

AIX-MARSEILLE UNIVERSITÉ
STMICROELECTRONICS
ED 353 – SCIENCES POUR L'INGÉNIEUR

INSTITUT MATERIAUX MICROÉLECTRONIQUE NANOSCIENCES DE PROVENCE

Thèse présentée pour obtenir le grade universitaire de Docteur

Discipline : Sciences pour l'ingénieur

Spécialité : Micro et Nanoélectronique

Guénolé LALLEMENT

**Extension of SoCs Mission Capabilities by Offering Near-Zero-Power
Performances and Enabling Continuous Functionality for IoT Systems**

Circuits à empreinte énergétique quasi nulle permettant une extension des profils de mission et un fonctionnement continu des systèmes destinés à l'Internet-des-Objets

Soutenue le 12/11/2019 devant le jury composé de :

Pr. Massimo Alioto, National University of Singapore – *Président & Rapporteur*

Pr. David Bol, Université Catholique de Louvain – *Rapporteur*

Dr. Édith Beigné, Facebook – *Examineur*

Dr. Pascal Vivet, CEA-Leti – *Examineur*

Dr. Martin Cochet, IBM Research – *Examineur*

M. James Myers, ARM Ltd. – *Invité*

Dr. Fady Abouzeid, STMicroelectronics – *Encadrant industriel*

Pr. Jean-Luc Autran, Aix-Marseille Université – *Directeur de thèse*

Dr. Daniela Munteanu, Aix-Marseille Université – *Co-Directrice de thèse*

Numéro national de thèse/suffixe local: 2019AIXM0573/033ED353



Cette œuvre est mise à disposition selon les termes de la Licence Creative Commons
Attribution-NonCommercial-ShareAlike 4.0 International

À mes parents.

Abstract

Recent developments in the field of low voltage Integrated Circuits (IC) have paved the way towards energy efficient electronic devices in a booming global network called the Internet-of-Things (IoT) or the Internet-of-Everything (IoE). However, the sustainability of all these interconnected sensors is still undermined by the constant need for either an on-board battery – that must be recharged or replaced – or an energy harvester with very limited power efficiency. The power consumption of present consumer electronic systems is fifty times higher than the energy available by cm^2 -size harvester or limited to a few months on a small battery, thus hardly viable for lifetime solutions. Upcoming Systems-on-Chip (SoCs) must overcome the challenge of this energy gap by architecture optimizations from technology to system level.

The technical approach of this work aims to demonstrate the feasibility of an efficient Ultra-Low-Voltage (ULV) and Ultra-Low-Power (ULP) SoC using exclusively latest industrial guidelines in 28 nm and 22 nm Fully Depleted Silicon On Insulator (FD-SOI) technologies. A hundred of nano watts leakage-based oscillator has been designed for synchronous digital clocking of the system's always-on parts. Dynamic self-biasing techniques enabled in FD-SOI are implemented to provide system speed-up during SoCs active operations, without incurring leakage current penalties during standby periods. Temperature compensation is also evaluated to increase robustness at low voltages. Several multi-power-domain SoCs based on Advanced RISC Machines (ARM) cores are implemented to demonstrate wake up strategies based on sensors inputs. By optimizing the system architecture, properly selecting and designing components with technology features chosen adequately, carefully tuning the implementation, a fully energy-optimized SoC is realized. Lastly, machine learning techniques coupled with the different power modes offered by the core enhance the system state restoration mechanisms. Depending on the application timing constraints (real-time vs. energy-saving) the proper standby mode can be selected to minimize the overall power consumption.

Keywords — 22 nm FDX, 28 nm FD-SOI, adaptive power management, body biasing, digital circuit design, energy-efficiency, machine learning, microcontroller (MCU), minimum energy point, ultra-low-power (ULP), ultra-low-voltage (ULV).

Résumé

Les développements récents dans le domaine des circuits intégrés (IC) à basse tension ont ouvert la voie à des dispositifs électroniques économes en énergie dans un réseau mondial en plein essor appelé l'Internet des objets (IoT) ou l'Internet des choses (IoE). Cependant, la durabilité de tous ces capteurs interconnectés est compromise par le besoin constant d'une batterie embarquée - qui doit être rechargée ou remplacée - ou d'un récupérateur d'énergie à rendement très limité. La consommation d'énergie des systèmes électroniques grand public actuels est en effet cinquante fois plus élevée que celle d'un collecteur d'une taille de l'ordre du cm^2 , ou limitée à quelques mois sur une petite batterie. Cela contraint la viabilité de solutions fonctionnant à l'échelle d'une vie humaine. Les systèmes sur puce (SoCs) à venir nécessitent donc de relever le défi de cette lacune énergétique en optimisant l'architecture de la technologie au niveau du système.

L'approche technique de ce travail vise à démontrer la faisabilité d'un SoC efficient, ultra-basse tension (ULV) et ultra-basse puissance (ULP) utilisant exclusivement les dernières directives industrielles en matière de technologies FD-SOI (Fully Depleted Silicon On Insulator) 28 nm et 22 nm. Un oscillateur à courants de fuite consommant une centaine de nanowatts a été conçu pour fournir aux éléments toujours actifs du système une horloge numérique synchrone. Des techniques de polarisation automatiques et dynamiques, disponibles en FD-SOI, sont mises en œuvre pour accélérer le SoC pendant les opérations actives, sans encourir de pénalité liée aux courants de fuite pendant les périodes de veille. La compensation de température est également évaluée pour augmenter la robustesse à basse tension. Plusieurs SoCs multi-domaines basés sur des cœurs Advanced RISC Machines (ARM) sont implémentés pour démontrer des stratégies de réveil basées sur les entrées des capteurs. Ainsi, en optimisant l'architecture du système, en sélectionnant et en concevant correctement les composants avec des caractéristiques technologiques choisies de manière adéquate, et en ajustant soigneusement l'implémentation physique, on obtient un SoC entièrement optimisé en énergie. Enfin, des techniques d'apprentissage machine couplées aux différents modes de consommation offerts par le SoC améliorent les mécanismes de restauration de l'état du système. En fonction des contraintes de temps de l'application (temps réel vs. économie d'énergie), un mode de veille approprié peut être sélectionné pour minimiser la consommation d'énergie globale.

Mots-clés — 22 nm FDX, 28 nm FD-SOI, apprentissage machine, conception de circuits numériques, efficacité énergétique, gestion de puissance adaptative, microcontrôleur (MCU), point énergétique minimum, polarisation du substrat, ultra-basse-consommation (ULP), ultra-basse-tension (ULV).

Résumé en français

POUSSÉ par la multiplication des applications sans fil, interconnectées et à faible coût, il est établi que l'Internet des Objets (*Internet-of-Things*, IoT) occupera une position prédominante dans nos sociétés de plus en plus connectées [1]. Cependant, cette croissance est limitée par la consommation d'énergie des éléments embarqués sur ces objets de plus en plus complexes.

Ainsi, dans nos sociétés qui cherchent à réduire leur consommation, un double problème se pose. D'un côté, une augmentation rapide des besoins en énergie est nécessaire pour répondre au nombre croissant d'appareils. De l'autre, les applications hétérogènes proposées souffrent d'une autonomie limitée et de la détérioration des batteries embarquées, ce qui impose un remplacement des appareils au fur et à mesure de leur utilisation.

Ces tendances établissent le contexte général de ce travail; promouvoir un fonctionnement continu des systèmes destinés à l'IoT, tout en garantissant un large éventail de profils de mission. Ce manuscript étant rédigé en langue anglaise, un résumé en français des recherches menées lors de cette thèse est proposé dans les pages suivantes.

Introduction

Les appareils électroniques reposant sur la simple utilisation de piles interchangeables n'étant plus une option pérenne [2], les prochains Systèmes-sur-Puce (*System-on-Chip*, SoC) doivent relever le défi de la collecte, du stockage et de la gestion de l'énergie pour une consommation et une distribution efficace et durable [3].

Un aperçu des différentes sources d'énergies qui peuvent être exploitées pour alimenter directement ou indirectement un système est proposé Figure 1. Pour chaque source, l'estimation de la puissance disponible est rapportée pour une surface unitaire de 1 cm^2 . En raison de l'efficacité limitée du dispositif qui collecte l'énergie (*energy harvester*), cette puissance disponible n'est jamais exploitée avec un processus de conversion parfait (*i.e.*: avec 100% d'efficacité). Afin d'estimer les fonctions actuellement réalisables avec ces budgets énergétiques, la consommation de divers appareils électroniques grand public est reportée au bas du graphique. Pour les SoCs destinés à l'IoT, un budget énergétique réaliste de $\sim 100\mu\text{W}/\text{cm}^2$ peut être défini, tandis que la consommation d'énergie des systèmes de type IoT est de l'ordre de 5 mW selon l'application demandée [1]. La consommation d'énergie de ces systèmes est donc 50 fois plus élevée que l'énergie disponible.

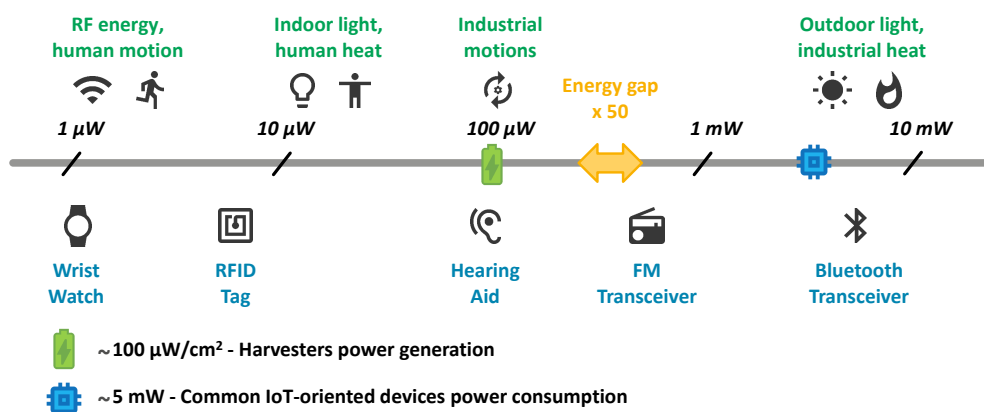


Figure 1: Puissance disponible par un collecteur d'énergie pour une surface équivalente de $1 cm^2$ comparée à la consommation de dispositifs électroniques standards.

Suite aux implications de la loi de Koomey – qui suppose que la quantité de puissance requise pour une charge de calcul donnée diminue d'un facteur 2 tous les 1,5 ans [4] – cet écart peut être comblé au delà de 9 années (~ 2026). Des innovations de conception sont ainsi nécessaires pour disposer rapidement de nouveaux objets intelligents avec des opérations sans charge.

Afin d'offrir des appareils autonomes en énergie, un premier levier consiste à réduire la tension d'alimentation. Cependant, pour les nœuds technologiques avancés, cette solution accroît la sensibilité aux variations PVT (*Process, Voltage, Temperature*) et aux courants de fuite, limitant en retour les opérations en mode actif et en mode repos [5, 6]. Pour pallier ces lacunes, des techniques ont donc été déployées, mais elles entraînent logiquement une augmentation de la consommation d'énergie et des défis supplémentaires en termes de conformité aux normes de l'industrie.

L'approche technique de cette thèse vise donc à démontrer la faisabilité d'un SoC efficace énergétiquement, à très faible consommation et tension, tout en utilisant exclusivement les dernières directives industrielles. Par conséquent le travail de manuscrit cherche à pousser le marché des microcontrôleurs (MCU) grand public vers une consommation de l'ordre du microWatt (*Ultra-Low-Power*, ULP). Pour cela, il convient de travailler sur l'optimisation de l'architecture, la sélection des composants, tout en suivant les caractéristiques technologiques adéquates. Une utilisation appropriée des différents modes de puissance offerts par le Cortex-M0+ est également associé à des solutions d'apprentissage machine (*Machine Learning*, ML).

Contexte Applicatif et Technologique

Une approche ascendante est abordée pour rechercher des techniques à tous les niveaux de conception afin d'optimiser l'efficacité énergétique et une diminution de la puissance consommée (voir Figure 2).

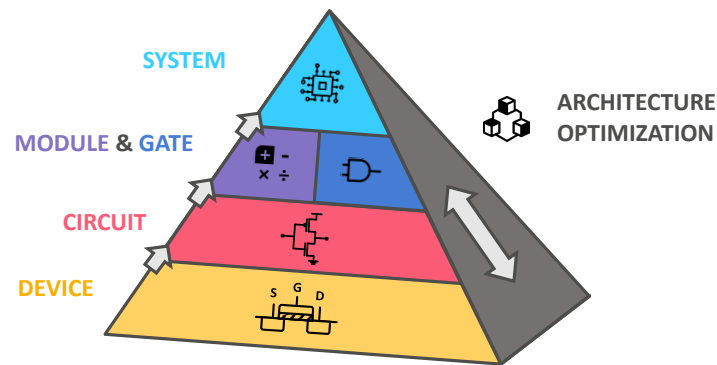


Figure 2: Pile d'implémentation d'un circuit intégré : de la technologie silicium au système.

Cette analyse de l'historique et de l'état de l'art nous éclaire sur le large éventail de solutions disponibles pour la réduction de la consommation d'énergie. Le fonctionnement à ultra-basse tension (*ultra-low-voltage*, ULV) appliqué à des technologies nanométriques n'est pas suffisant. En effet, l'efficacité énergétique doit aussi être optimisée jusqu'au niveau système, en tirant parti des interactions entre le matériel et le logiciel. De plus, pour permettre le déploiement industriel d'une architecture System-on-Chip (SoC) compatible avec les applications à faible consommation, certains critères ont été définis.

Premièrement, l'utilisation de la logique standard CMOS dans un nœud de technologie avancée comme le FD-SOI est nécessaire pour une efficacité en phase active maximale. L'ULV est également obligatoire pour atteindre le point d'énergie minimum du système (*Minimum Energy Point*, MEP). La sensibilité aux variations PVT qui en résulte doit être compensée par des méthodes simples ou en définissant des marges raisonnables.

La polarisation des caissons (*body-biasing*) possible en FD-SOI offre un levier supplémentaire entre vitesse, efficacité et fiabilité et doit donc être envisagée. Cependant, pour des applications ULP, des solutions simplifiées doivent être développées pour ne pas impacter le budget énergétique du système. Bien que ce travail vise principalement des optimisations architecturales, une analyse complète des composants basse consommation constitutifs d'un SoC est nécessaire. Cela mène notamment à re-concevoir des composants pour les opérations ULV/ULP.

Un partitionnement du système est obligatoire pour implémenter des techniques basse consommation mais aussi intégrer efficacement des modules. De même, cela facilite des techniques à haut-niveau de gestion de puissance qui se basent sur les différents états de fonctionnement du SoC. Enfin, la gestion dynamique de la puissance (*Dynamic Power Management*, DPM) est une option prometteuse pour exploiter les modes d'alimentation du système en fonction de l'activité et de l'application.

Definition d'un Système Ultra-Basse Consommation

Une description fonctionnelle du SoC est proposée pour intégrer tous les sous-systèmes nécessaires à une architecture relativement générique. De plus, un partitionnement lors de l'implémentation permet la mise en place de modes liés à l'activité du système et de la disponibilité énergétique (voir Figure 3).

Comme tous les composants essentiels à un SoC ne peuvent être couverts dans ce travail, une mise en œuvre simplifiée est proposée pour démontrer les optimisations effectuées dans les chapitres suivants de ce travail. Ces améliorations se focalisent notamment sur l'intégration de la technologie, des composants et des systèmes basse consommation.

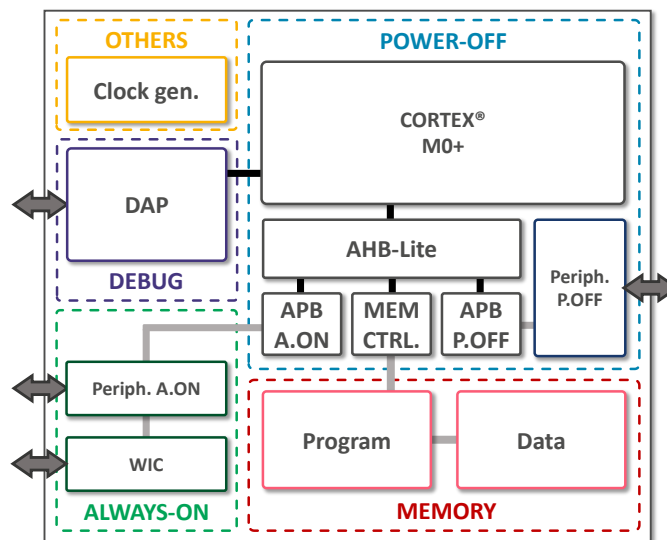


Figure 3: Partitionnement du système pour répondre à des applications Ultra-Low-Power.

Générateur d'Horloge Ultra-Basse Consommation

Pour assurer un fonctionnement correct, les systèmes de faible puissance nécessitent des sources de temps et de fréquence précises. De plus, afin de s'adapter aux budgets limités des applications ULP, des soins particuliers doivent être apportés à la réduction de la puissance globale, tout en améliorant la précision et la stabilité de l'horloge.

D'autre part, comme la référence de l'horloge reste active pendant la plupart des phases de fonctionnement du SoC, elle contribue de façon non-négligeable à la consommation totale du système. Ainsi, pour diminuer cet impact sans dégrader les performances, plusieurs solutions ont été explorées. Une solution compatible avec les technologie CMOS utilisant un oscillateur en anneau (*Ring-Oscillator*, RO) basé sur les courants de fuites apparaît comme le meilleur compromis en termes de puissance, de surface et de stabilité (voir Figure 4).

En raison de la haute sensibilité de l'oscillateur, une unité de compensation numérique nommée CLU pour *Control Logic Unit* est ajoutée. Construite autour d'un correcteur Proportionnel-Intégral (PI), elle intègre plusieurs optimisations pour une précision maximale avec un temps de convergence réduit (voir Figure 5).

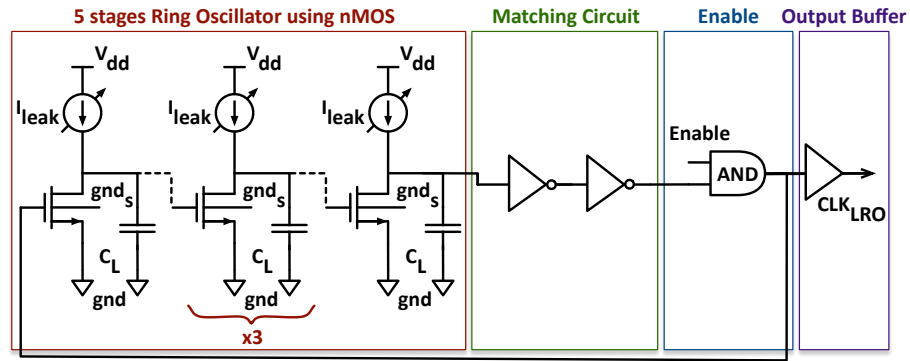


Figure 4: Schéma de l'oscillateur contrôlé en courant.

L'ensemble du système fonctionne à 0.5 V. Quarante-deux puces ont été testées et dix sélectionnées pour l'évaluation de la compensation numérique. Les résultats obtenus démontrent un système offrant un rendement énergétique élevé, combiné à de très faibles variations de tension et de température. Une consommation de 15 nW pour l'oscillateur et 125 nW pour la compensation est mesurée, tandis que le système assure une oscillation à 32.768 kHz avec 90 ppm/V pour $V_{dd} \pm 8\%$ et 1.9 ppm/°C de 0 à 50 °C. Enfin, les capacités de stabilité à long terme sont caractérisées par une déviation d'Allan de 0.1 ppm. En faisant la démonstration d'une horloge efficace et peu coûteuse, ce travail offre une référence temporelle et une source polyvalente pour les systèmes numériques standards.

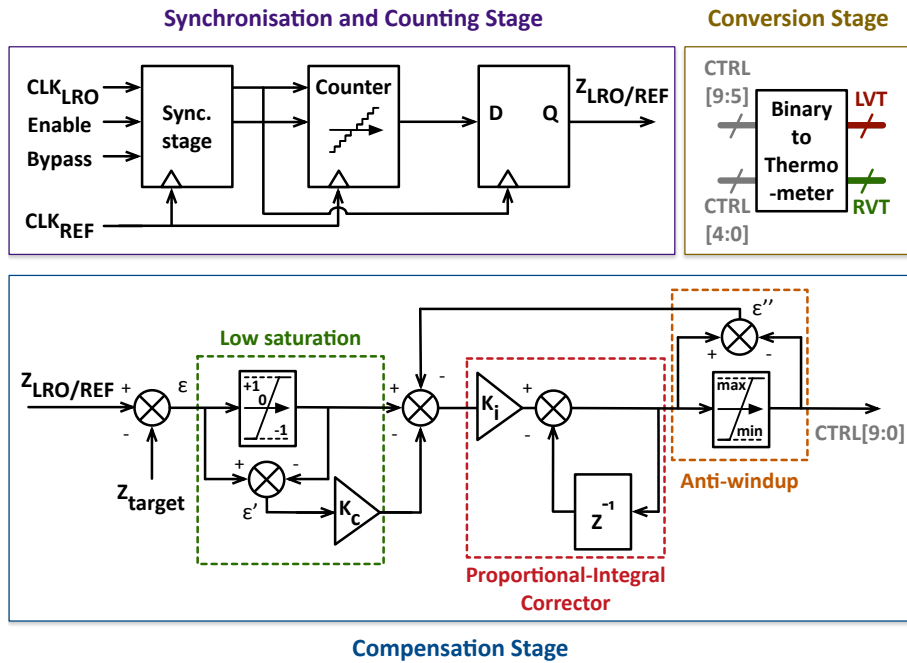


Figure 5: Schéma block de l'unité de compensation CLU.

Implémentation de Systèmes Ultra-Basse Consommation

L'approche technique de ce travail vise à démontrer la faisabilité d'un système efficace en énergie et adapté à l'énergie que peut collecter un récupérateur, tout en utilisant exclusivement les dernières directives ou outils industriels. Ainsi, les travaux présentés dans ce chapitre se distinguent des publications existantes par une approche reposant sur des méthodes simples mais très efficaces. En effet, afin de faciliter une adoption rapide des techniques basse consommation, le marché de masse des microcontrôleurs exige des solutions rentables et robustes.

Par conséquent, les anciennes techniques basse-consommation et les nouveaux concepts innovants sont explorés de la technologie jusqu'au niveau du système. Un SoC avec une consommation d'énergie active correspondant à une consommation de plusieurs centaines de micro watts est mise en œuvre grâce à l'utilisation de plusieurs techniques. Cela passe par une utilisation adéquate et un étalonnage des caractéristiques de la technologie, associés à un réglage spécifique du flot de conception. Dans un second temps, la sélection rigoureuse des composants et des techniques de faible consommation, l'optimisation de l'architecture système associée à un apprentissage des différents modes offerts par le Cortex-M0+ permettent de définir des modes de consommation pertinents.

La consommation finale obtenue laisse également une marge dans l'ensemble du budget énergétique pour étendre le système avec des capacités de communication et de détection, ouvrant ainsi la voie à des systèmes totalement autonomes (voir Figure 6).

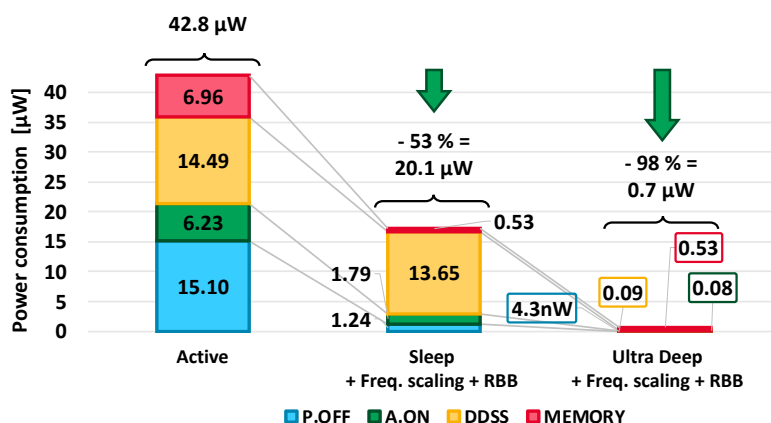


Figure 6: Répartition totale de la consommation du démonstrateur NZP28 – V1.0 à 0.5 V/25 °C.

