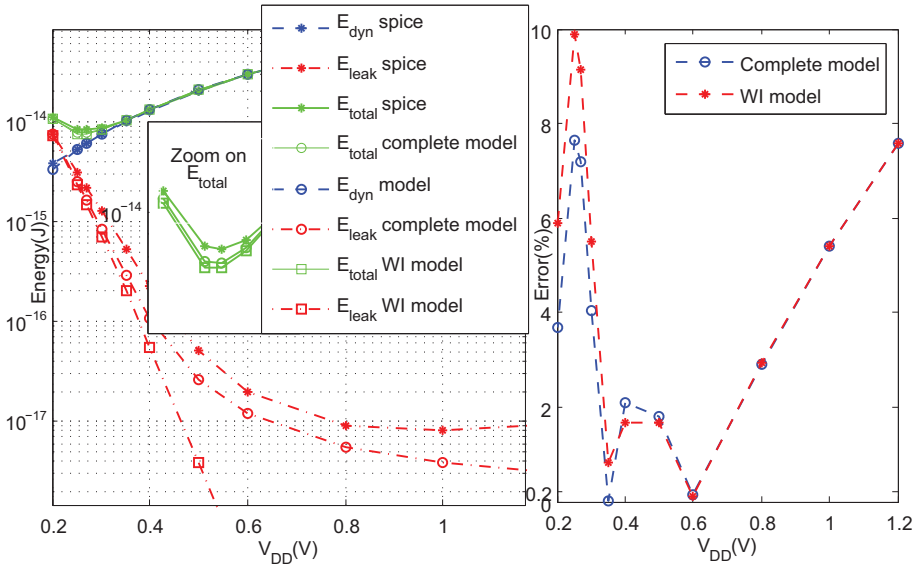


Figure 15: Energie consommée par le circuit test additionneur 32-bit en considérant les variations des procédés de fabrication.

Nous observons que l'énergie totale consommée est légèrement différente de celle déterminée par les modèles. L'erreur sur le point d'énergie minimale est de 7% et 9%, lorsqu'il est obtenu par le modèle complet et le modèle d'inversion faible, respectivement. L'erreur du modèle WI est comparable à celle du modèle complet. Ceci peut s'expliquer par la tendance opposée de la variabilité et du délai lorsqu'ils sont déterminés par le modèle WI. Ainsi, le modèle d'inversion faible reste valable pour l'estimation de l'énergie même en région d'inversion modérée.

Aussi, j'ai proposé une expression analytique qui détermine la tension d'alimentation optimale celle qui minimise l'énergie totale quand les variations des procédés de fabrication sont considérés. Puisque le modèle d'inversion faible estime bien l'énergie, j'ai utilisé les équations simplifiées du modèle de l'inversion faible et j'ai cherché la solution :

$$\frac{\partial E_{tot,var}}{\partial V_{DD}} = 0. \tag{12}$$

Pour le circuit additionneur 32-bit, la solution analytique de la tension d'alimentation optimale est de $0.252V$ et avec simulation spice est de $0.27V$, ce qui correspond à 6% d'erreur.

List of Figures

Introduction

"Low energy design", an interesting research area and a key word throughout this dissertation. Although it is widely investigated, one can not deny that we still need techniques and models in order to achieve better power-speed-density trade-offs.

Before, power consumption was not considered a so important issue in the design of Integrated Circuits (IC). Recently, with the increase of the complexity of communications systems and computing devices integrating high-speed computations and complex functionalities, power dissipation has become a third design constraint with speed and area.

Today, every circuit has to face problems related to power consumption increase. First, portable devices which are battery-powered systems require extended discharge time to ensure an acceptable duration of service. Unfortunately, with the slow rate of battery technology improvement, techniques aiming at low energy design are essential for such devices. Second, high-performance applications produce more heat as its clock frequency is getting higher. Since, operating frequencies keep increasing into the GHz range, heat generation becomes a very serious issue for IC design. Indeed, beyond a certain thermal limit, the reliability of the circuit can be severely affected. And dealing with cooling mechanisms is not only complex but also costly which constraints circuit commercialization. Obviously, the problem gets worse when we speak about portable devices with high performance requirements such as the new generation of cell-phones or laptops. For some other applications, energy is even the primary concern well before the constraint couple speed-density. Those systems, called Ultra-Low-Energy, require a minimum energy consumption for a successful operation. Take for instance implanted devices for health applications. A very long battery lifetime is needed to avoid the surgery for its replacement.

To help designing for low energy, a significant research interest in power optimization techniques has emerged. They are applied at many different levels of the design hierarchy. At each level, architectural and technological choices are made to produce an optimal solution that trades-off timing, area and power dissipation.

The main objective of the current work is to extend Low-Energy research area by proposing methods and models that facilitate the work of circuits and systems designers targeting "Low Energy Design".

An important factor that impacts the power dissipated in the circuit is the switching activity defined as the average number of transitions per cycle. Normally, In nominal synchronous logic, each net would change at most one time each clock period. However, multiple transitions can occur within the same clock cycle due to intermediate computation states. These spurious transitions are called glitches and they can contribute in the order of 30%-40% of the dynamic power consumption in CMOS circuits. This large amount of power dissipated by switching that do not contribute to the functioning of the circuit, has motivated a significant investigation on glitch power optimization. Gate–sizing and path–balancing are among the most known techniques to reduce glitches. Although, these methods allow an interesting glitch filtering, their application is very dependent on the circuit constraints. For instance, gate–downsizing, where smaller gates are used, may increase the delay of the circuit which limits its use when performance constraint is introduced. Likewise, path balancing, where extra elements are inserted, can cause area issues. So, clearly, there remains a need to expand glitch reduction research in order to achieve an optimum design.

Based on that fact, this work proposes a new method for glitch minimization using dual-threshold voltage assignment. The strategy is to use high threshold voltage to increase the inertial delay of glitchy gates. A heuristic algorithm to address this problem is presented. Simulations carried out on 6 ISCAS85 benchmark circuits provides a comparison between the proposed method and the well-known gate–sizing method. A combination of these two techniques proves their complementary for glitch reduction.

Although several works focus on glitch reduction, CAD tools for glitch analysis and optimization is still in its infancy. Hence, the current work also proposes a framework to analyze glitch and to test glitch optimization techniques based on the utilization of Cadence tools. However, the different steps can be reproduced using other CAD tools. Such glitch aware tool can be easily integrated in the top-down design flow to facilitate glitch analysis and optimization.

Another key strategy to reduce the energy is to scale the supply voltage until the required performance is met. Recently, it has been shown that the minimum energy consumption can be achieved using a supply voltage well bellow the threshold voltage. This sub-threshold operation has been an ideal solution for ultra low energy applications with low demand in speed requirements. As the interest for ultra low energy has increased, research related to sub-threshold logic has attained considerable importance. Modeling and characterization of sub-threshold operation for standard CMOS cell designs have been investigated for energy and performance analysis. However, lowering the supply voltage increases the sensitivity to process variations which can impact the robustness and effective performance of the circuit. On the other hand this sensitivity decreases as we move

towards near-threshold operation.

Based on that fact, this work investigates the impact of variability on sub-threshold and near-threshold circuit performance through analytical modeling and circuit simulation in a 65 nm industrial low power CMOS process. We show that variability moves the effective minimum energy point towards the near threshold region. Thus, we demonstrate that when variability is taken into account, a complete model that includes the near threshold (moderate inversion) region is necessary in order to correctly model circuit performance around the minimum energy point. However, we notice that the energy is perfectly estimated using the weak inversion model even in near-threshold voltage region. Hence, using the simple equations of the WI model, we developed an analytical expression for the optimum supply voltage that minimizes the total energy consumption under variability considerations.

The current work is organized as follows. Four chapters related through the keyword "Low Energy Design". First, the fundamental questions of Low Energy Design via reversible computing are presented in Chapter 1. Next, Chapter 2 surveys state-of-the-art power dissipation, power estimation techniques and power optimization techniques targeting Low Energy Design in CMOS circuits. Chapter 3 investigates glitch reduction for Low Energy Design. First, a glitch analysis tool is developed to help testing optimization algorithms for glitch minimization. Then, a new method for glitch filtering is introduced. In Chapter 4, sub-threshold design under variability aware analysis is examined. A complete model that takes into account process variations effects is introduced. And, an analytical solution for the optimum supply voltage targeting the lower energy is provided. Finally, we summarize the contributions made in this thesis and discusses some directions for future work.

Chapter 1

Reversible computing for low Energy Design

1.1 Introduction

Today, power dissipation has clearly been identified as a key limiter of continued CMOS device scaling. However, fundamental questions about heat dissipation were raised along before.

It all begins in 1961 in [2], where the question was : "How much heat generation to perform a certain computation?". In that paper, Rolf Landauer realized that the erasure of a bit of computational information requires inevitably the generation of an amount of physical entropy that he quantified as $K_b Log2$, where K_b is the Boltzmann constant. In other words, Landauer point out that logical irreversibility (bit erasure) in a computation should ultimately be accompanied with physical irreversibility (heat generation) in the system. Although, there is no proof on such principle, almost all researchers agree.

So, hopefully to find non-dissipative systems, researches concentrate first on proving the possibility of reversible computation. In this context, Bennett in 1973 [3], extended Landauer's work to prove, using a Turing machine model, that any irreversible computation can be performed in reversible way by saving a complete history of all the informations in each computational step and that the reversibly recorded history could also be cleared in a reversible manner except for the output result. Hence, he showed, even in agreement to Landauer's principle, that the energy dissipated in a computation can be proportional to the output size only.

This is far from the practical situation where the dissipation is proportional to the activity and complexity of the computational task.

Unfortunately, traditional CMOS logic devices as they are currently designed perform

irreversible elementary operations that consume (10^{12}) times more than the minimum energy limit imposed by Landauer's theory.

At this point, the question is : Can further improvements of this technology lead us to be closer to the optimum energy consumption. Or, is it impossible due to the variety of obstacles such as variability and leakage currents of the present semiconductor technology, to reach the fundamental physical limits of computation?. Or, maybe, we should look for other technologies.

Recently, efforts have been concentrated to build reversible logic elements based on today's electronic technology. The major motivation is to ameliorate the efficiency of today's computers through reversible computing.

In this chapter, we briefly describe some known reversible logic gates such as Fredkin and Toffoli gates and reversible technologies such as adiabatic circuits.

We will especially focus on Split Charge Recovery Logic (SCRL) adiabatic circuit developed by Younis and presented in [4], as it is the only attempt of non-dissipative computing using conventional CMOS logic. Then, we will discuss recent accomplishments in reversible computing.

1.2 Reversible computation

Based on the principle that logical irreversibility implies necessarily physical irreversibility and that physical irreversibility is synonym of dissipation, a demonstration that a reversible machine can not be built, would eliminate the only hope for dissipationless computation. Hence, efforts have been concentrated to prove first the feasibility of reversible machines.

Using two different approaches, Bennett, Fredkin and Toffoli [5] have demonstrated the possibility of reversible computation. While Bennett used a Finite State Machines, Fredkin and Toffoli relied on boolean functions instead.

1.2.1 Bennett's reversible machine

To build reversible machine, Bennett used a Turing machine. A Turing machine is used in computer science to simulate the logic of any algorithm.

It consists of two parts : a tape and a read/write head as illustrated in Figure 1.1. There is a finite set of internal states A_k and a finite alphabet of symbols Σ. The tape is composed of cells containing symbols from Σ. The head moves from one cell to another to allow reading and writing symbols on the tape.

To make a reversible Turing machine, Bennett adds an extra tape where a complete history of all past states is kept. Once the computation ends and the result is copied,

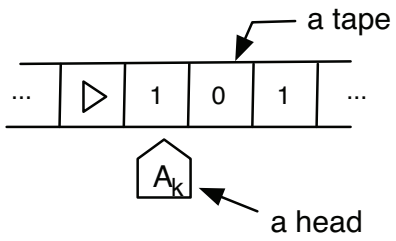


Figure 1.1: A Turing machine.

backward computation is performed to restore the history tape to its initial state without executing any irreversible operation.

At the end of the all process, the resulting Turing machine is exactly the same as it was at the beginning except for the copied result. Figure 1.2 illustrates the three phases of Bennett's reversible Turing machine [6].

Phase	*Work tape*	*History tape*	*Result tape*
1: Computation	_INPUT	_	_
	_OUTPUT	_HISTORY	_
2: Result copy	_OUTPUT	_HISTORY	_OUTPUT
3: History erasement	_INPUT	_	_OUTPUT

Figure 1.2: Phases of Bennett's reversible Turing machine.

1.2.2 Fredkin and Toffoli's demonstration

In order to demonstrate the ability of logical reversibility, Fredkin and Toffoli tried to perform computation using reversible logic elements. Figure 1.3 shows the truth table of Toffoli gate and its representation.

The first observation is that unlike traditional logic elements, where there is just one output, this gate has equal numbers of inputs and outputs. This is because elements which have fewer outputs than inputs can never be invertibles. Furthermore, using exactly 3 inputs is not a hazard. Indeed, Toffoli gate is a universal logic element that can be used to produce any boolean functions performing computation in a reversible manner. This is not possible using two input gates.

Figure 1.4 shows the truth table and the representation of Fredkin gate. In this gate, if

Inputs			Outputs		
p	q	r	p'	q'	r'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

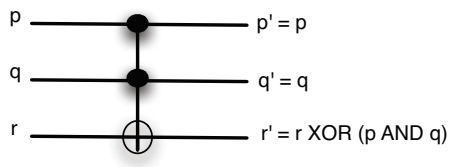


Figure 1.3: The Toffoli gate.

p is a zero, all inputs go through unchanged. Otherwise, the last two inputs are swapped to give the last two outputs, respectively.

Inputs			Outputs		
p	q	r	p'	q'	r'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

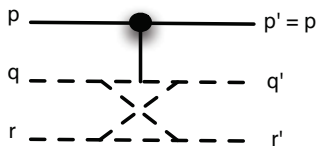


Figure 1.4: The Fredkin gate.

Like Toffoli gate, Fredkin gate is universal. For instance, to produce the AND gate, r should be set to zero. Hence, r' has the value of pq.

Fredkin gate is also known to belong to conservative logic in that the number of 1's in the outputs is equal to that in the inputs. Hence, according to them, any conventional combinatorial circuit can be constructed using conservative logic gates.

However, although their demonstration is convincing for combinatorial circuits, the generalization to sequential ones is weakly argued. Indeed, in their model, the stability is ensured by assumption, since the time and the space are discretized and every wire is a

memory [5].

1.2.3 Discussion

Bennett, Toffoli and Fredkin demonstrations are clearly important to still dream of dissipationless computation. However, they do not prove that this is physically possible.

In fact, as we have mentioned earlier, physical reversibility implies logical reversibility, but we should pay attention that the reverse is not true.

So, to clarify these points, we summarize possible implications in these equations:

$$no\ logical\ reversibility \Rightarrow no\ physical\ reversibility \tag{1.1}$$

$$logical\ reversibility \Leftarrow physical\ reversibility \tag{1.2}$$

$$physical\ reversibility \Leftarrow dissipationless\ computing \tag{1.3}$$

So, physical reversibility implies automatically that the system is also logically reversible. However, even if building logical reversible systems is possible, we can not say that there is a physical implementation of such system that will not dissipate.

Moreover, where in mathematics a simple demonstration is sufficient, in physics the only proof of the feasibility of an implementation is to build a functional realisation of it.

It is in this context that Fredkin and Toffoli introduce the Billiard Ball Model (BBM) as a physical implementation to their reversible elements, in order to give consistency to their demonstration.

We will not explain such system, however curious readers can found detailed explanation in [5].

This physical model remains an idealization and is still waiting for a functional realisation. Debates around the implementation of such system can be found in [7].

1.3 Adiabatic circuits

Precisely defining the term "adiabatic" in the context of hardware computing is sometimes difficult and may be confusing.

Originally in thermodynamics, adiabatic systems are those who are heat conservative, which means that there is no heat exchange between the system and the environment.

Also, the word adiabatic refers to thermodynamically reversible or near-reversible systems, those who do not generate new entropy (isentropic).

Weirdly, these two definitions can lead to completely opposite interpretations: a conservative system might not be a reversible one and vice versa.

In [4], adiabatic circuits are regarded as those who can generate asymptotically zero entropy per operation while imposing some limits such as low speed or low temperature.

Throughout this dissertation, we join the last definition that we explain the principle in the next paragraph.

1.3.1 Principle : quasistatic switching

As is widely known, conventional CMOS gates dissipate exactly CV_{DD}^2 to charge the node capacitance C.

In [4], Younis affirmed that such dissipation is due to a quick charge transfer between the supply voltage and the capacitance.

We add that this is especially due to the voltage difference across the resistance. Hence, the basic principles of adiabatic circuits are :

– perform quasistatic switching;

– never turn on any device as long as there is any voltage difference across it.

To explain how quasistatic charge transfer can reduce the energy consumption, let us consider the model in Figure 1.5, where the R is the ON resistance of the gate. Let us consider also that the load C is initially discharged.

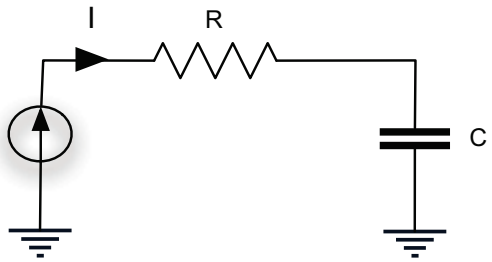


Figure 1.5: Basic RC model of CMOS gate.

Consider a constant-current source, the current I needed to charge C from 0 to V_{DD} in a time T is such that:

$$I = \frac{Q}{T} = \frac{CV_{DD}}{T} \tag{1.4}$$

Therefore, the energy dissipated is :

$$E_{diss} = \int (P)dt = R \int (I^2)dt = \frac{RC}{T}\ CV_{DD}^2 \tag{1.5}$$

We observe that the energy dissipation decreases as the charging time increases. So, if the charging process takes a large time ($T \gg RC$), the circuit can consume asymptotically

zero energy.

Actually, the constant-current source can be closely approached by a voltage ramp [4]:

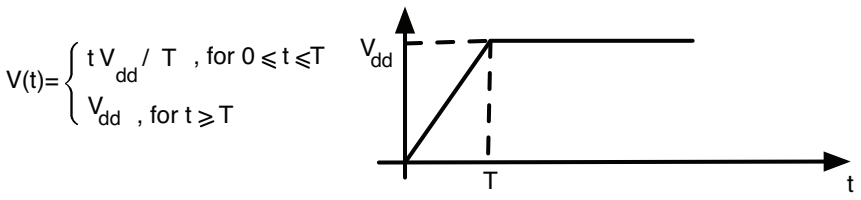


Figure 1.6: Voltage ramp to model constant current source.

For $T \gg RC$, energy dissipation is approximately given by Equation 1.5 , however when $T \ll RC$, the dissipation is that of ordinary circuits with constant supply voltage.

For more detailed analysis about voltage ramp and generation of voltage ramp see [4].

Note however, as mentioned earlier, quasistatic operation is not sufficient to guarantee asymptotically low dissipation. This later is only maintained if this technique is ultimately used for both charging and discharging of all nodes. This can only be possible if the previous value of the node is known in advance, which means in technical words "logical reversibility".

This affirms Equation 1.1 where no physical reversibility is possible if logical reversibility is not achieved. And this is why, to build adiabatic circuits, the second basic principle is to never turn on any device as long as there is any voltage difference across it, as connecting a node to another one of different voltages will throw away logical information.

One might note that starting from a purely physical reasoning, we land always in front of logical reversibility issues corresponding to the convergence in a FSM as shown in Figure 1.7.

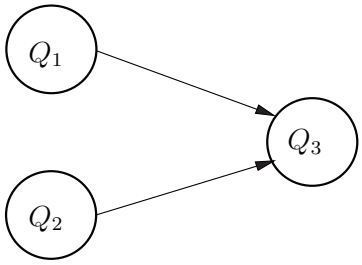


Figure 1.7: A convergence in an FSM.

In [4], Younis showed a method to connect gates in order to respect the second principle and achieves minimum energy consumption. However, like for Fredkin and Toffoli demonstration, such implementation are well convincing only when dealing with combinatorial

circuits. For sequential elements with "feedback", those techniques remain insufficient since a stable elementary bit-memory is traditionally obtained with a feedback on an amplifier. We believe that the question is not yet resolved and further researches in this field are needed.

1.3.2 The SCRL technique

SCRL is a technique proposed by Younis in [4] to build adiabatic circuits using standard CMOS technology. This technique is divided into two major tasks :

- implementing the appropriate components known as : SCRL gate;
- connecting these components in a non-dissipative way : SCRL pipeline.

1.3.2.1 The SCRL gate

The SCRL gate is simply derived from the conventional CMOS gate, except that the constant voltage supply rails (V_{DD}, GND) are replaced with swinging clock rails and a pass-gate is added at the output.

Figure 1.8 illustrates the SCRL gate for the 2 inputs logic NAND function. The p-FET network and the complementary n-FET network are those of an ordinary CMOS gate. Initially, the supply rails $(\phi, \bar{\phi})$ and all internal nodes are at $V_{DD}/2$. The control signal of the pass gate P_1 and $\bar{P}_1$ are at GND and V_{DD}, respectively, which means that the pass gate is turned off. After the inputs are set to either 0 or V_{DD}, representing respectively the logic 0 and 1, the control signal of the pass gate P_1 and $\bar{P}_1$ swing gradually to V_{DD} and GND to turn on the pass gate (step (1) in the figure). Now, the supply rail ϕ swings to V_{DD} and its logic inverse $\bar{\phi}$ swings to GND (step (2) in the figure), allowing the generation of the output signal (step (3) in the figure). Then, once the output is used by another gate in the SCRL pipeline, the inverse procedure start to reset the gate. Hence, the pass gate is turned off, then, the supply rails return to the initial level $V_{DD}/2$ and all internal nodes are restored to $V_{DD}/2$. The inputs can now change as there is no difference voltage across transistors which ensure the basic principle of asymptotically zero dissipation.

Note however, that the output is not yet restored to $V_{DD}/2$, and that to prevent dissipation, due to the potential difference, the pass gate should be kept turned off. Indeed, the reset of the output is ensured by another gate in the SCRL pipeline which will be explained in the next section.

1.3.2.2 The SCRL pipeline

The SCRL pipeline is a method to connect SCRL gates in order to restore the level of outputs in a non-dissipative way.

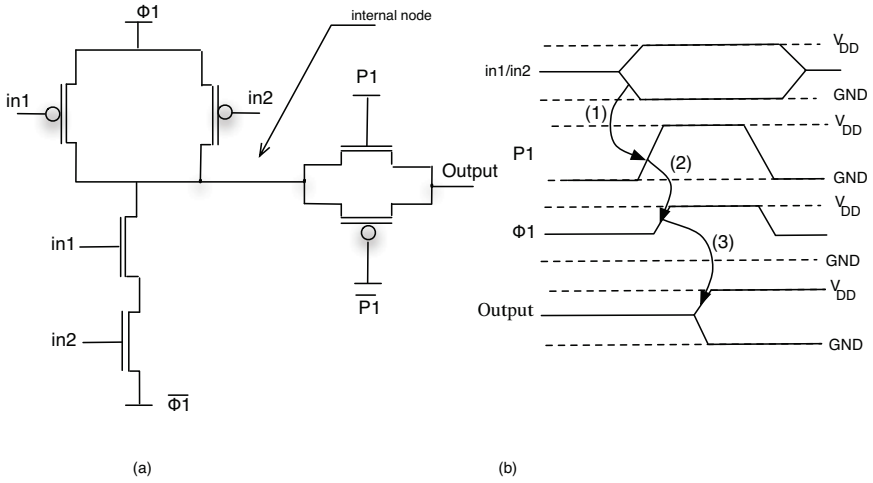


Figure 1.8: (a) The SCRL gate of NAND function (b) timing rails.

Figure 1.9 illustrates the structure of SCRL pipeline where F_i is the function realised by the gate and F_i^{-1} is its inverse function. In fact, it's the bottom half of the pipeline that performs the resetting of SCRL gate outputs when doing inverse computation.

The basic operational events are as follows : by turning on P_1, F_1 computes the output $F_1(a_0)$. Similarly, F_2 computes the output $F_2(F_1(a_0))$. Hence, F_2^{-1} produces $F_2^{-1}(F_2(F_1(a_0)))$ at the internal node of the SCRL gate F_2. Now, P_1 can be turned off to hand off node(a) to F_2^{-1} from F_1. Then, F_1 is restored by restoring ϕ_1 and so does F_2. Hence, F_2^{-1} restores node(a) to $V_{DD}/2$.

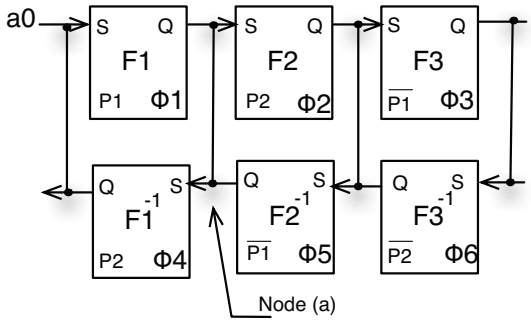


Figure 1.9: Structure of SCRL pipeline.

1.4 Conclusion

There has been many attempts to find a physical implementation of reversible circuits. Among them, the Billiard Ball Model, introduced by Fredkin in [5] to show that reversible

boolean logic gates can possibly be envisaged.

Currently, researchers forecast a promising future for reversible computing using quantum computation. For the moment, this field have achieved considerable results in theoretical computer science. However, it does not bring to us solution concerning dissipationless computing.

Nowadays, reversible logic is considered to be an important approach for low power design. Hence, many works have emerged in order to implement reversible logic gates in actual technologies such CMOS, pass-transistor [8] [9]. However, all these works forget that the practical objective is to build dissipationless elements and not just logically reversible ones. We believe that the fundamental question remains interesting and worth to be studied.

It is by the way remarkable to note that looking for the proof of the feasability or the infeasibility of dissipationless computing, we find more arguments of its possibility. To summarize, there is no proof for the necessity of dissipation in computing, but, on the otherhand, no real implementation is built that do not dissipate or that dissipate proportionally to the output size.

We might probably admit that we have not understand all and that something is missing on the understanding of dissipation of computation.

Personally, i choose to concentrate myself on questions that can have a chance to have a concrete application in the short term.

At this point, we leave reversible computing and start to look for low power design by focusing on current technologies especially CMOS one.

Chapter 2

CMOS circuits : Power dissipation, estimation and optimization for Low Energy Design

2.1 Introduction

Today, CMOS (Complementary Metal-Oxide Semiconductor) technology is clearly the most used technology due to its robustness and low power dissipation. It is also simple to construct : each CMOS logic gate consists of two complementary PMOS and NMOS networks that are connected trough the gate output as shown in Figure 2.1 for a CMOS NAND gate.

When the PMOS network is conducting, the NMOS network has a high resistance state such that ideally no currents flow between the output and the ground, and vice versa. This is why CMOS logic gates are supposed to consume only during switching operations and have no static power dissipation. However, MOS transistor is far from an ideal switch and there is always leakage currents. Earlier, this type of currents were not considered as they have small amount compared to the overall power dissipation. Until recently with sub-nanometer technologies, leakage currents become the major part of power dissipation.

As technology scales down, power dissipation has become a critical limit to design CMOS circuits. Hence, techniques for low power design are more and more needed. Specifically, there is a need for power estimation and optimization techniques that help reaching an optimal solution targeting low power design. An efficient design flow for low power design will include a ”design improvement loop” at each level of abstraction [10]. For each loop, a power estimator predicts the power dissipated by various design with different optimization options, searching for the potentially more effective solution in terms of power

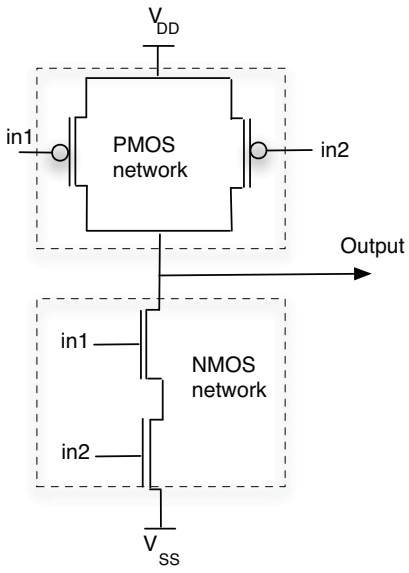


Figure 2.1: CMOS NAND gate.

dissipation. Recently, a wide variety of power estimation and optimization techniques have emerged to meet design goals. They are in all abstraction levels starting from the software-level to the transistor-level.

In this chapter, we survey state-of-the-art power dissipation, power estimation and power optimization techniques in static CMOS circuits. So, this chapter is divided in three parts. In the first one, the fundamental physical mechanisms of dynamic and leakage components of power consumption in CMOS circuit are reviewed. In the second part, a brief look at power estimation techniques at different abstraction levels is provided. And the last part is devoted to survey dynamic and leakage power reduction techniques.

2.2 Power dissipation in static CMOS circuits

Power dissipation in CMOS circuits is caused by three sources that are either static or dynamic. The static power consumption is due to leakage currents that flow when transistors are in OFF-state. The dynamic power consumption include short circuit power dissipation caused by currents which flow between the supply rails during output transitions and capacitive power dissipation required to charge and discharge parasitic capacitances in the circuit during logic transitions. Let us explain each component according to their physical origin.

2.2.1 Capacitive power dissipation

Dynamic capacitive power consumption represents the power consumed during logic transitions to charge and discharge parasitic capacitances in the circuit.

To illustrate the contribution of the dynamic dissipation, consider the structure of a generic CMOS gate shown in Figure 2.2.

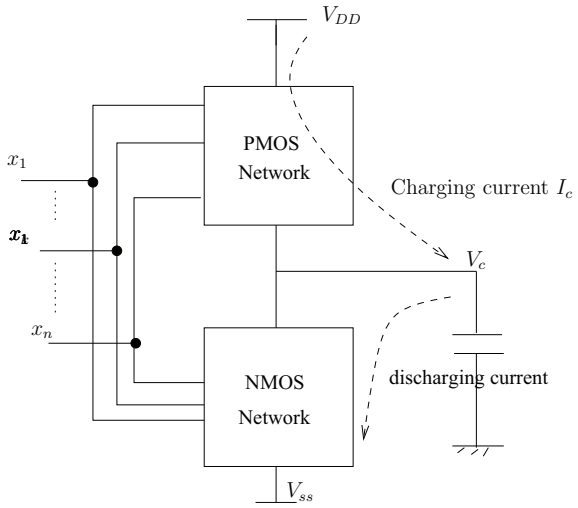


Figure 2.2: Structure of a generic CMOS gate.

Consider inputs combination in such a way output make a low-to-high transition. The capacitance C_L on the output incurs a voltage change $\triangle V_c$ drawing a current from the supply voltage given by :

$$I_c(t) = C_L \frac{dV_c}{dt} \tag{2.1}$$

The energy drawn from the supply voltage is therefore given by :

$$E_{Tot} = \int_0^{t_1} V_{DD} I_c(t) dt \tag{2.2}$$

Where t_1 denotes the time necessary to fully charge the output load C_L initially discharged. By substituting Equation 2.1 into Equation 2.2, the energy consumed during a low-to-high transition is as follows :

$$E_{Tot} = V_{DD} C_L \int_0^{t_1} \frac{dV_c}{dt} dt = V_{DD} C_L \int_0^{V_{DD}} dV = C_L V_{DD}^2 \tag{2.3}$$

Half of this energy is stored in the load capacitor C_L as given by the following expression:

$$E_c = C_L \int_0^{t_1} V_c(t) \frac{dV_c}{dt} dt = C_L \int_0^{V_{DD}} V_c(t) dV = \frac{1}{2} C_L V_{DD}^2 \tag{2.4}$$

And the second half is dissipated as heat energy in the PMOS network.

During the discharging phase, the voltage across the capacitance terminals swing from V_{DD} to 0 and the previously stored energy is dissipated in the NMOS network.

Thus, every output transition (high-to-low or low-to-high) results in an energy dissipation of $\frac{1}{2} C_L V_{DD}^2$.

Considering the switching activity a of the output, which is defined as the probability that the output makes a transition at each clock cycle, the total capacitive dissipation can be given by :

$$P_{dyn} = \frac{1}{2} . C_L . V_{DD}^2 . a . f \tag{2.5}$$

Where, f denotes the operating frequency.

C_L represents the equivalent capacitance of all internal parasitic capacitances which leads to consume the same power. These capacitances include, generally, MOS transistor capacitors and interconnect capacitance. Figure 2.3 shows parasitic capacitances of MOS transistor, where C_{gs}, C_{gd}, C_{bs} , C_{gb} represents intrinsic capacitance and C_{bd} and C_{bs} include the intrinsic and extrinsic capacitance terms. The intrinsic capacitances depends on the operating point of the transistor. C_{ovgs}, C_{ovgd} and C_{ovgb} are extrinsic capacitances resulting from the overlap between the gate and respectively the source, drain or bulk regions.

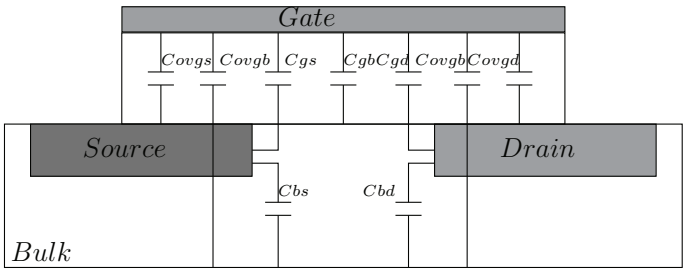


Figure 2.3: Parasitic capacitances of MOS transistor.

All of these parasitic capacitances are proportional to the channel width of the transistor.

2.2.2 Short circuit power dissipation

As the input rise/fall time is not null, there is a short period of time where the input voltage is between V_{tn} and $V_{DD} - |V_{tp}|$ (V_{tn} and V_{tp} are the threshold voltages of NMOS

and PMOS transistors).

Thus, both PMOS and NMOS networks are ON, resulting in a conductive path between V_{DD} and GND and currents which flow directly between the supply rails.

Figure 2.4 illustrates this component through the example of a CMOS inverter.

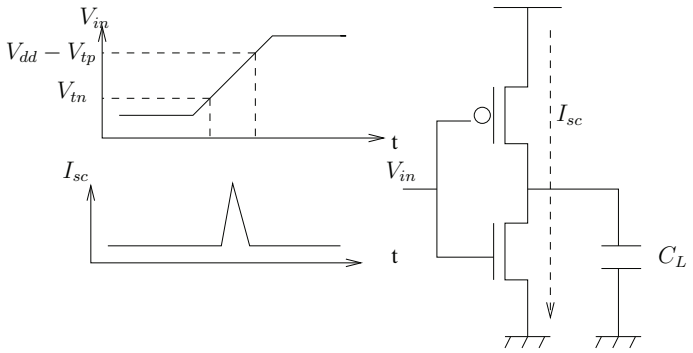


Figure 2.4: Short circuit power dissipation.

The short circuit current depends on the input slope, the output load and the gate transistors size. A well-known model for short circuit power dissipation is given in [11]:

$$P_{sc} = \frac{1}{12} k\tau (V_{DD} - 2V_t)^3 f\alpha \tag{2.6}$$

Where τ is the input transition time, V_t is the threshold voltage of transistors, α is the gain factor of transistors and k is the effective transconductance parameter of the logic gate.

Even if the short circuit power is a part of dynamic power consumption, in the literature the term dynamic denotes generally the sole capacitive power consumption. Mainly because the power consumed by the short-circuit currents is typically less than 7% of the total dynamic power [12].

2.2.3 Glitch power dissipation

In synchronous circuits, the activity is related to a clock signal. Normally, each net will at most have one transition per clock cycle. Unfortunately, some nets could have multiple transitions within the same clock cycle because they pass through intermediate computation states.

These unnecessary transitions known as glitches are generated by a logic gate when its input signals arrive at a different time due to paths of different delays.

Figure 2.5 shows an example with glitch appearance.

At t_0, the input C and the output O1 are both at the logic value 1 which allow the output O2 to make a (low-to-high) transition after 1 unit-delay corresponding to the inertial

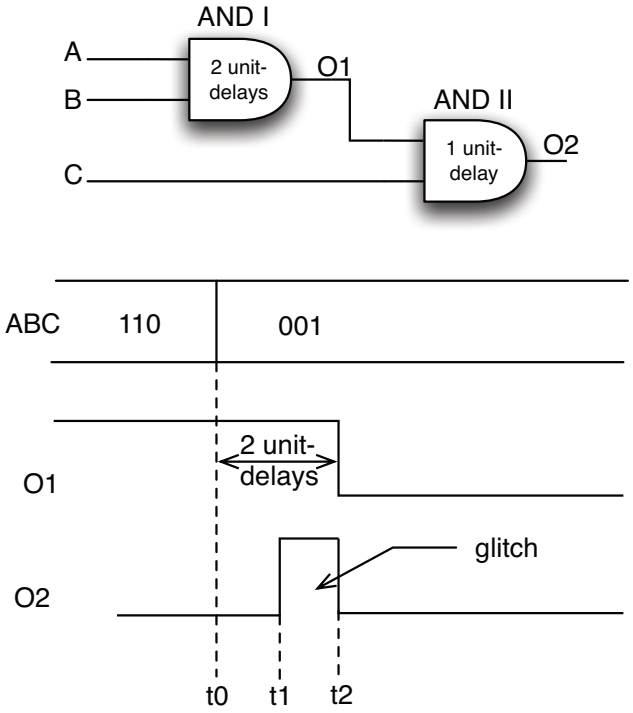


Figure 2.5: Example of glitch appearance.

delay of the gate ANDII. However, as the output O1 is delayed by 2 unit-delays compared to the input C, another transition of the output O2 takes place at t_2. The two extra transitions of the output O2 are unnecessary and represent the so called glitch.

By increasing the switching activity of the circuit, glitches add a dynamic component to the power dissipation.

2.2.4 Leakage power dissipation

Leakage power consumption, also called static power, is due to the leakage currents that flow in static conditions where all nodes in the circuit remain in a steady state. Possible leakage sources for a NMOS transistor are illustrated in figure 2.6. They include subthreshold leakage, gate oxide tunneling leakage and reverse-biased junction leakage [13].

Subthreshold leakage Subthreshold leakage current is the weak inversion current which flows between the source and the drain when the transistor operates with a gate voltage below the threshold voltage V_t. It can be modeled as follows [14]:

$$I_{DS} = 2n\mu C_{ox} U_T^2 \frac{W}{L} exp\frac{-V_t}{nU_T}(1 - exp\frac{-V_{DD}}{U_T}) \tag{2.7}$$

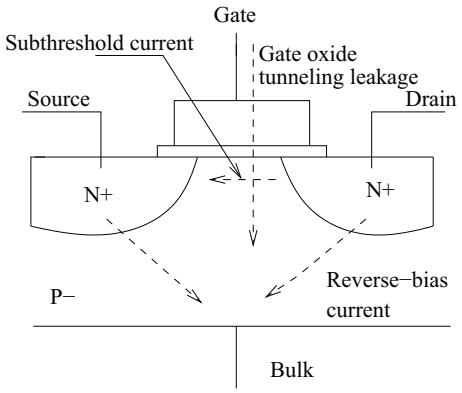


Figure 2.6: Leakage components in a MOS transistor.

Where n is the subthreshold slope factor, μ is the mobility, C_{ox} is the oxide capacitance, U_T is the thermal voltage and W/L denotes the channel width-length ratio of the transistor.

This equation shows the exponential dependency of subthreshold leakage on V_t and V_{DD}. So, if V_{DD} and V_t are scaled down subthreshold leakage power exponentially increases [14].

Also, this equation shows the inverse proportional relation between the channel length and the subthreshold current. These parameters will be exploited by leakage reduction techniques to achieve subthreshold leakage optimization.

Gate-oxide tunneling leakage Gate-oxide tunneling leakage is directly related to the oxide thickness t_{ox}. As technology scales down, the oxide thickness decreases resulting in the presence of a high electric field across the gate oxide. As a consequence, a tunneling of electrons can occurs leading to a current flow from or to the gate terminal [15].

A model of the direct tunneling current can be found in [16]. It shows especially the exponential dependency of current tunneling on oxide thickness.

Reverse-biased junction leakage The drain-bulk and source-bulk junctions are typically reverse biased when transistors are in OFF-state. Reverse bias means that the voltage applied to the N-type is higher than that applied to the P-type. Normally with such configuration, no currents will flow between the p–n junction. However, large reverse bias voltage results in a large depletion region causing minimal current flow across the p–n junction. And if the field across the reverse biased p–n junction exceed a critical limit, the depletion junction can break down resulting in a significant tunneling of electrons through this junction.

2.2.5 Discussion

As technology scales down, power consumption and especially leakage component is dramatically increasing. According to ITRS predictions [17], for technology nodes from 90nm to 45nm, the total power is rather determined by the dynamic component. While for 32nm and bellow, the leakage component has become a non-negligible and dominating part of the total power consumption.

Recently, new solutions in a technological level have been proposed to deal with leakage component issues. For instance, high-K dielectric is applied to moderate gate-oxide tunneling effect.

2.3 Power estimation techniques

There has been a significant research interest in power estimation techniques. Detailed surveys of existing techniques have been presented by Najm in [18] and Pedram in [10].

Power estimation techniques can be applied at different levels of the design flow and they differ especially in the accuracy and the time needed to estimate the power in the circuit.

Obviously, the more accurate the power estimation result is, the larger the time consumed is. Figure 2.7 illustrates the design space of power estimation starting from the software-level to the transistor-level.

Software-level power estimation is beyond the scope of this brief discussion but can be found in [10]. For other levels, we review the estimation concept in each level, although we use just the lowest transistor-level estimation as it is the most accurate one. Indeed, we have a need for a precise estimation of the power consumption as will be explained in Chapter 3.

In the next section, a brief review of different power estimation techniques at different abstraction levels is provided.

2.3.1 Behavioral-level power estimation

In behavioral level, a high-level description of system's behavior is given. The real challenge is to predict physical capacitance and switching activity of a circuit that is not yet determined. So, information about gate-level and RT-level components are unknown. In [19], authors propose to use information theoretic model, where entropy is used as a measure to estimate the total capacitance given the number of circuit inputs and outputs. Other works [20] [21] derive a power estimation model based on the circuit complexity in the behavioral description such as the number of states in an FSM machine or the number

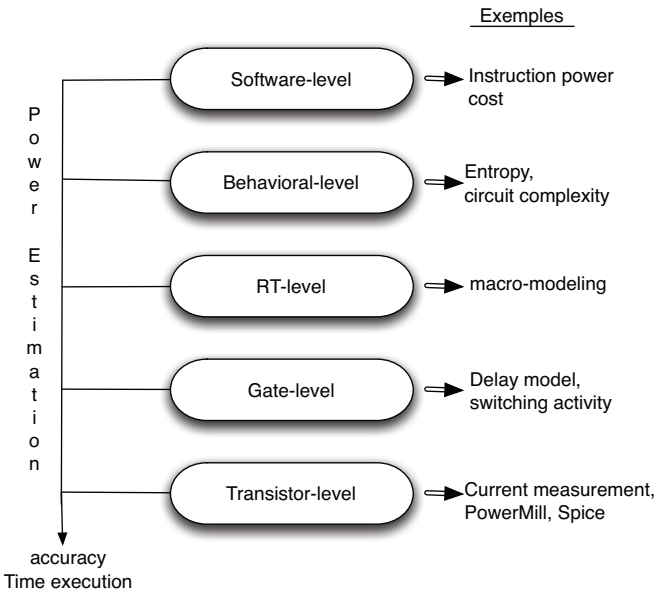


Figure 2.7: Abstraction levels of power estimation.

of arithmetic or boolean operations. Another approach for behavioral power estimation is to associate the high-level description to a RTL netlist using a quick synthesis program and to use after that any RTL power estimation techniques. Detailed analysis on how to obtain such netlist is beyond the scope of this discussion but it can be found in [10].

2.3.2 RT-level power estimation

In RT-level, the circuit is composed of combinatorial blocks such as adders, multiplexers, multipliers... and sequential blocs as registers, latches and memories.

Generally, RTL power estimation techniques use characterization based macro-modeling approach. The idea is to characterize the module by simulating it at a lower level implementation (gate or transistor level) under various input sequences. Based in this data, a multi-variable model which describes the power consumption of the module as a function of various parameters, is constructed.

The parameters of the macro-model can be the number of inputs, the input bit width and the supply voltage. In [10], Pedram presents many examples of previous macro-modeling RTL power estimation.

2.3.3 Gate-level power estimation

Gate-level power estimation is based on the measurement of the switching activity in every node of the circuit during a gate-level analysis. A typical gate-level design flow is illustrated in Figure 2.8.

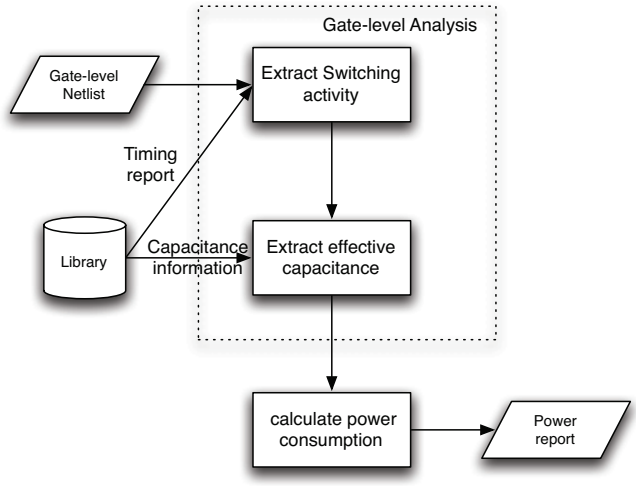


Figure 2.8: Design flow of gate-level power estimation.

In each node of the circuit, the switching activity is estimated using probabilistic approach.

To explain such approach, let us consider a logic signal x for which we define $prob(x)$ and $prob(\bar{x})$, the probability of x being equal 1 or 0.

Transition probabilities are the probability of x making a 0 to 1 or 1 to 0 transition, staying at 0 or staying at 1 between two times instants. We will represent these probabilities as $prob^{01}(x)$, $prob^{10}(x)$, $prob^{00}(x)$ and $prob^{11}(x)$ respectively. Assuming that consecutive input vectors are independent, which is generally referred as "temporal independence" assumption, then:

$$\begin{aligned} prob^{11}(x) &= prob(x) \cdot prob(x) \\ prob^{10}(x) &= prob(x) \cdot prob(\bar{x}) \\ prob^{01}(x) &= prob(\bar{x}) \cdot prob(x) \\ prob^{00}(x) &= prob(\bar{x}) \cdot prob(\bar{x}) \end{aligned} \tag{2.8}$$

The switching activity of signal x defined as the probability that signal x makes a transition is therefore given by:

$$\begin{aligned} E(x) &= prob^{01}(x) + prob^{10}(x) = 2\,prob(x) \cdot prob(\bar{x}) \\ &= 2\,prob(x) \cdot (1 - prob(x)) \end{aligned} \tag{2.9}$$

Given transition probabilities of the primary inputs, probabilistic methods propagate

this information through the logic circuit to determine transition probabilities and hence the switching activity at each node in the circuit.

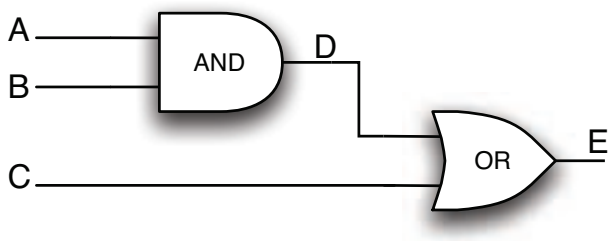


Figure 2.9: Example to explain probabilistic methods to determine switching activities.

Take for instance the example in Figure 2.9, the output (D) of the AND gate is at the logic value 1 if only its two inputs (A,B) have the logic value 1. Hence, the probability of D being 1 is as follows:

$$prob(D) = prob(A) \cdot prob(B) \tag{2.10}$$

The switching activity of the node D can be expressed as a function of inputs probabilities as follows:

$$E(D) = 2prob(A) \cdot prob(B)(1 - prob(A) \cdot prob(B)) \tag{2.11}$$

Once the switching activities are computed for all the design nets, the power can be evaluated using Equation 2.5.

Note that in this discussion, the gates are assumed to have zero delay. In that case, glitches are not taken into account and the dynamic power is under-estimated.

To generate an accurate estimation result where glitch contribution is considered, a real delay model is needed. This will be further explained in Chapter 3 where we will show how these two simulation mode "zero delay" and "real delay" are used to calculate glitches power contribution.

2.3.4 Transistor-level power estimation

At the transistor level, the average current drawn from the power supply during the transistor level simulation is calculated to estimate the power consumed.

The average power is therefore the multiplication of the estimated average current and the supply voltage. Although this approach can be applied to any circuit synthesized in any technology and performing any functionality, it is strongly pattern-dependent [22],

which means that the result depend strongly on the set of inputs vectors used during the simulation.

Generally, a large numbers of randomly generated inputs patterns are used for simulation in order to achieve more accurate result. This method is well-known as Monte Carlo simulation and unfortunately, it is computationally very expensive.

Note that the most accurate result of power estimation is provided at this level. But the execution time for such electrical simulations can be very long, which makes of the estimators in this level not practical for large circuits.

Spice is among the most known simulator at the transistor level.

2.4 Techniques for low power consumption

Optimization techniques for low power dissipation have recently emerged at all levels of the design hierarchy [23]. Generally, these techniques concentrate on the reduction of the capacitive and leakage power dissipation, since the short circuit contribution is typically less than 7% of the dynamic power consumption [12].

Most of recently reported surveys classify optimization techniques according to the abstraction level where they are applied. Figure 2.10 illustrates this approach and gives some examples of techniques at each level.

In this section, low power techniques will be arranged depending on the consumption component that we want to reduce as depicted in Figure 2.11.

First, we survey state-of-the art dynamic and leakage power reduction techniques.

2.4.1 Dynamic Power reduction techniques

As shown in Equation 2.5, dynamic power consumption is proportional to the load capacitance, to the square of the supply voltage V_{DD}, to the switching activity and to the clock frequency.

Consequently, power reduction techniques will attempt to reduce each parameter or a combination of them.

2.4.1.1 Voltage scaling techniques

As P_{dyn} is proportional to the square of V_{DD}, a small reduction in supply voltage causes quadratic decrease in dynamic power consumption. However, reducing the supply voltage can affect the system's performance since the delay is inversely proportional to $(V_{DD} - V_t)$ as illustrated in the following expression [24].

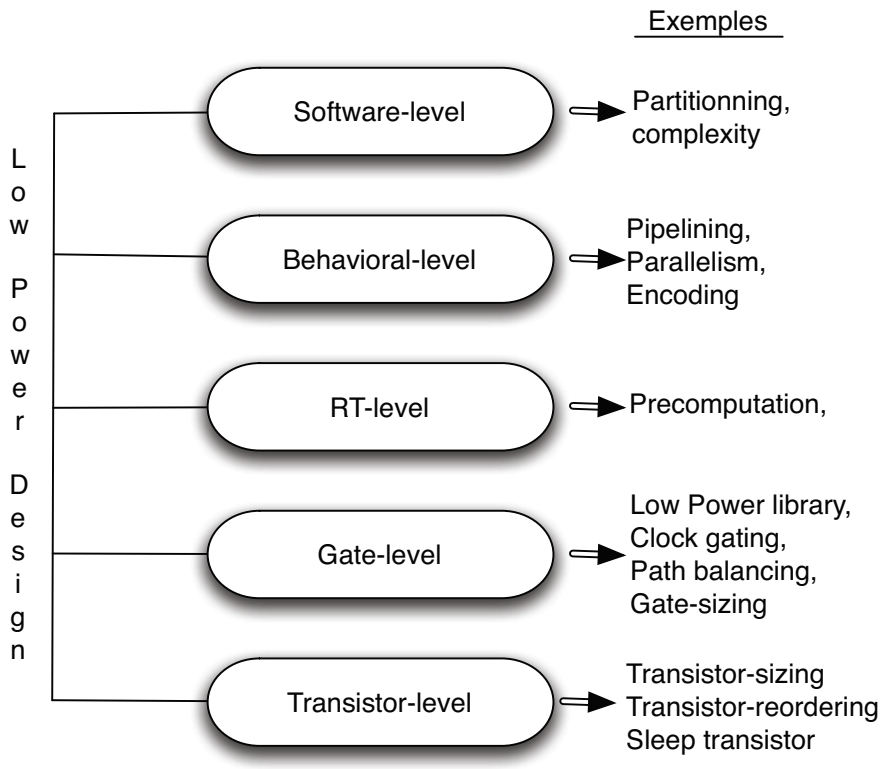


Figure 2.10: Low power design space.

$$t_{pd} = \frac{KV_{DD}}{(V_{DD} - V_t)^{\alpha}} \tag{2.12}$$

where K is a constant of proportionality, V_t is the threshold voltage of the transistor and α is the velocity saturation index. To preserve the system's performance while using lower supply voltages, three main approaches are used [25].