Cependant, les meilleurs modes ULP ne garantissent pas une consommation d'énergie efficace pendant toute l'activité du système. En raison de l'alternance entre les séquences actives et les séquences de repos, une gestion adaptative de la consommation est nécessaire pour une sélection optimale du mode du SoC en phase de repos. De plus, en raison des variations PVT, une caractérisation complète de la consommation électrique du système ne garantit pas la solution la plus favorable. Des efforts sont donc poursuivis pour mettre au point des techniques novatrices reposant sur l'apprentissage de l'activité du SoC.

Gestion Adaptative de la Consommation

Ce chapitre présente un nouveau modèle de gestion adaptative de la consommation (*Adaptive Power Management*, APM) basé sur un algorithme d'apprentissage par renforcement (*Reinforcement Learning*, RL) pour optimiser la consommation du système relativement à son activité. Le module APM offre la possibilité de sélectionner le mode Ultra-Low-Power (ULP) le plus efficace en fonction du temps passé en phase de repos, de sa consommation d'énergie et de son environnement externe.

La théorie générale de l'algorithme est définie sur une architecture générique. Cette analyse préliminaire permet de définir correctement les paramètres pour maximiser l'apprentissage par renforcement. Finalement, l'algorithme *Q-Learning* (QL) est sélectionné. Sur cette première architecture, les simulations démontrent un gain moyen de 17% comparé à une solution heuristique. Cependant, un temps d'apprentissage de 10 essais est requis, correspondant à un coût d'apprentissage de 87 nJ.

Le module APM a été synthétisé en technologie 28 nm FD-SOI. Au vu de l'impact de ce module générique sur un SoC basse consommation, des optimisations matérielles spécifiques ont été identifiées pour améliorer la consommation d'énergie du module.

Ainsi, une seconde version du module APM est évaluée sur une architecture de type *Near-Zero-Power* (NZP). Grâce à la réduction du nombre d'états du système, de meilleures performances sont obtenues. Moins de 5 essais sont requis par l'algorithme pour faire son apprentissage, ce qui aboutit à un coût de 75 nJ. Par rapport à une heuristique de sommeil profond, on obtient des économies d'énergie de 150 % (voir Figure 7). De plus, l'impact de l'APM est estimé à 42% de l'architecture SoC. Ainsi, les bons résultats montrés par la technique restent mitigés pour de minuscules ULP SoCs. Cependant, la technique ouvre la voie à une gestion efficace de l'énergie à l'aide de techniques d'apprentissage machine.

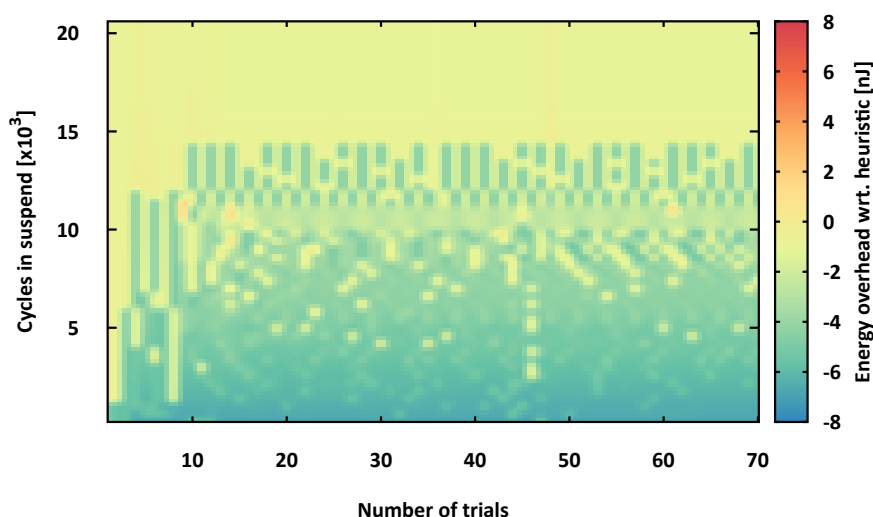


Figure 7: Comparaison des gains et coûts énergétiques du module APM (architecture de type NZP) relativement à une solution heuristique. L'énergie consommée [nJ] est donnée en fonction du nombre de tests (*trials*) et du nombre de cycles (*cycle in suspend*) passés en mode repos.

L'ensemble du système est simulé et évalué à l'aide d'un environnement de co-simulation Python et Verilog. Combiné à des techniques de code évolutives, le module APM peut être adapté à plusieurs architectures de SoC. L'implémentation physique du module utilise aussi les techniques ULP présentées dans le chapitre précédent pour assurer une consommation minimale. Un démonstrateur complet de SoC intégrant le module APM est en cours de fabrication en technologie 22 nm FD-SOI. Les travaux futurs se concentreront sur la mesure de la performance in situ du module, y compris l'impact du module sur la consommation globale. Ce travail est le premier à implémenter Q-Learning (QL) dans un démonstrateur complet SoC pour une sélection optimale du mode de puissance pendant les séquences de suspension.

Conclusions et Perspectives

Cette thèse aborde un problème qui dépasse les solutions dédiées aux objets connectés actuels. Afin de garantir une utilisation viable à long terme des systèmes électroniques, l'émergence d'appareils connectés entièrement autonomes et durables est nécessaire. Ainsi, ce travail développe des circuits et systèmes-sur-puce présentant une empreinte énergétique réduite au maximum.

Pour les applications qui dépendent de la récupération d'énergie ou de petites batteries, l'efficacité énergétique est une mesure clé. Contrairement aux dispositifs IoT standards, qui offrent des services à faible et haute performance, en tirant parti d'une réserve d'énergie macroscopique offerte par leur batterie, les systèmes NZP effectuent des opérations très spécifiques à faible coût énergétique tout en consommant peu ou pas d'énergie quand ils sont en mode veille. Ce travail met également un accent particulier sur la réduction de la consommation d'énergie des microcontrôleurs génériques en utilisant la technologie FD-SOI combinée avec les derniers flots standards de conception et outils industriels.

L'empreinte énergétique quasi nulle permettra non seulement des applications de faible activité, comme des capteurs sans fils qui analysent les variations physiques et chimiques de l'environnement ou qui surveillent la santé, mais aussi la robotique en essaim et les applications à distance. Pour de tels systèmes, la petite source d'énergie embarquée, souvent associée à des récupérateurs d'énergie, présente des dimensions minimales et une énergie disponible réduite. Ces limitations nécessitent des innovations à tous les niveaux de conception d'un circuit intégré conventionnel - dispositif, circuits, portes/modules, système et architecture - et cela, afin de fournir un service pertinent tout en garantissant l'autonomie énergétique du système.

La faisabilité d'un SoC ULP est donc démontrée par l'utilisation de plusieurs techniques. Cela débute par l'utilisation du catalogue industriel de cellules standards et des caractéristiques de la technologie FD-SOI. Un partitionnement du système est également proposé et combiné à l'utilisation de solutions optimisées pour la faible consommation ainsi que des techniques d'apprentissage machine pour permettre de minimiser la consommation du système.

Acknowledgment

As you might guess, carrying out a Ph.D. research project is a challenging task, especially when it comes to high-end technological topics such as microelectronics. Luckily, during three years of academic and industrial partnership between STMicroelectronics and Aix-Marseille Université I had the chance to meet outstanding people who filled my boundless curiosity, enriched my knowledge and supported my ideas. A few pages are definitely not enough to acknowledge what they have done but I will try to keep it short.

First, I would like to give infinite thanks to my industrial supervisor Fady Abouzeid. As far as I am concerned, he is THE most awesome supervisor a Ph.D. student might dream of. Sometimes using “forced autonomy”, I had to learn a lot in a short amount of time, but in the end he was always there even at silly hours to do last-minute design fixes, paper reviews or simply discuss some ideas or technical aspects of electronic design. Thank you for always supporting me. I also warmly show my appreciation to my academic supervisor Prof. Jean-Luc Autran for his academic guidance all along this project. Thank you also for giving me the means to promote my work and present it in world-class conferences. I also wish you the best to finally solve the Phaistos disc mystery.

My gratitude also goes to the people who accepted to be part of my thesis examination committee and for their valuable feedback and insightful questions: Prof. David Bol and Prof. Masimo Alioto, Édith Beigné, Pascal Vivet and James Myers.

Many thanks to Philippe Roche, the manager of the radiation hardening and ultra-low-voltage team at STMicroelectronics, who kindly welcomed me in this group. You gave me plenty of freedom and silicon opportunities to express my ideas. As a young designer eager to play with advanced technology nodes, I could not thank you enough for that. Moreover, your wise managing and technical advice also offered to my work credibility and visibility.

There are two people I consider as my “shadow” supervisors for their technical support and friendly discussions. Thank you Jean-Marc Daveau. Despite the years that separate us, I found an authentic nerd friend and sciences enthusiast. With you I could share my obscure computer and electronic related discoveries. Also bless you for the retard alerts. Take care of the triangle: Diling ♪ Diling ♪. Thank you Martin Cochet. As a former student of my master degree you contacted me to let me know about a research opportunity which resulted in this Ph.D. thesis. This manuscript would not exist without you. Thanks for all the debates about electronics, the silly imperial units, the Tomazou transistor humpf principle, PhD comics. You are a great source of inspiration and some of my best ideas come from our discussions.

My fellow Ph.D. students are also a great source of wise (and sometimes silly) discussions. Victor Malherbe (OK technically you are not a Ph.D. student anymore) many thanks for the discussions around the understanding of the transistor behavior. Now, I get it, “It’s only a model”. I am also glad I could share my thesis writing complaints with you. Next, thank you Ca-

pucine Lecat–Mathieu de Boissac my fellow office companion. Since I don't have the space to write you the full acknowledgement page you deserve, I will principally thank you for being my rubber duck debugging companion and comprehensive listener to all my thesis existential troubles. Finally, Sebastien Thomet, it was so nice having all these hacking discussions during our long biking commutes. Both of you, I wish you the best for the ending of your PhD.

I would like to extend my thanks to all my team member at STMicroelectronics. Thank you Dominic Zamora, for helping me when I was struggling with design tools. Thank you Dmitry Soussan, since you are a kind and friendly co-worker I wish you the best but I won't forget the painful gym sessions. Thank you Thomas Théry, even though I am still waiting for the 3D Tron game. Thank you Olivia Riewer for your precious advice on mastering SDC constraining. Thank you Vincent Lorquet, as a software enthusiast I am sure you will impose GIT to the team. Thank you Valérie Bertin for your software precious tips. Thank you Serge De Paoli, master of the digital debug. Thank you Anna Asquini and Calogéro Timineri, as digital designers who know the product experience, it was always nice exchanging with you. Lastly, thank you Gilles Gasiot, the Alpha and Omega of radiation hardening. I could not forget the interns and students who reminded me about my engineering school years. Thank you Patrick Temleitner, Mathieu Rodrigues, Florent Chartier, Mael Baudoin, Quentin Metras and Romain Demarets. A special thank to Yvan Debizet with whom I worked on Adaptive Power Management.

Thank you Soilihi Moindjie and Jeff Novakowski for your support during experimentation and testchips measurements. Thank you Gaele Passeron for the Cadence Support during stressful tape-out deadlines. Thank you Sylvain Clerc and Thierry Di Gilio. With you guys I had the best conversation on body-bias utilization and FD-SOI ever. A special thanks to Ricardo Gomez–Gomez to whom I wish the best for his PhD. You did such a great work and talking with you was always so delightful. Thank you Andrej Suller for sharing the ST Ph.D. experience during our long Express 1 bus commutes. Thank you Andrea Cathelin for your support on the after PhD thesis life. Thanks also to Robin Bennarouch for the visit of BWRC and the explanations of RF-design. Moreover, I really appreciate the always available administrative and logistic support of Pascale Maillet-Contoz at ST and Zolika Djellouli at Aix-Marseille.

In addition, many thanks to all other amazing people, Hans and other KUL students, Maxence and CEA members, Pranay and others, I have met during conferences and events all over the world for all the insightful discussions we had (often around a beer or two...). I also want to thank my friend Yannick, for his Sunday PlayStation support to climb the Rocket League ladder. Thank you Nicolas and Chloé for welcoming me when I returned back to Paris. And lastly, thank you to all my friends from Saint-Louis, Télécom Paris and Imperial College for supporting me during these years in spite of the distance.

Pour finir, je voudrais également profiter de cette occasion pour remercier (en français) les membres de ma famille. Ma sœur Léonie, pour son aide lors de la rédaction de ce manuscrit. J'ai également une pensée toute spéciale pour mon grand-père Jean qui était typographe. Tout comme tu pouvais ordonner des caractères d'imprimerie, j'organise des transistors pour "imprimer" des circuits. Par-dessus tout, je remercie chaleureusement mes parents pour leurs encouragements continus et leur soutien logistique. Ces trois dernières années n'auraient pas été possibles sans leurs dévouements continus. Ce travail vous est dédié !

Guénolé.



Figure 8: Picture of the first transistor ever assembled at Bell Labs in 1947.

Preface

DECEMBER 23th 1947, three scientists – John Bardeen, William Shockley, and Walter Brattain – present the progress of their semiconductor researches to the higher-ups of the Bell Labs, New Jersey (USA). As detailed in Brattain’s lab notes, a microphone and a headphone are connected to a PNP point-contact germanium device which operate as a speech amplifier “with no noticeable change in quality” [7]. A new keystone component of modern electronics is operating for the first time; the transistor is born.

Immediately convinced that a fundamental breakthrough had just occurred, an absolute secret is imposed, and the results disclosed to a limited number of collaborators. Fearing that other scientists could develop similar devices, patents are systematically filed. However, on June 30th, 1948, after having publicly announced their discovery, Bell Labs adopted a free policy of openness to avoid a military secret classification of their research. From there, the development of the transistor technology would be facilitated and stimulated by co-developments, communications, and conferences with other academic, industrial and military institutions. Less than 10 years later, the three scientists were honored “for their researches on semiconductors and their discovery of the transistor” with the 1956 Nobel Prize in Physics [8].

Nowadays, thanks to the progress of technology, the transistor is widely available and used in people’s life. From the personal computer to the CMOS based DNA sequencer, this component is integrated everywhere in electronic devices, yet its applications in more and more complex circuits have become a major issue. The following of Moore’s law [9] is declining and Dennard’s scaling [10] limited by a slowdown of the supply voltage reduction, hence putting into jeopardy the improvement of transistor integration and performances.

Popular solutions for modern embedded applications integrate more and more components into Solid-State platforms, known as System-on-Chip. Those actual systems’ power consumptions are limited in the milliwatt range, hardly sustaining lifetime operations. For perpetual operations, designers and manufacturers cannot rely on existing technology offers. Innovative design and Ultra-Low-Power solutions must be developed by taking challenges in the design of digital and analog circuits, systems and architectures for compact Internet-of-Things devices.

This current status sets the beginning of this research project. Seventy years after the prior discovery, now supplemented by thousands of contributions from researchers and engineers, this work humbly pursues the quest towards the significant reduction of electronic devices energy consumption. After reading this manuscript, I hope it will give plenty of ideas to others. Maybe one day, we will finally create the perpetual electronic machine.

Contents

Abstract	iii
Résumé en français	vii
Acknowledgment	xvii
Preface	xxi
Contents	xxiii
List of Figures	xxix
List of Tables	xxxv
Abbreviations	xxxvii
Introduction	1
New Paradigms of Computer Hardware	1
Autonomy and Power Constraints for IoT Systems	4
Power Scaling	5
Towards Near-Zero Power	7
Challenges	8
Thesis Outline	10
Chapter 1. Background	13
1.1 Brief History of Voltage Scaling	14
1.2 Technology for Energy-efficient Design	16
1.2.1 The MOSFET Transistor in CMOS Technology	16
1.2.2 Advanced Technologies for Nanometers Designs	21
1.2.3 FD-SOI for Minimized Energy Consumption	22
1.3 Energy-efficient Circuits	24
1.3.1 Circuit Topologies	24

1.3.2	Sources of Power Dissipation in CMOS Circuits	26
1.3.3	Minimum Energy Point	28
1.3.4	Circuit Techniques for Power Reductions	30
1.4	Gate-level Building Blocks and Modules for Low-Power Operations	32
1.4.1	Design Constraints for Low-Voltage Operations	32
1.4.2	Power Management Techniques	33
1.5	System and Architectural Design	36
1.5.1	Total Energy Consumption	36
1.5.2	System Level Power Optimizations	38
1.6	Summary	42
1.6.1	Review of Ultra-Low-Power Techniques	42
1.6.2	Existing SoC Implementations	43
1.6.3	Conclusion	44
Chapter 2.	Ultra-Low-Power System Definition	49
2.1	Ultra-Low-Power System-on-Chip Architecture	50
2.1.1	IoT Device Energy Flow	50
2.1.2	System and Processing Elements	51
2.1.3	Clock Generation and Time Keeping	52
2.1.4	Use Cases Scenarios	53
2.1.5	Resulting Design Considerations	56
2.2	Ultra-Low-Power System-on-Chip Functional Description	56
2.2.1	SoC Description	56
2.2.2	Activity Timeline	57
2.2.3	Derivation of Power Modes	58
2.3	Ultra-Low-Power System-on-Chip Implementation	59
2.3.1	System Partition	59
2.3.2	Resulting Memory Mapping	59
2.3.3	Adopted Ultra-Low-Power Demonstrator Architecture	60
2.4	Conclusions	61
Chapter 3.	Clock Reference for Ultra-Low-Power Systems	63
3.1	Ultra-Low Power Clock in IoT nodes	64
3.1.1	Design considerations: Power, Area and Stability	64
3.1.2	Evaluation of Existing Low Power Time References and Clock Sources	64
3.1.3	Design Targets and Use Case	67
3.1.4	System Architecture	68

3.2	Clock Generation using a CMOS-compatible Oscillator	69
3.2.1	Existing Solutions	69
3.2.2	CMOS Current Controlled Ring Oscillator	70
3.2.3	Current Sources Sizing: Process Variations Compensation	73
3.2.4	Leakage Sources Matching and Variability	75
3.2.5	Complete Physical Implementation	78
3.3	Digital Locking Scheme	79
3.3.1	Evaluation of Various Digital Compensations	79
3.3.2	Proportional-Integral Corrector Analysis and Optimizations	82
3.3.3	SPI Interface and Controls	84
3.4	Circuit Implementation and Measurements	86
3.4.1	LRO Free Oscillations and Jitter	86
3.4.2	Digital Compensation Evaluation	87
3.4.3	Power and Drift Trade-off Resulting from the Relocking Scheme	90
3.4.4	Comparison with the State of the Art	93
3.5	Summary	94
Chapter 4.	Ultra-Low-Power System Implementation	97
4.1	Ultra-Low-Power Design Methodology in FD-SOI	98
4.1.1	Minimum Energy Point Design Methodology	98
4.1.2	Transistor Features Selection	98
4.1.3	Clock Tree and Standard Cell Selections	104
4.1.4	Electronic Design Automation Flow Tuning	104
4.2	Static Power Reduction	105
4.2.1	Power Gating	105
4.2.2	State Retention with Double Power Gating	110
4.3	Dynamic Power Reduction	113
4.3.1	Dynamic Power Management	113
4.3.2	Clock Gating and Frequency Scaling	117
4.3.3	Ultra-Low Voltage Memories	120
4.4	Ultra-Low-Voltage Dynamic Body Biasing	121
4.4.1	Concept of Back-Biasing at Ultra-Low-Voltage	121
4.4.2	Evaluation in 28 nm FD-SOI	123
4.4.3	Evaluation in 22 nm FD-SOI	126
4.4.4	System Integration	130
4.5	Silicon Implementations and Measured Performances	134

4.5.1	Testchips Realizations	134
4.5.2	Power Consumption Evaluation	138
4.5.3	Impact of State Retention Power Gating	143
4.5.4	Technology Scaling Evaluation	145
4.5.5	Utilization of Self-Body Bias	146
4.5.6	Towards Adaptive Power Management	150
4.6	Summary of Performances and Conclusions	152
4.6.1	Summary of Performances	152
4.6.2	Conclusions	153

Chapter 5. Adaptive Power Management 157

5.1	Activity-oriented Power Management	158
5.1.1	Problem Statement	158
5.1.2	Existing Solutions	158
5.1.3	System Definition	159
5.1.4	System Representation	163
5.1.5	Reinforcement-Learning through Q-learning	164
5.1.6	Application on an ULP SoC	167
5.2	Generic SoC Application	172
5.2.1	Simulation Environment and Methodology	172
5.2.2	Benchmarking of Several Reinforcement-Learning Solutions	173
5.2.3	Algorithm Robustness and Validations	175
5.2.4	Algorithm Optimizations	177
5.2.5	Selection of Q-learning Coefficients	180
5.2.6	Partial Conclusions	181
5.3	Application to the NZP SoC	184
5.3.1	Architecture Optimizations	184
5.3.2	Simulated Performances	185
5.3.3	Area and Power Evaluation	186
5.4	APM Hardware Description and SoC Implementation	188
5.4.1	APM Design	188
5.4.2	Integration into an ARM Cortex-M0+ Based Architecture	193
5.4.3	SoC physical implementation	195
5.5	Measured Performances	196
5.5.1	Functional Validation	196
5.6	Summary and Perspectives	199

Conclusion	201
Appendix A. ARM Cortex-M0+ Presentation	209
A.1 Reminders on Microprocessor Architectures	209
A.2 ARM Cortex-M0+ Presentation	211
Appendix B. Introduction to the Allan deviation	217
B.1 Stability and Accuracy	217
B.2 Frequency Stability Characterization	218
B.3 State of the Art Analysis	219
Appendix C. Electronic Design Automation Flow	221
C.1 Synthesis, Place-and-Route, Sign-Off	221
C.2 Unified Power Format	223
C.3 Simulations and Validation	224
Appendix D. UPF Descriptions for Power Gating	227
D.1 Power Gating Implementation	227
D.2 Double-Gated SRPG implementation	228
Appendix E. Measurement Setup	233
Appendix F. Register Configurations of the Hardware Power Management Modules	235
F.1 Power Management Unit Register Mapping	235
F.2 Wake-Up Controller Register Mapping	236
F.3 Adaptive Power Management Register Mapping	236
F.4 C-code memory mapping	237
Appendix G. Summary of Contributions	239
Biography	243
Bibliography	245

List of Figures

8	Picture of the first transistor ever assembled at Bell Labs in 1947.	xx
9	Computer hardware paradigms shifting and available electronic devices.	2
10	Computer systems according to their performances and power consumption. . .	3
11	Required autonomy of IoT devices depending on their area of application. . . .	4
12	Harvesters power generation and power consumption of electronic devices. . . .	5
13	Evolution of the computing efficiency at peak performance over time.	6
14	Evolution of battery energy density over the years.	7
15	Application contexts for NZP applications.	8
1.1	Integrated Circuit implementation stack.	13
1.2	Scaling of technology feature size and nominal supply voltage over time.	15
1.3	pMOS and nMOS transistors cross-sections in standard CMOS bulk process. . . .	16
1.4	Summary of leakage current mechanisms in a deep-submicrometer transistors. .	17
1.5	Transistor dynamic current according to the gate overdrive.	18
1.6	Visualization of the different process corners.	19
1.7	Body biasing capabilities in standard CMOS bulk process.	20
1.8	Schematic views of three types of transistor: planar Bulk, UTBB FD-SOI and FinFET.	21
1.9	28 nm UTBB FD-SOI CMOS transistors cross section and biasing capabilities. . .	23
1.10	Transient characteristics of an inverter.	24
1.11	Impact of the I_{ON}/I_{OFF} on the behavior of a standard inverter.	25
1.12	Charge and discharge currents during the switching operation of a CMOS inverter.	27
1.13	Energy according to the supply voltage.	29
1.14	Illustration of conventional DT-CMOS scheme.	30
1.15	Schematic illustration of conventional biasing and SWBB configurations.	31
1.16	Nominal cycle period and additional PVT margins.	32
1.17	Automatic clock gating performed by Synopsys design compiler tool.	33
1.18	Power gating implementations.	34