Pipelining Pipelining is among the first proposed low power techniques [26]. As depicted in Figure 2.12, pipelining consists in the insertion of storage elements in the circuit datapath to compose smaller combinatorial sub-blocks, where the output of one block is used as the input of the next one.

Usually, pipelined structures are used to maximize the throughput. To reduce power consumption, pipelining helps working at a reduced supply voltage while maintaining the same operating frequency as sub-blocks have shorter critical paths. Pipelining is also useful for glitch power reduction, where the storage element barrier breaks the propagation of the glitches.

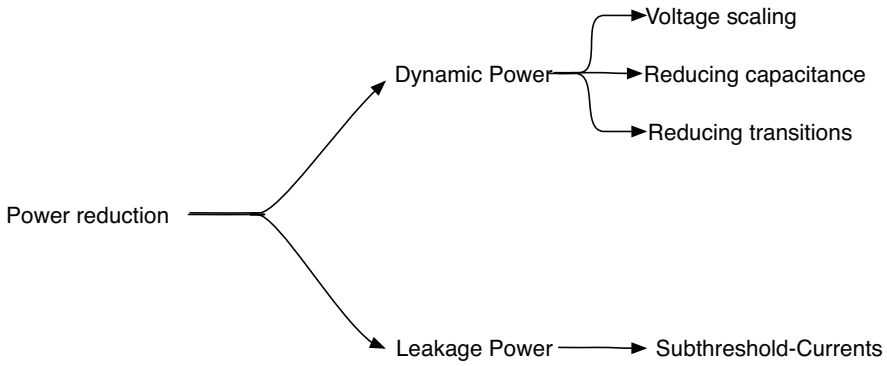


Figure 2.11: Techniques for low power design.

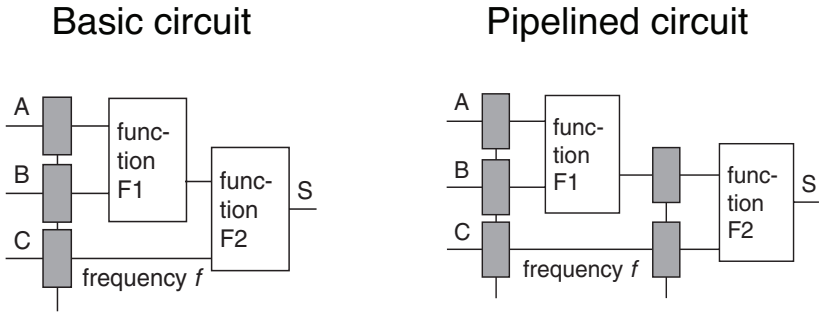


Figure 2.12: Pipelining concept.

Dual supply voltage approach It consist in using two different supply voltages (V_{DDH} and V_{DDL}). The lower supply voltage V_{DDL} is assigned to gates that can run at lower frequency without affecting the circuit performance.

An output of V_{DDH} cell can be connected to an input of V_{DDL} ones, however, an output of V_{DDL} cell cannot be directly fed to V_{DDH} ones. Indeed, the rise and fall times of V_{DDH} supplied cells can be severely degraded with an under-driven signals at its inputs. And a slower transition means an increase in the short circuit current and a reduction in the noise margins.

To avoid this problem, a voltage level conversion (LC) cell is inserted between outputs of V_{DDL} supplied cells and inputs of V_{DDH} supplied ones.

Clustered Voltage Scaling (CVS) is a well-konwn structure proposed in [27] to implement the dual-V_{DD} approach while minimizing the numbers of Level converters needed.

In this technique, cells of the circuit should be placed in this particular order: Primary inputs $\dashrightarrow$ V_{DDH} gates $\dashrightarrow$ V_{DDL} gates $\dashrightarrow$ level converters $\dashrightarrow$ primary outputs.

To determine the V_{DDL} supplied cells, cells are traversed in a topological order from the primary outputs to the primary inputs. Initially, all cells in the circuit are V_{DDH} cells,

then, each visited cell is replaced by a V_{DDL} cell as long as the timing constraints are respected.

Supply and threshold voltage adjustment approach To maintain circuit performance when reducing the supply voltage, threshold voltage should be reduced too. However, a reduced threshold voltage results in an increase on subthreshold leakage current. Kao in [24], developed a theoretical model that determine the optimal operating point (V_{DD},V_t) that minimizes the total active power consumption.

It should be noted that for traditionnal CMOS circuits V_{DD} is always beyond V_t. However, we will see in Chapter 4, with sub-threshold operation, how working with a supply voltage bellow the threshold voltage is an efficient solution for Ultra-Low-Energy applications.

2.4.1.2 Reducing the output capacitance

The output load capacitance includes all parasitic capacitances of existing MOS transistors in the gate. Techniques to reduce this term belong generally to transistor-level techniques.

Transistor sizing Transistor sizing technique consist in setting the transistor channel width to a width that achieves minimum power consumption.

A reduction of the size of the transistor results in a decrease in the capacitive power consumption but increases the signal rise/fall times at the gate output and thus penalise the timing performance.

In [28], Borah developed an algorithm which looks for the optimal transistor size that minimizes power dissipation while still preserving a given delay constraints. The idea is to start with minimum transistor sizes for all the gates in the circuit. Then, if the required delay constraint is satisfied, the process is terminated. Else, transistors of the gates on-critical path are re-sized such that delay constraints are satisfied.

Transistor reordering In a serially connected MOSFET chain, the inputs can be permuted without any change in the output values. Transistor reordering techniques exploit this freedom to optimize propagation delays and/or power dissipation of the gate.

In [29], [30] and [31], transistor reordering schemes are developed to find the best configuration that optimize power consumption in the gate. Generally, inputs with high switching activity should be placed closest to the output terminal [31].

Figure 2.13 illustrates an example where various combination of pin reordering in the NMOS-network of a 4-input NAND gate are tested. It shows 80% leakage current decrease

just by putting inputs in a judicious order.

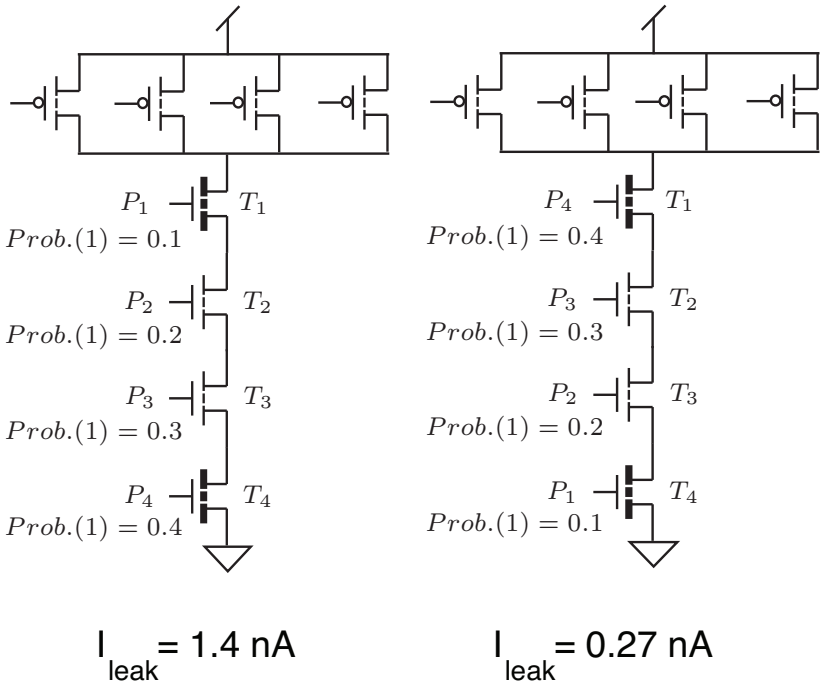


Figure 2.13: Example taken from [1] to illustrate transistor reordering.

2.4.1.3 Reducing switching activity

Optimization techniques which target switching activity reduction are generally gate-level techniques.

Clock gating Clock gating is one of the most well-known and widely used low-power technique. It consists in disabling the clock feeding a sub-circuits that are in the idle mode. Hence, the transitions related to the clock tree and to the logic fed by the disabled Flip-Flops are suppressed, achieving capacitive power savings.

The concept of the clock gating is illustrated in Figure 2.14(a). A clock-gate signal (CLKG) is derived from the main clock signal (CLK). It is controlled by an external signal (CTRL) and turned off to inhibit capacitance charge/discharge when the functional unit is not used.

A simple implementation of the clock-gating (CG) function is given in Figure 2.14(b), where the CG block is just an AND gate.

Today, all design almost benefit from clock gating approach. Indeed, it becomes simple and easy to implement : specific clock gating cells, generally included in the library, are

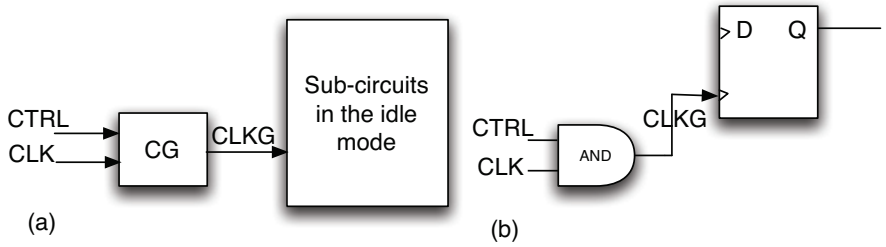


Figure 2.14: Clock gating: (a) Principle, (b) AND implementation .

automatically inserted without the requirement to change RTL circuit description [32].

Encoding Encoding techniques consist in choosing a judicious encoding for State Transition Graphs that minimize the average number of signal transitions on the state lines.

For example grey code counter, where only one bit changes for two successive states, can be used to minimize switching activity.

Bus encoding can also be considered. It consists in transforming the information transferred on the bus to a form having a lower transition activity than the original form. For example, if the value `0000` is previously transferred and the current value to be transferred is `1011`, then its complement, the value `0100`, is transferred instead. An extra line is added to know if the output needs to be complemented.

Precomputation This optimization method was first proposed by Devadas in [33]. The idea is to selectively precompute the output logic values one clock cycle before they are required and use the precomputed values to reduce switching activity in the next clock cycle.

Figure 2.15 shows a general precomputation structure, where g_1 and g_2 are the precomputation logic that predict the output values for a subset of inputs. If the prediction is possible, the original circuit is turned off in the next clock cycle, reducing, consequently the switching activity. To achieve power savings, the additional power consumed by the precomputation logic should not be greater than that consumed by the original circuit.

2.4.2 Leakage Power reduction techniques

Because leakage currents are dramatically increasing with technology scaling and because components may often be in an idle state, methods to reduce leakage power had been developed and will continue to be an interesting research area in deep submicron technologies.

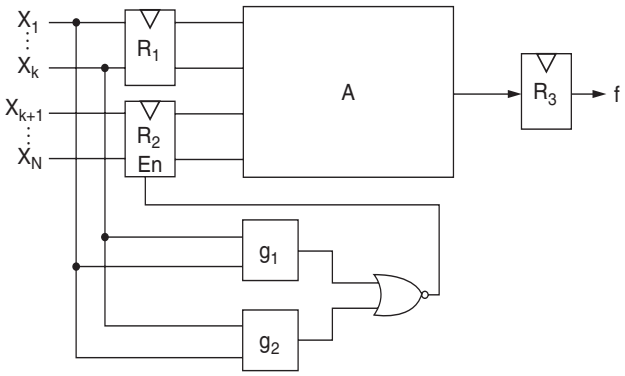


Figure 2.15: General precomputation structure.

An obvious idea to reduce standby leakage currents has been to put the system or parts of the system in a low leakage mode when they are not needed.

Another approach is to use Multi-threshold voltage in a single chip, where high V_t transistors are used to suppress subthreshold leakage currents and low V_t ones are used needing to maintain high performance.

In this section, we review the following leakage power reduction techniques : Dual-V_t, Variable-V_t and sleep transistor technique.

Dual-threshold CMOS Dual-threshold CMOS (DTCMOS) technique is a well-known technique in which high V_t are assigned to transistors in noncritical paths to reduce leakage current, while maintaining the circuit performance by using low V_t transistors in critical paths [34]. In [35], Wei demonstrated that DTCMOS technique can achieve 68% of leakage power reduction without any delay or area increase. Many heuristic algorithms and linear programming models have been proposed to find an optimal solution to Dual-V_t selection [36] . The objective function is to use the largest possible number of high V_t gates while the critical delay remains within the timing constraints.

Other works have proposed to use gate upsizing to improve performance while using high V_t transistors [37]. However, upsizing will increase the switching power by increasing the parasitic capacitances. A trade–off against leakage power increase when using low V_t transistors have to be found.

Using multiple threshold voltage in the same circuit is very common for 45nm and below circuit design. Usually, an initial synthesis with high threshold voltage that ensures a high performance is performed. Then, cells off-critical paths are swapped to operate in lower leakage mode.

Variable-threshold CMOS To reduce leakage current, variable threshold CMOS (VTCMOS) technique uses high V_t in standby mode, while standard V_t is used in active mode to preserve the system's performance.

This can be achieved using a substrate bias control circuit that fix the body bias. For instance, a reverse body bias is applied in standby mode in order to reach higher threshold voltage with lower leakage currents. Figure 2.16 illustrates VTCMOS techniques.

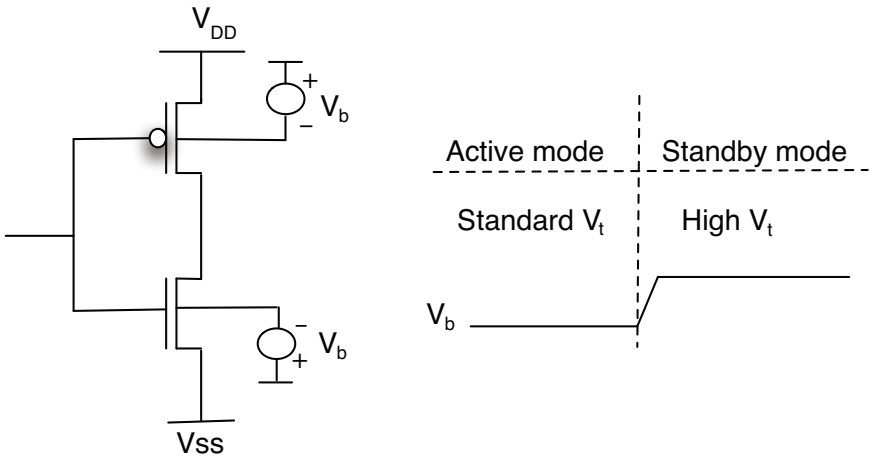


Figure 2.16: Variable-threshold CMOS technique.

According to Kaijian Shi, a solution architect in Cadence design systems, substrate bias is no longer applied for 45nm and bellow circuits. This is because, substrate bias acts to reduce sub-threshold leakage currents while the dominant leakage source for recent technology nodes is rather gate leakage component.

Sleep transistor Sleep transistor technique consist in adding extra-transistors, called sleep transistors, to cut off pull-up/pull-down networks from the supply rails during the standby mode. Figure 2.17 illustrates the principle of this technique.

In the active mode, the sleep transistors are turned on to operate with regular supply voltages. While, in the standby mode, the sleep transistors are turned off to cut off leakage current flow.

In fact, it is the stacking effect between the extra transistor and the transistors in the networks that achieves leakage savings. Indeed, turning off more than one transistor in a stack raises the internal voltage (source voltage) of the stack, which acts as reverse biasing the source [26].

Moreover, in [38] Mutoh propose to use multi-threshold (MTCMOS) technique where high-V_t are used for the sleep transistors to achieve extra leakage savings, while low-V_t transistors are used in the logic circuit to maintain speed requirement.

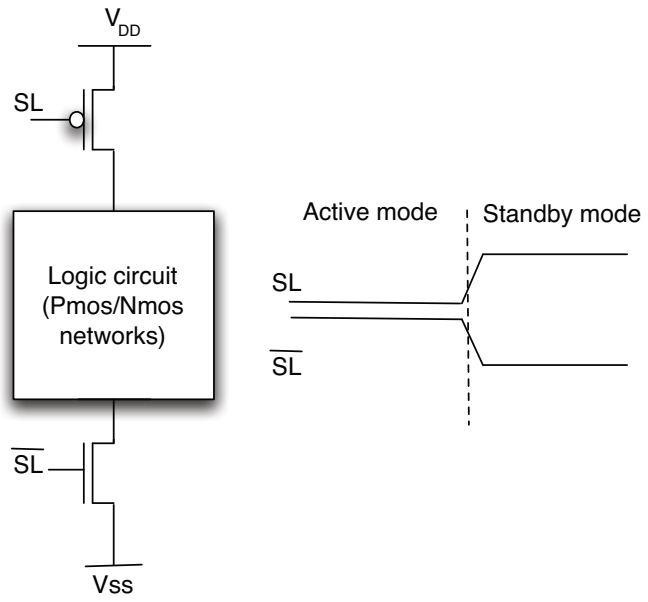


Figure 2.17: Sleep transistor technique.

2.4.3 Discussion

The most efficient technique to reduce dynamic power is to reduce the supply voltage. However, the problem with V_{DD} scaling is that it decreases the drive current, resulting in lower performance. An effective way to maintain the performance is to decrease the threshold voltage. But, with lower V_t, leakage currents increase. This is why there exists always a (V_{DD},V_t) trade off that achieves the optimum design in terms of total power consumption.

Although most techniques for leakage reduction are in a technological level (multiple V_t, high K, Finfet transistors...), leakage component is also linked to the switching activity because reducing the switching activity with a judicious choice of architectural transformations, will allow us to easily increase the supply voltage and hence the threshold voltage, which as consequence reduce leakage currents.

In other terms, the opposite trends of the leakage energy and the dynamic energy lead to an optimum V_{DD} that minimizes the total energy E_{tot}, as illustrated in Figure 2.18(a).

Roughly speaking, the minimum of E_{tot} occurs near the crossing-point of the leakage energy and the dynamic one ($E_{leak} \simeq E_{dyn}$). Now, if the switching activity is reduced, which is illustrated by the translation of the dynamic curve in the Figure 2.18(b), the new optimum supply voltage (V_{DD2}) will be surely greater than the old one (V_{DD1}), in order to balance the leakage and the dynamic energy.

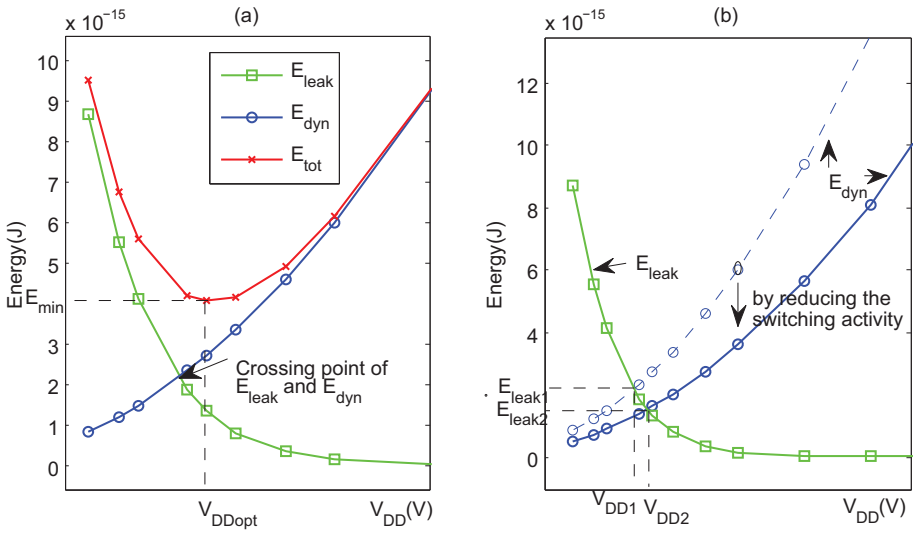


Figure 2.18: Leakage and dynamic energy trends.

2.5 Conclusion

In this preliminary chapter, we reviewed power dissipation, power estimation techniques and power optimization techniques targeting low energy design in static CMOS circuits. We showed that power estimation and optimization are emerged at all levels of the design hierarchy in order to reach an optimal solution in terms of power dissipation.

It should be noted that with technology scaling, leakage currents become a critical problem and contribute to the major part of power dissipation in today's sub-nanometer technologies. However, this is a purely technological problem and it is treated in a technological level. For instance, there are potential solutions proposed by the founders like the proposed Intel tri-gate transistor.

Nevertheless, leakage component remains very related to the switching activity which is purely a logical term. For that reason, we consider that reducing leakage power is also involved in an architectural level.

In our work, we focus in the dynamic part as we still need to reduce power when the circuit is no longer in a steady state.

Chapter 3

Glitch power reduction for Low Energy Design

3.1 Introduction

Although leakage currents are dramatically increasing with technology scaling [39], we still need techniques to reduce power when the circuit is no longer in a steady state. So, in our work, we focus on dynamic energy reduction by acting in the design level.

A change on the output level, causes a capacitance charge or discharge and consumes $P = 1/2.C_{eff}V_{DD}^2$, where C_{eff} is the load capacitance on the output. In nominal synchronous logic, the output would change at most one time each clock period. However, there are unnecessary signal transitions, called glitches, that occur due to the differential path delay (DPD) of input signals. Obviously these transitions, like any other transitions in the circuit contribute to the global power dissipation. Some researches prove even that they consume about 30%-40% of the dynamic power consumption in CMOS circuits [40].

Recently, several works have focused on glitch power optimization techniques. These methods try to balance the differential path delays to guarantee that signals arrive at the same time at the inputs of the logic gates.

In this chapter, we propose a new method for minimizing glitches in the combinational part of CMOS circuits based on a dual-threshold voltage (dual–V_t) technique. We present a heuristic algorithm that minimizes glitches using dual–V_t assignment. The results show a 16% average glitch reduction on ISCAS85 benchmark circuits. We compare the proposed method to the well-known gate–sizing technique [41]. Then, we propose to combine the two techniques into a single optimization process. Up to 18% of glitch reduction is achieved compared with the conventional gate–sizing technique.

To test the optimization algorithm, we developed a glitch analysis tool that uses post-

layout timing report. We then integrate this tool in a glitch aware top-down design flow.

This chapter is organized as follows. Section 3.2 reviews glitch minimization techniques. In section 3.3, we describe CAD tools for glitch power estimation and we propose a new methodology for glitch analysis. The heuristic dual–V_t algorithm and the mixed gate–sizing/dual–V_t algorithm are presented in Section 3.4. Finally, some simulation results are presented.

3.2 Glitch reduction techniques

Over the last decade, several works concerning glitch power optimization were proposed. Most of them apply either path balancing approach like the work presented by Kim in [42] or glitch filtering approach proposed by Agrawal in [43]. Figure 3.1 illustrates these techniques.

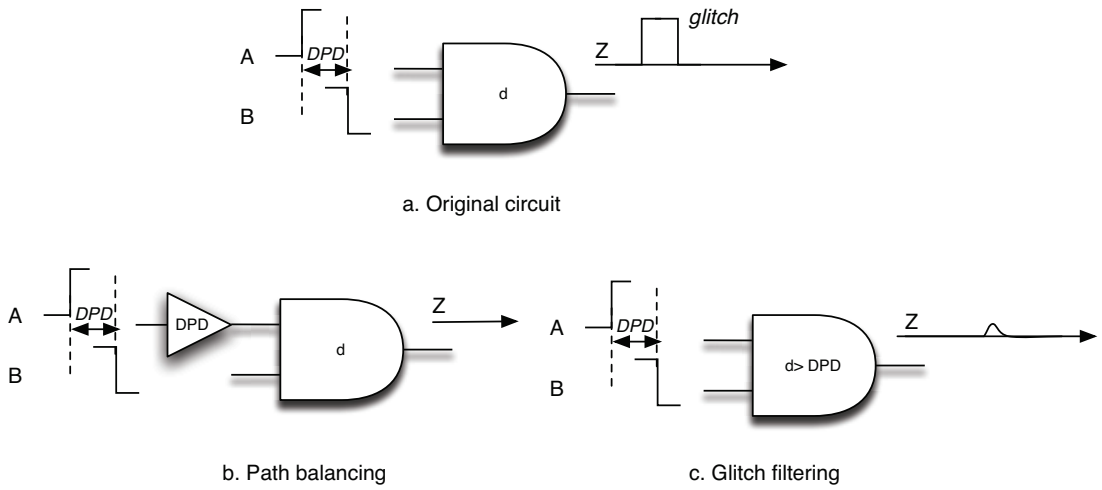


Figure 3.1: Glitch minimization techniques.

Path balancing (Figure 3.1.b) consists in adding extra delay buffers to early arriving signals such that DPD is decreased, while glitch filtering (Figure 3.1.c) adjusts the inertial delay of gates such that :

$$t_i - t_j < d \tag{3.1}$$

This inequality has been presented initially by Agrawal in [44] where they demonstrate that no glitch appears at any gate as long as the DPD of its input signals is smaller that its inertial delay.

Gate sizing is a well-known technique that applies the glitch filtering approach. It scales the size of gates in the circuit to impact path delays. A combination of gate sizing and

path balancing techniques is demonstrated by Kim in [42] to achieve an optimum design.

Next, we explain path balancing and gate–sizing techniques and we show the drawbacks and advantages of each of them.

3.2.1 Path balancing technique

Inserting delay buffers to avoid glitch appearance is not as simple as it appears. Indeed, extra elements inserted consume themselves extra power and hence reduce the achievable power reduction. The challenge is to determine the number of buffers to be inserted and the exact position where they will be inserted in order to achieve the optimum solution and reduce the overhead.

To do so, Agrawal in [44] has proposed to use linear programming (LP) techniques. It is a linear formulation of an optimization problem under linear constraints. Unlike heuristic approaches, it gives a globally optimal solution. It is thus, an interesting technique to solve optimization problem. Furthermore, there are many software solutions, called solvers, that are available and can be used to find the solutions for LP problems [45].

The objective function for LP path balancing is to minimize the number of delay buffers while maintaining the required performance on the overall circuit delay. A suitable linear formulation for this problem would be :

$$\begin{aligned} & min \sum_{i,j} \delta_{i,j} \\ & subject\ to \\ & T_{critic} \leq T_{max} \end{aligned} \tag{3.2}$$

Where $\delta_{i,j}$ is the delay of the element inserted between the gates i and j, T_{critic} is the critical path delay of the circuit and T_{max} is the maximum delay that still maintain the throughput constraint.

To derive a solution, each gate in the circuit is characterized by three variables:

- T_i : the latest time at which the output of gate i can produce an event after the occurrence of an event at the primary input;
- t_i : the earliest time at which the output can produce an event after the occurrence of an event at the primary input;
- d_i : the inertial delay of gate i.

Besides the overall circuit delay constraint which ensures that the critical path delay of the circuit should be lesser than a predefined limit (T_{max}). There are gate constraints that are applied to each gate in the circuit including the inserted delay buffers. These

constraints can be written as follows :

$$T_i \geq T_j + \delta_{i,j} \tag{3.3}$$

$$t_i \leq t_j + \delta_{i,j} \tag{3.4}$$

$$T_i - t_i \leq d_i \tag{3.5}$$

Where i is the corresponding gate and j is its fan–in gates.

The Equation 3.5 means that the differential path delay should be smaller than the inertial delay of the gate, which corresponds to the condition in Equation 3.1 that satisfies glitch filtering.

To better explain these constraints, let us consider the circuit in Figure 3.2 where the number inside each gate corresponds to its inertial delay and the numbers above are identifying them. Let us focus on timing constraints of gate (1). Two delay buffers are inserted on the inputs of this gate. It's constraints are :

$$T_1 \geq T_0 + \delta_{1,0} \tag{3.6}$$

$$T_1 \geq 0 + \delta_{1,PI} \tag{3.7}$$

$$t_1 \leq t_0 + \delta_{1,0} \tag{3.8}$$

$$t_1 \leq 0 + \delta_{1,PI} \tag{3.9}$$

$$T_1 - t_1 \leq d_1 \tag{3.10}$$

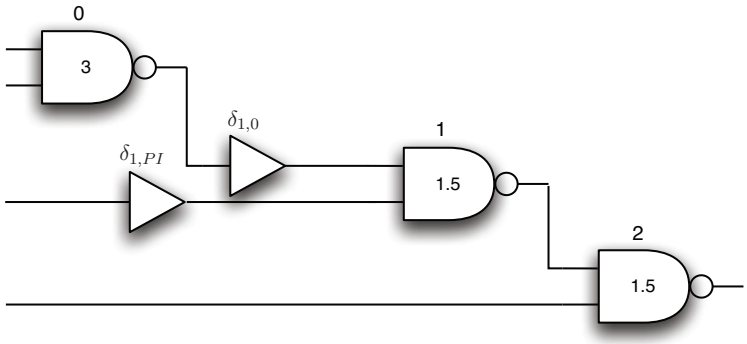


Figure 3.2: Test circuit to explain LP constraints.

It's now to the linear program solver to find a solution. For instance, an optimized solution where there is no timing constraints relaxation (ie. $T_{max} = 0$) is given in Figure 3.3.

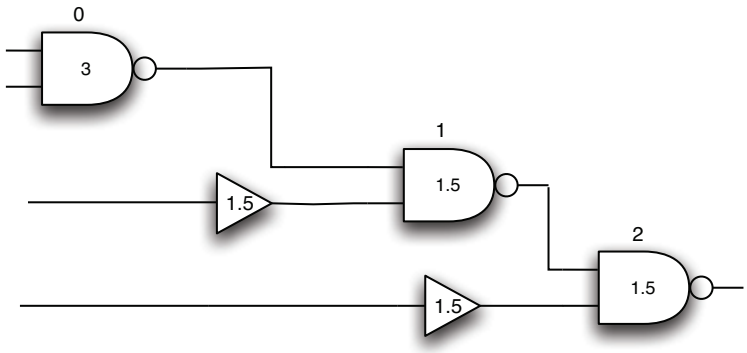


Figure 3.3: Optimized circuit using delay buffers.

3.2.2 Gate–sizing technique

As explained before, gate–sizing is a method that scales the size of gates to modify their timings.

In [46], Lee distinguish 3 types of gate sizing:

– downsizing the gates that produce the earliest input signals to hold them up;

– up–sizing the gates that produce the latest input signals to hurry them up;

– downsizing the gates where glitches may appear to increase their inertial delay.

Generally, gate downsizing is mostly used for the simple reason that up–sizing adds extra power consumption due to the use of larger transistors. Moreover, in the literature, the third type of gate sizing is the most commonly used and is referred to as gate–sizing technique. As a matter of fact, the first type is quite complex to implement since the earliest signals for some gates could be the latest signals for others. Throughout this thesis, like in previous works, gate–sizing technique will signify downsizing to increase the inertial delays.

So, gate–sizing for glitch minimization scales down gates where glitches may appear to increase their inertial delay and satisfy the Equation 3.1, such that glitches are filtered. Besides glitch power reduction, gate–sizing achieves capacitance power reduction since internal capacitances scale down as the size of the gate decreases. However, unlike path balancing, gate sizing and glitch filtering can, in general, increase the critical delay of the circuit and thus fail to meet timing constraints.

Many heuristic gate–sizing algorithms have been proposed to address glitch power reduction problem. We can cite for example, works done by Coudert [47] and Wang [48].

However, as explained by Hashimoto in [49], glitch reduction by gate–sizing technique is an ill–behaved problem. Indeed, modifying the delay of the gates may change path delays and result in the generation of glitches in other gates. Wang in [48] suggest to

use Hill–climbing or perturbation techniques to get out of a bad local solution. A recent heuristic gate–sizing algorithm has been presented by Wang and can be found in [48].

Notice that, there is no LP formulation proposed for gate–sizing to reduce glitches. This can be explained by the fact that gate–sizing is not a static problem in that changing the size of just one gate can affect all arrival times and path delays.

3.3 CAD tools for power estimation

In this section, we describe the design flow used to derive an estimation of the switching activity and the power consumption of a combinational circuit. Then, we illustrate how to estimate glitch power consumption using multiple delay simulation modes (zero and real). In the second part, we illustrate the flow used in this thesis to derive an accurate estimation of the power consumption based on electrical simulation.

3.3.1 Design Flow

Figure 3.4 illustrates the CAD flow to analyse power consumption. The results presented here are based on the utilization of Cadence tools but the different steps can be reproduced using other CAD tools. Starting from the HDL of the benchmark circuits, RTL compiler is used to synthesize the design. We obtain a gate netlist and a Standard Delay Format (SDF) file that contains estimation of the delays for all the cells and interconnects in the design. Using a test-bench and the SDF file, an RTL simulator can generate an accurate switching activity report (Toggle Count Format (TCF) file) used to annotate all the nets of the circuit. The power report generated is thus more accurate than what can be obtained without back-annotation.

3.3.2 Glitch power Estimation

To calculate glitch power consumption, two simulation modes are used: zero delay mode and real delay mode. In ”Zero delay mode”, all the gates are assumed to have zero inertial delays. Therefore, all signals (inputs and outputs) are generated simultaneously and no differential path delays can be detected. As a result, transitions corresponding to glitches are not taken into account when generating TCF file with zero delay mode and so does the estimated power.

Now, using the TCF generated with real delay mode, where inertial delays are those taken from an SDF file, all glitches are captured and a part of the dynamic power report contains this time glitch contribution. It’s clear that the difference between the dynamic power generated by the two simulation modes gives an estimation of the glitch power consumption.

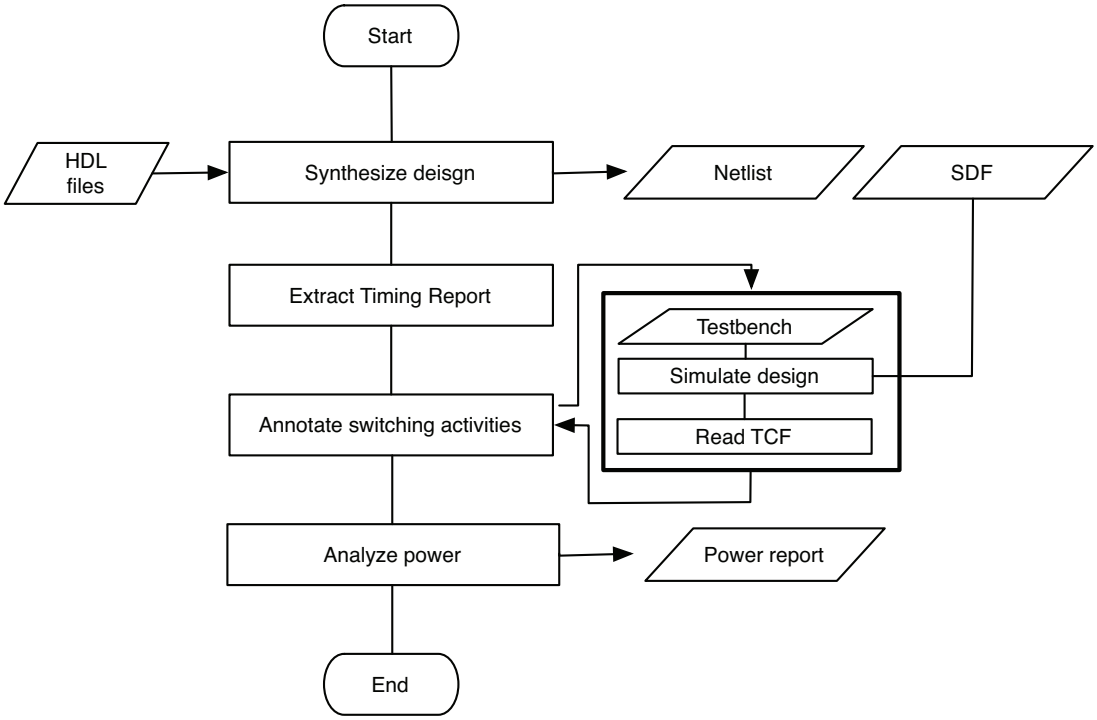


Figure 3.4: Power analysis in the Top-Down design flow.

Previous works done by Wang [48] and Lu [50] used this method to derive an estimation of glitch power dissipation. To illustrate that, we show an example of these simulations for the c17 ISCAS85 benchmark circuit with zero delay mode and real delay mode. Figure 3.5 shows the waveforms of the output of the gate "NAND3" under the two simulation modes. As expected, glitches are omitted when simulating with zero delay mode.

Table 3.1 shows dynamic power report of c17 benchmark circuit with 10000 random input vectors generated after back-annotation of zero delay and real delay switching activity.

Table 3.1: Glitch power estimation using zero delay and real delay simulation modes.

Circuit	**Dynamic power (μW)**		**Glitch power (μW)**
	Zero delay	**Real delay**	
c17	4.4	7.6	3.2

Column 4 gives the glitch power, which is the difference between dynamic power with zero delay mode simulation and real delay mode simulation. Unfortunately, such value can be under-estimated. Indeed, transitions counted in the TCF file are just complete ones from rail to rail. However, for gates whose DPD are not that large relatively to the inertial

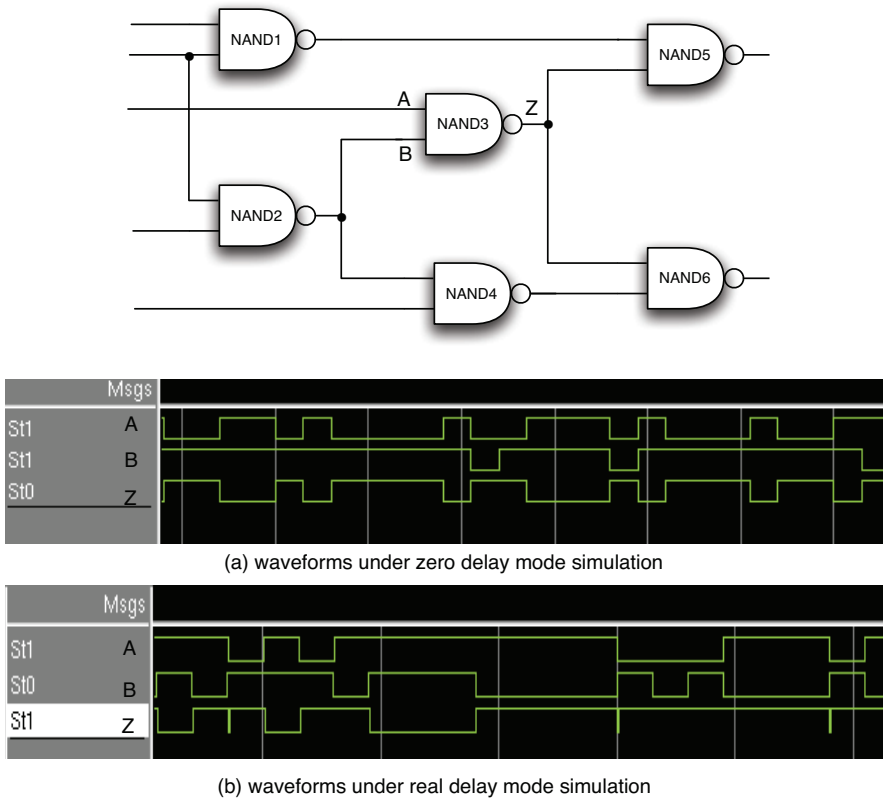


Figure 3.5: Simulation with (a) zero delay mode, (b) real delay mode of c17 benchmark circuits.

delay, the output can make transition to an intermediate state ($V_{DD}/2$, $V_{DD}/4$ ...), and these kind of transitions also consume power.

In the next section, we explain the design flow used in this work to generate a precise estimation of power consumption, based on electrical simulation.

3.3.3 Accurate power Estimation

An accurate estimation of power consumption can be obtained using electrical simulations. Figure 3.6 shows the flow overview. First, a spice netlist is generated from the gate verilog netlist obtained after circuit synthesis. A value change dump (VCD) file that contains all changes of the input signals is obtained from the verilog event simulation. The electrical simulation using transistor level models gives the current delivered by the supply voltage and thus allows power computation.

However, even such results miss some accuracy since wire capacitances are not yet included, especially for glitch analysis where path delays are an important factor. Indeed wire capacitances affect the delays in the circuit and can change the differential path delays resulting in more or less glitches in the circuit.

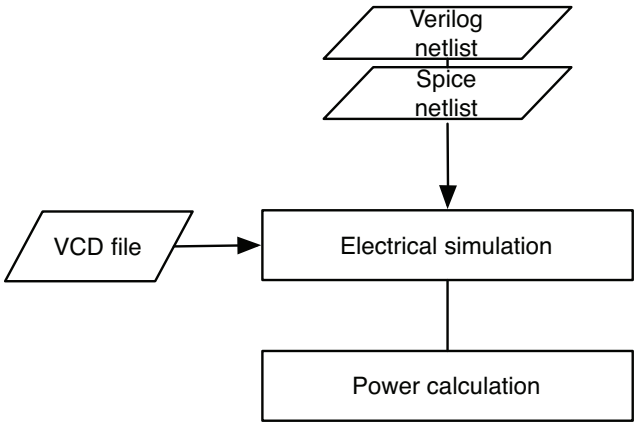


Figure 3.6: Electrical simulation for accurate power estimation.

Figure 3.7 illustrates the design flow used in this work to calculate power consumption. After layout and parasitic extraction, the simulator generates a SPEF (Standard Parasitic Exchange Format) file that contains parasitic resistance and capacitance of all wires in the circuit. Then, the Spice model, generated from the SPEF file, is simulated in combination with the VCD file.

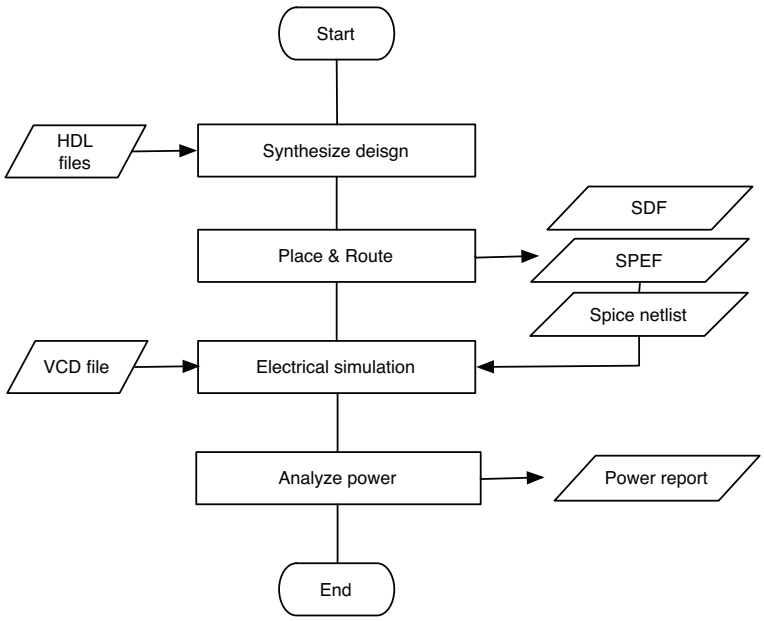


Figure 3.7: Post-layout electrical simulation.

3.3.4 Glitch analysis tool

For all optimization algorithms, the earliest and the latest arrival times should be computed at each node in the circuit. To do so, previous works use generally an iterative procedure where gates are traversed in a topological order from the primary inputs to the outputs. Hence, the earliest and the latest arrival time at the output of any gate are the accumulation of the inertial delay and the earliest and the latest arrival times of input signals, respectively.

Figure 3.8 illustrates such procedure.

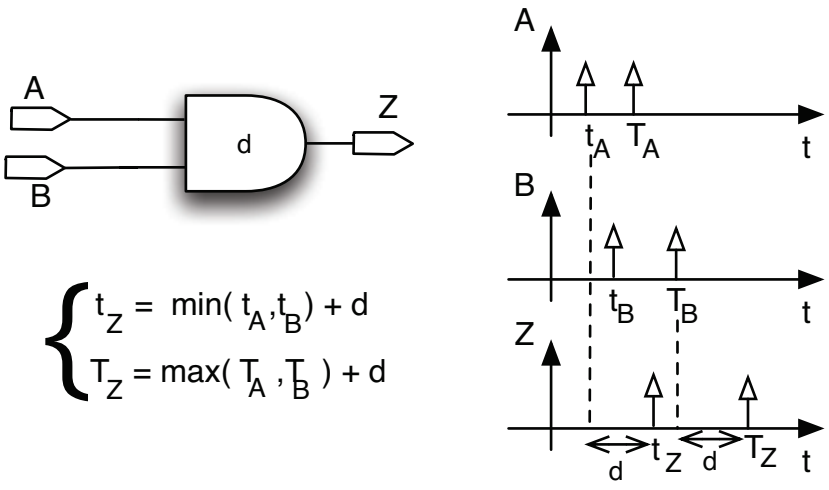


Figure 3.8: Determination of the earliest (t_i) and the latest (T_i) signal arrival times.

Generally, the inertial delay is derived from a delay model that uses fitting parameters such as load capacitance, input transition time, temperature, numbers of inputs/outputs of the gate..etc. However, such approach may not have the sufficient accuracy since wire capacitances are not computed. This lack of accuracy can lead to wrong results in the glitch optimization procedure. For that reasons, we have preferred to use simulation-based approach.

To analyze glitches in a circuit, we examine the timing report generated after layout and parasitic extraction. In the results presented here, the extraction has been done using Cadence's Encounter toolset. We have developed a software (mostly Python routines) that reads the timing report and provides a simplified report that contains, for each gate in the circuit, the following information :

- The inertial delay (ID);
- The maximum differential path delay (DPD) of input signals.

This report is further modified to include just gates that do not satisfy the condition in Equation 3.1. The report is now called glitch report and it contains all gates with glitch

appearance. Figure 3.9 shows the glitch report of ISCAS85 c17 benchmark circuit.

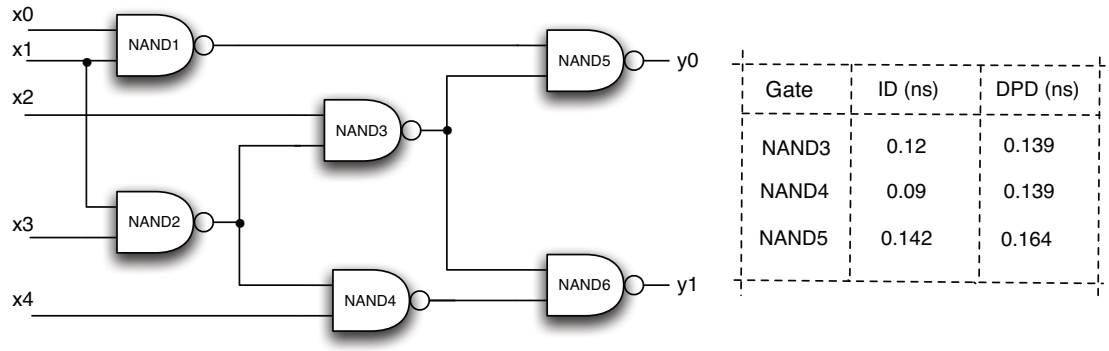


Figure 3.9: Glitch report of c17 benchmark circuit.

For our analysis, the percentage of glitches is simply the percentage of gates with glitch appearance. And it can be computed using the number of gates in the glitch report. This metric can over/under estimate glitch switching activity which depends deeply on transition probability of input signals. However, it's obvious that decreasing the number of glitchy gates decreases the switching activity.

To test glitch minimization algorithms, we propose to integrate glitch analysis tool in the top-down design flow as shown in Figure 3.10. Hence, we have a glitch aware top-down design flow, where glitch optimization routine is repeated until no further optimization is possible.

3.4 Dual-V_t for glich reduction

As mentioned earlier, dual–V_t has been proposed as an efficient solution for leakage power reduction. The concept is simple : high threshold voltages are assigned to gates off-critical paths to decrease subthreshold leakage currents while low threshold voltages are used to maintain the required performance. Many heuristic algorithms and linear programming (LP) functions have been proposed to search for an optimal solution for this problem and the results show a great leakage power reduction [51] [52].

However, since threshold voltage deeply impacts the delay of the gate, dual–V_t, as it is applied for leakage reduction, can be the source of glitches and may increase the dynamic power. None of the previous works have considered glitches due to dual–V_t assignment. Although, simulations show over 5% increase of glitches on c2670 ISCAS85 benchmark circuit.

In this work, we propose to use dual–V_t technique rather for glitch reduction. In the next sections, we explain the basic idea and we present the proposed heuristic dual–V_t

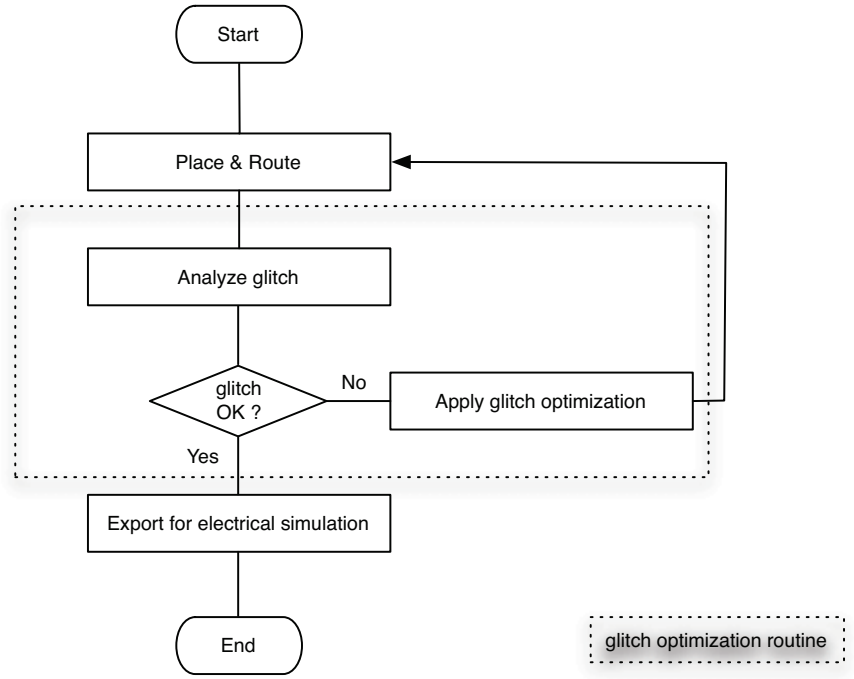


Figure 3.10: Glitch aware top-down design flow.

algorithm.

3.4.1 The basic idea

The basic idea of dual–V_t for glitch minimization is to use high threshold voltage in order to increase the inertial delay of gates such that glitches are filtered. As can be inferred from Equation 3.11, the delay of the gate increases as threshold voltage increases.

$$t_{pd} = \frac{C_L V_{DD}}{(V_{DD} - V_t)^{\alpha}} \tag{3.11}$$

Where C_L and α are, respectively, the load capacitance and the velocity saturation index.

Table 3.2 shows the inertial delays of some gates from a 65nm HVT and SVT industrial library with 2 fanouts. HVT and SVT denote respectively High and Standard threshold voltages. Results are obtained using a spice simulator with a nominal power supply voltage $V_{DD} = 1.2$V.

We observe an average increase of 42% in the gate delay, moving from standard to high V_t gates. Hence, applying high V_t can result in the complete filtering of the glitch if the DPD is not larger than 1.4 the inertial delay on average. Even if the DPD is too large, the use of high V_t for the candidate gates may decrease the magnitude of the glitch, which

Table 3.2: Inertial delays of low and high V_t gates.

	Inertial delay (ps)		
Gates	SVT	HVT	% increase
AND	45.1	65.5	45.2
OR	55.2	80.1	45.1
NAND	20.4	28.1	37.7
NOR	25.4	36.1	42.1
		Average	42.5

can also achieve switching power savings.

To illustrate the effect of high-V_t on glitch removal, let us consider the circuit with a glitch formation shown in Figure 3.11(a) . Glitches appear due to the difference of input arrival times to the AND gate. Simulation results using SVT and HVT AND gates are shown in Figure 3.11(b).