1.19 Activity scenario of a duty-cycled Ultra-Low-Power system.	36
1.20 Impact of system-level power management techniques on the power consumption. 40	
1.21 Schematic of SRPG implementation.	41
2.1 Energy flow for an ULP IoT device.	50
2.2 Relatively general architecture of IoT nodes with detailed sub-systems.	51
2.3 Clock modulation during the duty-cycled operation of an IoT device.	53
2.4 Energy driven use case.	54
2.5 Activity driven use case.	55
2.6 From activity to energy driven mode.	55
2.7 ULP SoC functional description.	57
2.8 Timeline of the Figure 2.7 activated domains according to the activity.	58
2.9 Architecturally defined memory map of the Cortex-M0+ implementation.	60
2.10 Adopted system partition for ULP applications.	61
3.1 Oscillator classification based on their physical type of operation.	65
3.2 Relocking principle to achieve ULP operations.	67
3.3 Block diagram of the clock generator architecture.	68
3.4 Schematic of the current controlled ring oscillator.	71
3.5 Current control schemes of a pMOS transistor.	71
3.6 Current modulation of a blocked pMOS transistor.	72
3.7 Schematic of the digitally controlled leakage sources.	72
3.8 Current sources selection scheme to reach a given target frequency.	73
3.9 Simulated LRO output frequency.	74
3.10 Evaluation of the LRO accuracy.	75
3.11 Main sources of variability for a FD-SOI transistor.	76
3.12 Silicon thickness evaluation of a wafer used in 28 nm FD-SOI technology.	76
3.13 PSD functions describing the surface topological variations of an SOI wafer.	77
3.14 Layout of the fine array composed of 31 leakage structures.	77
3.15 Chosen random distribution of the 31 leakage structure.	78
3.16 Testchip micrograph and view of LRO implementation.	78
3.17 DCO architecture using a SAR based locking circuit.	79
3.18 Evaluation of the tuning word in a SAR relocking scheme and impact of jitter.	80
3.19 Multi-level bangbang calibration.	80
3.20 Structural diagram of an Integral corrector.	81
3.21 Control Logic Unit block diagram.	83
3.22 Stability analysis of the compensation stage.	84

3.23 Stability analysis of the compensation stage with a corrector gain $K_c = -5$	84
3.24 Block diagram of the complete system.	85
3.25 SPI and communication interface protocol with the system.	85
3.26 Comparison of simulated and measured LRO frequency ranges	86
3.27 Measured peak-to-peak and RMS jitter for 10 dice.	87
3.28 Evaluation of the locking scheme.	87
3.29 Evaluation of the output frequency when locked.	88
3.30 Measured output frequency when locked.	89
3.31 Measured LRO Allan deviation.	89
3.32 Example of a wireless sensor node operation scenario.	90
3.33 Evaluation of the total oscillator power and drift trade-off.	91
3.34 Power overhead resulting from the utilization of the relocking scheme.	93
3.35 Comparison of Allan Deviation/Power with State of the Art clock generators.	95
4.1 qFO4 logical path schematic.	100
4.2 Evaluation of Cortex-M0+ critical paths length.	100
4.3 qFO4 frequency according to the PVT conditions.	101
4.4 Energy performance of qFO4 paths in 22 nm FD-SOI.	102
4.5 Comparison of FD-SOI technology nodes using qFO4 paths.	102
4.6 Subset of Cortex-M0+ Synthesis loops performances.	103
4.7 Global power switch design using LVT flavor pMOS.	105
4.8 Layout view of the global pMOS power switch.	106
4.9 Integration of the global power switch into the core power grid.	107
4.10 Performances evaluation of the global power switch.	108
4.11 Schematic floor plan for distributed power switches.	108
4.12 Power gating standard cell layout and integration.	109
4.13 Performance evaluation for two types of RVT header cells in 22 nm FD-SOI.	109
4.14 Schematic of a double-gated SRPG implementation.	110
4.15 Double-gated SRPG implementation in 22 nm FD-SOI.	111
4.16 Layout abstraction for double-gated 22 nm FD-SOI SRPG implementation	112
4.17 Interruption handling during suspend modes and PMU control signals.	114
4.18 PMU control sequence and signal generation during suspend mode transitions.	115
4.19 Context restoration and IRQ handling during suspend mode with no retention.	116
4.20 Simulated behavior of the PMU	117
4.21 Block diagram of the DDSS clock generator.	118
4.22 Block diagram of the DFS implementation at system level.	119

4.23 Proposed 28 nm FD-SOI body-biasing modes.	121
4.24 Evaluation of the supply voltage before well junction leakage occurs.	122
4.25 qFO4 frequency using NOM and FBB modes according to the supply voltage. . .	123
4.26 Evaluation of the energy per cycle according to the frequency.	124
4.27 Leakage power of a qFO4 according to the Polysilicon Biasing.	124
4.28 Frequency of a qFO4 according to the temperature and the supply voltage. . . .	125
4.29 Power consumption of a qFO4 according to the temperature and the supply voltage.	126
4.30 Proposed 22 nm FD-SOI body-biasing modes.	127
4.31 Evaluation of a qFO4 frequency using NOM and FBB modes.	127
4.32 Performance analysis of the SWBB modes in 22 nm FD-SOI	128
4.33 Rising edge propagation along a qFO4.	129
4.34 Frequency of a qFO4 according to the temperature and the supply voltage. . . .	129
4.35 Power consumption of a qFO4 according to the temperature and the supply voltage.	130
4.36 Self body bias generator using standard inverters.	131
4.37 Overview of the system floorplan with integrated self body-biasing.	132
4.38 Layout abstraction of the SWBB implementation.	133
4.39 Self body bias generator mode selection through PMU controls.	133
4.40 NZP28 V1.0 – 28 nm FD-SOI implementation.	135
4.41 NZP28 V2.0 – 28 nm FD-SOI implementation.	135
4.42 NZP28 V3.0 – 28 nm FD-SOI implementation.	136
4.43 NZP22 V1.0 – 22 nm FD-SOI implementation.	137
4.44 NZP22 V2.0 – 22 nm FD-SOI implementation.	137
4.45 Cumulative percentage of working parts of NZP28 V1.0.	138
4.46 Measured mean core frequency of NZP28 V1.0 according to the supply voltage. .	139
4.47 NZP28 V1.0 power evaluation according to the supply voltage.	140
4.48 NZP28 V1.0 SoC power breakdown.	140
4.49 Influence of frequency scaling on the power consumption.	141
4.50 Reduction of the leakage power using RBB.	142
4.51 Total power breakdown of the NZP28 – V1.0 SoC.	142
4.52 Power impact of SRPG in 28 nm.	143
4.53 Area impact of SRPG in 28 nm on the whole SoC.	144
4.54 NZP V2.0 and V3.0 power consumption in 28 nm FD-SOI.	144
4.55 Energy/Frequency Trade-off analysis in 28 nm and 22 nm FD-SOI	145
4.56 Leakage Power according to the supply voltage in 28 nm and 22 nm FD-SOI . . .	146
4.57 Measured NZP22 V1.0 frequency according to the supply voltage and bias. . . .	147
4.58 NZP22 V1.0 measured SoC's performances.	147

4.59	NZP22 V1.0 measured maximum frequency.	148
4.60	NZP22 V1.0 SoC's power with temperature compensation scheme enabled. . . .	149
4.61	Evaluation of self-body bias scheme on NZP28 V3.0 for energy/frequency trade-off.	149
4.62	NZP28 V3.0 SoC's power with FBB scheme enabled for temperature compensation.	150
4.63	Evolution of the energy consumption depending on the selected power state. . .	151
5.1	System activity describing various suspend scenarios.	160
5.2	Energy consumed by the system according to the heuristic solution selected. . .	161
5.3	Color map representation of the deep sleep heuristic solution.	162
5.4	Example of a simple finite Markov Decision Process.	163
5.5	TiSMDP representation using 3 states.	164
5.6	Basic agent and environment interactions used in RL algorithms.	165
5.7	Generic SoC architecture used as a template for the APM implementation. . . .	168
5.8	Generic system Markov Decision Process	168
5.9	MDP diagram of the system and Q-matrix associated.	169
5.10	Definition of the energy consumed during the system transitions.	170
5.11	Mixed HDL–Python conception and validation flow	172
5.12	Evaluation of different RL-algorithms.	173
5.13	Energy color map of a the Q-learning algorithms with two system representations.	174
5.14	Comparison of the heuristic solution with the Q-learning algorithm.	175
5.15	Evaluation of the robustness and stability of the Q-learning.	176
5.16	Evaluation of three different exploration policies.	178
5.17	Energy colormap of the implemented Q-learning solution.	179
5.18	APM performances according to the learning rate and discount factor.	180
5.19	Impact of the learning rate on the APM performances.	181
5.20	Comparison of the energy consumption gains and losses of the generic APM. . .	182
5.21	Simplified NZP System MDP	184
5.22	Comparison of the energy consumption gains and losses of the NZP-oriented APM.	185
5.23	Interaction model between the APM and the PMU of the SoC.	188
5.24	FSM of the APM HW module.	189
5.25	Suspend sequence timing diagram of the APM – Beginning.	190
5.26	Suspend sequence timing diagram of the APM – End.	190
5.27	Block diagram of APM HW module.	191
5.28	Verilog code description to generate the Q_{matrix} coefficients.	192
5.29	System integration of PMU, WUC and APM modules.	194
5.30	Block diagram of the NZP22 – V2.0 integrating the PMU, WUC and APM modules.	195

5.31 Simulated behavior of the APM module.	197
5.32 Simplified flow chart describing the functional validation.	198
5.33 List of the main contributions from device up to system level.	203
A.1 Typical Microprocessor Unit architecture including its subsystems.	209
A.2 Functional block diagram of the ARM Cortex-M0+.	212
A.3 Two-stage pipeline representation of the Cortex-M0+.	215
B.1 Accuracy and stability evaluation.	217
B.2 Allan deviation of timers with regard to their power consumption.	219
C.1 Electronic Design Automation flow for digital circuit implementations.	221
C.2 Dummy system for power intent basic definitions.	224
D.1 UPF standard description to define a power gating strategy.	227
D.2 Power switches liberty files attributes.	228
D.3 Triple power grid description for SRPG implementation in 22 nm FD-SOI.	229
D.4 Power intent description for SRPG implementation (1/3).	230
D.5 Power intent description for SRPG implementation (2/3).	231
D.6 Power intent description for SRPG implementation (3/3).	232
E.1 Picture of the daughter board and instrumentation setup.	234
E.2 FPGA board instrumentation setup under utilization.	234
F.1 C-code header file giving the HPM modules memory mapping.	238

List of Tables

1.1	Effect of device scaling on some of the transistor intrinsic parameters.	14
1.2	Logic Core Device Technology Roadmap based on IRDS projections.	15
1.3	Suspend mode consumption and autonomy for two standard BLE devices.	37
1.4	Summary of the Ultra-Low-Power techniques encountered.	42
1.5	ULP SoC and referencing of the low-power optimizations (1/2).	45
1.6	ULP SoC and referencing of the low-power optimizations (2/2).	46
2.1	SoC power modes and corresponding peripheral states.	58
3.1	Qualitative study of electronic oscillators.	66
3.2	CMOS-based oscillators summary of performances	70
3.3	LRO \Leftrightarrow CLU Interface	73
3.4	Oscillator frequency specifications in TT.	74
3.5	Design parameters of the devices used.	75
3.6	Qualitative study of digital compensations.	81
3.7	Evaluation of the drift and corresponding mismatch power of the relocking scheme.	92
3.8	Summary of the achieved performances and state of the art comparison.	94
4.1	Shut-off performance analysis for a single pMOS transistor.	106
4.2	System partitions and proposed ULP modes.	113
4.3	DFS and clock generator modes.	120
4.4	NZP28 V1.0 SoC energy/cycle and total power breakdown.	141
4.5	Summary of the NZP V1.0 power modes and configuration.	142
4.6	Estimation of the SRPG impact on the power consumption.	143
4.7	SRPG impact on the area.	145
4.8	NZP22 – V1.0 power breakdown according to the ULP modes.	148
4.9	ULP SoC and referencing of the low-power optimizations performed.	154
5.1	Evaluation of different systems for Adaptive Power Management implementation.	162

5.2	Comparison of Reinforcement Learning algorithms implementations.	166
5.3	Performance evaluation of the APM according to the minimum achievable energy.	175
5.4	Performance evaluation of the APM relatively to the Deep Sleep heuristic solution.	182
5.5	Evaluation of the APM HW module on a complete SoC.	183
5.6	Performance evaluation of the NZP-oriented APM.	185
5.7	Performance comparisons of two versions of the APM.	186
5.8	Performance evaluation of the APM relatively to the NZP22 – V2.0.	186
B.1	Corresponding references and data associated to Figure B.2.	220
E.1	PMUCFG registers description.	235
E.2	WUCCFG register description.	236
E.3	ALPHA register description.	236
E.4	ALPHAGAMMA register description.	237
E.5	ALPHACOMPLEMENT register description.	237
G.1	Chips design gallery.	241

Abbreviations

A.ON	Always-ON
ABB	Adaptive Body Biasing
AC	Alternative Current
ADC	Analog-to-Digital Converter
AFS	Adaptive Frequency Scaling
AHB	Advanced High-performance Bus
AI	Artificial Intelligence
ALU	Arithmetic Logic Unit
AMBA	Advanced Microcontroller Bus Architecture
APB	Advance Peripheral Bus
API	Application Programming Interface
APM	Adaptive Power Management
ARM	Advanced RISC Machines
AVFS	Adaptive Voltage and Frequency Scaling
BoM	Bill of Material
BOx	Buried Oxide
C-ABI	C-Application Binary Interface
CAD	Computer Aided Design
CEF	Constant Electric Field
CISC	Reduced Instruction Set Computer
CLU	Control Logic Unit
CMOS	Complementary Metal Oxide Semiconductor
cocotb	COroutine based COsimulation TestBench
CPU	Central Processing Unit
CV	Constant Voltage
DAC	Digital-to-Analog Converter
DAP	Debug Access Port
DCO	Digitally Controlled Oscillator
DDSS	Direct Digital Sampling and Synthesis
DFS	Dynamic Frequency Scaling
DG	Double Gate

DIBL	Drain Induced Barrier Lowering
DMA	Direct Memory Access
DPM	Dynamic Power Management
DRC	Design Rules Check
DSP	Digital Signal Processor
DT-CMOS	Dynamic-Threshold CMOS
DUT	Device Under Test
DVFS	Dynamic Voltage and Frequency Scaling
DVS	Dynamic Voltage Scaling
EDA	Electronic Design Automation
EWS	Electrical Wafer Sorting
FBB	Forward Body Biasing
FD-SOI	Fully Depleted Silicon On Insulator
FET	Field Effect Transistor
FF	Fast-Fast
FinFET	Fin Field Effect Transistor
FMC	FPGA Mezzanine Card
FO4	Fan-Out of 4
FoM	Figures of Merits
FPGA	Field-Programmable Gate Array
FS	Fast-Slow
FSM	Finite State Machine
GAA	Gate-All-Around-Device
GDS	Graphic Database System
GIDL	Gate Induced Drain Leakage
GPIB	General Purpose Interface Bus
GPIO	General Purpose Input Output
GPU	Graphics Processing Unit
HDD	Hard Disk Drive
HDL	Hardware Description Language
HPM	Hardware Power Management
HVT	High Voltage Threshold
HW	Hardware
I2C	Inter-Integrated Circuit
IC	Integrated Circuit
ICT	Information and Communications Technology
IDM	Integrated Device Manufacturer
IEEE	Institute of Electrical and Electronics Engineers

IIR	Infinite Impulsion Response
IO	Input/Output
IoT	Internet-of-Things
IP	Intellectual Property
IRDS	International Roadmap for Devices and Systems
IRQ	Interrupt ReQuest
ISA	Instruction Set Architecture
ITRS	International Technology Roadmap for Semiconductors
JTAG	Joint Test Action Group
LGAA	Lateral Gate-All-Around-Device
LRO	Leakage-based Ring Oscillator
LSB	Least Significant Bit
LVT	Low Voltage Threshold
MC	Monte Carlo
MCML	Mos Current Mode Logic
MCU	Microcontroller Unit
MDP	Markov Decision Process
MEM	Memory
MEMS	Micro Electro-Mechanical System
MEP	Minimum Energy Point
ML	Machine Learning
MOSFET	Metal-Oxide-Semiconductor Field Effect Transistor
MPPT	Maximum Power Point Tracking
MPU	Microprocessor Unit
MT-CMOS	Multi-Threshold CMOS
NMI	Non-Maskable Interrupt
nMOS	n-doped MOSFET
NN	Neural-Networks
NOM	NOMinal Biasing
NVIC	Nested Vectored Interrupt Controller
NVM	Non-Volatile Memory
NZP	Near-Zero-Power
P.OFF	Power-OFF
PB	Polysilicon Biasing
PCB	Printed Circuit Board
PD-SOI	Partially Depleted Silicon On Insulator
PDK	Product Design Kit
PDN	Pull-Down Network

PG	Power/Ground
PI	Proportional Integral
PLL	Phase Locked Loop
pMOS	p-doped MOSFET
PMU	Power Management Unit
PoC	Proof-of-Concept
PSD	Power Spectral Density
PUN	Pull-Up Network
PVT	Process, Voltage and Temperature
PVTSC	Process Voltage Temperature Slope Capacitance

qFO4	quasi Fan-Out of 4
QL	Q-Learning

RAM	Random Access Memory
RBB	Reverse Body Biasing
RF	Radio Frequency
RFID	Radio Frequency Identification
RISC	Reduced Instruction Set Computer
RISC-V	Reduced Instruction Set Computer five
RL	Reinforcement Learning
RMS	Root Mean Square
RO	Ring Oscillator
RTC	Real Time Clock
RTL	Register-Transfer Level
RTOS	Real Time Operating System
RVT	Regular Voltage Threshold
RXEV	Receive Event

SAR	Successive Approximation Register
SARSA	State-Action-Reward-State-Action
SBGA	Super Ball-Grid Array
SCE	Short Channel Effects
SCR	State Control Register
SF	Slow-Fast
SLVT	Super Low Voltage Threshold
SMA	SubMiniature version A
SMDP	Semi Markov Decision Process
SoC	System-on-Chip
SOI	Silicon On Insulator
SON	Silicon On Nothing
SPI	Serial Peripheral Interface
SQUAL	Space Qualification

SRAM	Static Random Access Memory
SRPG	State Retention with Power Gating
SS	Slow-Slow
STA	Static Timing Analysis
STM	STMicroelectronics
STSCL	Sub-threshold Source-Coupled Logic
SW	Software
SWBB	Swapped Body Biasing
SWD	Serial Wire Debug
TD	Time Difference
TDP	Thermal Design Power
TiSMDP	Time-indexed Semi Markov Decision Process
TT	Typical-Typical
UART	Universal Asynchronous Receiver Transmitter
UCB	Upper Coefficient Bound
ULP	Ultra-Low-Power
ULV	Ultra-Low-Voltage
UPF	Unified Power Format
USB	Universal Serial Bus
UTBB	Ultra Thin Body and BOx
UWVR	Ultra Wide Voltage Range
VGAA	Vertical Gate-All-Around-Device
VHDL	Very High Speed Integrated Circuit Hardware Description Language
WFE	Wait For Event
WFI	Wait For Interrupt
WIC	Wake-up Interrupt Controller
WSN	Wireless Sensor Node
WUC	Wake-Up Controller
XO	Crystal Oscillator
ZIF	Zero Insertion Force

Introduction

PULSED by the multiplication of wireless, battery operated, interconnected, and low cost applications, it is established that connected objects will occupy a predominating position in the electronic landscape [1]. However, this continuous growth is limited by the power consumption of the various elements embedded on these increasingly complex objects. Henceforth, this is a twofold problem. First, the fast increase in energy requirements due to the growing number of devices will be substantial. Second, the heterogeneous applications will suffer from limited autonomy and the deterioration of on-board batteries.

This trend sets the general context of this work. In this general introduction, the notion of Internet-of-Things and the power constraints associated are exposed. Then, starting from the evolution of computing power consumption, we will move towards the NZP concept to understand how a SoC can target utmost energy efficiency and extend the sensor's lifetime from weeks to years.

New Paradigms of Computer Hardware

Over the past sixty years the Information and Communications Technology (ICT) industry has significantly evolved (see Figure 9). From the mainframe era in the mid-60s, when one generally tremendous equipment was operated by many people, the computer hardware paradigm has shifted in the 2000s towards the personal computing era when personal computers were widely spreading [11]. Nowadays, the user-to-device ratio is inverting with one user using many devices.

From personal vehicles to home appliances through smartphones and watches, people are outnumbered by the devices they access. More importantly they are surpassed by the number of nodes such as servers and other infrastructures they rely on. All these items are now embedding electronics, software, sensors, actuators, and connectivity functions which enables them to connect and exchange data, creating an advanced network of physical devices called the Internet-of-Things (IoT).

Accordingly, the power consumption and computing power has ushered a quest for power efficiency across the whole spectrum of computer systems (see Figure 10). From supercomputers like the US champion Summit with 143.5 PFLOPS and ~ 9.8 MW power consumption [17] to tiny implantable electronic devices with low activity and micro watt power consumption, power constraints and budgets are becoming a major issue.

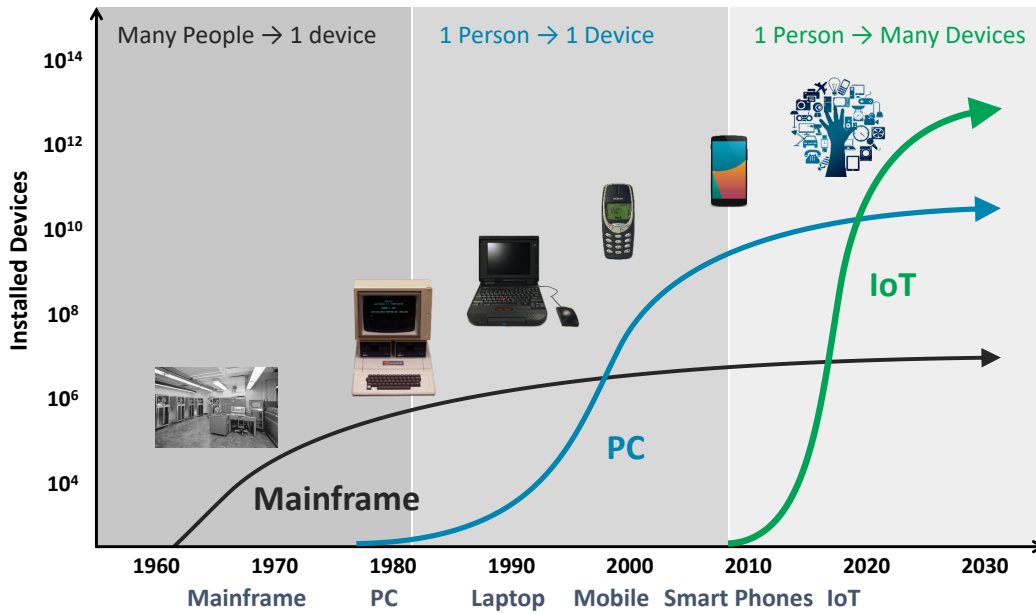


Figure 9: Computer hardware paradigms shifting and available electronic devices over the years. Based on data from [12–14]. The superimposed S-curves shows the adoptions trends [15]. Following Bells Law, a new "minimal" computer class emerges approximately every decade [12, 16].

Since November 2007, the TOP500 [18] project which ranks and details the 500 most powerful non-distributed computer systems in the world has been complemented with the unveiling of the Green500 project [19]. This new list puts a premium on the top 500 supercomputers in the world considering their performance-per-watt in order to target sustainable super-computing. As a matter of fact, the “performance-at-any-cost” has led to the emergence of supercomputers that consume vast amounts of electrical power. So much heat is generated that large cooling facilities are required for proper performance.

Moreover, with the increase of the Internet and cloud-based data storage, server type computers are spreading, and data centers energy-efficiency is becoming critical [20]. In 2005, the power and cooling cost reached \$26.7Bn [21] representing an estimated 1% of the world electricity [22]. Besides, with the current environmental issues and energy resources availability, companies are redesigning data centers to maximize efficiency and reliability [23]. For instance, Facebook relocated one of its data center in Luleå, Sweden, near the Arctic Circle [24], and Microsoft is now operating a data center under the sea for natural cooling [25]. Both entities aiming to reduce their electricity bill and environmental impact.

Desktop computing, laptops and all the consumer electronic products associated are also changing. Whereas 2000s’ advertisements were selling computers with the highest processor frequency and operations per unit of time, current products are branding quieter and lighter devices with extended autonomy. Thermal Design Power (TDP) which reflects the maximum amount of heat generated by a component is also newly showcased on processor specifications. Heat dissipation techniques have therefore been extensively investigated for noise and comfort yet it has also resulted in an increased power constraining.

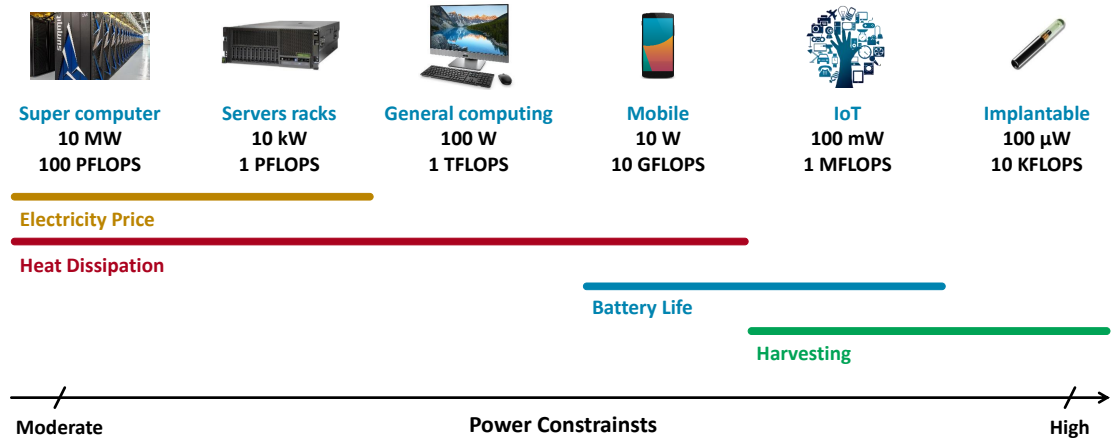


Figure 10: Computer systems according to their performances/power (in gray) and related power constraints.

For smartphones and Internet-of-Things (IoT) devices, the power budget is the most constrained. In this case, the autonomy has become the principal concern since technological advancements in term of battery are relatively slow compared to other electronic trends. Moreover, with a growing demand for compact devices, power restrictions are mandatory to reduce the need for bulky heat sinks. However, the advent of smartphones has favorably contributed to reduce the energy consumed by Integrated Circuits (ICs) by offering multiple solutions: using multiple cores to perform tasks in parallel [26, 27], reducing the frequency for tasks with low computing power requirements [28], or disabling unused elements by the task being performed [29]. Nevertheless, these innovations have for context the specific case of the smartphone, its homogeneous composition and performance needs.

Although beneficial for IoT systems, these techniques are not sufficient given the numerous contexts of end application and do not fully meet the growing demands for autonomy [30]. With foresight projections of 30.7 billion devices expected in 2020 [31], IoT is considered to be the next decade market, allowing a wide variety of low cost and connected applications in the field of health care and wellness, wireless monitoring to observe and process information from various sensors, or for wearable equipment such as watches and intelligent clothing [32].

Lastly, battery-operated only devices are no longer the only sustainable option [2]. Solutions called energy-harvesters are able to extract unused energy from their environment and convert it into useful power in the form of voltage or current. However, the energy collected is relatively limited for small devices such as wearable and implantable devices. Upcoming SoCs must therefore overcome the challenge of collecting, storing and managing energy for efficient consumption and distribution [3].

Hence, the core of this thesis research will focus on devices targeting IoT applications and beyond. As a consequence, only battery-operated systems with or without complementary energy-harvesters are going to be taken into account.

Autonomy and Power Constraints for IoT Systems

To evaluate the power constraints for IoT-oriented devices, it is mandatory to consider a broad spectrum of applications. Figure 11 gives an overview of the required autonomy for various use cases.

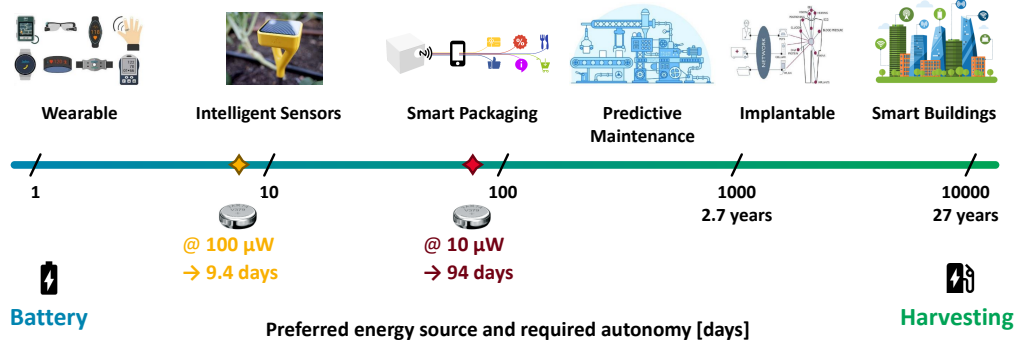


Figure 11: Required autonomy of IoT devices depending on their area of application. Based on data from [33–35],. The autonomy of a standard 1.5 V silver-oxide SR63 battery ($\sim 1 \text{ cm}^2$) is also reported for two types of power consumption (10 μ W and 100 μ W).

On the one side, with a lifetime expectancy up to twenty days, applications such as wearable devices, smart packaging or active Radio Frequency Identification (RFID) tags are the less constrained. Generally, it corresponds to devices that can be easily recharged or the energy source replaced. On the other side, for operations aiming to monitor physical or chemical values within buildings, the lifetime follows the restoration works (expected once every 25 to 30 years for commercial buildings [36]). As an example, the autonomy provided by a standard 1.5 V/14.5 mAh silver-oxide SR63 battery ($\sim 1 \text{ cm}^2$) for two devices, consuming respectively 10 μ W and 100 μ W, is given. It leads to a lifetime of respectively 93.6 days and 9.4 days.

Whenever the power needed or the physical constraints can be fulfilled with a battery, a battery would be preferred above an energy harvester. Indeed, battery technologies are well proven and the existing eco-system of solutions would reduce the costs. Moreover, for replaceable or disposable devices, rechargeable batteries can provide the necessary power and be recharged to re-use the device [34].

For non-accessible devices such as implantable trackers or monitors embedded inside the structure of a building, due to the environment constraints and a limited physical access, the autonomy required is larger. In that case, the straightforward battery-based solution (which could include recharging possibilities) is not explicit and a complementary energy-harvester might be required.

An overview of the different types of available energy sources that can be harvested is given in Figure 12. Because of the limited efficiency of the energy-harvesting power management this available power cannot be fully harvested with a perfect conversion process. For instance, [33] reports 0.1% to 3% physical efficiency for a thermal energy source, 10% to 24% physical efficiency for a light source and 50% for electro-magnetic (RF) energy. The distance from the source has also an impact on the energy collected. The estimation of available harvested power is reported for a standard area of 1 cm^2 : 1 μ W is obtained from RF harvesting,

10 μW from indoor light, 100 μW from vibrations, 10 mW for outdoor light or industrial temperature difference [37]. At the bottom are reported various consumer electronic devices and their present power consumption.

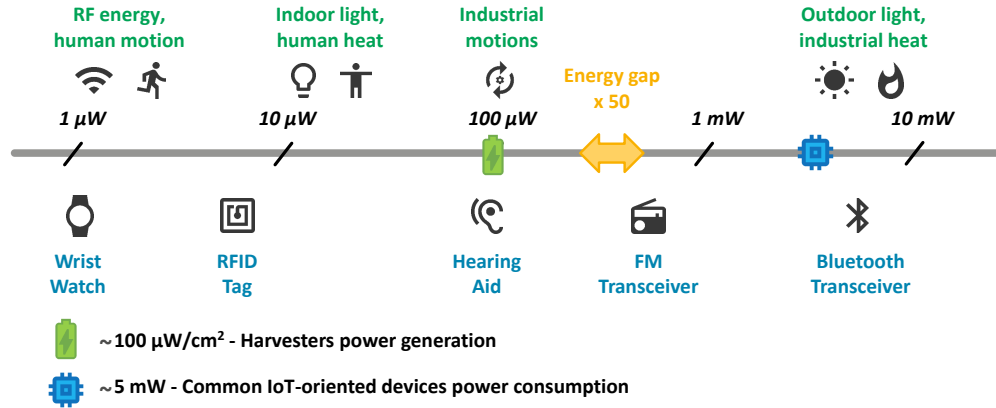


Figure 12: Harvesters power generation per cm^2 compared to the power consumption of common electronic devices.

The energy harvesting option drastically restricts the application power level. Within the context of indoor IoT's SoCs, a realistic $\sim 100 \mu\text{W}/\text{cm}^2$ power budget can be defined. As an illustration, the high-performance processor Intel i9-9900K reaches 95 W TDP [38] and a smartphone processor like the Snapdragon 855 reaches 4.4 W [39].

However, depending on the application [1], autonomous sensor nodes and IoT-oriented devices' energy consumption are above 5 mW. Those systems' power consumption are 50 times higher, thus hardly sustainable by integrated energy harvesting solutions, leading to an energy gap. Hence, it is mandatory to bring back the energy consumption of electronic circuits in the range of the energy harvested in order to reach battery free operation.

Power Scaling

From mainframe computers to personal devices, the number of computation per kilowatt-hour (i.e., the computing efficiency) over the years has followed a remarkably stable trend since the 1950s (see Figure 13). Indeed, from 1946 to 2009, the number of calculations per joule of energy spent has been doubling every 18 months (1.57 years). This conjectures that the amount of power required for a given computing load will fall by a factor of 2 every 1.5 years, as stated by Jonathan Koomey in [4].

This evolution, also known as Koomey's law, has been initially led by a race to increase the computational speed and reliability issues of the first vacuum-tube computers. Later, it was followed by transistor based computers¹, thanks to the reduction of manufacturing costs and increasing speed operated by decreasing the transistor size (which in return brought power reduction as explained in Section 1.1).

¹In April 1955, IBM announced the IBM 608; the first completely transistorized computer available for commercial installation [40].

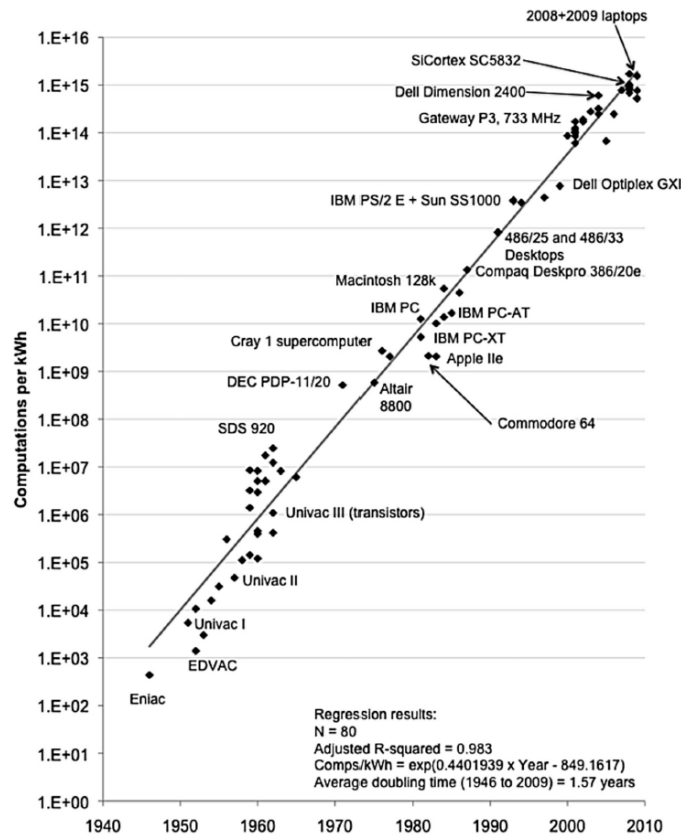


Figure 13: Evolution of the computing efficiency at peak performance over time. From 1946 to 2009, regression results show a doubling of the efficiency every 1.57 years [4].

Following the implications of this phenomenon, and assuming a steady harvester power generation, the x50 gap previously stated would be filled after almost 9 years (~ 2027). Hence, in order to rapidly enable new smart objects with charging-free operations, designers and manufacturers must offer design innovations to reach harvesters capabilities.

Looking at the battery autonomy point of view, the energy density evolution is reported in Figure 14 for various technologies. The data points are obtained using scientific literature for advanced Technology and/or manufacturer data-sheets. Extrapolation based on these results and estimations on chemical and material developments helps to overview future trends [34].

A x2 or x3 growing factor is observed per decade considering the last ten years. However, cutting edge publications in this field of research are claiming that improvements in device concepts or new material [41, 42] will lead to an acceleration of energy density capabilities. Yet, packaging limitations are restraining the commercial availability of such technologies.

Moreover, as shown in Figure 14, the energy density of micro batteries, thin-film and 3D devices are increasing at faster paces [43]. Nonetheless due to their smaller density these solutions would not replace standard macro batteries for many years. These trends confirm the need for advanced low power electronic systems.

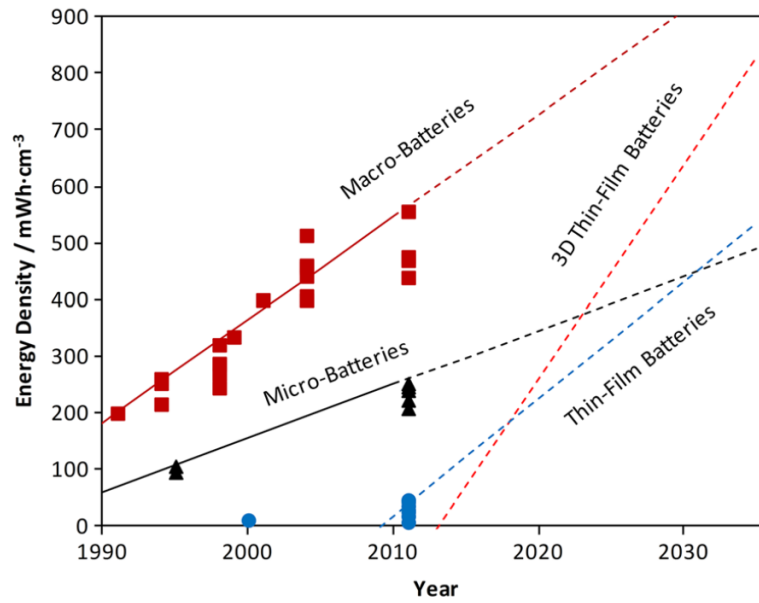


Figure 14: Evolution of battery energy density over the years [34].

Towards Near-Zero Power

Minimizing the SoCs power consumption has become the cornerstone for a wide range of applications. The generalized growth of smartphone usage over the last decade had opened a way to efficient and powerful electronic devices. However, the smartphone-oriented technical solutions developed do not fully meet the growing demands for autonomy.

On the contrary, for IoT applications, regarding to the very limited energy budget, performance requirements are relaxed and Ultra-Low-Voltage (ULV) digital circuits become relevant. By targeting SoCs that operate in the range of tens of MHz, some room is offered to imagine battery-powered applications with a prolonged lifetime. Combined with energy harvesters, ULP SoCs can be designed to avoid costly and time-consuming redeployment of any power-depleted sensor.

An overview of various application contexts where we could benefit from the IoT-oriented specifications is given in Figure 15. Autonomous interconnected sensors ensure a constant physical and chemical monitoring of an area for the health of ecological environments and building safety. For instance, by signaling when cracks develop in bridges or fire ignites in structures, severe degradations are prevented as soon as possible [44].

Presence detection and perimeter monitoring is also of a great interest. The analysis and sorting of various objects' patterns (vehicles, humans, events, etc.) is done by hundreds of sensors deployed over a specific area. Such information help to redesign cities and improve dwellers quality of life by tackling the challenges of modern urban growth [45].

Lastly, swarm robotic is a challenging and promising approach. Instead of being seen as one large complex system, robots can be seen as many simple elements with limited functions. Like ants, swarm robots show a collective behavior allowing complex tasks realization. To implement these robots, a very power efficient SoC is required as the main unitary block.

This panel of applications is theorized in the “Near Zero Power RF and Sensor Operations (N-Zero)” program from DARPA [46]. It follows the goal of developing the technological foundations to overcome the power limitations of the current Wireless Sensor Nodes (WSNs). While focusing on the radio and sensors, the program also intends “to develop underlying technologies to continuously and passively monitor the environment and activate an electronic circuit only upon detection of a specific signature”. The sensors should remain dormant yet be aware until an event of interest awakens them.

From this perspective, this work extends the concept to the digital core to avoid power consumption gaps between the radio, sensors and processing elements. Near-Zero power consumption when inactive combined with an ULP consumption extend the system lifetime from weeks to years. It cuts down the costs of maintenance and make energy-harvesting powered systems widely available. Alternatively, it leads to reduce the battery size for a standard battery-operated sensor while still guaranteeing its current operational lifetime.

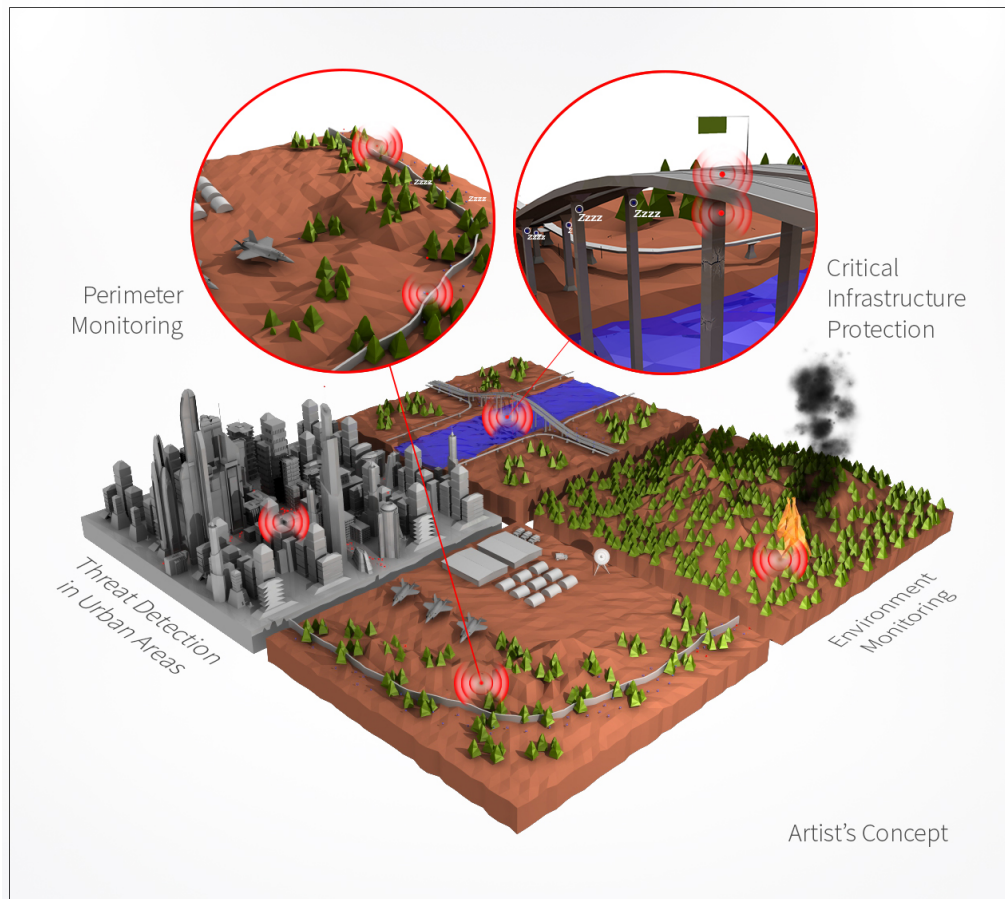


Figure 15: Application contexts for Near-Zero-Power (N-ZP) applications [47].

Challenges

The N-Zero project supports that connected devices and applications will be accessible by shifting the power consumption paradigms towards Near-Zero-Power energy-efficient SoCs.

In this vein, this dissertation answers the following questions:

1. At system level, how the Near-Zero-Power concept can be applied to produce low-cost energy-efficient SoCs?
2. What are the techniques to reduce the mass market microcontrollers power consumption down to the micro watts, based on the FD-SOI technology and the latest industrial workflows?

Additionally, through the realization of circuits with reduced energy footprint, this work targets solutions that goes beyond present connected objects applications. Unlike common IoT devices, which take advantages of a macroscopic energy reserve offered by their battery and offer low or high-performance services, NZP SoCs target specific operations at low energy cost while minimizing the power consumption during suspend modes.

Current technological constraints limit the power budget in the order of hundreds of microwatts and force innovations in the design of conventional circuits at all levels: cells, processors, system architecture, communication, software and energy storage. Since all areas cannot be covered, the following challenges are addressed:

- **ULP microcontroller implementation:** Generic portable applications require a standard microprocessor core with reconfiguration capabilities. Using appropriate benchmarking of the FD-SOI technology features, standard cells, Intellectual Properties (IP) and implementation flow, a ≥ 10 s of MHz ULV and ULP digital core with high energy-efficiency must be demonstrated.
- **ULP clock references and generators:** Clock sources are required for precise time and frequency keeping and correct operations of low power SoCs. Focus should be done on implementing ULP clocking solution for constrained budgets, while maintaining the accuracy and stability.
- **Efficient power island partition:** After identification of the mandatory components of a SoC, architecture optimizations and power island partitions are needed to wake up the system modules useful for the execution of a task while switching off the unnecessary parts. Context restoration and implementation of state saving should be mitigated to avoid unnecessary power consumption. An improved architecture is mandatory for versatile and efficient power modes.
- **Reduce the Always-On power consumption:** The power consumption of the constantly powered parts (i.e., Always-on) of the system should be greatly reduced. Hence, it would avoid energy draining during the system suspend sequences.
- **Power Management Unit:** Based on the previous features, an efficient Power Management Unit (PMU) is required to enable suspend modes. Then, depending on the application requirements, parts of the design are disabled and wake up on certain hardware events. Such technique is used at system level for dynamic power reduction.
- **Adaptive Power Management:** A SoC with ULP power modes requires a close interaction with its application and the environment to determine the optimal suspend mode according to the system activity and wake-up sequences. Intelligence should be added to generic microcontrollers, through Adaptive Power Management (APM) combined with embedded learning, to optimize the power consumption based on the system activity.

Thesis Outline

Chapter 1 – Background An introduction of all the necessary background knowledge to understand this work is done along with a focus on the latest State of the Art ULP SoCs. Since power optimizations are done on multiple scale, a brief review of low power techniques is performed from the transistor up to the system level.

Chapter 2 – System Definition The ULP SoC architecture developed during this work to reach maximum energy efficiency is detailed. It results from several uses cases analysis and activity scenarios. This system employs optimizations at all levels of implementation in the context of energy limited resources.

Chapter 3 – Clock Reference for Ultra-Low-Power Systems Since low power systems require precise time and frequency sources to ensure correct operations, an ULP clock reference is introduced in this chapter. It uses a leakage-based Ring Oscillator combined with a digital compensation logic for optimal power, area and stability trade-offs.

Chapter 4 – ULP System Implementation Several ULP techniques are implemented in SoCs fabricated in 28 nm and 22 nm FD-SOI technologies. Components selection based on adequately chosen FD-SOI technology features, architecture optimizations and design trade-offs are bench-marked to improve the system energy efficiency. Static and dynamic power reduction techniques are combined with Dynamic Power Management (DPM) to implement versatile system suspend modes.