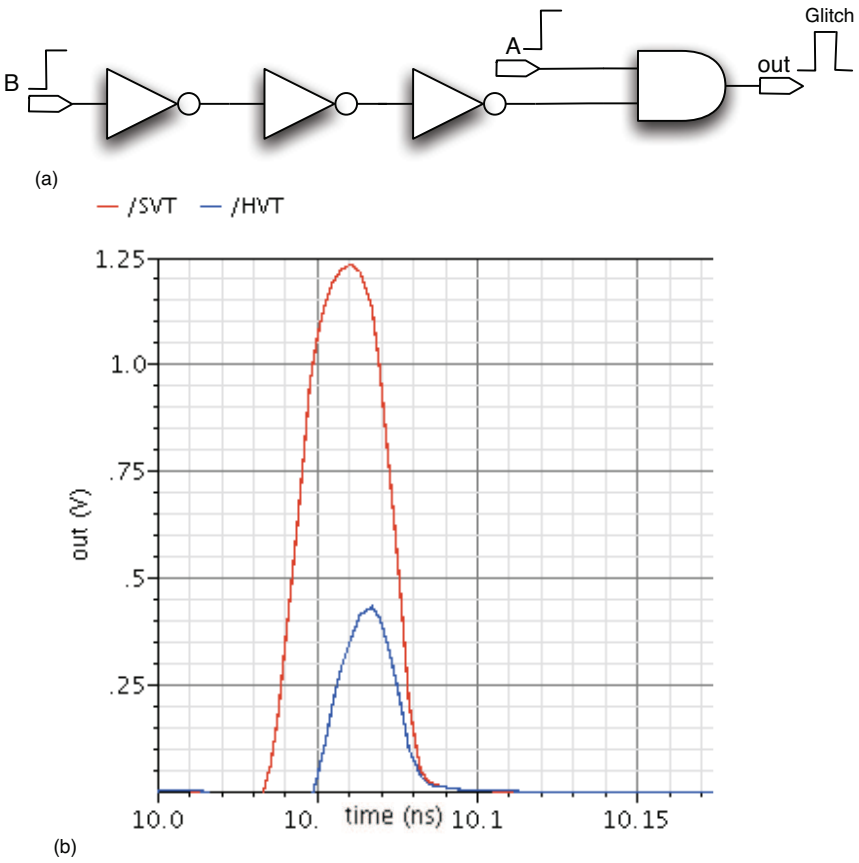


Figure 3.11: Output voltage waveforms for different threshold voltages.

We observe that the glitch is partially filtered when using HVT gate and this is due to the inertial delay increase.

3.4.2 Optimization algorithm

Let $S = (s_1, s_2, ..., s_n)$ be an implementation of a given combinational circuit G synthesized in a standard cell library L operating in low V_t.

Let $s^{svt}_{i\subseteq 1..n}$ and $s^{hvt}_{i\subseteq 1..n}$ be the same functional gate as $s_{i\subseteq 1..n}$ operating in low V_t and high V_t, respectively. The proposed heuristic dual–V_t algorithm is presented in Algorithm 2.

Algorithm 2 Heuristic Dual-V_t algorithm

```
GlitchOptimization(G,S,L) begin
Criticalpathdelay(G, S)
if C ← glitchreport(G, S) ≠ 0 then
    D ← Sort(C)
    S' ← S
    for all ν_j ⊆ D do
        s_j^svt ← search(ν_j, S)
        s'_j ← s_j^hvt
        Criticalpathdelay(G, S')
        if chektiming(G, S') = failed then
            s'_j ← s_j^svt
        end if
    end for
end if
end
```

Actually, the $glitchreport(G, S)$ routine generates the list of all gates where glitches can appear as explained in Section 3.3.4.

Since using dual–V_t is more efficient for gates whose DPD is much smaller, before the optimization, gates in the glitch report are sorted from the lowest to the highest DPD. Then, each gate is modified to operate in high V_t as long as timing constraints are respected.

Like gate–sizing technique, glitch reduction by dual–V_t is an ill-behaved problem [49]. Indeed, modifying the delay of the gates may change path delays and result in the generation of glitches in other gates. To get out of a bad local solution, we use Hill-Climbing technique where the optimization procedure is repeated until no further improvements can be found.

3.4.3 Unified Dual-V_t/gate–sizing algorithm

We combine the proposed dual–V_t technique and the gate–sizing technique to achieve further glitch reduction. Algorithm 3 describes the overall optimization algorithm.

First, we apply the heuristic dual–V_t algorithm described in algorithm 2. We use Hill-Climbing algorithm to reach a global optimum solution. The gates that remain in the glitch report have their glitches not suppressed during dual–V_t optimization. Then, we use

Algorithm 3 Heuristic Dual-V_t/gate–sizing glitch reduction algorithm

```
UnifiedGlitchOptimization(G,S,L) begin
S' ← Hill − Climbing(GlitchOptimization(G, S, L))
Criticalpathdelay(G, S')
if C ← glitchreport(G, S') ≠ 0 then
    S'' ← S'
    for all ν_j ⊆ C do
        s''_j ← downsizing(ν_j, S'')
        Criticalpathdelay(G, S'')
        if chektiming(G, S'') = failed then
            s''_j ← upsizing(ν_j, S'')
        end if
    end for
end if
end
```

downsizing to increase the delays of these gates. And each time, we update and verify path delays to make sure that timing constraints are not violated.

3.5 Experimental results

The proposed heuristic dual–V_t and dual–V_t/gate–sizing algorithms were implemented in Python. Glitch optimization was carried out on 6 ISCAS85 benchmark circuits synthesized in a 65 nm Standard Threshold Voltage Low Power (SVTLP) CMOS library.

Table 3.3 reports the percentage of glitch reduction using dual–V_t, gate–sizing and dual–V_t/gate–sizing optimization, under 5% of timing constraints relaxation.

Table 3.3: Glitch reduction on ISCAS85 benchmark circuits under 5% of timing constraints relaxation.

Circuits		% of glitch	% of glitch reduction		
Name	Gates		Dual-V_t	Gate–sizing	Dual-V_t/gate–sizing
c432	160	28.5	41.6	22.9	41.6
c499	202	42.8	12.2	12.2	21.1
c880	383	44.9	8.7	12.2	18.2
c1908	880	30.6	8.0	15.4	17.4
c2670	1193	23.9	21.7	25.7	38.6
c3540	1669	33.6	5.5	2.1	5.6
Average:		34.0	16.2	13.0	23.7

The results show 16% average glitch reduction by means of the proposed dual–V_t algorithm. We observe that the proposed dual–V_t present a better average glitch reduction than gate–sizing technique. This is due to the fact that gate–sizing is just used for gates which have the possibility to be downsized. Thus, not all gates with glitches can be treated

with gate sizing, while dual–V_t can be applied to all gates.

On the other hand, the delay increase resulting from a downsized gate is 1.5 times more important than that resulting from the use of high threshold voltage. Hence, the inertial delay increase can be much more than necessary, which can impact the critical delay and stop gate–sizing process for time-constrained circuits.

We remark that c432 benchmark circuit does not benefit from the combination dual–V_t/gate–sizing, since the glitch reduction is the same as dual–V_t. In fact, most of glitches in the circuit are caused by relatively small DPD that can be supressed by applying just dual–V_t. While other gates that still figure in glitch report have too large DPD that can not even be balanced by gate downsizing.

For some benchmark circuits like c2670, gate–sizing appears to be more efficient. Indeed, glitches that may persist using dual–V_t can be filtered by gate–sizing due to the important delay increase. We believe that dual–V_t and gate–sizing techniques are complementary as long as dual–V_t is more adapted for gates with small DPD, while gate sizing is more efficient for gates with large DPD. Indeed,Table 3.3, last column reports more than 18% of glitch reduction improvement compared to the gate–sizing results.

Table 3.4 reports the percentage of total energy reduction for c432 and c3540 ISCAS85 benchmark circuits. These results are obtained trough spice simulations with random input stimuli.

Table 3.4: Energy reduction due to dual–V_t, gate–sizing and Dual-V_t/gate–sizing.

Circuits		% of total energy reduction		
Name	Gates	Dual-V_t	Gate–sizing	Dual-V_t/gate–sizing
c432	160	27.3	33.4	48.9
c3540	1669	9.2	16.8	27.1

Instead of what may be expected, gate–sizing presents a better total energy reduction than dual–V_t. This can be explained by the amount of energy saved due to the use of smaller transistors. Nevertheless, this does not contradict, under no circumstances, the results in Table 3.3 where dual–V_t corrects more the differential path delay and hence allows more glitch filtering.

Results in Table 3.3 show just 5.6% of glitch reduction for c3540 benchmark circuit after dual–V_t/gate sizing optimization. While spice simulation shows more than 27% of total energy savings. Once again, this can be explained by the percentage of dynamic energy savings when using smaller transistors and by the percentage of leakage energy savings when using high threshold voltages.

We notice that after dual–V_t/gate sizing optimization, the total energy consumption

is diminished by up to 15% compared to gate–sizing optimization only. This proves again that gate–sizing combined with the proposed dual–V_t technique is effective for glitch minimization.

3.6 Conclusion

In this chapter, we propose to use dual-threshold voltage technique to reduce glitches. We showed that dual-threshold voltage and gate sizing techniques are a good combination for a better glitch filtering. Together, they achieve up to 15% total energy reduction compared to gate–sizing optimization.

It should be noted that choosing to apply such or such glitch reduction techniques (path-balancing, gate–sizing and dual–V_t) or a combination of them depends totally on the circuit characteristics. One should just remember that :

- path balancing adds an extra power due to the inserted elements;
- gate–sizing and dual–V_t can increase the delay of the circuit;
- dual–V_t is more adapted for smaller differential delays while gate–sizing is more efficient when dealing with large difference;
- besides glitch reduction, dual–V_t achieves leakage energy savings due to the use of high threshold voltage gates while gate sizing allows more capacitive energy reduction due to the use of smaller transistor.

Chapter 4

Subthreshold Operation for Low Energy Design

4.1 Introduction

Over the last decade, sub-threshold logic has been used as an ideal option to achieve Ultra Low Energy consumption for applications with low demand in speed requirements. Here, sub-threshold term refers to the weak inversion (WI) region where there is an exponential dependence between drain current and gate voltage [53] and where the minimum energy can be achieved using a supply voltage V_{DD} well below the threshold voltage. Traditionally digital circuits have a supply voltage V_{DD} well above the threshold voltage. In this case transistors when are on, operate in the strong inversion region where the dependence between drain current and gate voltage is quadratic (becoming almost linear when velocity saturation effect tends to dominate [54]). In between weak and strong inversion there is the moderate inversion region or near-threshold region, occurring for V_{DD} voltages around the threshold voltage, where the transistor I_D-V_G characteristic is neither exponential nor quadratic [54]. Recently it has been shown that the near-threshold region provides convenient trade-offs [55] in the design of ultra low energy digital circuits and this thesis contributes to this topic.

As the interest for ultra low energy has increased, research related to sub-threshold logic has attained considerable importance. Modeling and characterization of sub-threshold operation for standard CMOS cell designs have been investigated for energy and performance analysis [53] [56]. However, several works [57] [58] [59] have shown that variability in sub-threshold logic is a critical limit to achieve robust ultra low energy devices. As current in weak inversion region exponentially depends on threshold voltage (V_t), random V_t variations significantly affect the on and off currents and gate delays and can affect the output

swings and result in functional failures of some gates. Moreover, the minimum energy point can be strongly affected by variability. Models considering variability have been developed in previous works specifically for use in subthreshold, considering, therefore, just the weak inversion region. [60] [59]. Through Monte Carlo Spice simulations, we show that variability moves the effective minimum energy point towards the near-threshold region (Moderate Inversion). Thus, models restricted to the weak inversion (exponential drain current vs. gate voltage) region can no longer model circuit performance around the minimum energy point.

In this chapter, we apply a complete and compact transistor model valid from weak to strong inversion in order to correctly model the circuit performance in a variability aware analysis. We provide an analytical solution for the optimum supply voltage that minimizes the total energy per operation while considering variability effects. Simulations of the transistor and inverter chain characteristics are required. Then, the model is validated for predicting the behavior of more complex circuits, like a 32-bit adder applied as test case. Section 4.2 presents the concept of sub-threshold operation and briefly reviews previous work in digital circuits. In section 4.3, we investigate the effects of variability to achieve robust subthreshold circuits and its impact on minimum energy point. In section 4.4, we propose a complete model that includes the near threshold region. The verification of the model in the test case of a 32-bit adder and the analytical expression for the optimum V_{DD} are also included in section 4.4.

4.2 Sub-threshold Operation

Sub-threshold operation consists in reducing the power supply voltage V_{DD} below the threshold voltage V_t in order to achieve minimum energy consumption. The concept is simple: as P_{dyn} is proportional to the square of V_{DD}, a small reduction in supply voltage causes quadratic decrease in dynamic power consumption at the cost of a significant increase in delay. This results in an increase in the leakage energy as it is proportional to the delay of the circuit. The opposing trends of the two forms of energy (dynamic and leakage) lead to a minimum energy point achieved at an optimum supply voltage. This point often occurs in the weak inversion region where sub-threshold leakage currents are used as the active drain current. Figure 4.1 plots the I_D versus V_{GS} for a NMOS transistor in a 65nm technology. This curve points out two main characteristics of subthreshold operation. First, in the WI region, the drain current depends exponentially on V_{GS}. Second, lowering the supply voltage causes the degradation of active to idle currents ratio (I_{on}/I_{off}), where I_{on} is the subthreshold current when V_{GS}=V_{DD} and I_{off} is the leakage current derived when V_{GS}=0. The decrease of this ratio depends on the subthreshold swing S defined as :

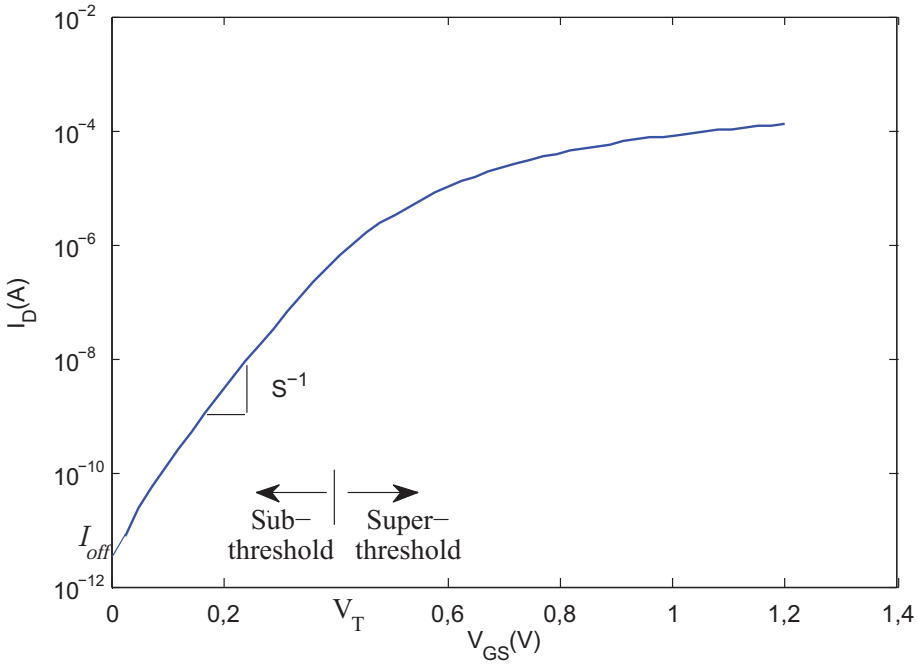


Figure 4.1: I_D versus V_{GS} curve for NMOS transistor.

$S = nV_t ln10$, where n is the subthreshold slope factor [53]. Moreover, a reduced I_{on}/I_{off} ratio can impact output swing and result in a functional failure especially when process variation effects are introduced. This will be further explained in Section 4.3.2.

Note that the exponential decrease of I_{on} in the weak inversion region leads to an exponential increase of the delay as shown in Figure 4.2. That limits the application of subthreshold logic to circuits requiring low to medium throughput constraints. For instance, wireless-sensor networks are one of those applications that benefits well from subthreshold operation. Indeed, energy consumption is the primary concern in wireless-sensor node where small batteries with a long lifetime are needed.

Figure 4.3 shows the Spice results for the evolution of active, leakage and total energy consumption of a chain of inverters with a logic depth equals to 23, like in [61] where the circuit is a standard 8-bit ripple-carry-array (RCA) multiplier. As expected, the dynamic energy (E_{dyn}) decreases quadratically with the supply voltage while the leakage energy (E_{leak}) goes up exponentially due to the exponential increase of the delay at lower supply voltages. These opposing trends lead to a minimum energy point achieved at the optimum supply voltage $V_{DDopt} = 0.2V$.

Several works have emerged to characterize minimum energy operation for subthreshold circuit design. In [62] authors suggests to plot constant energy and delay contours to evaluate optimum (V_{DD},V_t) voltages for a desired clock frequency under varying activity factor. Later, Calhoun in [63], provides analytical solution for the optimum (V_{DD},V_t) that

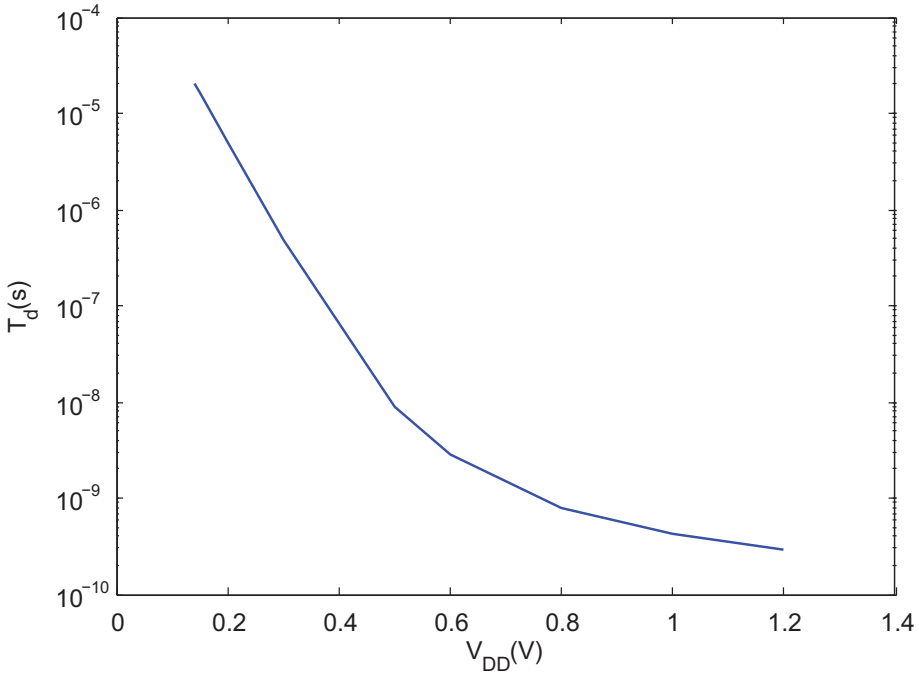


Figure 4.2: The delay T_d of a chain of inverters (L_D=23) across the full range of supply voltages.

minimize the energy for a given operating frequency. It shows that V_{DDopt} depends on the activity factor and not on the operating frequency.

Other works have tried to use different logic families such as Pseudo-NMOS (P-NMOS) and domino logic for subthreshold design [64] [65]. Due to their reduced delay compared to static CMOS logic, P-NMOS logic is demonstrated to achieve lower energy solution. An adaptive filter for hearing aids implemented in P-NMOS logic and operating in subthreshold regime is reported in [64].

Various CMOS circuits were designed to operate in subthreshold region. The subthreshold FFT processor implemented in [56] was the first major digital circuit operating in subthreshold region. Table 4.1 surveys recently developed circuits in subthreshold CMOS logic.

Table 4.1: Survey of developed circuits in subthreshold CMOS logic.

Reference	Circuit	Process-technology	Operating-voltage	Frequency	Energy/cycle
Bol2012 [66]	microcontroller	$65nm$	$0.4V$	25 MHz	7 nJ
Seok2011 [67]	FFT	$65nm$	$0.27V$	30 MHz	17.7 nJ
Kwong2008 [68]	Processor	$65nm$	0.3V-0.6V	1.04 MHz (@0.6V)	29.9 pJ (@0.6V)
Kim2008 [69]	SRAM	$0.13m$	$0.2V$	100 kHz	2 pJ
Seok2008 [70]	Processor	$0.18m$	0.5V	106 kHz	2.8 pJ

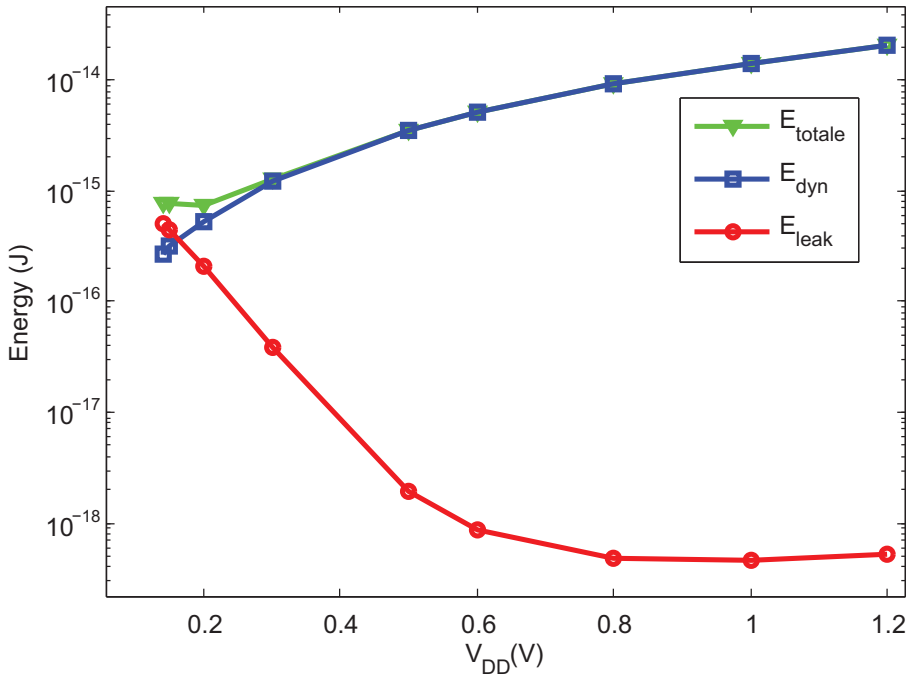


Figure 4.3: Dynamic, leakage and total energy evolution for a chain of 23-inverters.

4.3 Variability in subthreshold design

In this section, we identify the different sources of process variations and we analyze their impact on subthreshold circuit design first and on minimum energy point second.

4.3.1 Process variations

Process variations are fluctuations around the intended value of a design parameter, caused by manufacturing process. They are typically classified into global (inter-die) and local (intra-die) variations [71]. Global variation affects every structure on a die equally and can induce different characteristics from one die to another. They result generally from factors such as temperature effects and equipment properties. Local variation, however, affects structures on the same die differently. Three sources are of particular importance : Random Dopant fluctuations (RDF :random placement of dopant atoms in the channel region), Line Edge roughness and oxide thikness T_{ox} variations. Usually, these variations are identified by the ratio, σ/μ, where, σ and μ are the standard deviation and the mean of a process parameter, respectively.

4.3.2 Variability impact on subthreshold circuits

Since subthreshold leakage currents depends exponentially on the threshold voltage, operating the circuit in the weak inversion region will result in a more sensitivity to V_t

variation. Both global and local variations can induce V_t variations and affect severely the functionality of subthreshold circuits. However, in [72], authors show that global V_t variation can be compensated using adaptive body biasing technique (ABB). A further analysis of the efficiency of this technique for subthreshold circuits can be found in [60]. For local variation, it is shown in [60] that RDF has the most significant impact on V_t variability. It can be modeled through a normal distribution where the standard deviation is inversely proportional to the square root of the channel area [73].

Figure 4.4 plots the delay distribution of a 32-bit adder in sub-threshold (0.2V) and above-threshold (1.2V) resulting from 1K Monte Carlo simulation in 65nm process technology.

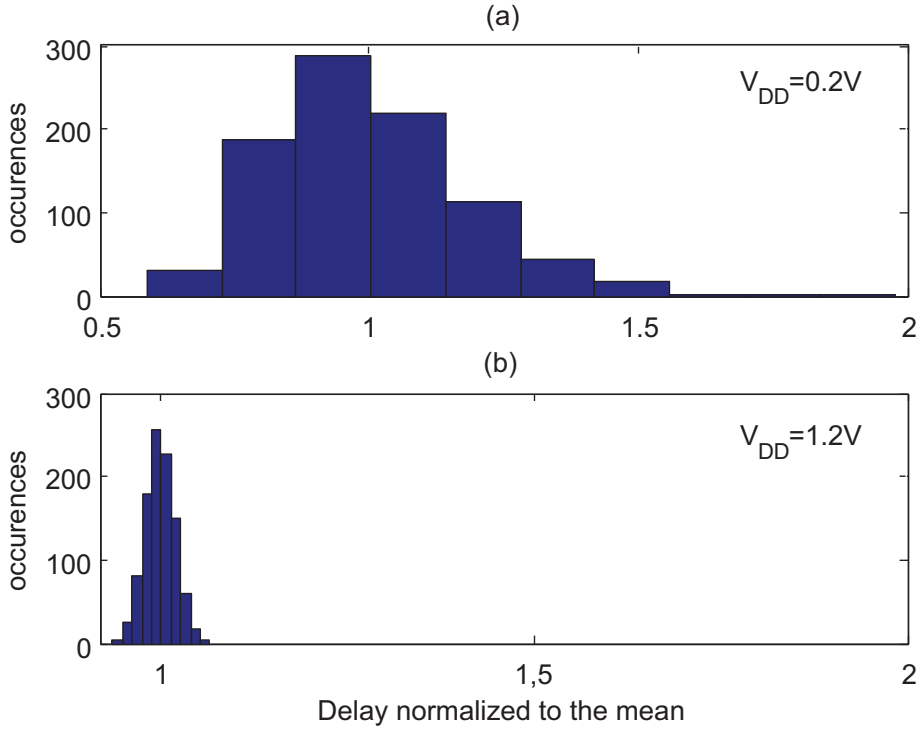


Figure 4.4: Delay distribution of a 32-bit adder (a) in sub-threshold (0.2V) (b) in above-threshold (1.2V).

We observe that the standard deviation in sub-threshold voltage is two orders of magnitude larger than that at nominal voltage. This implies a higher variability of sub-threshold operation that should be taken into account when designing subthreshold circuits. A major effect of variability is the correct functionality of subthreshold circuits. For example, if variations let NMOS devices stronger than PMOS ones, the pull up network can fail to drive the output to the correct logic value V_{DD}, and vice versa.
Many previous work have been proposed to design subthreshold circuit while considering failure due to process variation and low ratio of on to off currents.