Chapter 5 – Adaptive Power Management An investigation of APM for optimal selection of the SoC ULP modes is done. Using Machine Learning (ML) techniques based on Reinforcement Learning (RL) algorithms, a complete module is integrated into a SoC architecture for selection of the best suspend mode depending on the system activity.

Chapter 5.6 – Conclusion The developments presented in this thesis are summarized. Aspects of possible improvements and future works are finally established.

Appendices Several appendices are available at the end of this manuscript to highlight or describe some of the concepts used in this manuscript. A presentation of the digital core implemented in this work is provided in Appendix A. The Allan deviation, a clock reference stability metric is explained in Appendix B. The standard Electronic Design Automation (EDA) flow used to design SoCs is given in Appendix C. Appendix D illustrates with Unified Power Format descriptions the techniques presented in Chapter 4. The FPGA setup implemented to get measurement on the system presented in this work is shown in Appendix E. A description of the configuration registers of the Hardware Power Management (HPM) modules, specified in Chapter 5, is provided in Appendix F. Finally, Appendix G lists the publications arising from this work.

Chapter 1

Background

SINCE existing SoCs solutions do not reach our power consumption goals, it is necessary to review the full ICs implementation stack (see Figure 1.1). From the silicon technology and circuit perspective up to the system level, a bottom-up approach is covered in this chapter to seek design choices for energy-efficiency and power reduction on all design levels.

First, in Section 1.1 is recalled how the constant voltage scaling operated between silicon technology node has resulted in a constant energy reduction. Then, in Section 1.2 is introduced the MOSFET transistor in CMOS technology for low power and low voltage applications. Advanced technology nodes are also presented, with a special emphasis on FD-SOI for its benefits in energy-efficient design. The next section (Section 1.3) reviews the CMOS circuit topology and sources of power dissipation. Some low power circuit techniques are also introduced. The power management techniques available at gate and module level are extensively presented in Section 1.4. Section 1.5 concludes this background overview of low power optimizations available at the system level. A summary of all the encountered solutions is finally given in Section 4.6.1, with a review of the last ten years ULP SoCs performances.

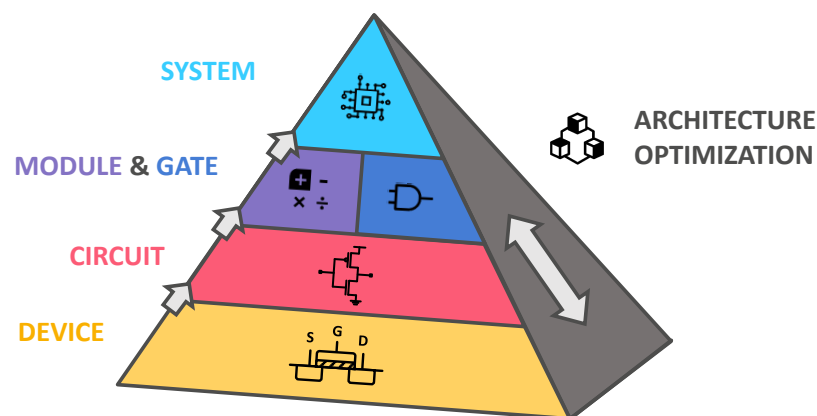


Figure 1.1: Integrated Circuit implementation stack: from silicon technology to the system level.

1.1 Brief History of Voltage Scaling

Since the first commercial release of the Intel 4004 microprocessor in 1971 [48], the number of transistor on a chip had been roughly doubling every two years. This exponential growth known as the Moore's law [9], is translated by the reduction of transistor sizes by approximately $\sqrt{2}$ or 30% every technology nodes. Accordingly, length scaling helps to reduce transistors delay, power consumption and manufacturing costs¹.

However, proper scaling of transistors also requires adjusting of all other geometric dimensions and physical parameters in order to guarantee a correct operation and increase the performances. The geometrical and physical parameters scaling is reported in Table 1.1. Other relevant geometric parameters are also given for two type of scaling: Constant Voltage (CV) scaling and Constant Electric Field (CEF) scaling [51, 52].

	Parameter	Symbol	CV-scaling <i>Constant Voltage</i>	CEF-scaling <i>Constant Electric Field</i>
Geom.	Gate length	L	$1/\alpha$	$1/\alpha$
	Gate width	W	$1/\alpha$	$1/\alpha$
	Oxide thickness	t_{ox}	$1/\alpha$	$1/\alpha$
Physical	Doping concentration	N_a, N_d	α^2	α^2
	Electric field	\mathcal{E}	α	1
	Transition frequency	f_r	α^2	α
	Voltage	V	1	$1/\alpha$
	Current	I	α	$1/\alpha$
	Power	P	α	$1/\alpha^2$
	Area	\mathcal{A}	$1/\alpha^2$	$1/\alpha^2$

Table 1.1: Effect of device scaling on some of the transistor intrinsic parameters.

With CV scaling, supply voltage remains constant while the dimensions of the device are shrunk. Consequently, the delay scales leading in return to an increase of the maximum operating frequency yet the power density remains constant. The electric field also increases with scaling in the device resulting in a device breakdown tendency. This solution has been the preferred scaling method in older circuit technologies since it provides voltage compatibility over different technology nodes and ensures continuity in Input/Output (IO) voltages. Historically, it has been employed until the beginning of the 1990s, with a fixed supply voltage of 5 V [53]. However, beyond sub micrometer nodes, the benefit on delay scaling are mitigated due to the velocity saturation [10].

Suggested in 1974 [54], CEF scaling or Dennard's scaling assume a constant electric field over technology nodes. Both voltage, geometric dimensions and intrinsic characteristics like the doping concentration of the device are reduced proportionally². CEF results in the largest reduction in power, avoids oxide breakdowns while guaranteeing the physical stability of the Metal-Oxide-Semiconductor Field Effect Transistor (MOSFET).

¹ Moore's law is above all an economic law which results from the rush to push down production costs by reducing the price per transistor [49]. Widely adopted by manufacturing companies it ensures increasing of the transistor density by two for a given area [50].

² To preserve the electrostatic integrity of the transistor, $L_g \searrow$ goes with $W \searrow$, $t_{ox} \searrow$, $N_a \nearrow$, $N_d \nearrow$ and $V_{dd} \searrow$.

As seen in Figure 1.2, Dennard's scaling has been adopted at the beginning of the 1990s. Operating supply has started to decrease following the node gate length. This solution was then considered in the International Technology Roadmap for Semiconductors (ITRS) [55].

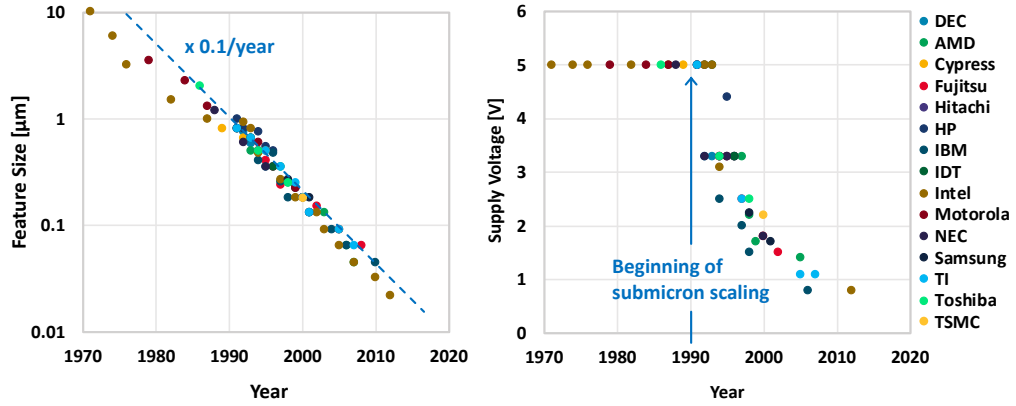


Figure 1.2: Scaling of technology feature size and nominal supply voltage over time [53].

However non-idealities (see Section 1.2) in advanced technology nodes lead to slow down the transistor voltage sizing trends and consequently slowing down the gain in power efficiency over time. A generalized scaling has later been introduced which tolerates a small increase of the electric field to compensate for the slower scaling of the supply voltage [56]. For the last decade technology nodes the supply scaling is much lower than feature shrinking.

Nowadays, the International Roadmap for Devices and Systems (IRDS) has replaced the ITRS which stopped in 2016 [57]. The previous road map had become irrelevant because the race for smaller devices is not sufficient to ensure a continuity of the semiconductor industry growth. By extending its identifications of new devices key trends to systems and architectures, the IRDS is expecting to go beyond the Moore law and proposes further developments in traditional ICs (see Table 1.2).

Year of production	2017	2019	2021	2024	2027	2030	2033
“Node range” [nm]	10	7	5	3	2.1	1.5	1.0
Logic Device structure options	FinFET FD-SOI	FinFET LGAA	FinFET LGAA	LGAA VGAA	LGAA VGAA	LGAA VGAA 3D-VLSI	LGAA VGAA 3D-VLSI
Mainstream device	FinFET	FinFET	LGAA	LGAA	LGAA	VGAA	VGAA
Supply Voltage [V]	0.75	0.70	0.65	0.65	0.65	0.60	0.55

Acronyms used in the table (in order of appearance):

Fin Field Effect Transistor (FinFET), Fully Depleted Silicon On Insulator (FD-SOI), Lateral Gate-All-Around-Device (LGAA), Vertical Gate-All-Around-Device (VGAA), Fine-pitch 3D logic sequential integration (3D-VLSI).

Table 1.2: Logic Core Device Technology Roadmap based on IRDS projections [57].

1.2 Technology for Energy-efficient Design

With the advance of digital applications in modern electronic circuits, MOSFETs have gained a strong foothold – compared to other type of transistors – thanks to their low leakage current and their availability in a Complementary Metal Oxide Semiconductor technology [58, 59]. This standard process easily allows the integration of inverters through one n-doped MOSFET (nMOS) and one complementary p-doped MOSFET (pMOS).

As shown in the previous section, to increase the operating speed of the transistor, it is necessary to reduce its feature sizes. On the one hand, this idea implies the evolution of fabrication techniques and methods, allowing small device availability, but resulting to more complex and expensive fabrication costs. On the other hand, it also leads to Short Channel Effects (SCE); the device can no longer be seen as an ideal ON/OFF switch.

1.2.1 The MOSFET Transistor in CMOS Technology

MOSFET stands for Metal-Oxide-Semiconductor Field Effect Transistor (FET) and refers to a lightly doped Silicon substrate with two high doping implantation areas named source and drain and associated to a top contact defined on a thin oxide, known as the gate. As shown in Figure 1.3, a n-channel FET or nMOS (left) consists of a p doped substrate with two highly doped n+ regions. For a p-channel FET or pMOS (right), the dopings are reversed. The n-type doping requires to add negatively charged electrons using donors atoms. The p-type doping needs a hole excess (i.e., electron deficit) which is done with acceptor atoms. For Silicon ($_{14}\text{Si}$), Phosphorus ($_{15}\text{P}$) and Boron ($_{5}\text{B}$) atoms can be used respectively for n and p type doping.

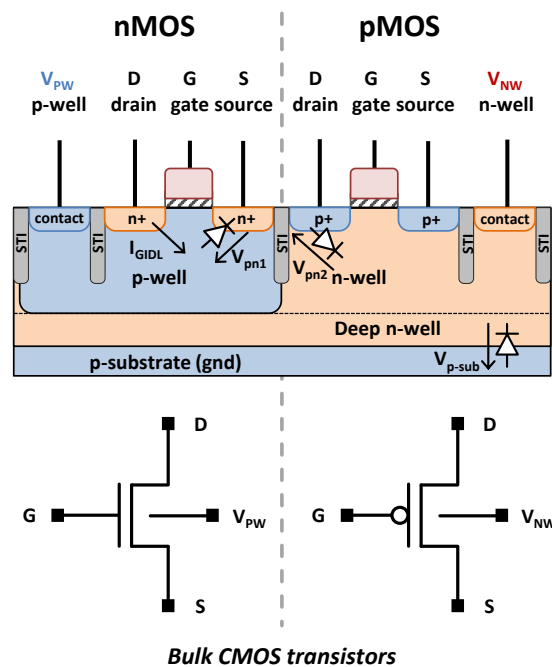


Figure 1.3: pMOS and nMOS transistors cross-sections in standard CMOS bulk process.

Nominal Operation

When a voltage is applied on the gate, the resulting field effect induces carriers in the channel. A current can consequently flow between source and drain. A fourth contact to the bulk substrate is mandatory to all MOSFETs to operate. For an nMOS, it is generally connected to the ground and tied to the source to set a reverse-biased drain-bulk junction. This solution ensures minimum leakage current through the substrate. The same scheme is applied to the pMOS with the source connected to the supply voltage.

At nominal voltage, the dynamic current of a MOSFET in active operation (i.e., saturation mode) is given by [51]:

$$I_{\text{saturation}} = \frac{W}{L} \cdot \frac{\mu_{\text{eff}} \cdot C_{\text{ox}}}{2} \cdot (V_{\text{GS}} - V_{\text{TH}})^2 \cdot (1 + \lambda V_{\text{DS}}) \quad (1.1)$$

with; W and L the width and length of the device, μ_{eff} the effective mobility, C_{ox} the oxide capacitance of the grid, V_{TH} the threshold voltage, V_{DS} the source drain voltage, and λ a channel length modulation factor.

Leakage Currents

In contrast to the dynamic current I_{ON} , a leakage current I_{OFF} is defined as any “non-useful” currents that can occur in a transistor when it is in a blocked or in passing state [60]. In Figure 1.4 are reported the six main leakage current contributions, each resulting from different mechanisms.

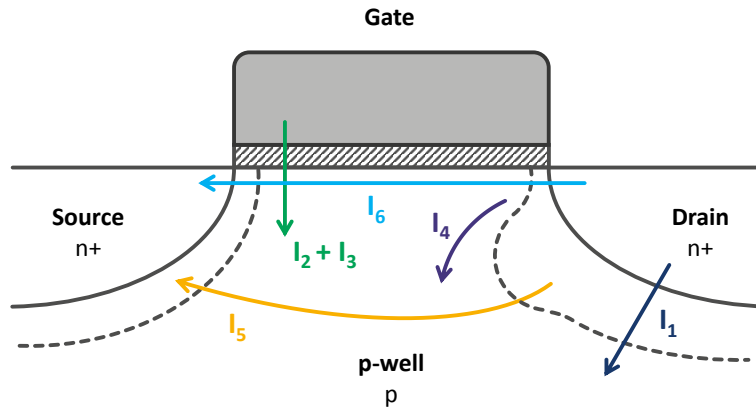


Figure 1.4: Summary of leakage current mechanisms in a deep-submicrometer transistors.

First, the typically reverse biased PN-junctions between the drain/source and the wells create a junction leakage current I_1 . Then, a gate current I_G results from two contributions: a gate oxide current I_2 , due to electron tunneling from the channel to the gate, and a hot-carrier injection current I_3 due to highly energy-charged electrons crossing the interface potential barrier and entering into the oxide layer. Next, a Gate Induced Drain Leakage (GIDL) I_4 , results from the coupling between the high field effect in the drain junction and the electric field of

the gate. A punchthrough current I_5 might also appear in short-channel devices when the depletion regions of source and drain merge. Lastly, the most dominating contribution is the sub-threshold leakage current I_6 . It corresponds to a sub-threshold (or weak-inversion) state of the transistor where the transistor carriers are moving from drain to source thanks to diffusion instead of drift. All these contributions and the mechanisms associated are extensively explained in [60].

Low Voltage Operation

The on-current I_{ON} depends on the operating region of the device as shown in Figure 1.5. Contrary to the nominal operation (above-threshold), when the supply voltage is decreased the devices enter in the near-threshold or sub-threshold regions. For an nMOS transistor, they are generally given for $V_{GS} - V_{TH} \in [-50\text{mV}; 200\text{mV}]$ and $V_{GS} \leq V_{TH}$ [61].

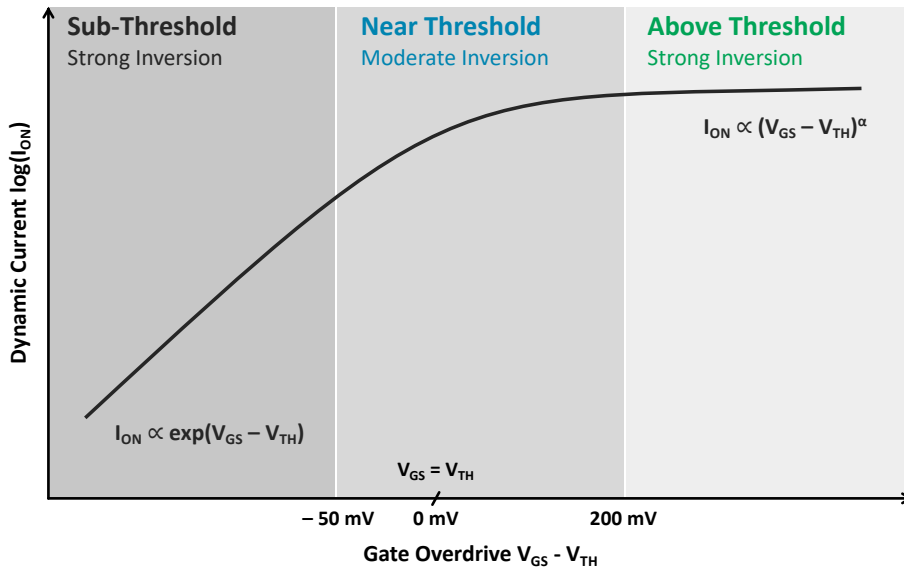


Figure 1.5: Qualitative trend showing the transistor dynamic current I_{ON} evolution according to the gate overdrive $V_{GS} - V_{TH}$. In modern technologies, $\alpha \in [1.1, 1.5]$ due to carrier velocity saturation [62].

In this region, the MOSFET operates in weak or moderate inversion and the previously quasi-linear expression of the current becomes exponential as defined in [63]:

$$I_{\text{sub-TH}} = \frac{W}{L} \cdot \mu_{\text{eff}} \cdot C_{\text{ox}} \cdot (m-1) \cdot v_T^2 \cdot e^{\left(\frac{V_{GS}-V_{TH}}{m \cdot v_T}\right)} \cdot \left(1 - e^{-\frac{V_{DS}}{v_T}}\right) \quad (1.2)$$

where; L is the effective length of the gate, m is the slope coefficient of the current below sub-threshold, and v_T the thermal voltage kT/q . At low voltage this current becomes the dynamic current I_{ON} ruling the operations of the transistor. Conversely, the leakage current I_{OFF} is defined as the current flowing through a transistor when it is blocked.

For simulation purposes, mathematical models are developed to reflect the behavior of the transistor. In particular, the EKV model has proven to be very accurate when the MOSFET is operating the sub-threshold region [64]. Moreover, it takes into consideration many of specialized effect that occur in sub-micrometer technologies.

Process Variations

Due to the exponential dependencies of the sub-threshold current, devices operating at low voltage are highly sensitive to voltage and temperature variations. Although it does not appear directly in (1.2), process variability has also a critical impact on the devices' operations [63]. In fact, manufacturing introduces fluctuations resulting in transistor critical dimensions inaccuracies, substrate doping variations or roughness irregularities between interfaces [65]. In return, these effects significantly change the threshold voltage V_{TH} of the MOSFETs. Such variations are classified in two types.

First, intra-die variations (intrinsic or local variability) regroup variations on device parameters between transistors on the same die. Because of their random nature, they are characterized using *normal* or *Gaussian* and estimated using Monte-Carlo simulations [66].

Second, inter-die variations (extrinsic or global variability) imply systematic fluctuations on device parameters from one die to another. Technology manufacturers quantify these deviations by lumping the collective effects of all variations into their effect on the nMOS and pMOS transistors in three categories: fast, typical, slow. Thereupon, process corners are defined as a combination of these effects (see Figure 1.6). From the nominal corner Typical-Typical (TT), the Fast-Fast (FF) and Slow-Slow (SS) exhibit the largest deviation, thus deeply impacting the speed and strength of the devices while the crossing corners Fast-Slow (FS) and Slow-Fast (SF) mainly effect the circuit speed [67].

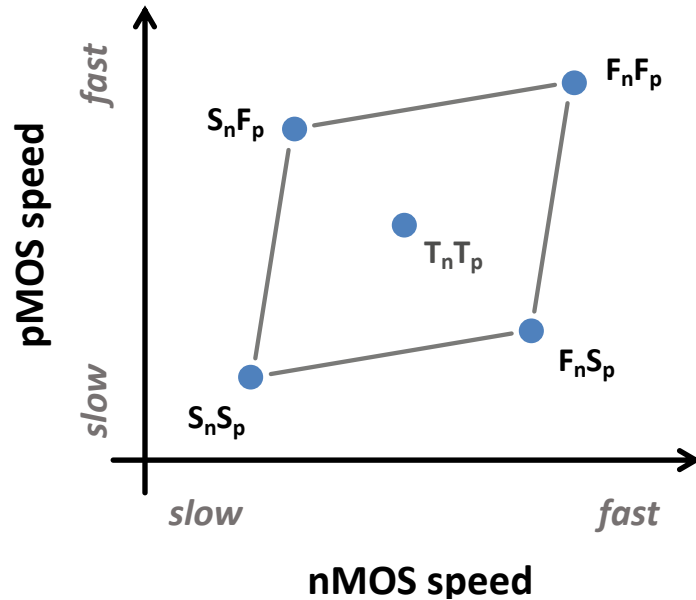


Figure 1.6: Visualization of the different process corners. The first abbreviation defines the nMOS and the second the pMOS. The Typical-Typical (TT) is the nominal case, while Fast-Fast (FF), Fast-Slow (FS), Slow-Fast (SF), Slow-Slow (SS) characterize extreme cases. The process box is generally not square as characteristics are shared by both nMOS and pMOS devices.

1.2.2 Advanced Technologies for Nanometers Designs

While the size of the transistor is reduced in nanometer technology nodes traditional bulk silicon technologies suffer from characteristic degradation due to Short Channel Effects that undermine efficient operation [72]. Due to a weaker electrostatic control of the channel by the transistor gate, the current leakage of the devices increases, reducing the I_{ON}/I_{OFF} ratio. This limitation requires the utilization of technologies that provide better electrostatic control of transistors in order to preserve and amplify the efficiency for their low voltage operations.

Therefore, energy-efficient circuits are also achieved by choosing an appropriate fabrication process. The technologies available in mainstream products to overcome the bulk defects³ are the Fin Field Effect Transistor (FinFET) and the Fully Depleted Silicon On Insulator (FD-SOI) processes (see Figure 1.8). Alternatives such as the Silicon On Nothing (SON), Double Gate (DG) and Partially Depleted Silicon On Insulator (PD-SOI) were also explored yet did not overtake the competition due to a reduced gain in energy efficiency or manufacturability/costs.

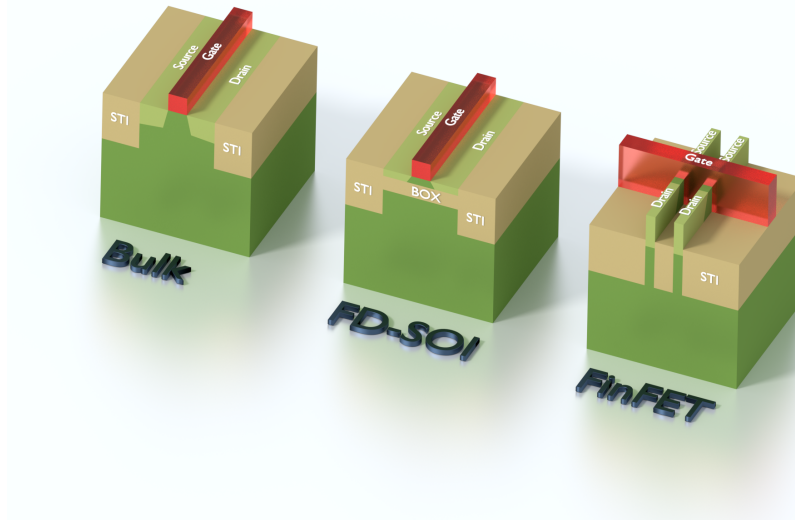


Figure 1.8: Schematic views of three types of transistor: planar Bulk, UTBB FD-SOI and FinFET. Courtesy of V. Malherbe.

Fin Field Effect Transistors FinFET are using a vertical 3D structure where the gates are discretized and wrapped around the channel. This structure improves the electrostatic control resulting in a better energy efficiency [73]. This is currently the main commercial solution for nodes below 20 nm and it has been widely adopted by manufacturers such as Intel and TSMC. However, the increased complexity results in higher manufacturing costs, mostly due to the higher number of required processing masks [74]. An Extension of the Fin Field Effect Transistor (FinFET) technology is currently being developed for nodes below 7 nm named Gate-All-Around-Device (GAA) [75]. In this case, the channel is completely enclosed inside the gate. Moreover, due to reduced channel and bulk contact area, these devices are less sensitive to biasing techniques [76].

Fully Depleted Silicon On Insulator (FD-SOI) is a technology based on an extremely thin

³Up to 20 nm nodes, SCE can be reliably compensated using layer and implantation optimizations.

silicon channel associated with a buried oxide (Ultra Thin Body and BOx (UTBB)). The use of this silicon film between source and drain ensures effective current control for a given voltage by reducing parasitic losses mostly due to charge sharing and Drain Induced Barrier Lowering (DIBL) [77]. The absence of carriers in this volume (fully depleted technology) also considerably reduces low voltage variability, which is mainly due in bulk technologies to the fluctuation of the doping particles in the useful volume [78]. The isolation of the channel from the bulk silicon leads to an increased range of body biasing options, offering an increased effective grid length at no additional surface cost (see next section). This technology – as of December 13, 2019 – is commercially available in 28 nm and 22 nm, with a future 12 nm node announced [79].

1.2.3 FD-SOI for Minimized Energy Consumption

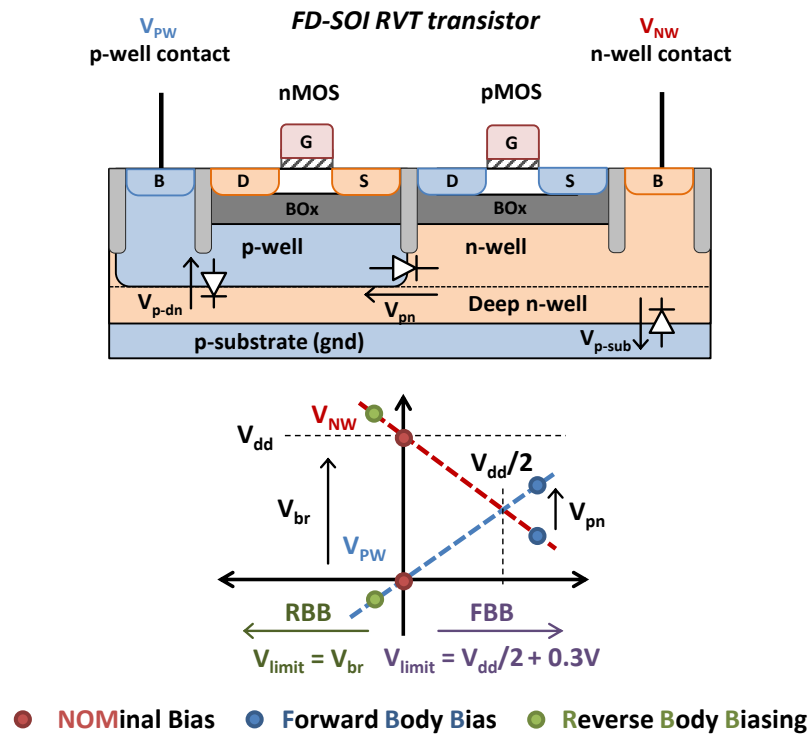
In addition to the aforementioned intrinsic benefits, the 28 nm FD-SOI technology – proposed by STMicroelectronics (STM) – offers small changes in design methodologies. Moreover, it presents the advantage of a back-end compatibility with the bulk technologies of a given node.

As seen in Figure 1.9, FD-SOI also offers a wider range of design solutions through the possibility of using differentiated well architectures. Standard n-well and p-well structures are available for Regular Voltage Threshold (RVT) transistors whereas a “flip-well” structure is used for Low Voltage Threshold (LVT) transistors. This time, the body biasing limitations are restricted to the PN-diode between the p and n wells, located under the Buried Oxide (BOx). Hence, an extended back-biasing polarization range allows the performance of transistors to be modulated with greater amplitude in a static and/or dynamic manner.

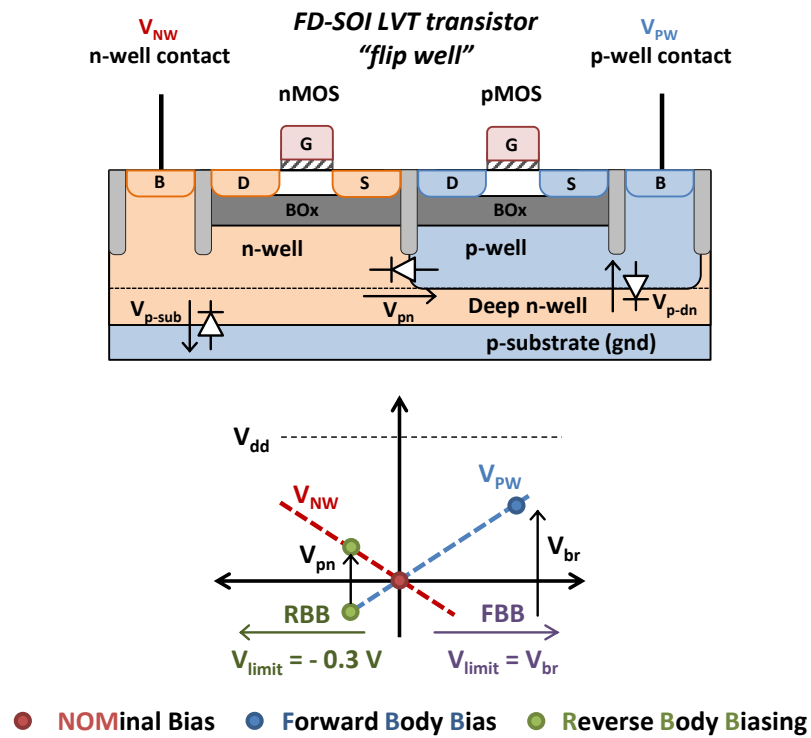
In 28 nm FD-SOI, the maximum RVT – FBB is around $V_{dd}/2 + 300$ mV while the LVT – RBB is around -300 mV. RVT – RBB and LVT – FBB are both limited by the oxide layer and/or diode breakdown (to about $V_{br} \pm 3$ V) as well as the transistor aging [80]. In practice, beyond 1.5 V, the RVT – RBB and LVT – FBB body biasing schemes become inefficient. The leakage resulting from the FBB overcomes the speedup benefit and the RBB reaches the MOSFET intrinsic leakage floor which is independent of the effective device voltage threshold [81]. These body biasing effects are also subsequently stronger than in BULK. Considering an nMOS device, a threshold voltage shift of 85 mV per 1 V of FBB appears in FD-SOI while 25 mV/V exists in BULK [82].

Global Foundry has proposed a 22 nm FD-SOI implementation based on the STMicroelectronics 14 nm transistor but using the 28 nm back-end. In this case, the RVT nominal biasing conditions are the same than LVT devices, i.e., $V_{PW} = V_{NW} = 0$ V. IBM also proposed a double BOx technology where the p and n transistors are completely isolated from the neighboring opposed wells and the substrate. It extends the forward and reverse body biasing capabilities up to ± 2 V [83].

At this point, we already perceive that leakage and dynamic power could be traded-off. At a fixed supply voltage V_{dd} , the operating frequency F_{op} can be increased at an increased leakage cost whereas at a fixed F_{op} , V_{dd} can be reduced which directly affects the total power dissipated in the system (see Section 1.3).



(a) Regular Voltage Threshold (RVT) transistor



(b) Low Voltage Threshold (LVT) transistor.

Figure 1.9: 28 nm UTBB FD-SOI CMOS transistors cross section and biasing capabilities.

1.3 Energy-efficient Circuits

Using the MOSFET as the basic building block, this section explores the CMOS circuit topology and its sources of power dissipation. Due to this wide availability in standard technology and its integration in industrial flows, this topology is preferentially used in this work. Some low power circuit techniques for power mitigations are also introduced.

1.3.1 Circuit Topologies

Standard CMOS Logic

The standard CMOS logic⁴ involves the opposition between two complimentary networks composed of passing and blocked transistors. Using pMOS transistors, a Pull-Up Network (PUN) connects the outputs to the supply voltage when the input is such that a logic '1' is required. Similarly, a Pull-Down Network (PDN) that uses nMOS transistors connects the output to the ground node to realize a logic '0'. The input signals are connected to the gates of the transistors in such a way that for any input patterns only one network is passing (ON), whereas the other one is blocked (OFF). The CMOS logic is inherently inverting, thus the simple cell that can be made with only one stage of PUN and PDN is the inverter.

An important characteristic for CMOS logic is the time required for a given cell to transmit an information from its inputs to its outputs. This is the propagation delay t_p that is measured at 50% of a signal transition between inputs and outputs. It is defined as the average of the low to high transition time t_{LH} and high to low time t_{HL} (see Figure 1.10). The rise and fall time, respectively t_r and t_f are also useful metrics of the slope of a waveform.

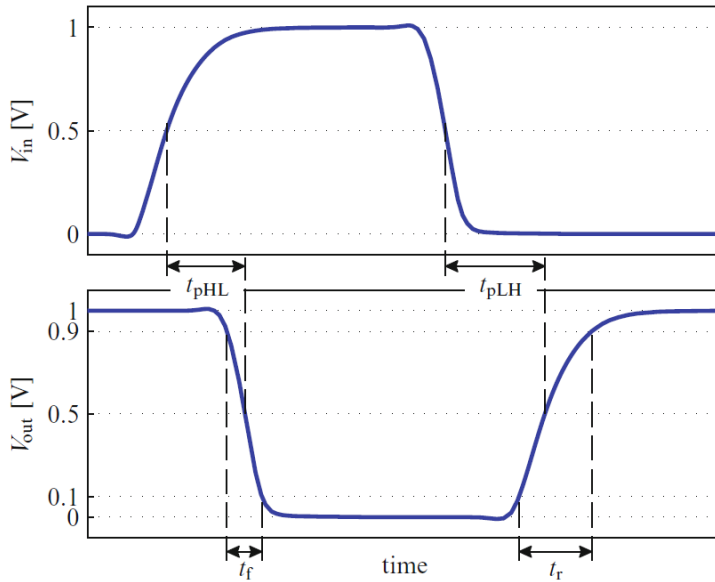


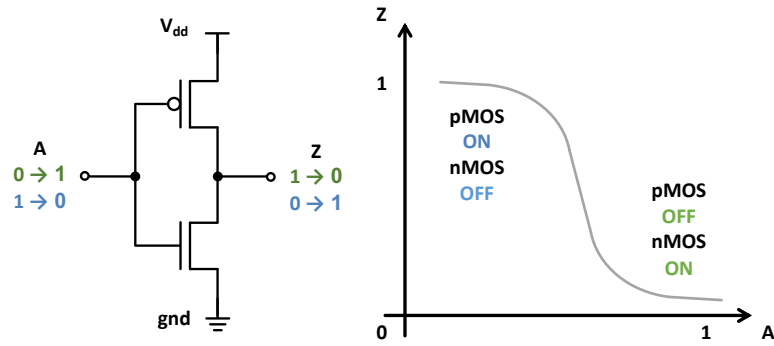
Figure 1.10: Transient characteristics of an inverter [66].

⁴The Complementary Metal Oxide Semiconductor (CMOS) logic replaced the former nMOS logic which was area and power hungry due to the utilization of wide resistors [68].

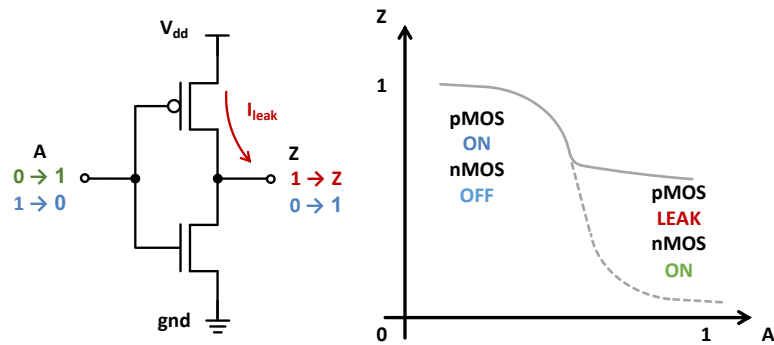
At low voltage, the transition time of a CMOS stage is exponentially dependent of the supply voltage (see (1.13)). [84] reports performance penalties in the sub-threshold region from 2 to 4-5 orders of magnitude, for a given microarchitecture⁵. Hence, an important speed penalty is observed at low voltage on CMOS logic, resulting in low power operation at the cost of high-performance degradations.

A resulting issue of CMOS logic operating at low voltage is the increased imbalance between the nMOS and pMOS strength and thus between PDN and PUN [86, 87]. To ensure adequate noise margin and reasonable symmetric rise and fall transitions, the strength of these networks must be comparable [52]. Hold violations also appear due to clock skews resulting from the increased local variability in the clock tree [88].

For correct operation of a logic gate, the dynamic current I_{ON} must be higher than the leakage current I_{OFF} (see Figure 1.11a). At low voltages, if the blocked transistor array has a leakage current equal to, or greater than, the dynamic current of the passing network, it is short-circuited and the output information is incorrect (see Figure 1.11b). This undesired behavior is particularly re-enforced at low voltages due to the increased leakage current as previously explained in Section 1.2.1.



(a) Nominal operation of an inverter.



(b) Impact of leakage current on the inverter operation.

Figure 1.11: Impact of the I_{ON}/I_{OFF} on the behavior of a standard inverter.

Various solutions are proposed to recover the effect of low voltage on CMOS logic. Such solutions include transistor sizing and stacking [86, 89], body biasing [87], threshold voltage selection [90], supply voltage tuning and fine-grained boosting [14].

⁵The logic depth which often increases for energy-efficient design also impacts the total delay of a datapath[85].

Other Topologies

To be fair with fellow researchers in ULP circuit design and understand the State of the Art comparison table of Section 1.6, it is worth mentioning other CMOS topologies that have been explored for ULV operations⁶. The Pseudo-nMOS logic replaces the resistance of the nMOS by gate-grounded pMOS [95]. This solution offers a reduction of the load capacitance and thus improves speed and area performance for a given supply voltage. However, this topology shows an increased power consumption combined with an high sensitivity to variations [96].

Pass Transistor Logic allows driving of the gate along with source/drain transistor terminals. Since, it is inherently unsuitable for ULV operation due to voltage drop, the Transmission Gate topology has been derived for low voltage operations [66]. However, its industry adoption is limited by the effort required to re-design and characterize standard cells with this exotic logic [97].

Contrary to static logic where the outputs are connected to the supply rails, the Dynamic logic also known as Domino uses intermediary nodes [96]. Transistors controlled by a clock preload and evaluate signal values on the capacitance of high-impedance circuit nodes. Increased speed and improved efficiency are achieved, yet the power consumption is still deeply related to the activity of the circuit.

1.3.2 Sources of Power Dissipation in CMOS Circuits

The sources of power consumption in digital CMOS circuits can be subdivided between a dynamic and a static power consumption (see (1.3)). The dynamic power P_{dyn} consists of the power consumed when the device is active. The static power P_{stat} covers the power consumed when the device is powered up but no signals are changing value [98].

$$P_{\text{tot}} = P_{\text{dyn}} + P_{\text{stat}} \quad (1.3)$$

Dynamic Power

The dynamic power is consumed when the system is actively switching, i.e., when signals are changing values. It is often split between switching P_{switch} and short-circuit P_{short} contributions, as follows;

$$P_{\text{dyn}} = P_{\text{switch}} + P_{\text{short}} \quad (1.4)$$

On the one hand, the switching power P_{switch} , also known as capacitive power, results from the charging and discharging of load capacitance as logic gates are switching. Figure 1.12 illustrates the contributing currents which result in the switching power for a basic CMOS inverter. Then, the energy required to load an output capacitance C_L at a fixed V_{dd} is given by;

$$E_{\text{switch}} = C_L \cdot V_{\text{dd}}^2 \quad (1.5)$$

⁶For the curiosity of the reader, other topologies are also available, such as: the Pass Transistor Logic [91], Sub-threshold Source-Coupled Logic (STSCl) [92], adiabatic logic [93] and the Mos Current Mode Logic (MCML) [94].

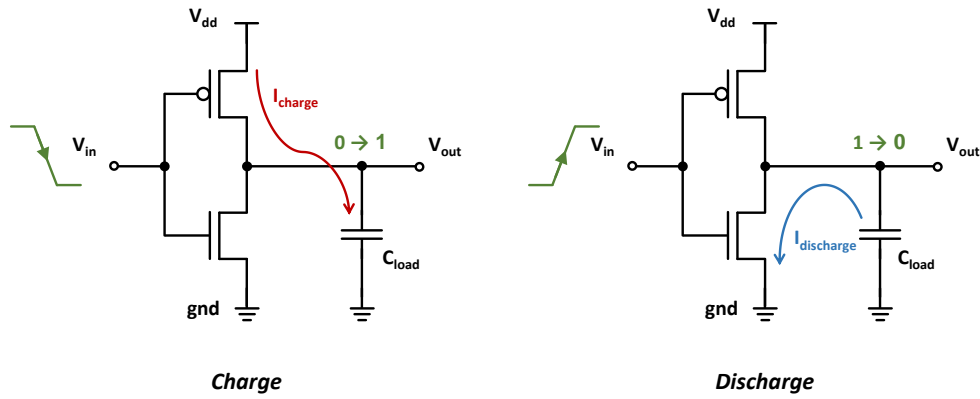


Figure 1.12: Charge and discharge currents during the switching operation of a CMOS inverter.

Consequently, if the working clock frequency of the system is f_{clk} and considering an activity factor ⁷ α and C_{tot} is the total capacitance of the system, the previous expression can be rewritten as;

$$P_{\text{switch}} = \alpha \cdot C_{\text{tot}} \cdot V_{\text{dd}}^2 \cdot f_{\text{clk}} \quad (1.6)$$

On the other hand, P_{short} is the dissipation power due to the short-circuit currents flowing when a direct current path from supply to ground is created during switching of the devices. This short-circuit is caused by the finite slope of the input signal during switching. For a very short period of time, it leads the pMOS and nMOS to be turned ON simultaneously during a logic transition. These currents are generally small and highly sensitive to the supply voltage and almost negligible in advanced technology node. Moreover, at ULV, as $V_{\text{dd}} \leq V_{\text{THn}} + |V_{\text{THp}}|$, the p and n devices are never fully ON at the same time, which eliminates the short-circuit dissipation [99]. Therefore, it is often considered that the overall dynamic power is dominated by the switching power only.

$$P_{\text{dyn}} \approx P_{\text{switch}} \quad (1.7)$$

Static Power

The static power consumption results from the current when the system is inactive. When the transistor is OFF, we have seen that a leakage current I_{leak} still exists in the device, which directly results in a static power dissipation. Thus, we have:

$$P_{\text{stat}} = I_{\text{leak}} \cdot V_{\text{dd}} \quad (1.8)$$

In advanced nanometer CMOS technologies, the smaller feature size of the transistor induces an absolute increase on I_{leak} (which is also function of V_{dd} as seen in Section 1.2.1). Therefore, P_{stat} is becoming increasingly important in advanced technologies' designs.

⁷Since all gates are not always switching, the activity factor evaluates the average switching activity of the system. It ranges between 0 when no gates are switching, to 1 when all gates are changing state at every clock cycle.

Total Power

Finally, the total power consumption of a digital system is given by:

$$P_{\text{tot}} = \alpha \cdot C_{\text{tot}} \cdot V_{\text{dd}}^2 \cdot f_{\text{clk}} + I_{\text{leak}} \cdot V_{\text{dd}} \quad (1.9)$$

1.3.3 Minimum Energy Point

When we are referring to devices with a limited energy source or storage, which are mostly battery-operated devices, the power consumption is not always the most interesting metric. Indeed, by reducing the frequency and taking an infinite amount of time to finish an operation the power consumption could be reduced to almost leakage only. Hence, in order to evaluate the consumption of a system during its active modes, the total energy E_{tot} , or consumption for a service provided is preferred. It is obtained by multiplying the total power by the amount of time to perform an operation or a clock cycle (i.e., the clock period t_{clk}).

$$E_{\text{tot}} = P_{\text{tot}} \cdot t_{\text{clk}} \quad (1.10)$$

Using (1.9), it results that the total energy consumption for a digital system has also a dynamic and a leakage component, as follows;

$$E_{\text{tot}} = \underbrace{\alpha \cdot C_{\text{tot}} \cdot V_{\text{dd}}^2}_{E_{\text{dyn}}} + \underbrace{I_{\text{leak}} \cdot t_{\text{clk}} \cdot V_{\text{dd}}}_{E_{\text{stat}}} \quad (1.11)$$

On the one hand, the dynamic energy follows a quadratic trend on V_{dd} ($E_{\text{dyn}} \propto V_{\text{dd}}^2$) then, it decreases when the supply voltage is reduced. On the other hand, assuming an ideal device⁸ with a leakage current independent of V_{dd} , the static energy E_{stat} is dependent on the supply voltage and the clock period t_{clk} . However, this last part is also related to the supply voltage by:

$$t_{\text{clk}} = \frac{C \cdot V_{\text{dd}}}{I_{\text{ON}}} \quad (1.12)$$

In Section 1.2.1, it has been shown that the dependence on the current I_{ON} is determined by the transistor operating region. At low voltage an exponential relationship is observed while it quadratically evolves at nominal voltage.

$$\begin{aligned} t_{\text{clk}} \propto \frac{V_{\text{dd}}}{e^{V_{\text{dd}}}} &\Rightarrow E_{\text{stat}} \propto \frac{V_{\text{dd}}^2}{e^{V_{\text{dd}}}} && \text{at low voltage} \\ t_{\text{clk}} \propto \frac{1}{V_{\text{dd}}} &\Rightarrow E_{\text{stat}} \propto 1 && \text{at nominal voltage} \end{aligned} \quad (1.13)$$

Then, following the dependencies in (1.13), we can conclude that the static energy increases exponentially at low voltage, while it is constant at higher supply.

Figure 1.13 qualitatively plots the various energy contribution of E_{tot} according to the supply voltage. The evolution of the static and dynamic energy are counteracting and so creating a local Minimum Energy Point (MEP) obtained for an optimal supply point $V_{\text{dd,MEP}}$. The energy reduction obtained by voltage scaling is thus limited by the leakage contribution that

⁸SCE effects such as DIBL, GIDL or channel-length modulation are not considered here.

appears at low voltage.

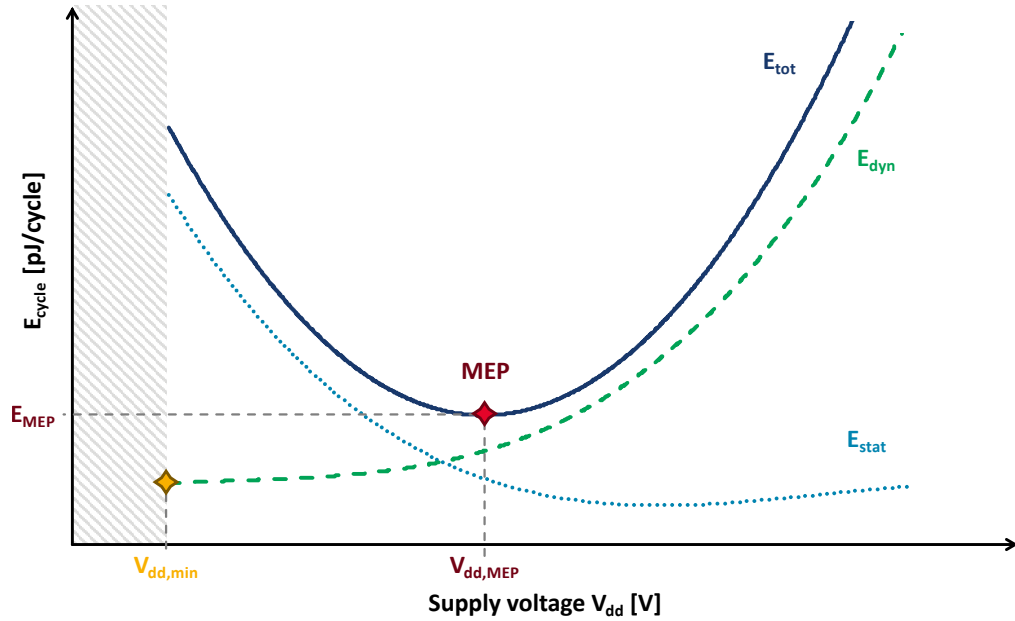


Figure 1.13: Qualitative trend describing the static, dynamic and total energy according to the supply voltage.