In [56], a process corners analysis is proposed to determine the minimum operation voltage and the minimum transistor sizing which ensures a good functionality of the gate. The method consists in searching for the minimum ratio of the pMOS width to nMOS width of a standard cell library that still assures an output swing of 10%-90% of the supply voltage in the worst case corners. We have applied this analysis to an inverter from a 65nm Low Power industrial library. Figure 4.5 shows the limits for the ratio of the pMOS width to nMOS width, which must be between the minimum set by the Fast NMOS-Slow PMOS (FS) corner and the maximum set by the Slow NMOS-Fast PMOS (SF) corner that still drive the output voltage to 90%-10% of the supply voltage respectively. In fact, the curve marked with circles shows the minimum size of PMOS that still drive the output voltage to be at least $0.9V_{DD}$ in the presence of leakage currents through the fast NMOS device. Similarly, the curve marked with square shows the maximum PMOS size that allows the output voltage to be driven low than 10% of V_{DD} when idle currents of a large PMOS limits the drive current of small NMOS to pull down the output node. This restriction determines a minimum possible operating voltage, which for this technology occurs at 143mV by sizing the PMOS width to be 1.83 the NMOS one. We see that the inverter of the standard cell library ($W_p/W_n = 1.4$)is guaranteed to operate down to 160mV. Thus, if the minimum energy point is achieved by a supply voltage between 143mV and 160mV, a modified logic cell library should be created.

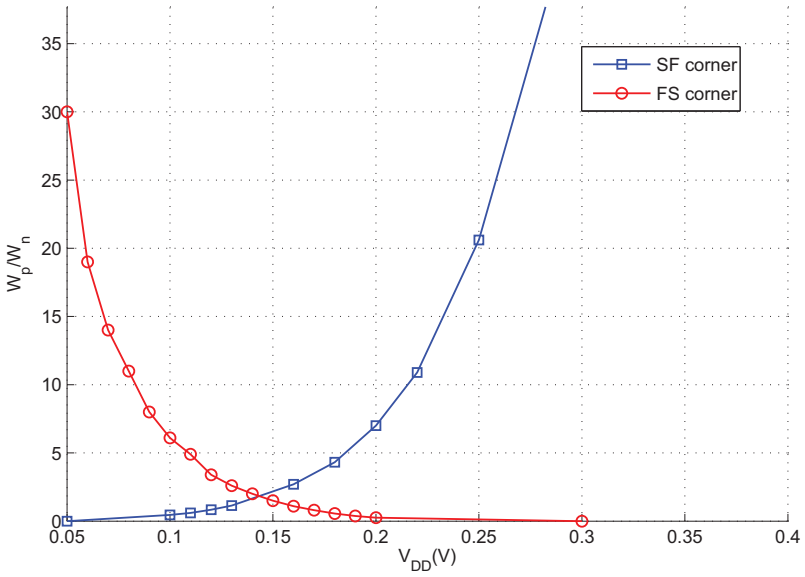


Figure 4.5: Worst case corners analysis of the inverter for different supply voltage.

Later, Kwong et al in [58] applied an efficient method to verify logic gate output levels (V_{OL} and V_{OH}) based on an SNM (Signal Noise Margin) approach, it's called the butterfly

plot. It consist to superimpose the VTC (Voltage Transfer Characteristic) of the gate in question with the mirrored VTC of NOR gate to verify V_{OL} and with the mirrored VTC of NAND gate to verify V_{OH}, as they have the worst case V_{IL} and V_{IH}, respectively. A negative SNM means that the V_{OL} of the gate is above the required input voltage (V_{IL}) of the succeeding gate. This implies a functional failure and thus the gates can not be operated at this supply voltage. Figure 4.6 shows the butterfly plot of NAND gate with functional and failing V_{OL}. Simulations are carried out with cells from an industrial 65nm technology for typical and worst process corners, respectively.

In [63], device sizing is proposed to allow for operation at lower supply voltage. It consist to increase device widths in order to operate in the optimal V_{DD} while still maintaining throughput constraints. This is specifically benefitial when the minimum sized circuit operating at the optimal supply voltage can not satisfy the required constraints of the circuit, an upsized circuit is used therefore to allow operation with minimum energy and satisfied throughputs. However, since device upsizing increase both the total switched capacitance and the leakage energy, a fine search for the suitable size that really guarantee a lower energy is required. In this thesis, we will not consider device upsizing, the approach was, instead, to evaluate the feasibility of applying an industrial standard library on the subthreshold and near-threshold regions. Therefore, all circuits are synthesized in a 65nm Low Power industrial library intended to work in nominal supply voltage (1.2V).

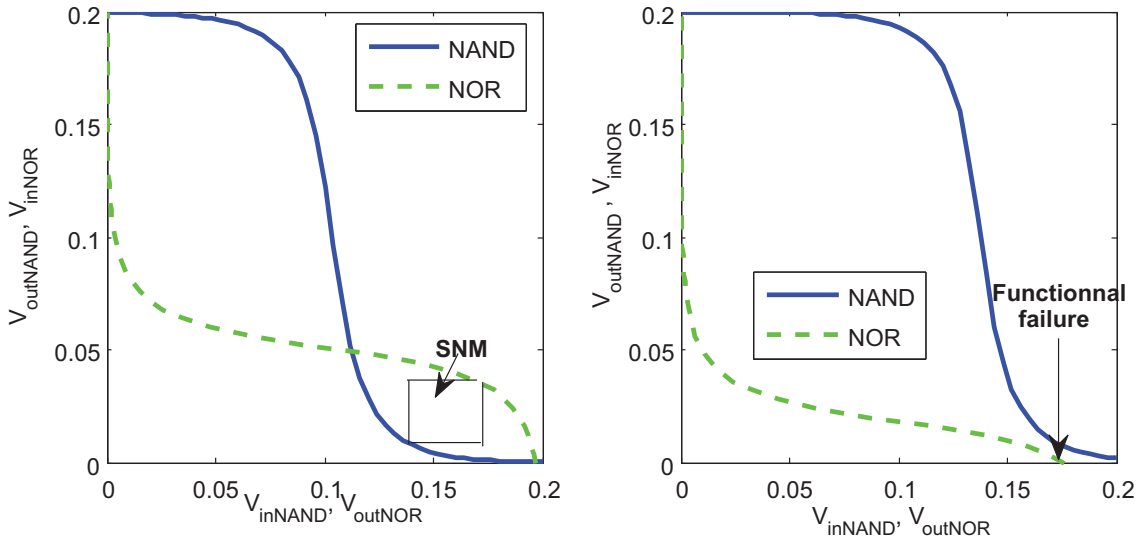


Figure 4.6: Butterfly plot of NAND gate with functional and failing output levels, simulation with gates from 65nm LP technology.

4.3.3 Variability impact on minimum energy point

To show the impact of variability on the minimum energy point, Monte Carlo Spice simulations with 1000 points have been performed for different values of V_{DD}. Simulations are first carried out in a chain of inverters where the first four inverters are not considered in order to correctly calculate the static current, increased due to the degeneration of stable states as mentioned in [53]. Since delay variability depends on the logic depth of the circuit, an appropriate choice of the logic depth is essential. For our case study, we have chosen, a logic depth of 50 corresponding to the logic depth of a 32-bit adder. Figure 4.7 shows the evolution of total energy consumption without and with variability effect, considering typical and 3σ worst case delay, respectively. We observe that variability leads to a considerable increase in leakage energy. This results in the increase of the minimum energy point by 50%, located now in the moderate inversion region.

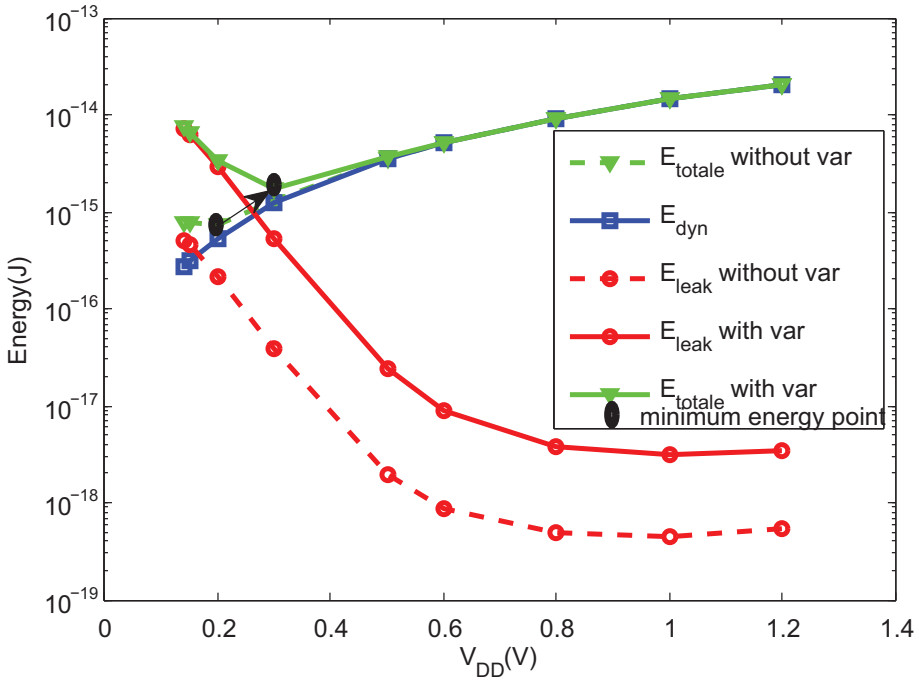


Figure 4.7: The evolution of dynamic, leakage and total energy consumption of a 32-bit adder with and without variability considerations.

4.4 Variability aware Modeling in sub-threshold Operation

Modeling the minimum energy point has been addressed in previous work [53] [60] [59] based on the simple sub-threshold current model (WI Model : exponential drain current vs. gate voltage). However, as we have seen in section 4.3.3, variability considerably affects the minimum energy point which moves towards the moderate inversion region. In [53]

modeling in moderate inversion applying an all region transistor model is also considered, but the impact of variability is not analyzed in this case. [55] is another prior work where authors suggest to work in the near threshold voltage region in order to recover some of the delay performance at the expense of a little energy increase. An energy-delay modeling framework that extends over all inversion regions is developed in this paper. But variability is still not considered in these models.

In this section, we present a variability aware model that extends over the weak and moderate inversion region. This model is based on the EKV model expressions [53]. We start with a simple model based on the characteristics of a transistor and an inverter and then show that this model allows to predict the behavior of more complex circuits. We show through Spectre and Matlab simulations that the WI model is no longer sufficient to model the performance of a system exposed to process variations. The proposed model is intended as a simple tool for assessing the trade-offs between energy, speed and variability faced when designing in the sub and near threshold regions.

4.4.1 Current and delay model under variability analysis

In weak and moderate inversion region, the drain current can be expressed as [53]:

$$\begin{aligned} I_{DS} =& I_S(ln^2[1+exp\frac{V_{GS}-V_t}{2nU_T}]- \\ & ln^2[1+exp\frac{V_{GS}-V_t}{2nU_T}exp\frac{-V_{DS}}{2U_T}]) \end{aligned} \tag{4.1}$$

Where n is the subthreshold slope factor, V_{GS} and V_{DS} are respectively the gate to source and drain to source voltages and V_t is the threshold voltage. I_S is the specific current given by

$$I_S = 2n\mu C_{ox}U_T^2 W/L = 2n\beta U_T^2 \tag{4.2}$$

Where μ is the mobility, C_{ox} is the oxide capacitance, U_T is the thermal voltage and W/L denotes the channel width-length ratio of the transistor. Equation 4.3 tends to the classical exponential WI model when $V_{GS} - V_t$ is negative:

$$I_{DS} = I_S exp\frac{V_{GS}-V_t}{nU_T}(1-exp\frac{-V_{DS}}{U_T}) \tag{4.3}$$

The model of Equation 4.1 is a long channel model that does not include effects such as mobility reduction, velocity saturation and drain induced barrier lowering (DIBL). The former two are mainly of impact in the strong inversion region. The last one (DIBL), has some impact on the performance in the WI and MI regions [60]. Nevertheless, for simplicity sake, we will not include DIBL in the model, and as shown later, reasonable agreement

is achieved anyway. Leakage currents can be determined from the I_{DS} expression when $V_{GS} = 0$.

Figure 4.8 shows the I_D versus V_{GS} for NMOS transistor determined with the weak inversion model, the complete model and Spice simulations. As expected, the weak inversion model is not sufficient to model the current in near-threshold region. Moreover, we observe that the complete model is not so accurate in strong inversion region due to the lack of modeling of mobility reduction and velocity saturation [74].

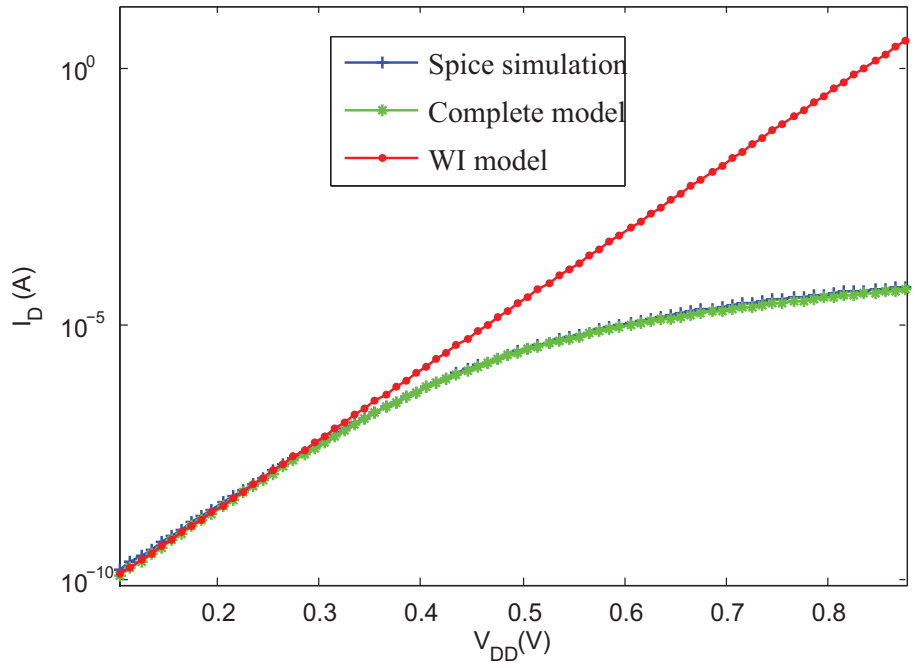


Figure 4.8: I_D versus V_{GS} curves for NMOS transistor.

The expression of inverter's delay can be estimated from the current model as follows:

$$T_d = \frac{C_L V_{DD}}{I_{on}} \tag{4.4}$$

Where C_L is the equivalent load capacitance, V_{DD} is the supply voltage and I_{on} is the saturated on-current. Since the delay is fitted using the equivalent load capacitance value (C_L), it proves to be enough to consider just the characteristic of one transistor type, the nMOS in our case.

To extract model parameters, we have applied the method based on the gm/ID curve described in [74]. The following values were obtained for the nMOS transistor of a basic inverter of the considered 65nm LP industrial library:

- n = 1.22;
- $V_t = 0.38$(V);
- $\beta = 4.83e^{-4}(A/V^2)$.

Now, for variability analysis, we will consider just random V_t and β variations, modeled as a normal distributions with parameters (μ_{V_t}, σ_{V_t}, μ_β, σ_β), determined through Monte Carlo simulations. Current model considering process variations is derived from Equation 4.1 by replacing β and V_t by values that follow the normal distribution. We do not consider global variations in this work. As shown in [72], adaptive body biasing can be efficiently used to compensate such variations.

It is well known that the current and hence the delay are normally distributed in the Strong Inversion region. It is also noted that in the Weak Inversion region, they have lognormal distribution. However, what is less known is what distribution have they in the moderate inversion region. This is very essential to correctly model the circuit characteristics as they have different expressions depending on the considered distribution.

4.4.2 Current and delay distribution in different operating regions

A normal distribution is a very familiar statistic distribution. Its probability density function (PDF) is given by :

$$P(X) = \frac{1}{\sigma\sqrt{(2\pi)}} \exp^{\frac{-1}{2}(\frac{x-\mu}{\sigma})^2} \tag{4.5}$$

Where μ and σ represent the mean and the standard deviation of the variable X, respectively. A lognormal distribution is not as well-known as the normal one. But they are closely related. If $X = log(Y)$ is normally distributed with mean μ and standard deviation σ, then the random variable Y is a lognormal distribution characterized by the mean m and the variance v given by :

$$m = \exp(\mu + \frac{\sigma^2}{2}) \tag{4.6}$$

$$v = \exp(2\mu + \sigma^2)(exp(\sigma^2) - 1) \tag{4.7}$$

Figure 4.9 (a) and (b) plot the PDF of a lognormal and a normal distribution with μ=0 and $\sigma = 1$, respectively.

Visually, what differentiate the two distributions is the symmetry: while the normal distribution looks symmetric, the lognormal distribution is highly assymetric with a tail. However, this might not be always obviously remarked. For instance, Figure 4.10 shows the delay distribution of 32-bit adder for different supply voltages from a 1k-point Monte Carlo simulation with mismatch effect. As expected, due to the exponential dependence of the sub-threshold currents on V_t, the delay histogram in the WI region ($V_{DD} = 0.2$) resembles the one of lognormal distribution. It is also clear that the delay in the SI region (V_{DD}=1.2V) looks like normally distributed and this is due to the linear dependence of the currents on V_t in this region. However, the judgment is more difficult when looking at

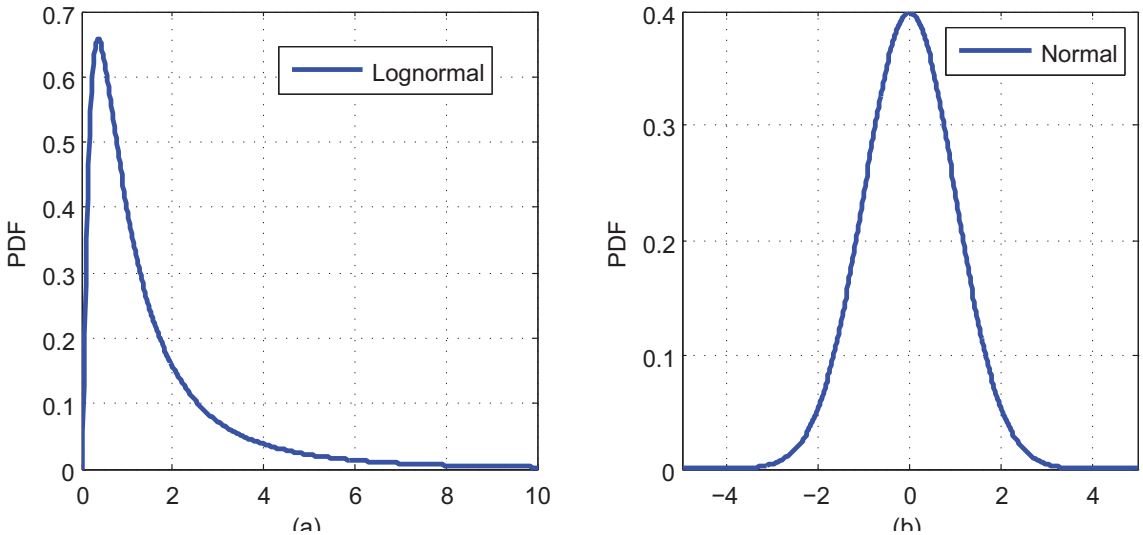


Figure 4.9: (a) Lognormal distibution, (b) Normal distribution.

the delay distribution for V_{DD}=0.4V and V_{DD}=0.6V.
To find the distribution that better fits our data, we have used QQ-plots [75], a graphical method that compare two distribution by plotting their percentiles against each other. For instance, Figure 4.11 (a) and (b) show the QQ-plots of the Monte Carlo data of the 32-bit adder delay for V_{DD}=0.2V fitted using normal and lognormal distribution, respectively. We observe that the lognormal distribution is effectively the best fit for the delay data on sub-threshold region.

We have applied QQ-plots technique for all delay data in each supply voltages to know what distribution gave the best fit in the moderate inversion region. The results show that the distribution from V_{DD}=0.2V until V_{DD}=0.7V is better fitted with a lognormal distribution. Beyond V_{DD}=0.7V, the data will be considered as normally distributed.

4.4.3 Model vs. the logic depth

For a complex circuit with a logic depth L_D corresponding to the number of inverter delays that compose the critical path of the circuit, the delay will be:

$$T_{circuit} = L_D.T_d \tag{4.8}$$

Simulating the circuit's critical path delay and normalizing to the delay of an inverter with the same supply voltage provides L_D.

As previously stated, the current and hence the delay can have normal or lognormal distribution depending on the operating region. Consequently, the delay model will have different expressions depending on the nature of the considered distribution.

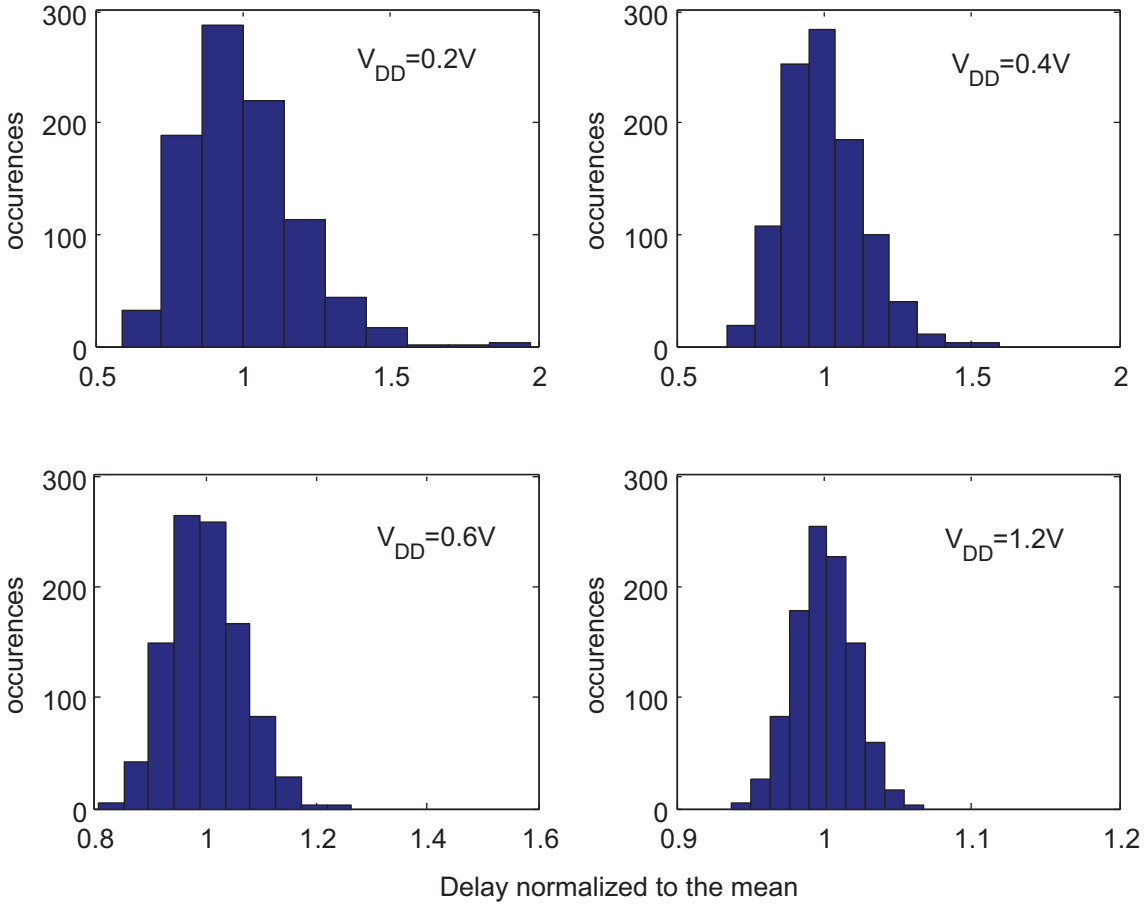


Figure 4.10: 32-bit adder delay distribution for different supply voltages.

Case of normal distribution If T_d has a normal distribution defined by (μ_{delay}, σ_{delay}), $T_{circuit}$, which is the sum of L_D normally distributed T_d, will be a normal distribution too defined by ($L_D.\mu_{delay}$, $\sqrt{(L_D)}.\sigma_{delay}$). Thus, the delay variability($\sigma_{delay}/\mu_{delay}$) of the circuit can be obtained as follow:

$$var_{circuit-delay} = (1/\sqrt{(L_D)}).var_{inverter-delay} \tag{4.9}$$

Table 4.2 lists delay variability for different Logic depths at $V_{DD} = 1.2V$.

Table 4.2: Delay variability for different logic depths at V_{DD}=1.2 V.

L_D	Delay variability ($\sigma_{delay}/\mu_{delay}$)	
	Spice simulation	**Analytical model**
1	0.085	—
8	0.025	0.03
15	0.019	0.021
23	0.014	0.016

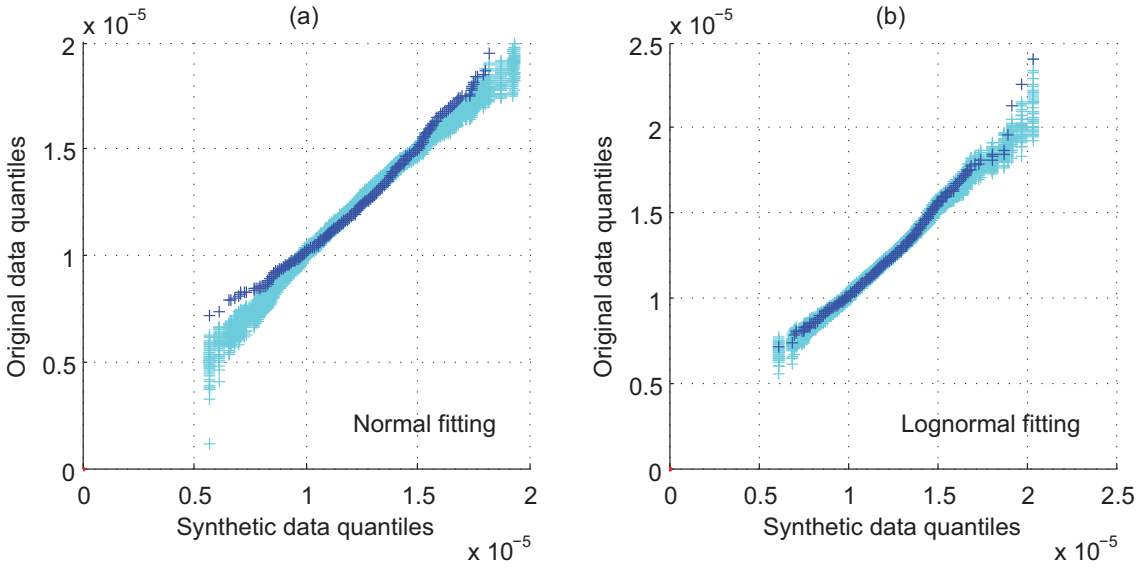


Figure 4.11: Fitting of the Monte Carlo data of the 32-bit adder delay for V_{DD}=0.2V (a) normal distribution (b) lognormal distribution. The blue points are original data while the green points correspond to a set of 10000 synthetic data forming the enveloppe-QQplot.