The Minimum Energy Point (MEP) tends to exist in sub or near-threshold operation region of the transistor [63]. However, as its exact position depends on the relative importance of dynamic versus static energy, the MEP is related to various design choices resulting from the specifications or the technology node selected. For instance, considering a certain design, the utilization of smaller CMOS technology nodes tends to increase the static energy ratio over dynamic. Consequently, even though the energy value of the MEP tends to decrease, the voltage of this point shifts towards higher voltages as shown in [100].

The activity of a circuit also affects the MEP position. Indeed, a small activity factor leads to the reduction of the dynamic energy with no consequence on the static energy. It results in the increase of the relative position of $V_{dd,MEP}$. On the contrary, with a high activity factor, the dynamic energy augments. This time the optimal point is shifted to lower voltages.

Proper examples of two systems with different activity are on the one side, Digital Signal Processors (DSPs) which are mainly built on datapaths and, on the other side, generic processors which are using memories paths. Datapaths are good examples of systems dominated by dynamic energy, since all cells are switching simultaneously. On the contrary, memories are dominated by static energy, as only a few cells are accessed simultaneously. [81] reports for instance a MEP at 350 mV for a DSP in 28 nm CMOS while in a equivalent technology node the CPU of [101] exhibits a MEP at 450 mV.

The activity of a system⁹ may change as a function of time so the MEP position can also move along the supply voltage. However, the optimal point is located in a flat region [66]. This observation greatly helps to relax the constraints on the operating supply voltage of the system. In fact, a minimum energy is obtained without the need for a strict $V_{dd,MEP}$ supply.

⁹The activity of a microprocessor is for instance directly dependent of the running software [102].

It also reveals that a minimum effort should be placed on a design to dynamically track the MEP. Current efforts to do so often result in an increased circuit complexity at the expense of an increased overall power consumption [103].

In summary, leakage dominated circuits present a relatively high voltage (around 0.5 V) MEP, whereas dynamic energy dominated systems MEPs occur at lower voltages. Therefore, aggressive supply voltage reduction is not a fast forward solution to improve the energy efficiency of a circuit. As a simple rule of thumb, reducing the supply voltage for optimal energy efficiency is most favorable for circuits dominated by dynamic energy consumption. However, it also affects the operating frequency of the design. Therefore, the application for ultra-low-voltage targets systems with relaxed performance constraints or energy-constrained systems.

1.3.4 Circuit Techniques for Power Reductions

Multi-Threshold CMOS Logic

This technique, widely used in analog designs, mixes transistor types at circuit level to tune a circuit and reach optimal power and performance trade-off [69]. For digital circuits, starting from an LVT design targeting high performances, the overall leakage is reduced by replacing with RVT type (or HVT) every transistor or cells not limiting the speed. On the contrary, starting from a power limited design, speed is improved by replacing some devices by LVT transistors.

In 28 nm FD-SOI, this technique is limited by the n and p well shortcuts that might appear below the BOX due to the body-biasing capabilities. To solve this, a deep n-well is required which limits the application of Multi-Threshold CMOS (MT-CMOS) at cell level. Hence, this work will prefer multi-threshold utilization at system level through partitions of blocks or modules (see Section 1.5.2).

Dynamic Threshold CMOS

Dynamic-Threshold CMOS (DT-CMOS) logic introduced in [104] relies on a connection between the substrate contact and the transistor gate (see Figure 1.14). According to the input signal, it dynamically modulates the device's threshold voltage, thus improving the transition from one state to another.

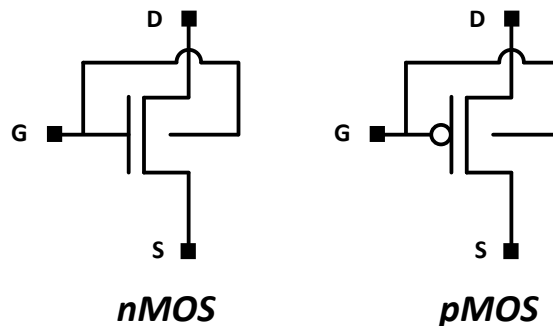


Figure 1.14: Illustration of a conventional Dynamic-Threshold CMOS (DT-CMOS) scheme.

In bulk technology, the technique is limited to very low voltage to avoid leakage due to

forwarded PN-junctions. In FD-SOI this phenomenon is contained thanks to the BOxs which isolate the devices, yet increasing the overall surface. Moreover, for digital circuits it requires standard cells specific design thus discarding its utilization. However, it is highly efficient for custom cells like power gates.

Swapped Body Biasing

At low-voltage the Swapped Body Biasing (SWBB) technique can be applied to alleviate the limits of the maximum operating frequency due to the supply voltage reduction and thus improves the energy efficiency of the circuit; At 0.5 V, [105] reports a frequency improvement up to $\times 4.4$ combined with a 75% power decreasing.

Contrary to the nominal biasing scheme, the SWBB swapped the conventional body connections of the nMOS and pMOS devices. As described in Figure 1.15, the pMOS body is connected to the ground (instead of the supply voltage) and the nMOS body connected to the main supply (instead of the ground). From this configuration, all devices receive an amount of FBB equal to V_{dd} , increasing the operating frequency. Low-voltage operation below the voltage threshold of the wells PN-junction is assumed to avoid excessive diode leakage (see Section 1.2.1).

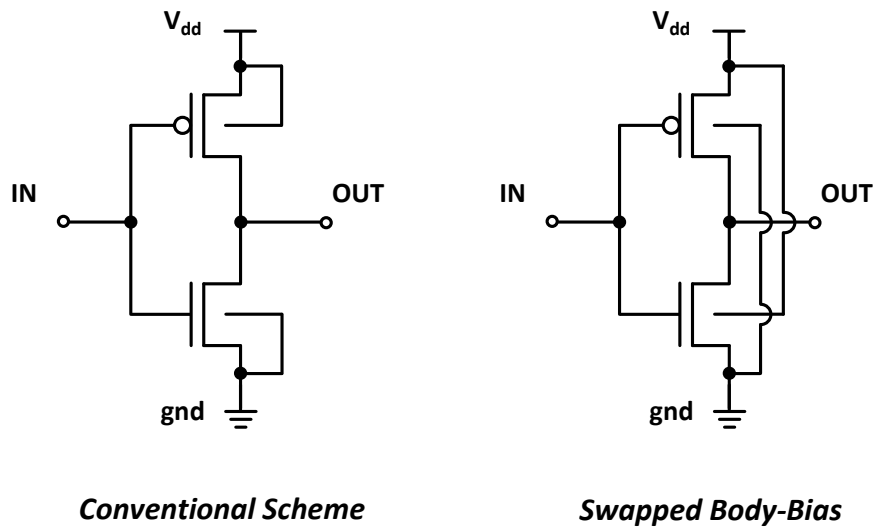


Figure 1.15: Schematic illustration of conventional biasing and Swapped Body Biasing configurations. The pMOS body is connected to the ground (instead of the supply voltage) and the nMOS body connected to the main supply (instead of the ground).

Traditional body biasing techniques require bias generators and body bias grid routing to apply a constant body-to-source voltage (independent of V_{dd}) to all devices for RBB or FBB operations. With SWBB, such generators are removed leading to a negligible area overhead compared to no body-bias designs. SWBB can also be used dynamically with body connections swapped during circuit active mode and brought back to the nominal configuration during standby. Such technique will be described at system level in Section 1.5.2

1.4 Gate-level Building Blocks and Modules for Low-Power Operations

Using the CMOS logic, standard cells and IPs are designed and used as building block of digital designs. After a short highlighting on the ULV design constraints and margins, this section lists some standard power management techniques available at gate and module level for energy reduction.

1.4.1 Design Constraints for Low-Voltage Operations

At ULV operations, the devices increased sensitivity to variations compromises the robustness of circuits and the overall yield (see Section 1.2.1). To counteract these effects, design constraints and mitigation strategies are implemented.

Due to the sub-threshold currents dependence to temperature, it is mandatory to control the evolution of the functionality and performance as a function of temperature variations. Since pMOSs and nMOSs have different size and doping, the effect of these variations may differ between devices. The ageing of components also degrades performance over the time. Aging models are thus required for long term reliability¹⁰.

To limit the process variation sensitivity, it is demonstrated that the utilization of greater logical depth is beneficial. Indeed, the random component due to local variation is averaged along the path. It is also recommended to increase the length of the device's channel. This is confirmed by Pelgrom's modelling of the impact of dimension on variability given in (1.14) where: $\sigma_{V_{TH}}$ is the standard deviation of V_{TH} , $A_{V_{TH}}$ an area proportionality constant and W, L the device size features [106].

$$\sigma_{V_{TH}} = \frac{A_{V_{TH}}}{\sqrt{W \cdot L}} \quad (1.14)$$

For purely digital designs, these Process, Voltage and Temperature (PVT) and aging constraints might impact the cell delay causing setup or hold violations. Hence, they are translated into additional design margin, as described in Figure 1.16.

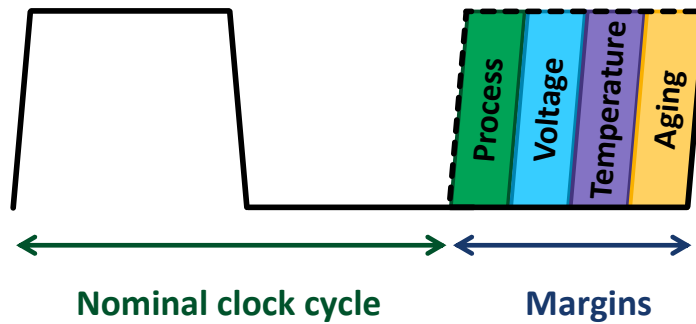


Figure 1.16: Nominal cycle period and additional margins added to compensate for Process, Voltage, Temperature and Aging variations.

¹⁰The phenomena implied in aging are exponentially dependent on the supply voltage. In this particular case, the reduction to sub-threshold voltage substantially mitigates them [14].

This conservative approach guarantees the correct functionality and performance specifications in the worst-case die and environmental conditions. In return, it impacts the power budget by forcing a whole lot of devices to work in degraded conditions [107]. Researches have been carried out on “better than worst case” ULV circuits to overcome margin induced overhead through *in situ* PVT monitoring or timing error detection [108–110].

1.4.2 Power Management Techniques

This subsection introduces techniques used in low power designs, with a particular emphasis on solutions fully compatible with standard industrial flows and EDA tools.

Clock Gating

Clock gating is a well-known low power technique used in synchronous circuits to reduce the dynamic power [111]. It works by disabling or pruning the clock signals sent to flip-flops or logic elements.

In synchronous digital circuits, the power consumed by the clock network effectively contributes to the overall dynamic power. The usually highly loaded clock signals are often distributed using a clock tree, which results in most of the switching activity of the system [112]. In suspend modes, without any precautions, the clock is still fed to logic elements, resulting in a persisting switching activity. Besides, when logic elements are maintaining the same value (the system is idling), the input clock signal will result in power dissipation.

The clock gating technique saves dynamic power by not feeding inactive modules with the clock or disabling part of the clock tree, thus reducing the overall gate activity. This technique is applied on gate-level building block, module or over a total system.

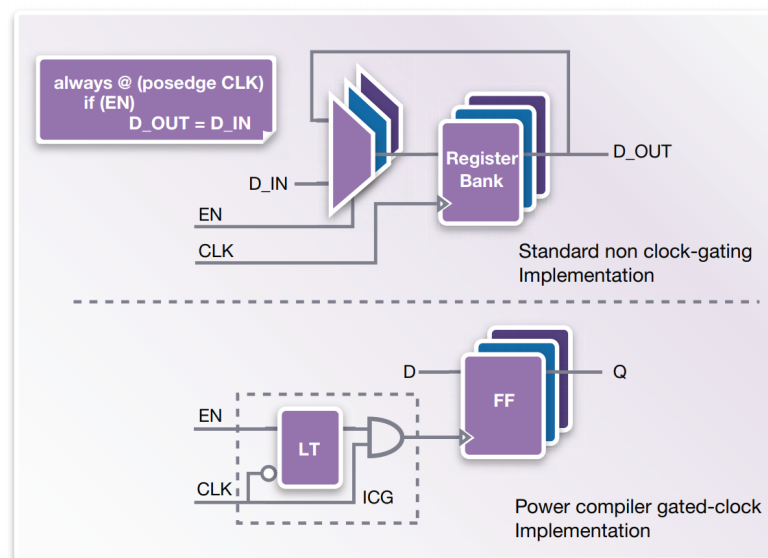


Figure 1.17: Automatic clock gating performed by Synopsys design compiler tool to reduce dynamic power consumption [113].

Currently, the insertion of clock gating is automatically performed by modern EDA tools

as described in Figure 1.17. It does not affect the functional behavior of the circuit while presenting a low hardware overhead and little performance penalties [113]. In an industrial flow with a wide standard cells offer, libraries often contain specific clock gating cells. This simple and reliable solution for dynamic power reduction is systematically applied on modern SoC designs. However, in suspend mode, clock gating still results in power dissipation due to the inherent leakage of the devices.

Power Gating

The power gating technique has been proposed to reduce both dynamic and leakage power by turning down the power supply for idling modules of a design [114]. Compared to clock gating, it is advantageous for system remaining in suspend modes for long periods of time [115].

However, power gating leads to more complex implementations which brings area penalties. The first challenge is to create a proper power grid to connect the cells power supply to switchable power rails. Power switches are distributed along the digital design or on top of it [116]. Then, sleep transistors are used to switch-off the power supply when the circuit is in a suspend mode. pMOS devices are used as “header switches” and control the power supply rail when nMOS device control the ground rail and are often called “footer switches”, as shown in Figure 1.18.

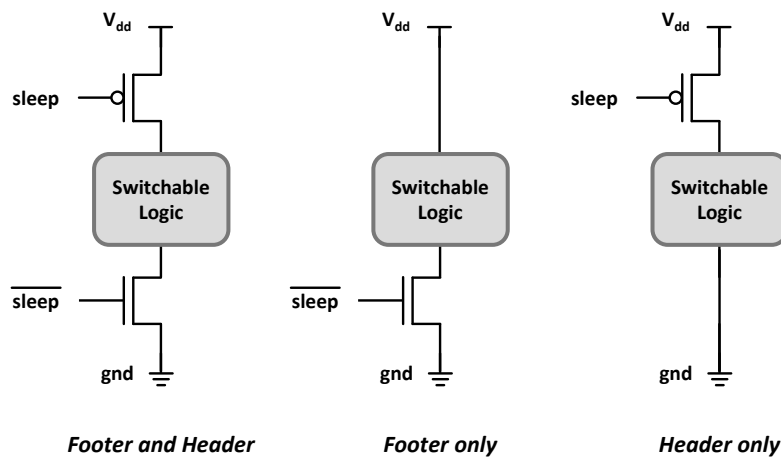


Figure 1.18: Power gating implementations.

Both headers and footers can be used yet, due to the area and performance penalties only one type of device is selected [14]. The choice between these two options is the result of a trade-off. For a given size, pMOS transistors are less leaky (small I_{OFF}) than nMOS transistors but present a lower drive current (small I_{ON}) [72]. As a result, a header switch implementation is more area consuming for a given load but shows better power saving in suspend mode. Even though small area and low leakage are obtained for footer switch based implementations, an increase sensitivity to ground noise also results from the virtual ground [116]. Power switches are sized to benefit from leakage saving while catching up the area and power penalties of the extra transistors. This requires the sleep transistors to provide a large on-current during the active mode of the system but, to deliver a reduced off-current during inactive mode. For this reason, large high threshold voltage devices are selected because they present a larger I_{ON}/I_{OFF}

ratio. However, as seen in Section 1.3.1, this ratio is degraded at low voltage. Then, techniques as in [90, 117] have been proposed to recover the I_{ON} current by boosting the gate voltage during active modes.

A common implementation of sleep transistor relies on MT-CMOS logic. LVT devices are generally used in the switchable logic where speed to sustain the target frequency is required and High Voltage Threshold (HVT) or RVT devices ensure the power gating while reducing the static power.

“Fine-grained” power gating has also been proposed, where a power switch is directly incorporated within every gate creating a virtual power supply or ground [118]. The standard PUN or PDN of the gate is connected to these virtual powers. This solution greatly reduces the problem of logic partition, transistor sizing and simplifies the integration of power gating strategies in a digital design. However, a large area overhead is added due to both the inclusion of the sleep transistor in every gate, and the creation of sleep signal distribution network. This last network also required particular attention to avoid skew between power gate.

Power gating induces system delays and thus power penalties, due to the transition time necessary for a system (or block) to switch between power gated modes. For time constrained systems (e.g., real time applications), power gating scheme might be detrimental for the correct operation of the system [119]. Power gating also affects the inter-module and system’s communication. During design time, isolation cells are added between blocks, state retention registers potentially included (see Section 1.5.2), always-on buffers placed and a power gating controller designed. These implementation requirements often restrict the power gating scheme to the more power consuming blocks.

Asynchronous Designs

Contrary to synchronous circuits, asynchronous design assumes that circuit timing events are continuous. Signal information are exchanged between modules and sub-systems with mutually negotiated times and data transfer protocols, thus removing the need for external timing regulation (i.e., a clock). For this reason, they are also called self-timed circuits [120].

Amongst other possible benefits, asynchronous circuits show: no clock skew and natural clock gating, averaged-case instead of worst performances and automatic adaptation to PVT variations, lowered power consumption since no transistors are switching (unless useful computation is needed), automatic adaptation to PVT variations [121]. Unfortunately, although highly attractive, asynchronous design suffer from a detrimental lack of industry support and design-focused commercial EDA tools. Purely asynchronous systems also require a complete overhaul of circuit design strategies [122]. For these two reasons, its utilization has been discarded for this work.

1.5 System and Architectural Design

1.5.1 Total Energy Consumption

The ULP applications targeted in this work typically need to process a low amount of data after a repetitive amount of time. Consequently, they operate following duty-cycled operations [123] which alternate between active and suspend¹¹ phases as seen in Figure 1.19.

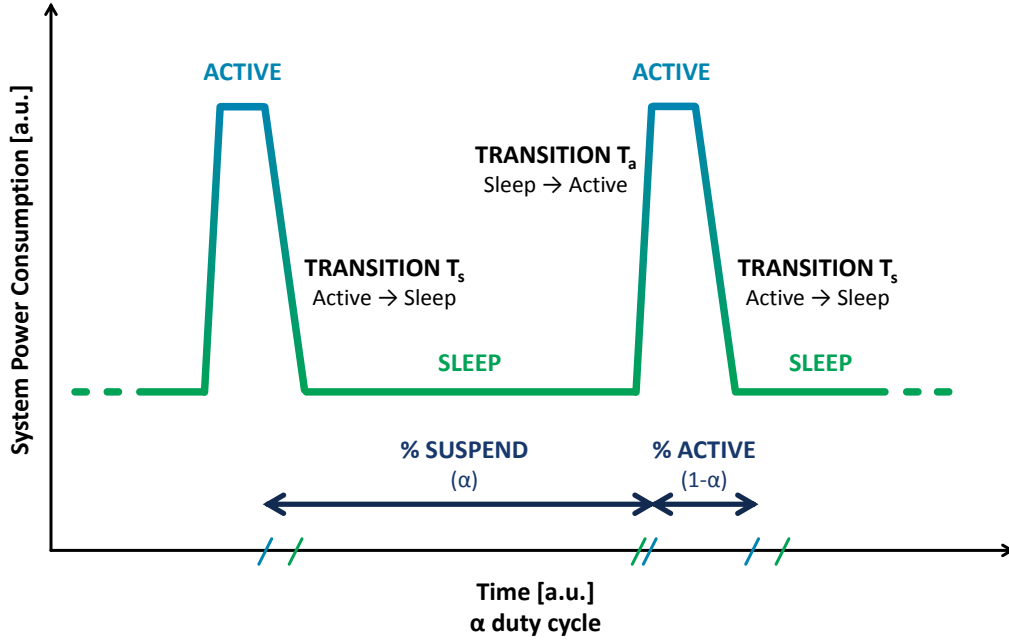


Figure 1.19: Activity scenario of a duty-cycled Ultra-Low-Power system.

As the systems are entering the active mode in a relatively short amount of time, it helps to greatly reduce the average power consumption and fit within the systems' constrained power budgets. In fact, for a battery-operated system, the lifetime depends on the battery autonomy T_{batt} which is given by:

$$T_{batt} \approx \frac{Q_{batt}}{I_{avg}} \approx \frac{E_{batt}}{P_{avg}} \quad (1.15)$$

with E_{batt} the energy that can be delivered by the battery and Q_{batt} the battery capacity expressed in mAh. It assumes a power conversion with 100% current efficiency, an energy delivered over a reasonable range of voltages and including non-idealities like memory effects [124]. P_{avg} is the averaged power given by:

$$P_{avg} = P_{Active} \times \%Active + P_{Suspend} \times \%Suspend \quad (1.16)$$

Particular attention must be paid to derive the average current consumption I_{avg} from P_{Active} and $P_{Suspend}$ (instead of P_{avg}) since the supply voltage might change between modes.

¹¹Here, the suspend sequence includes the low power modes and the transitions associated to enter/leave the modes (sleep and wake-up transition) and swith between them.

$P_{Suspend}$ is the power consumed during the suspend mode which is mostly due to the Always-On power domain and the leakage currents of the other blocks. To simplify the previous equation, the power consumed during the transitions between modes has been included into the power consumption of the sleep mode. The total power consumed during the transitions and the sleep determines the overall power of the suspend mode. However, to reduce the overall power consumption and minimize wasted power, mode transition optimizations are crucial. The impact of mode transitions will be thoroughly explored in Section 4.5.6.

The suspend and active modes are not unique. Indeed, for most of digital SoCs, the activity changes over time depend on the computational load. Then, the system presents different active modes according to the computational task. Generally, compute-intensive, low-speed and idle operations are proposed, which result in non-continuous power consumption during the whole active phase. The same distinction can be done on the suspend modes. The total power can be the result of low power mode, for instance, sleep, deep sleep, etc.

Using a generic SR63 coin-cell battery selected for its small size ($\sim 1 \text{ cm}^2$), a relevant analysis of the trade-offs between system's autonomy and power modes utilization is done in Table 1.3. The methodology can be adapted to compare SoCs' autonomy with a common reference battery or a known harvester/power storage couple.

Device	Activity per day	Active current	Sleep current	Average current	SR63 14.5 mA h battery autonomy ¹
STMicroelectronics BlueNRG-2 [125]	1% – 15 min. 0.07% – 1 min.	8300 μA	2.1 μA	85.08 μA 7.86 μA	3 months, 4 days 3 years, 3 months, 4 days
NORDIC nRF51822 [126]	1% – 15 min. 0.07% – 1 min.	10 500 μA	0.6 μA	105.59 μA 7.89 μA	2 months, 27 days 3 years, 3 months, 0 days

¹ Battery autonomy = (battery capacity) / (average current consumption) = life duration – assuming a linear regulator with 100% current efficiency.

Table 1.3: Suspend mode consumption and autonomy for two standard BLE devices using an SR63 coin-cell battery

Considering 1% spent in active mode per day, which corresponds to 15 minutes active over 24 hours, standard SoC lasts around 3 years and 3 months for a STMicroelectronics BlueNRG-2 [125] and a Nordic nRF51822 [126]. To target a 25 years utilization with the same battery, it would require an average power consumption in the micro watt order, justifying the need for aggressive power management techniques.

Lastly, continuous operation with an energy harvesting device requires the average power consumed by the system to be lower than the average power scavenged. In this case, contrary to a full battery-operation, the system autonomy is determined by the energy harvesters rather than the battery capacity. For system only powered through the energy-harvester (i.e., battery-less systems), the instantaneous power consumption will be the parameter of interest. Such scenario needs to be explored and leads to the Chapter 2, Section 2.1.4 discussions and ULP architecture.

1.5.2 System Level Power Optimizations

Low Power IPs

To reach maximum energy efficiency at system level, ULP IP are mandatory yet require significant design effort for ULV operations. Current challenges are around the following components; memory macros and bit cells with low writing and retention power which maintain a very low area [127], ULP clock generators and time references [128], ULP always-on radios [129] and sensors [130].

Multi-Voltage and Frequency Islands Partitions

To deal with power and performances, multiple voltage and frequency islands – often called power islands – can be defined at SoC level. While some portions of the design need high speed operations, others may operate at lower frequency, thus saving energy. For instance, partitions can be performed between the system computational core, the memory and the peripherals. It does, however, require one tunable or multiple supply voltage generators (as well as clock generator) which automatically impact the power budget. Partitions are also required for power gating implementation.

Multi V_{TH} islands

While MT-CMOS suffers from technology limitations at circuit level (see Section 1.3.4), the technique is mitigated at system level [131]. Entire module are incorporated into separated power island and implemented with a particular transistor flavor depending on the specifications requirements (speed or power saving). Contrary to voltage or frequency islands partition, no additional hardware is required. Moreover, it has little impact on the standard design flow since modern EDA tools usually support multi-threshold design correctly.

Dynamic Voltage and Frequency Scaling (DVFS)

Dynamic Voltage and Frequency Scaling (DVFS) is an open-loop power management technique, where the voltage and/or the frequency used in a component are increased or decreased, depending upon circumstances [132]. A performance manager is used to predict the performance requirements of the system tasks and define a functional operating mode, consisting in practice of a voltage supply and frequency point close to the MEP. Then dynamic regulation is performed by controlling voltage/frequency actuators to increase the speed or to reduce the system power.

As described in Figure 1.20, through frequency scaling the circuit slack is reduced and the energy per operation saving equals to the fraction of leakage overhead power removed. With voltage scaling, both static and dynamic power is saved following Section 1.3.2 equations. Still, it might impact the operating frequency leading to intrinsic limitations. DVFS needs to divide the system into several power domains working with an independent supply and frequency generators. It is also compatible with standard design flow and EDA tools, thus making good use of the performance needs of the system workload and achieving performance and power trade-off in real time. Moreover, DVFS does not impact the highest performance of the system. However, DVFS implementations at ULV require custom performance monitor, power

management and clock generator modules. Beyond the inherent complexity of the technique, the estimation of the system workload is challenging and limit the DVFS efficiency.

Adaptive Voltage and Frequency Scaling (AVFS)

Adaptive Voltage and Frequency Scaling (AVFS) is a closed-loop extension of DVFS, which solves the reliability limitations of the DVFS algorithms. AVFS takes into consideration the applicative constraints regardless of the system intrinsic abilities and PVT variations [109, 133].

An *in-situ* performance module is added that monitors the workload of the system in the current environmental conditions. In basic implementation, an estimation of the real delay needed by the system is performed through a critical path replica, then transmits the correct voltage and frequency operating conditions to the power management and clock modules, which in return regulates the voltage supply and the clock frequency of the sub-module.

Dynamic Threshold Scaling

Using the capability to modulate the threshold voltage of the devices (see Section 1.2), dynamic implementation can be performed to deal between speed and energy trade-offs during the active and suspend mode of a system. In basic variable threshold CMOS implementations, an additional stabilization circuit is added to detect the leakage currents of the transistors and feedback the variations to actuator circuits [134]. In return, they apply a voltage to the substrate to compensate for leakage. Such system requires a charging pump to provide above supply voltage power rails, thus resulting in increased complexity with higher surface costs.

Instead of using leakage sensing, performance monitors can be implemented to offer a second open-loop compensation. Combined with voltage and frequency actuator it is called Adaptive Body Biasing (ABB) (echoing the AVFS technique). In FD-SOI, thanks to the increase of the biasing range possibilities the technique shows improved performances for dynamic PVT compensations [135, 136].

Lastly, SWBB has been demonstrated in a dynamic fashion in [137]. This time, discrete biasing points are available, and used to modulate performances between active and suspend mode of the system. This time the inherent biasing generators are removed making it a simple yet highly efficient biasing scheme.

Dynamic Power Management (DPM)

From Section 1.5.1 it is established that SoCs operate following duty-cycled operations. DPM operates mode management to mainly improve the system sequences efficiency [138]. By applying the previous techniques onto the several power islands, like cutting off the clock signal or power supply to the circuits when they are in suspend mode, low power operating modes become available (e.g., idle, standby, power down, etc...). In these modes, the frequency or supply voltage can also be reduced while the unnecessary modules are halted to save power dissipation.

Some additional hardware regrouped in a PMU drive the gating/isolation signals and peripherals to execute a specific mode. For large SoCs, complex or re-programmable solutions are necessary. Hence, a complementary digital core can be added as shown in [139]. For smaller

system, programmable state machines are preferred as they result in small area and energy footprints.

DPM requires knowledge on the system behavior and along with support from the hardware and software for both suspend or active optimizations [140]. Based on predictive and stochastic policies, the software predicts whether a module can go to sleep long enough to save energy or what are the minimum hardware modules needed to meet the different computation tasks. Then, orders are asserted to the PMU to execute the proper suspend mode or disable components while still complying with the system workload.

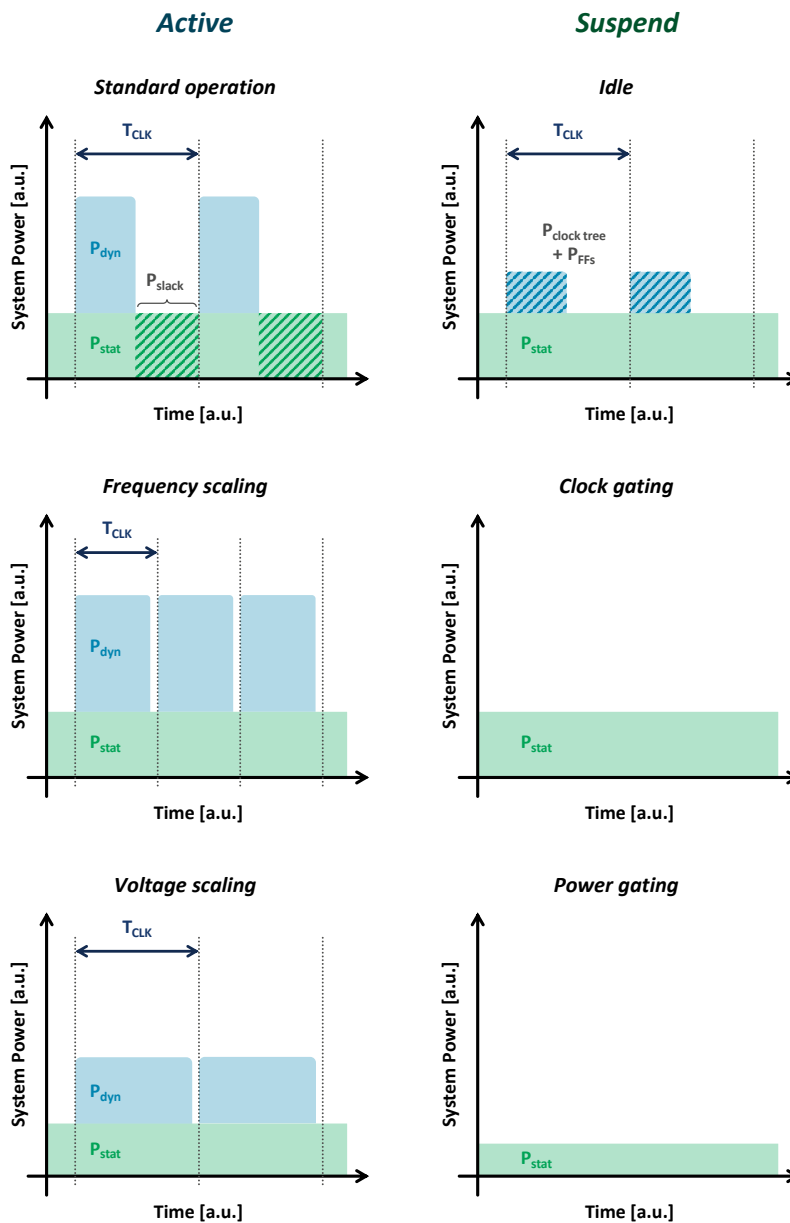


Figure 1.20: Impact of system-level power management techniques on the power consumption during Active (left) and Suspend (right) modes. Adapted from [71].

State Retention

When the SoC or parts of it are powered down, some critical sequential elements are required to retain the current data for proper wake-up. State retention is thus required to save the information. While system memories are prone to perform such task, their utilization results in latency before and after powering down, which might be critical in term of timing and power consumption. Therefore, retention registers are available to directly save the data into the flip-flop.

Combined with power gates, this technique is referred as State Retention with Power Gating (SRPG). It requires proper definition of the sensitive registers that need retention and additional signal controls and sequencing. Such implementation is described in Figure 1.21. The retention flip-flop uses the switchable main power supply V_{dd} and an additional always-on V_{RET} supply for information saving (controlled using CMD RET). Inherently this increases the power management complexity and floorplan integration due to the dual power supply rails required. However, it helps to reach the lowest suspend mode without loss of information.

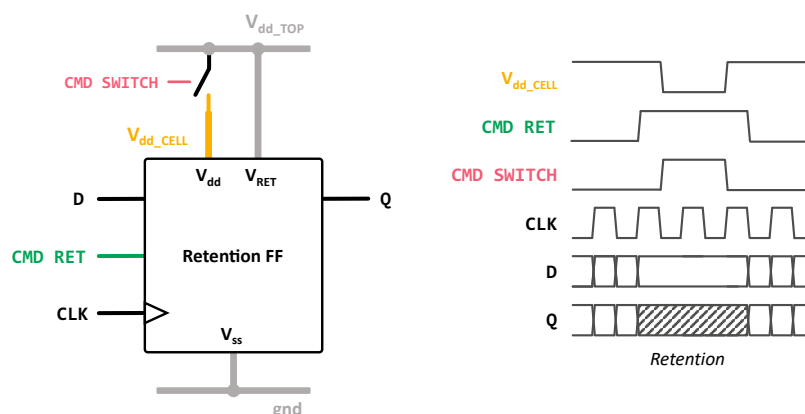


Figure 1.21: Schematic description for SRPG implementation with corresponding control signals and sequencing. Retention control is ensured by CMD RET, CMD SWITCH for power gating of the main power supply.

Core Microarchitectural design

Although out of the scope of this thesis, core microarchitectural design optimizations are available to enable ULP systems. For instance, to improve energy efficiency, processor predictions and speculations must be avoided, and STALL instructions implemented instead of simple NOP. Data and architectural gatings are also highly efficient to reduce unnecessary dynamic power when inputs of logic paths are not changing. Lastly, improvements are also possible on the pipeline such as dynamic pipeline adjustments.

1.6 Summary

1.6.1 Review of Ultra-Low-Power Techniques

All along this chapter, ULP techniques have been presented to reduce the power. These solutions are available at different design levels, starting from the technology and device up to the system level. Such techniques are not mutually exclusive since they target different level of power savings, and seek to improve energy efficiency during the different steps of the system lifetime, as shown in Table 1.4.

At design time, the optimizations are performed during implementation. The power reduction is independent of the architecture and the applications running on the system but it must ensure that the performance specifications are reached. Since the power consumption of the system follows the computational load and system activity, power optimizations related to the running application (i.e., during run time) are available. Lastly, suspend phases are critical on the overall power consumption. In standby time are regrouped the techniques to control the power consumption during inactivity periods and reduce the power of the modules non necessary for the tasks to execute.

	ULP Techniques	Design Time	Run Time	Standby Time
Device	Advanced CMOS technology	✓	–	–
	CMOS process options	✓	–	–
	Biasing techniques	✓	–	–
	Sub/Near-Threshold operation	✓	–	–
Circuit	Exotic Topology	✓	–	✓
	Multi- V_{TH} implem.	✓	–	–
	CMOS logic options	✓	–	–
	Swapped Body Biasing	✓	–	✓
Mod./Gate	Clock gating	–	✓	✓
	Power gating	–	–	✓
	Asynchronous design	✓	–	–
	Non-conventional designs	✓	–	–
System	Voltage/Frequency Partitions	✓	–	–
	Multi- V_{TH} islands	✓	–	–
	State Retention	–	–	✓
	DVFS	–	✓	✓
	AVFS	–	✓	–
	Dynamic Treshold Scaling	–	✓	✓
	DPM	–	✓	✓
	Microarchitectural Optim.	✓	✓	–

Table 1.4: Summary of the Ultra-Low-Power techniques encountered.

1.6.2 Existing SoC Implementations

The ULP optimizations listed in the previous sections have been presented independently from each other. In practice, low power designing call for employing a combination of various solution. To highlight this phenomenon, a summary of recent single-core energy efficient microprocessors targeting IoT applications is provided in Table 1.5 (continuing in Table 1.6). The ULP techniques of Table 1.4 used in each system are also listed and sorted.

Several conclusions can be drawn concerning techniques utilizations. All the proposed designs are covering a sub or near threshold voltage operation to achieve ULP performances. However, the supply voltage range is not limited to ULV and recent designs are also covering an Ultra Wide Voltage Range (UWVR) [141–145], generally up to the nominal voltage of the technology. This choice is often justified by a need for a target performance or a too low operating frequency at low voltage.

The number of techniques applied has increased over the years and even moving from device and circuit-oriented solutions to the system-level and architecture optimizations. A distinction can eventually be done on the type of demonstrators encountered. On the one side, some are “low-level” Proof-of-Concepts (PoCs) focusing on one or two new techniques or a specific block and exclude system approach [97, 146]. These solutions are often proposed by academic teams. On the other side “high-level” demonstrators, proposed directly or in partnership with industrial or big university groups, aim system and architecture optimizations. These SoCs integrate more and more components and IPs and are getting closer to a final product with a given application [143].

Dynamic optimizations are developing. Voltage, frequency and bias modifications are now implemented to generally compensate from PVT variations [136, 147] or reduce the power consumption of the system during its lifetime, whereas [109] proposes dynamic solutions. Retention techniques using custom memories and system integration are demonstrated in [144, 148] leading to very low minimum suspend power. Nonetheless, adaptive optimizations depending on the activity or software-related are yet sparse leaving rooms for architectural optimizations. Software and co-integration of IPs (e.g., radio, Wake-up Interrupt Controller (WIC), energy harvester, etc...) is still limited.

The design effort has also changed over the years. Former solutions were mostly targeting dynamic power reduction which was dominant in older technologies. Then, with the advent of new CMOS technologies, the emphasis has shifted towards leakage reduction and even restarted the interest for exotic circuit topology with a focus on custom digital library [145, 149].

However, the industrial standard flow compatibility and eco-system seems to be a determining factor. Although optimizations are developing and have shown good results, the current design effort required by Integrated Device Manufacturers (IDMs) to integrate these solutions make them non-conclusive. Industrial seems to be particularly reluctant to exotic circuit topology, non-conventional design or Instruction Set Architecture (ISA) optimizations.

Standardization also led to the acceptance of 32-bit microprocessor architectures over 16 or 8-bits. This choice is motivated by the need for a broad range of applications requiring more memory storage for data and computation. Indeed, even though they are more complex, 32-bits architectures help to reduce the code size to process sensor data, resulting in more energy-efficient architectures for a given computational load. Finally, the academic availability of the Cortex ARM M0 has helped to spread a standard architecture for IoT applications. This solution is now challenged by open source architecture such as the RISC-V core and derivatives.

Due to the non-homogeneity of the SoCs presented (technology chosen, various architecture, memory size, peripheral embedded, etc.) it is still complicated to compare designs on their performances only. However, from these tables some trends are observed. For all these solutions, the constraints are relaxed to target very low power consumption. Yet, it leads to non-realistic solutions which present a nW or even pW power consumption but at a kHz operating frequency. The sub-10 pJ/c. at the kHz range can be easily accessible however efforts are mandatory to reach a sub-1 pJ/c. at the MHz range which will enable IoT spreading and mainstream utilizations. Moreover, older technology node solutions present interesting power figure in suspend mode at the expense of a reasonable operating frequency whereas, a frequency boost can be brought by technology scaling. Although in return it inevitably leads to an increase of the minimum suspend power.

1.6.3 Conclusion

This background and state of the art analysis sheds light on the wide panel of solution for power reduction. Supply scaling and ULV voltage operation in the nanometer technology area is not sufficient; energy efficiency must be optimized up to the system level by taking strong advantages of the hardware and software interactions. To allow industrial deployment SoC architecture compatible with NZP applications, this work will respect the following criteria.

First, the utilization of the standard CMOS logic and cells in an advanced technology node like FD-SOI in 28 nm and 22 nm is needed for maximum active efficiency. The application of ULV operations is also mandatory to reach the system MEP. The resulting variation sensitivity should be compensated with simple methods otherwise “smart” margins will be defined.

Then, the utilization of body-bias as an additional FD-SOI trade-off knob between speed, efficiency and reliability should be considered. However, for NZP applications, the extra bias generators directly impact the strict power budget. Simplified solutions in the vein of SWBB must be preferred. Although this work primarily targets architectural optimizations, a comprehensive analysis of the low power components and IPs is still required. This might lead to component designing for ULV/ULP operations.

Moreover, power island partitions are mandatory for ULP techniques and modules integration. Not only it does allow integration of clock and power gating techniques compatible with standard EDA flow, it also facilitates high level power management techniques which leverage the system operating states. Similarly, frequency and supply generators should not be multiplied to fit the constrained power envelope. This work will prefer a single power supply for the whole system and discrete frequency modes.

Lastly, DPM is a promising option to leverage the system power modes according to the SoC activity and running applications. Beyond the conception of a pure NZP SoC, such techniques should be implemented at system level. However, the need for costly power and software bench-marking is constraining. Some adaptability to environmental variations must be added.

Reference	ISSCC 2007 [137]	JSSC 2009 [150]	ESSCIRC 2011 [141]	JSSC 2013 [109]	ISSCC 2013 [151]	JSSC 2013 [152]	ISSCC 2015 [136]
Features	Technology	0.18 μm CMOS	65 nm CMOS	65 nm CMOS LP/GP	0.13 μm CMOS	0.18 μm CMOS	28 nm FD-SOI
	CPU	32-bit RISC core	32-bit Custom CPU	16-bit MSP430 compatible	32-bit RISC core	32-bit ARM Cortex M3	32-bit SPARC V8
	Memory	N/A	52 x 40-bit data SRAM 64 x 10-bit inst. SRAM 64 x 10-bit inst. ROM	64 B I\$ + 18 KB DC/DC conv., PMU, Clock gen., SPI, timers, UART, TDC, GPIOs, HW accel.	N/A	5 KB SRAM	2 KB I\$ 2 KB D\$ 16 KB SRAM
	Peripherals	N/A	SPI, I2C, Timers, WIC		N/A	DC/DC conv., LDO, Temp. sensor, PMU, WUC, Harvester	Clock gen., Temp. sensor, DC/DC conv., Bias gen., UART, WIC, GPIOs
Optimizations	Device	Sub/Near-Threshold	Sub/Near-Threshold	Sub/Near-Threshold	Sub/Near-Threshold	Sub/Near-Threshold	Sub/Near-Threshold
	Circuit	Swapped Body Biasing	–	Multi- V_{TH} implem.	–	–	Advanced CMOS Techno.
	Gate	Clock Gating Power Gating	Power Gating	Clock Gating Power Gating	Power Gating	Power Gating	Body-Biasing Techniques
	Modules	–	–	Glitch Masking I\$	–	Custom bitcell	–
Performances	System	Dynamic Swapped Body Biasing	Custom bitcell Reduced Energy ISA	AVS Multi-voltage islands	State Retention	–	ULP Clock Generator Body Bias Generator Dynamic Body Biasing
	Area [mm^2]	0.84 (core) 2.25 (die)	N/A	0.42 (core) 0.66 (die)	0.846 (core)	N/A	0.62 (system) 15.8 (die)
	Temperature [$^{\circ}\text{C}$]	N/A	N/A	-40/85	-5/65	25	-40/40
	Supply Voltage [V]	0.23 - 0.5	0.54 - 1.2	0.32 - 0.48	0.3 - 0.5	0.4 - 0.5	0.33 - 0.45
Performances	Memory Voltage [V]	N/A	0.4 - 1.2	1	N/A	0.4 - 0.6	0.33 - 0.45
	Frequency	375 kHz - 16 MHz	540 kHz - 82.5 MHz	25 MHz	5 MHz - 25 MHz	72 kHz - 1 MHz	1 MHz - 20 MHz
	Min. Suspend Power	3.02 μW	N/A	1.7 μW	584 nW	100 pW	N/A
	MEP of SoC	55.69 $\mu\text{J}/\text{c.}$ @ 0.35 V/5 MHz	10.2 pJ/c. @ 0.54 V/540 kHz	6.92 pJ/c. @ 0.40 V/25 MHz	17.2 pJ/c. @ 0.30 V/5 MHz	28 pJ/c. @ 0.40 V/72 kHz	26 pJ/c. @ 0.45 V/20 MHz

Table 1.5: Ultra-Low-Power (ULP) System-on-Chip (SoC) and referencing of the low-power optimizations (1/2).

Reference	ISSCC 2015 [146]	ISSC 2016 [142]	JSSC 2017 [143]	VLSI 2017 [144]	JSSC 2017 [97]	ISSCC 2018 [145]	ISSCC 2018 [147]
Technology	0.18 μ m CMOS	65 nm CMOS	14 nm FinFET	65 nm CMOS	40 nm CMOS	0.18 μ m CMOS	28 nm FD-SOI
CPU	32-bit ARM Cortex M0+	32-bit ARM Cortex M0+	32-bit ARM x86 IA	32-bit ARM Cortex M0+	32-bit ARM Cortex M0	16-bit MSP430 compatible	32-bit ARM Cortex M0
Memory	256 B	2 KB ROM 8 KB ULV SRAM 16 KB SRAM	8 KB DTCM 64 KB SMEM 16 KB BootROM	12 KB SRAM 2 KB ROM	256 KB SRAM	2 KB	32 KB ULP SRAM 32 KB SRAM
Peripherals	Clock gen.	DC/DC gen., LDO, RTC, PMU, GPIOs, Debug, SPI, 128b AES	Clock gen., I2C, Timer, GPIOs, IOMUX	Clock gen., PMU, GPIOs, SPI, DMA, IRQ, LDO, 128b AES	UART, GPIOs, Debug	Clock gen. / GPIOs	Clock gen., FFT accel., WIC, GPIOs, Body-bias gen., DC/DC gen.
Optimizations	Device	Sub/Near-Threshold Thick Oxide Devices	Sub/Near-Threshold Advanced CMOS Technology Thick Oxide Devices	Sub/Near-Threshold Thick Oxide Devices	Sub/Near-Threshold	Sub/Near-Threshold	Sub/Near-Threshold Advanced CMOS Technology Thick Oxide Devices
	Circuit	Dynamic Leakage Suppression Logic	Multi- V_{TH} implem.	Multi- V_{TH} implem.	Transmission Gate Topology	Custom Circuit Topology	Body-Biasing Techniques
	Gate	–	Clock Gating Power Gating Custom bitcell	Clock Gating Power Gating Custom bitcell	–	–	Power Gating Custom bitcell
	Modules	–	Multi-voltage/frequency islands	State Retention, DVFS, Multi-Voltage/Frequency Islands	–	Custom bitcell	State Retention, Adaptive-BB, Multi-Voltage/Frequency Islands
	System	–	–	–	–	–	–
Performances	Area [mm²]	2.04 (core)	1.28 (core)	1.15 (core)	0.16 (core)	5