As expected, the delay variability of a circuit decreases as its logic depth increases, and the decrease follows perfectly $1/\sqrt{(L_D)}$ law.

Case of lognormal distribution If T_d is lognormally distributed, the sum of several lognormal random variables can be approximated by another lognormal random variable as shown in [76]. Matching the first moments as shown in [76] and better explained in [77], we obtain

$$\mu(lnT_{circuit}) = \mu_{log} + \frac{1}{2}\sigma_{log}^2 + \frac{1}{2}ln\frac{L_D^3}{L_D - 1 + exp(\sigma_{log}^2)} \tag{4.10}$$

$$\sigma(lnT_{circuit}) = \sqrt{ln\left(1 + \frac{1}{L_D}(exp(\sigma_{log}^2) - 1)\right)} \tag{4.11}$$

Where μ_{log} and σ_{log} are respectively the mean and the standard deviation of the normal distribution $log(T_d)$. See Appendix A for calculation details to arrive to these equations.

Similar equations have been shown in the work of Zhai [78]. However, while they consider just V_t variations with WI models, our concern is to derive a model of generic circuit based on the characteristic of an inverter.

We use Equation 4.6 and Equation 4.7 to calculate respectively the mean and the standard deviation of the distribution $T_{circuit}$ itself.

Figure 4.12 shows delay variability of a chain of inverters with different logic depths at $V_{DD} = 0.2V$ and $V_{DD} = 0.5V$. We observe that the model closely matches Spice simulation for both of the supply voltages.

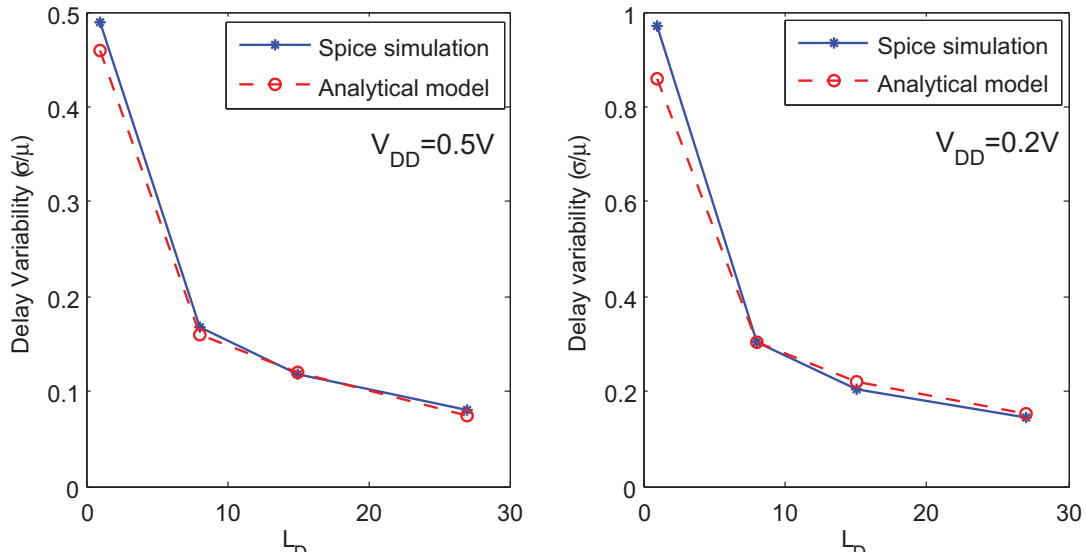


Figure 4.12: Delay variability at sub-threshold voltages($V_{DD} = 0.5V, V_{DD} = 0.2V$) for different logic depths

One of the interesting parameter is the 3σ worst-case delay defined as the value from which just 0.3% of the data are above. This parameter is also computed differently depending on the considered distribution. Indeed, if the delay T_d is normally distributed with mean μ and standard deviation σ, the 3σ worst case delay is calculated as :

$$T_{d,3\sigma} = \mu + 3\sigma \tag{4.12}$$

Otherwise, for V_{DD} where T_d follows a lognormal distribution with mean μ and standard deviation σ , the 3σ worst case T_d is given by :

$$T_{d,3\sigma} = exp(\mu_{log} + 3\sigma_{log}) \tag{4.13}$$

Where μ_{log} and σ_{log} are respectively the mean and the standard deviation of the normal distribution $log(T_d)$ and they can be computed as follows :

$$\mu_{log} = ln\left(\frac{\mu^2}{\sqrt{\sigma^2 + \mu^2}}\right) \tag{4.14}$$

$$\sigma_{log} = \sqrt{ln\left(\frac{\sigma^2}{\mu^2} + 1\right)} \tag{4.15}$$

Figure 4.13 shows the evolution of typical and 3σ Worst Case (WC) delay of a chain of inverters with logic depth $L_D = 23$. As expected, the delay increases exponentially in the sub-threshold region. We observe that the complete model with and without variability

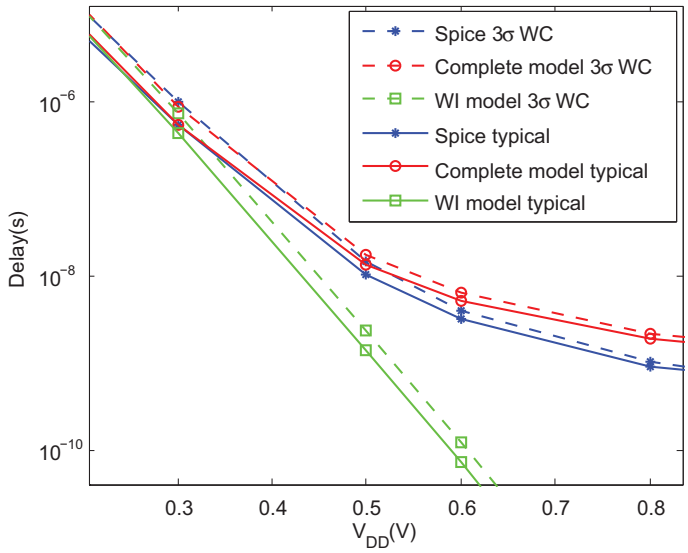


Figure 4.13: Evolution of typical and 3σ WC delay of a chain of inverters with logic depth $L_D = 23$ for different V_{DD} .

consideration follow perfectly Spice simulations whereas the delay of the WI model deviates from $0.3V$ of V_{DD}. This proves first that the proposed variability model considering just V_t and β variations predicts well the effect of mismatch on the circuit delay. And second that the WI model is a limited one for near-threshold voltages and above modeling. This is further proven by Figure B.1 where the delay variability obtained with the WI model, remains constant at different supply voltages, while the one obtained with the complete model presents the same shape as Spice simulations and has even close values.

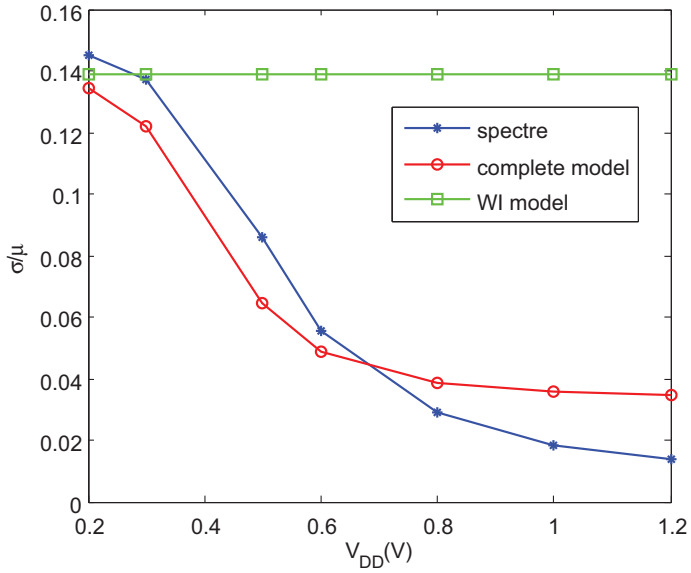


Figure 4.14: Delay variability for different V_{DD}(V) of a chain of inverters with $L_D = 23$.

Figure 4.15 shows how the WC delay determined by Zhai equations where just V_t variations are considered, is less accurate than the one derived by our model taking into account both V_t and β variations. We conclude that β effects on delay variability are not negligible.

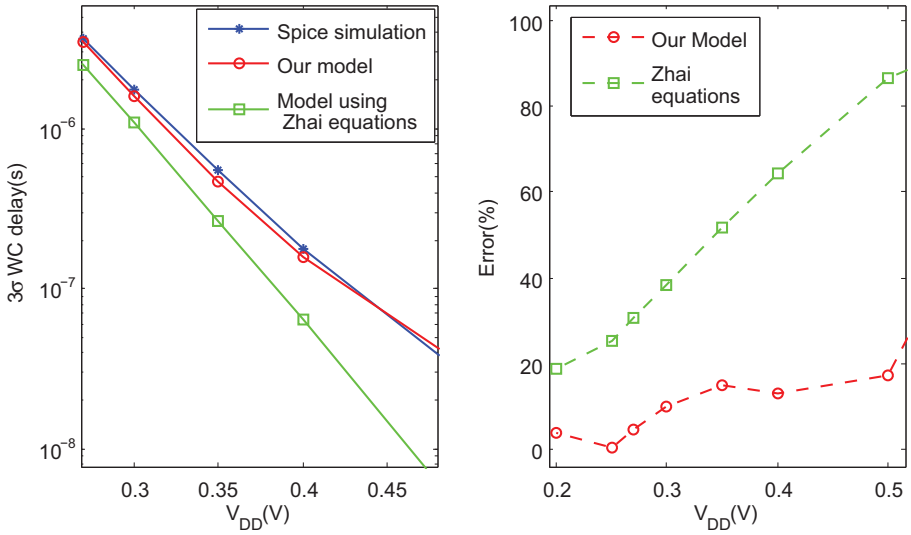


Figure 4.15: Comparison of our model (V_t and β variations) and the model with Zhai equations (V_t variations) .

Test case : 32-bit adder The considered benchmark circuit is a 32-bit carry save adder used in a FIR filter for biomedical application. It has been synthesized in a 65nm Low Power process technology. It has 55 logic gates (inverters, Nand, FA, HA...).

Table 4.3 lists all values of the parameters of our model for this benchmark circuit. L_D is obtained by normalizing the delay of the circuit to that of the inverter at $V_{DD} = 0.2$V. σ_{V_t} and σ_β are obtained by 1K Monte-Carlo simulation.

Table 4.3: Model parameters values for 32-bit adder.

Prameter	value
n	1.22
U_T	0.026
β	4.83e-4 (A/V^2)
V_t	0.38 (V)
σ_{V_t}	20e-3 (V)
σ_β	80e-6 (A/V^2)
L_D	50
C_{eff}	8.34e-14 (F)
W_{eff}	300

Figure 4.16 compares delay models (WI and complete) vs Spice simulation for the considered benchmark circuit.

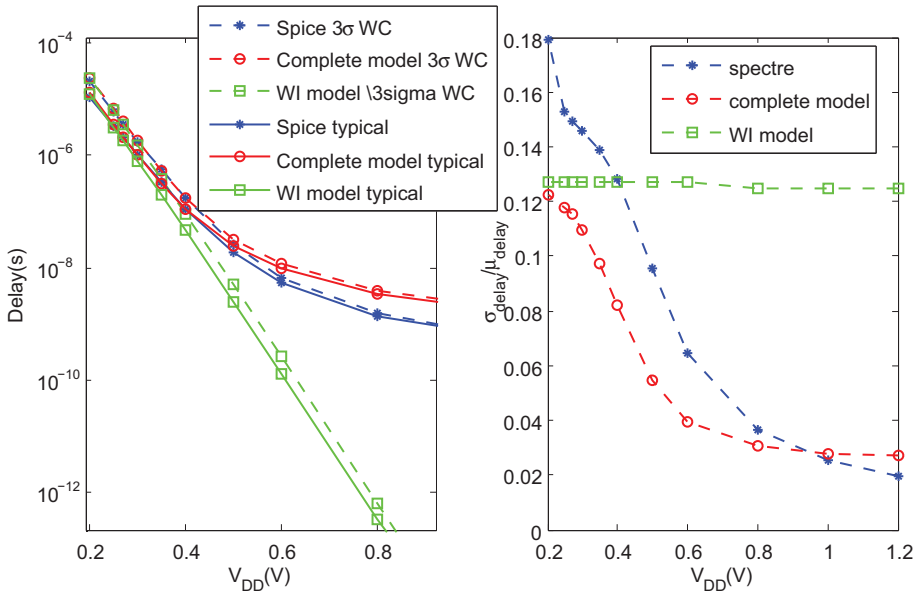


Figure 4.16: Evolution of typical and 3σ WC delay for the 32-bit adder (right) and the delay variability ($\sigma_{delay}/\mu_{delay}$) obtained by Spice simulation, complete model and WI model (left).

4.4.4 Energy model under variability analysis

The total energy consumed by the circuit is the sum of the dynamic energy E_{dyn}, required to charge and discharge parasitic capacitances during logic transitions, and static energy E_{stat} due to leakage currents (see section 2.2). This can be summarized in the following expression as shown by Calhoun in [79]:

$$E_{tot} = C_{eff}V_{DD}^2 + V_{DD}W_{eff}I_{leak}T_{circuit} \tag{4.16}$$

Where C_{eff} denote the average switched capacitance of the circuit. It is estimated by simulating the circuit for a typical simulation and solving $C_{eff} = I_{avg}T_{circuit}/V_{DD}$. W_{eff} denotes the normalized width that contributes to leakage current. It is calculated by measuring the average leakage current of the circuit normalized to the inverter's leakage current (I_{leak}).

Without loss of generality, we consider $Just-in-time$ operation [80], where the circuit works in its maximum frequency, i.e., the period is set to be the critical path delay of the circuit.

To consider variability in energy analysis, 3σ worst-case delay and mean I_{leak} are

considered in the static energy calculation as follows :

$$E_{stat} = V_{DD}\mu_{I_{leak}}T_{circuit,3\sigma} \tag{4.17}$$

Fig. 4.17 shows the consumed energy under process variations consideration of the benchmark 32-bit Carry Save Adder (CSA).

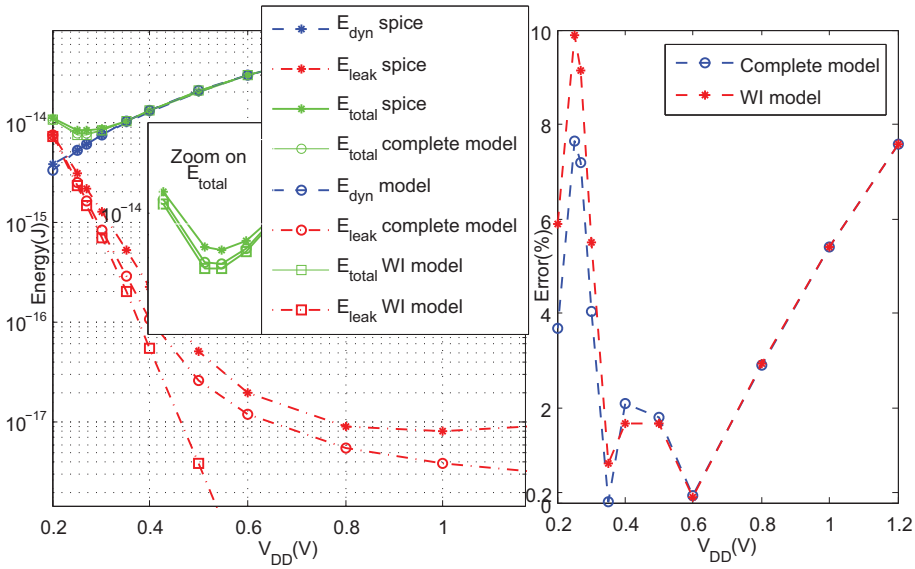


Figure 4.17: Consumed energy under process variations (left), and % of complete and WI model error compared to Spice simulations (right) of 32-bit CSA.

We observe that the total energy consumed is slightly different from that determined by the models. The error of the energy on the minimum point is of 7% and 9%, when obtained by the complete and the WI model, respectively.

Not what we expect, the error of the WI model is comparable to that of the complete model as shown in Figure 4.17 (right). This can be explained by the inverse tendency of variability and delay determined by the WI model. On the one hand, the variability of the WI model is constant whatever the value of the supply voltage. It is therefore overestimated in the moderate and strong inversion regions. On the other hand, the delay obtained by the WI model is underestimated with respect to the one obtained by Spice simulations as observed in Figure 4.13. As the energy contains the product of variability and the delay, there is a compensation that let the WI model remain a good model of energy consumption even under variability analysis.

4.4.5 Analytical solution of minimum energy point under variability analysis

In this section, we derive an analytical expression for the optimum voltage V_{DDopt} corresponding to the minimum energy point under variability analysis. In [79], authors have provided such expression but without considering effects of process variations. To our knowledge, this is the first attempt to derive analytical solution of V_{DDopt} while considering mismatch effects.

The expression, determined by Calhoun in [79], is a solution of a set of equations derived from a WI model. As we have demonstrated in section 4.4.4 that the WI model is still a good model for energy consumption under variability consideration, we will use the simple WI model and we will follow the same steps as in [79].

Let us start with the expression of the total energy E_{tot} shown in Equation 4.16. Under variability consideration, this expression is:

$$E_{tot,var} = C_{eff}V_{DD}^2 + V_{DD}W_{eff}\mu_{leak}T_{circuit,3\sigma} \tag{4.18}$$

Using the WI model as in Equation 4.3, μ_{leak} and $T_{circuit}$ can be written as follow :

$$\mu_{leak} = 2n\beta U_T^2 exp(\frac{-V_t}{nU_T}) \tag{4.19}$$

$$\begin{aligned} T_{circuit} &= L_D T_d \\ &= L_D \frac{CV_{DD}}{2n\beta U_T^2 exp(\frac{V_{DD}-V_t}{nU_T})} \end{aligned} \tag{4.20}$$

Since we know that $T_{circuit}$ is a lognormal distribution in the WI region, the 3σ WC $T_{circuit}$ is

$$T_{circuit,3\sigma} = exp(\mu_{log,circuit} + 3\sigma_{log,circuit}) \tag{4.21}$$

Where $\mu_{log,circuit} = mean(log(T_{circuit}))$ and $\sigma_{log,circuit} = StdDev(log(T_{circuit})$ and they can be written as a function of μ_{log} and σ_{log} the mean and the standard deviation of the delay of one inverter T_d, as shown in Equation 4.10 and Equation 4.11, respectively.

Applying the function logarithm to T_d, we get

$$\mu_{log} = ln(V_{DD}) - \frac{V_{DD}}{nU_T} + ln\left(\frac{C}{2n\beta U_T^2}\right) + \frac{V_t}{nU_T} \tag{4.22}$$

$$\sigma_{log} = \sqrt{\frac{1}{(nU_T)^2}\sigma_{V_t}^2 + \frac{\sigma_{beta}^2}{\mu_{beta}^2}} \tag{4.23}$$

Now, substituting Equation 4.19 and Equation 4.21 into Equation 4.18 gives the expression

of total energy under variability analysis $E_{tot,var}$, which can be written as follows:

$$\begin{aligned} E_{tot,var} &= C_{eff}V_{DD}^2 + AV_{DD}\left[exp\left(ln(V_{DD}) - \frac{V_{DD}}{nU_T} + B\right)\right] \\ &= C_{eff}V_{DD}^2 + AV_{DD}^2 exp\left(B - \frac{V_{DD}}{nU_T}\right) \end{aligned}$$

$Where$

$$\begin{aligned} A &= 2n\beta U_T^2 W_{eff} exp\left(\frac{-V_t}{nU_T}\right) \\ B &= \frac{V_t}{nU_T} + ln\left(\frac{C}{2n\beta U_T^2}\right) + 0.5ln\frac{L_D^3}{L_D - 1 + exp\left(\sqrt{\frac{\sigma_{V_t}^2}{(nU_T^2)} + \frac{\sigma_{beta}^2}{\mu_{beta}^2}}\right)} \\ &\quad + 0.5\sqrt{\frac{\sigma_{V_t}^2}{(nU_T^2)} + \frac{\sigma_{beta}^2}{\mu_{beta}^2}} + 3\sqrt{ln\left(1 + \frac{1}{L_D}\left(exp\left(\sqrt{\frac{\sigma_{V_t}^2}{(nU_T)^2} + \frac{\sigma_{beta}^2}{\mu_{beta}^2}}\right) - 1\right)\right)} \end{aligned} \tag{4.24}$$

To search for V_{DDopt} that gives the minimum energy point, we solve $\frac{\partial E_{tot,var}}{\partial V_{DD}} = 0$. The derivative of $E_{tot,var}$ with respect to V_{DD} is:

$$\frac{\partial E_{tot,var}}{\partial V_{DD}} = 2C_{eff}V_{DD} + AV_{DD}(2 - \frac{V_{DD}}{nU_T})exp(B - \frac{V_{DD}}{nU_T}) \tag{4.25}$$

Equating Equation 4.25 to 0 and making a simple change of variables yields:

$$(2 - \frac{V_{DDopt}}{nU_T})exp(2 - \frac{V_{DDopt}}{nU_T}) = \frac{-2C_{eff}}{Aexp(B-2)} \tag{4.26}$$

Therefore, the analytical expression of V_{DDopt} is :

$$V_{DDopt} = nU_T\left(2 - lambertW\left(\frac{-2C_{eff}}{Aexp(B-2)}\right)\right) \tag{4.27}$$

Where, W=lambertW(x) function is the solution to $Wexp(W) = x$, see Appendix B for more details about lambertW function and it's constraints.

Analytical solution vs.Simulation Calculating V_{DDopt} from Equation 4.27 for the considered 32-bit adder benchmark circuit, we find the optimum V_{DD} values under variability analysis equals to 0.252V which corresponds to just 6% error from the value obtained by Spice simulation (0.27V).

4.5 Conclusion

This chapter has examined modeling under variability consideration for sub-threshold operation. We have shown that in a variability aware analysis a complete model is needed to correctly estimate the delay performance in the near-threshold region. However, we have noticed that for energy modeling the WI model remain a good one for both sub and near-threshold operations. Thus, we have used the simple equations of the WI model to introduce an analytical solution for the optimum V_{DD} that minimizes the total energy considering process variation effects. The analytical value matched Spice simulation to within 6%.

Conclusions

The work described in this thesis investigated several aspects of "Low Energy Design". Two major contributions are presented in this dissertation with the eventual goal of helping the advances of Low energy design. This section summarizes each contribution and discusses possible directions for future work.

First, glitch-less area is investigated. A new methodology for glitch analysis using post-layout timing report is demonstrated. To reduce glitches, we propose to use high threshold voltage to increase the inertial delays of "glitchy" gates. However, low threshold voltage should be used for gates in the critical path to maintain the required performance. Using this dual–V_t assignment, the developed heuristic algorithm achieves 16% average glitch reduction when tested on 6 ISCAS85 benchmark circuits. The implementation of the combined dual–V_t/gate-sizing algorithm shows a better glitch filtering. Together, they achieve up to 15% total energy reduction compared to gate-sizing optimization.

Although the glitch analysis tool proposed in this work is not very accurate compared to Spice simulation, it can be used to make rapid comparison or decision about the combination of glitch reduction techniques (path-balacing, gate–sizing, dual–V_t) that better minimizes the total energy. But, clearly more investigations in CAD tools for glitch analysis and optimization would be helpful to circuit and systems designers. For instance, the proposed tool in this work can be more extended to take into consideration the characteristics of the circuit such as the operating frequency and the idle time and the characteristics of the used library such as gate delays and capacitances, to make a choice about the appropriate combination that achieves minimum energy consumption.

The second part of this work is devoted to sub-threshold operation considered as an ideal solution for Ultra-Low-Energy applications that do not require high performance. Modeling and characterization of sub-threshold circuits under variability aware analysis were investigated. First, it was demonstrated that a complete model that take into account moderate inversion region is necessary to correctly estimate the delay performance when considering process variations. However, it was shown that, even under variations effects, the restricted WI model remains a good model for analyzing the minimum energy point. Hence, the simple equations of the WI model were used to develop an analytical ex-

pression of the optimum V_{DD} that gives the minimum energy point under variability-aware design. Simulations on a 32-bit adder synthesized in a 65nm industrial library matched the complete model for the delay analysis and the analytical expression for the optimum V_{DD} within a few percent.

Further exploration of sub-threshold operation under variability considerations would help for designing more robust sub-threshold circuits. Several directions for future work remain opened. First, modeling and characterization should be extended to take into account synchronous elements. For instance, timing characterization of Flip-Flops under variability considerations would be helpful to verify timing constraints in sub-threshold logic. Another opportunity for future work would involve CAD tools for sub-threshold analysis. Such tools would help to verify large circuits operating in sub-threshold logic while spending a reasonable lapse of time.

The research performed during this thesis originated several publications as can be seen bellow.

- **Analyse d'architectures de multiplieurs en vue de la basse consommation** [81] - Poster presentation at JNRDM, Marseille, France, 2010;
- **Multiple threshold voltage for glitch power reduction** [82] - Oral presentation at the 10th edition of IEEE Low Voltage Low Power (FTFC) 2011;
- **Variability-speed-consumption trade-off in near threshold operation** [83] - Oral presentation at the 21st Power and Timing Modeling, Optimization and Simulation (PATMOS) 2011;
- **A Dual threshold voltage technique for glitch minimization** [84], - Oral presentation at the 19th IEEE International Conference on Electronics, Circuits, and Systems (ICECS) 2012;
- **Variability aware modeling for sub-threshold circuits** - To be submitted to Journal of Low Power Electronics and Applications (JLPA).

Appendix A

Calculation details for equation 4.10 and equation 4.11

For a complex circuit with a logic depth L_D corresponding to the number of inverter delays that compose the critical path of the circuit, the delay will be:

$$T_{circuit} = L_D.T_d \tag{A.1}$$

If T_d is lognormally distributed with mean μ_{log} and standard deviation σ_{log}, $T_{circuit}$, the sum of several lognormal random variables, can be approximated by another lognormal random variable as shown in [76].

The mean and the variance of $ln(T_{circuit})$ denoted respectively $(\mu_{lnTcircuit}, \sigma_{lnTcircuit})$, are obtained by matching the first moments (m_1, m_2) as explained in [77], where :

$$\begin{aligned} m_1 = E(T_{circuit}) &= exp(\mu_{lnTcircuit} + \frac{\sigma^2_{lnTcircuit}}{2}) \\ &= \sum_{i=1}^{L_D} exp(\mu_{log} + \frac{\sigma^2_{log}}{2} \end{aligned} \tag{A.2}$$

$$m_2 = E(T^2_{circuit}) = exp(2\mu_{lnTcircuit} + \sigma^2_{lnTcircuit}) \tag{A.3}$$

Resolving this set of equations, we obtain :

$$\begin{cases} \mu_{lnTcircuit} = 2ln(m_1) - 0.5ln(m_2) \\ \sigma^2_{lnTcircuit} = ln(m_2) - 2ln(m_1) \end{cases} \tag{A.4}$$

It is shown in [77] that $\sigma_{lnTcircuit}$ has the following expression:

$$\sigma_{lnTcircuit} = ln\left(1 + \frac{1}{L_D}\left(exp(\sigma^2_{log}) - 1\right)\right) \tag{A.5}$$

Hence, from equation A.4 $\mu_{lnTcircuit}$ will be:

$$\mu_{lnTcircuit} = ln(m_1) - 0.5\sigma_{lnTcircuit} \tag{A.6}$$

Let us calculate $ln(m_1)$:

$$\begin{aligned} ln(m_1) &= ln\left(\sum_{i=1}^{L_D} exp(\mu_{log} + \frac{\sigma_{log}^2}{2})\right) \\ &= ln\left(L_D exp\left(\mu_{log} + \frac{\sigma_{log}^2}{2}\right)\right) \\ &= ln(L_D) + \mu_{log} + 0.5\sigma_{log}^2 \end{aligned} \tag{A.7}$$

Substituing equation A.7 and equation A.5 into equation A.6, $\mu_{lnTcircuit}$ can be written as:

$$\begin{aligned} \mu_{lnTcircuit} &= \mu_{log} + 0.5\sigma_{log}^2 + ln(L_D) - 0.5ln\left(1 + \frac{1}{L_D}\left(exp(\sigma_{log}^2) - 1\right)\right) \\ &= \mu_{log} + 0.5\sigma_{log}^2 + 0.5ln(L_D)^2 - 0.5ln\left(\frac{L_D + \left(exp(\sigma_{log}^2) - 1\right)}{L_D}\right) \\ &= \mu_{log} + 0.5\sigma_{log}^2 + 0.5ln(L_D)^2 + 0.5ln\left(\frac{L_D}{L_D + \left(exp(\sigma_{log}^2) - 1\right)}\right) \\ &= \mu_{log} + 0.5\sigma_{log}^2 + 0.5ln\left(\frac{L_D^3}{L_D + \left(exp(\sigma_{log}^2) - 1\right)}\right) \\ &= \mu_{log} + 0.5\sigma_{log}^2 + 0.5ln\left(\frac{L_D^3}{L_D - 1 + exp(\sigma_{log}^2)}\right) \end{aligned} \tag{A.8}$$

Appendix B

LambertW function

This appendix is based on Appendix B of Calhoun's dissertation [85]. It is reprised here in order to ease the reading on this thesis.

The lambertW function W=lambertW(x) function is the solution to $Wexp(W) = x$, in the same way that W=ln(x) is the solution to exp(w)=x.

Figure plots lambertW and Wexp(W) functions. For real $x \geq 0$, the function W=lambertW(x) has exactly one real solution. For real $-e^{-1} < x < 0$, there are exactly two real solutions, called branches. The upper branch increases monotonically in $[-1, \infty]$ for $x \in [-e^{-}1, \infty]$, and the lower brach decreases monotonically in $[-1, -\infty]$ for $x \in [-e^{-}1, 0]$. For the solution in equation 4.27, the argument to lambertW is always negative, so two real solutions exist. The lower branch gives the minimum energy solution, and the upper branch solution is the local maximum.

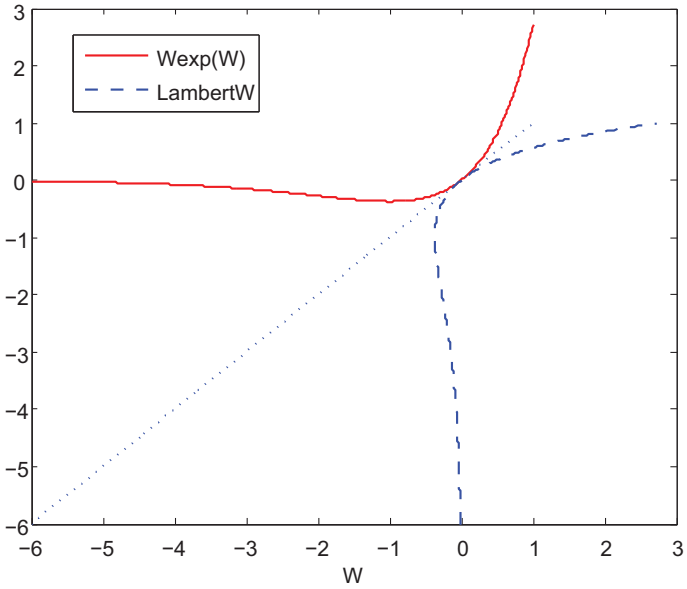


Figure B.1: The lambertW function, W=lambertW(x), gives the solution to Wexp(w)=x.

Bibliography

[1] A. Sultania, D. Sylvester, and S. Sapatnekar, "Transistor and pin reordering for gate oxide leakage reduction in dual t_{ox} circuits," in *Computer Design: VLSI in Computers and Processors, 2004. ICCD 2004. Proceedings. IEEE International Conference on*, pp. 228–233, IEEE, 2004.

[2] R. Landauer, "Irreversibility and heat generation in the computing process," *IBM journal of research and development*, vol. 5, no. 3, pp. 183–191, 1961.

[3] C. Bennett, "Logical reversibility of computation," *IBM journal of Research and Development*, vol. 17, no. 6, pp. 525–532, 1973.

[4] S. Younis, *Asymptotically zero energy computing using split-level charge recovery logic*. PhD thesis, Massachusetts Institute of Technology, 1994.

[5] E. Fredkin and T. Toffoli, "Conservative logic," *International Journal of Theoretical Physics*, vol. 21, no. 3, pp. 219–253, 1982.

[6] P. Matherat, "Où en est-on de la dissipation du calcul? retour à bennett," *Annals of Telecommunications*, vol. 62, no. 5, pp. 690–713, 2007.

[7] W. Porod, R. Grondin, D. Ferry, and G. Porod, "Dissipation in computation," *Physical Review Letters*, vol. 52, no. 3, pp. 232–235, 1984.

[8] D. Vasudevan, P. Lala, J. Di, and J. Parkerson, "Reversible-logic design with online testability," *Instrumentation and Measurement, IEEE Transactions on*, vol. 55, no. 2, pp. 406–414, 2006.

[9] T. Hisakado, H. Iketo, and K. Okumura, "Logically reversible arithmetic circuit using pass-transistor," in *Circuits and Systems, 2004. ISCAS'04. Proceedings of the 2004 International Symposium on*, vol. 2, pp. II–853, IEEE, 2004.

[10] M. Pedram, "Power simulation and estimation in vlsi circuits," *The VLSI handbook*, pp. 18–27, 1999.

[11] H. Veendrick, "Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits," *Solid-State Circuits, IEEE Journal of*, vol. 19, no. 4, pp. 468–473, 2002.

[12] Y. I. Ismail, E. G. Friedman, and J. L. Neves, "Dynamic and short-circuit power of cmos gates driving lossless transmission lines," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 46, no. 8, pp. 950–961, 1999.

[13] W. Liu, "Techniques for Leakage Power Reduction in Nanoscale Circuits: A Survey," *IMM Report, Dept. of Informatics and Mathematical Modeling, Technical University of Denmark*, 2007.

[14] X. Qi, S. Lo, A. Gyure, Y. Luo, M. Shahram, K. Singhal, and D. MacMillen, "Efficient subthreshold leakage current optimization - leakage current optimization and layout migration for 90- and 65- nm asic libraries," *Circuits and Devices Magazine, IEEE*, vol. 22, pp. 39 –47, sep. 2006.

[15] S. Mohanty, N. Ranganathan, E. Kougianos, and P. Patra, *Low-Power High-Level Synthesis for Nanoscale CMOS Circuits*. Springer Verlag, 2008.

[16] C. Piguet, *Low-power electronics design*. CRC, 2005.

[17] `http://www.itrs.net`.

[18] F. Najm, "A survey of power estimation techniques in vlsi circuits," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 2, no. 4, pp. 446–455, 1994.

[19] D. Marculescu, R. Marculescu, and M. Pedram, "Information theoretic measures for power analysis [logic design]," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 15, no. 6, pp. 599–610, 1996.

[20] K. Muller-Glaser, K. Kirsch, and K. Neusinger, "Estimating essential design characteristics to support project planning for asic design management," in *Computer-Aided Design, 1991. ICCAD-91. Digest of Technical Papers., 1991 IEEE International Conference on*, pp. 148–151, IEEE, 1991.

[21] P. Landman and J. Rabaey, "Activity-sensitive architectural power analysis for the control path," in *Proceedings of the 1995 international symposium on Low power design*, pp. 93–98, ACM, 1995.

[22] S. Kang, "Accurate simulation of power dissipation in vlsi circuits," *Solid-State Circuits, IEEE Journal of*, vol. 21, no. 5, pp. 889–891, 1986.

[23] S. Devadas and S. Malik, "A survey of optimization techniques targeting low power vlsi circuits," *32nd ACM/IEEE Design Automation Conference*, 1995.

[24] J. Kao, M. Miyazaki, and P.Chandrakasan, "A 175-mv multiply-accumulate unit using an adaptive supply voltage and body bias architecture," *IEEE Journal of Solid-State Circuits*, vol. 37, November 2002.

[25] D. Soudris, "Circuits Techniques for Dynamic Power Reduction," *Low-power electronics design*, 2005.

[26] C. Piguet, *Low-power electronics design*, vol. 1. CRC, 2004.

[27] K. Usami and M. Horowitz, "Clustered voltage scaling technique for low-power design," in *ISLPED '95: Proceedings of the 1995 international symposium on Low power design*, (New York, NY, USA), pp. 3–8, ACM, 1995.

[28] M. Borah, R. M. Owens, and M. J. Irwin, "Transistor sizing for low power cmos circuits," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 15, no. 6, pp. 665–671, 1996.

[29] R. Hossain, M. Zheng, and A. Albicki, "Reducing power dissipation in serially connected mosfet circuits via transistor reordering," in *ICCS '94: Proceedings of the1994 IEEE International Conference on Computer Design: VLSI in Computer & Processors*, (Washington, DC, USA), pp. 614–617, IEEE Computer Society, 1994.

[30] E. Musoll and J. Cortadella, "Optimizing CMOS circuits for low power using transistor reordering," in *Proceedings of the 1996 European conference on Design and Test*, p. 219, IEEE Computer Society, 1996.

[31] A. A. R. Hossain, M. Zheng, "Reducing power dissipation in cmos circuits by signal probability based transistor reordering," in *IEEE Transactions on Computer-Aided Design of integrated Circuits And Systems*, vol. 15, March 1996.

[32] M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi, *Low power methodology manual: for system-on-chip design*. Springer Publishing Company, Incorporated, 2007.

[33] S. Devadas, J. Carlos, J. Monteiro, and A. Monteiro, "A Computer-Aided Design Methodology for Low Power Sequential Logic Circuits," 1996.

[34] K. Yeo and K. Roy, *Low voltage, low power VLSI subsystems*. McGraw-Hill Professional, 2005.

[35] L. Wei, Z. Chen, K. Roy, M. Johnson, Y. Ye, and V. De, "Design and optimization of dual-threshold circuits for low-voltage low-power applications," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 7, no. 1, pp. 16–24, 2002.

[36] Y. Lu and V. Agrawal, "Leakage and dynamic glitch power minimization using integer linear programming for v th assignment and path balancing," *Integrated Circuit and System Design. Power and Timing Modeling, Optimization and Simulation*, pp. 909–909, 2005.

[37] P. Pant, R. Roy, and A. Chattejee, "Dual-threshold voltage assignment with transistor sizing for low power cmos circuits," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 9, no. 2, pp. 390–394, 2001.

[38] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-v power supply high-speed digital circuit technology with multithreshold-voltage cmos," *Solid-State Circuits, IEEE Journal of*, vol. 30, no. 8, pp. 847–854, 1995.

[39] I. Group *et al.*, "Itrs report on system drivers," 2007.

[40] T. Raja, V. Agrawal, and M. Bushnell, "Variable input delay cmos logic for low power design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 10, pp. 1534–1545, 2009.

[41] M. Hashimoto, H. Onodera, and K. Tamaru, "A power optimization method considering glitch reduction by gate sizing," in *Low Power Electronics and Design, 1998. Proceedings. 1998 International Symposium on*, pp. 221–226, IEEE, 2005.

[42] S. Kim, J. Kim, and S.-Y. Hwang, "New path balancing algorithm for glitch power reduction," *Circuits, Devices and Systems, IEE Proceedings* -, vol. 148, pp. 151 –156, jun. 2001.

[43] V. D. Agrawal, "Low-power design by hazard filtering," in *VLSI Design, 1997. Proceedings., Tenth International Conference on*, pp. 193–197, IEEE, 1997.

[44] V. Agrawal, M. Bushnell, G. Parthasarathy, and R. Ramadoss, "Digital circuit design for minimum transient energy and a linear programming method," in *Proceedings of the Twelfth International Conference On VLSI Design*, pp. 434–439, IEEE, 1999.

[45] `http://lpsolve.sourceforge.net`.

[46] H. Lee, H. Shin, and J. Kim, "Glitch elimination by gate freezing, gate sizing and buffer insertion for low power optimization circuit," in *Proceedings of the 30th Annual Conference of IEEE Industrial Electronics Society, IECON.*, vol. 3, pp. 2126–2131, IEEE, 2004.

[47] O. Coudert, "Gate sizing for constrained delay/power/area optimization," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 5, no. 4, pp. 465–472, 1997.

[48] L. Wang, M. Olbrich, E. Barke, T. Buchner, M. Buhler, and P. Panitz, "A gate sizing method for glitch power reduction," in *IEEE International SOC Conference (SOCC)*, pp. 24–29, IEEE, 2011.

[49] M. Hashimoto, H. Onodera, and K. Tamaru, "A power optimization method considering glitch reduction by gate sizing," in *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 221–226, IEEE, 1998.

[50] Y. Lu and V. Agrawal, "Cmos leakage and glitch minimization for power-performance tradeoff," *Journal of Low Power Electronics*, vol. 2, no. 3, pp. 378–387, 2006.

[51] M. Ketkar and S. Sapatnekar, "Standby power optimization via transistor sizing and dual threshold voltage assignment," in *IEEE/ACM International Conference on Computer Aided Design, ICCAD.*, pp. 375 – 378, nov. 2002.

[52] Y. Lu and V. Agrawal, "Cmos leakage and glitch minimization for power-performance tradeoff," *Journal of Low Power Electronics*, vol. 2, no. 3, pp. 378–387, 2006.

[53] E. Vittoz, "Weak inversion for ultimate low-power logic," in *Low-Power CMOS Circuits, Ed. Christian Piguet, CRC, 2006.*

[54] Y. Tsividis and C. McAndrew, *Operation and Modeling of the MOS Transistor*, vol. 2. Oxford University Press New York, 1999.

[55] D. Markovic, C. Wang, L. Alarcon, and J. Rabaey, "Ultralow-power design in near-threshold region," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 237–252, 2010.

[56] A. Wang and A. Chandrakasan, "A 180-mV subthreshold FFT processor using a minimum energy design methodology," *Solid-State Circuits, IEEE Journal of*, vol. 40, no. 1, pp. 310–319, 2004.

[57] S. Hanson, B. Zhai, D. Blaauw, D. Sylvester, A. Bryant, and X. Wang, "Energy optimality and variability in subthreshold design," in *Low Power Electronics and Design, 2006. ISLPED'06. Proceedings of the 2006 International Symposium on*, pp. 363–365, IEEE, 2006.

[58] J. Kwong and A. Chandrakasan, "Variation-driven device sizing for minimum energy sub-threshold circuits," in *Proceedings of the 2006 international symposium on Low power electronics and design*, pp. 8–13, ACM, 2006.

[59] D. Bol, D. Kamel, D. Flandre, and J. Legat, "Nanometer MOSFET effects on the minimum-energy point of 45nm subthreshold logic," in *Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design*, pp. 3–8, ACM, 2009.

[60] 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*, vol. 17, no. 10, pp. 1508–1519, 2009.

[61] D. Bol, *Pushing ultra-low-power digital circuits into the nanometer era.* PhD thesis, University of California, 2008.

[62] A. Wang, A. Chandrakasan, and S. Kosonocky, "Optimal supply and threshold scaling for subthreshold cmos circuits," in *VLSI, 2002. Proceedings. IEEE Computer Society Annual Symposium on*, pp. 5–9, IEEE, 2002.

[63] B. Calhoun, A. Wang, and A. Chandrakasan, "Modeling and sizing for minimum energy operation in subthreshold circuits," *Solid-State Circuits, IEEE Journal of*, vol. 40, no. 9, pp. 1778–1786, 2005.

[64] C. 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*, vol. 11, no. 6, pp. 1058–1067, 2003.

[65] H. Soeleman, K. Roy, and B. Paul, "Sub-domino logic: ultra-low power dynamic subthreshold digital logic," in *VLSI Design, 2001. Fourteenth International Conference on*, pp. 211–214, IEEE, 2001.

[66] D. Bol, J. De Vos, C. Hocquet, F. Botman, F. Durvaux, S. Boyd, D. Flandre, and J. Legat, "A 25mhz 7μw/mhz ultra-low-voltage microcontroller soc in 65nm lp/gp cmos for low-carbon wireless sensor nodes," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, pp. 490 –492, feb. 2012.

[67] M. Seok, D. Jeon, C. Chakrabarti, D. Blaauw, and D. Sylvester, "A 0.27v 30mhz 17.7nj/transform 1024-pt complex fft core with super-pipelining," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pp. 342 –344, feb. 2011.

[68] J. Kwong, Y. Ramadass, N. Verma, M. Koesler, K. Huber, H. Moormann, and A. Chandrakasan, "A 65nm sub-v_t, microcontroller with integrated sram and switched-capacitor dc-dc converter," in *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International*, pp. 318–616, IEEE, 2008.

[69] T. Kim, J. Liu, J. Keane, and C. Kim, "A 0.2 v, 480 kb subthreshold sram with 1 k cells per bitline for ultra-low-voltage computing," *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 2, pp. 518–529, 2008.

[70] M. Seok, S. Hanson, Y. Lin, Z. Foo, D. Kim, Y. Lee, N. Liu, D. Sylvester, and D. Blaauw, "The phoenix processor: A 30pw platform for sensor applications," in *VLSI Circuits, 2008 IEEE Symposium on*, pp. 188–189, IEEE, 2008.

[71] D. Boning and S. Nassif, "Models of process variations in device and interconnect," *Design of high performance microprocessor circuits*, p. 6, 2000.

[72] J. Kao, M. Miyazaki, and A. Chandrakasan, "A 175-mv multiply-accumulate unit using an adaptive supply voltage and body bias architecture," *Solid-State Circuits, IEEE Journal of*, vol. 37, no. 11, pp. 1545–1554, 2002.

[73] M. Pelgrom, A. Duinmaijer, and A. Welbers, "Matching properties of mos transistors," *Solid-State Circuits, IEEE Journal of*, vol. 24, no. 5, pp. 1433–1439, 1989.

[74] P. Jespers, *The Gm/ID Methodology, a Sizing Tool for Low-voltage Analog CMOS Circuits: The Semi-empirical and Compact Model Approaches*. Springer Verlag, 2009.

[75] H. Thode, *Testing for normality*, vol. 164. CRC, 2002.

[76] N. Beaulieu, A. Abu-Dayya, and P. McLane, "Comparison of methods of computing lognormal sum distributions and outages for digital wireless applications," in *Communications, 1994. ICC'94, SUPERCOMM/ICC'94, Conference Record,'Serving Humanity Through Communications.'IEEE International Conference on*, pp. 1270–1275, IEEE, 1994.

[77] N. El Faouzi and M. Maurin, "Sur la loi de la somme de variables log-normales: application à la fiabilité de temps de parcours routiers," tech. rep., Working Paper, INRETS, 2006.

[78] B. Zhai, S. Hanson, D. Blaauw, and D. Sylvester, "Analysis and mitigation of variability in subthreshold design," in *Proceedings of the 2005 international symposium on Low power electronics and design*, pp. 20–25, ACM, 2005.

[79] B. Calhoun and A. Chandrakasan, "Characterizing and modeling minimum energy operation for subthreshold circuits," in *Low Power Electronics and Design, 2004. ISLPED'04. Proceedings of the 2004 International Symposium on*, pp. 90–95, IEEE, 2005.

[80] M. Seok, S. Hanson, D. Sylvester, and D. Blaauw, "Analysis and optimization of sleep modes in subthreshold circuit design," in *Proceedings of the 44th annual Design Automation Conference*, pp. 694–699, ACM, 2007.

[81] M. Slimani and P. Matherat, "Analyse d'architectures de multiplieurs en vue de la basse consommation," in *Journées Nationales du Réseau Doctoral en Microélectronique*, (Montpellier), 2010.

[82] M. Slimani and P. Matherat, "Multiple threshold voltage for glitch power reduction," in *Faible Tension Faible Consommation (FTFC), 2011*, pp. 67–70, IEEE, 2011.

[83] M. Slimani, F. Silveira, and P. Matherat, "Variability-speed-consumption trade-off in near threshold operation," *Integrated Circuit and System Design. Power and Timing Modeling, Optimization, and Simulation*, pp. 308–316, 2011.

[84] M. Slimani, P. Matherat, and Y. Mathieu, "A dual threshold voltage technique for glitch minimization," in *IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, (Séville, Espagne), Dec. 2012.

[85] B. Calhoun, *Low energy digital circuit design using sub-threshold operation.* PhD thesis, Massachusetts Institute of Technology, 2005.

Conception Basse Consommation de Circuits Numériques

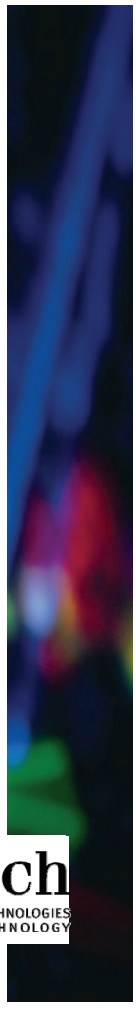
RESUME : Ce travail de thèse traite différents aspects de la conception basse consommation. Tout d'abord, le concept du calcul réversible, considéré comme le premier essai pour un calcul sans dissipation, est présenté. Puis, je me suis intéressée aux dissipations des circuits complémentaires MOS puisque c'est la logique la plus couramment utilisée dans les circuits numériques. J'ai proposé deux approches pour réduire la consommation de ces circuits numériques. La première approche porte sur la réduction de la dissipation due aux glitchs. J'ai proposé une nouvelle méthode qui consiste à adapter les tensions de seuil des transistors pour assurer un filtrage optimal de ces glitchs. Les résultats de simulation montrent que nous obtenons jusqu'à 16% de réduction des glitchs, ce qui représente une amélioration de 18% par rapport à l'état de l'art sur la base des circuits de référence ISCAS85. La deuxième approche porte sur la réduction de la dissipation obtenue en faisant fonctionner les transistors MOS en régime d' inversion faible (sous-seuil). Les circuits fonctionnant dans ce régime représentent une solution idéale pour les applications ultra-basse-consommation. Par contre, l'une des préoccupations majeures est qu'ils sont plus sensibles aux dispersions des processus de fabrication, ce qui peut entraîner des problèmes de fiabilité. Je propose un modèle compact qui détermine le point d'énergie minimum de façon analytique, donc sans recourir à une simulation type SPICE, tout en étant suffisamment précis et robuste vis-à-vis de la variabilité (due à la dispersion). L'écart de résultat entre le modèle compact et un modèle SPICE complet est de 6%.

MOTS-CLEFS : Conception basse consommation, réduction d'énergie, réduction des transitions parasites, circuits sous-seuil, variation des procédés de fabrication

Low Energy Design of Digital Circuits

ABSTRACT : This thesis focuses on different aspects of "Low Energy Design". First, reversible logic, as it is the first attempt for low energy computing, is briefly discussed. Then, we focus on dynamic energy saving in the combinational part of CMOS circuits. We propose a new method to reduce glitches based on dual threshold voltage technique. Simulation results report more than 16% average glitch reduction. We also show that combining dual-threshold to gate-sizing technique is very interesting for glitch filtering as it brings up to 27% energy savings. In the third part of this dissertation, we have been interested in sub-threshold operation where the minimum energy can be achieved using a reduced supply voltage. Sub-threshold operation has been an efficient solution for energy-constrained applications with low speed requirements. However, it is very sensitive to process variability which can impact the robustness and effective performance of the circuit. We propose a model valid in sub and near threshold regions in order to correctly estimate the circuit performance in a variability aware analysis. We provide an analytical solution for the optimum supply voltage that minimizes the total energy per operation while considering variability effects. Spice simulations matches the analytical result to within 6%.

KEY-WORDS : Low Energy Design, energy savings, glitch reduction, sub and near threshold design, variability modeling



TELECOM ParisTech

EDITE

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY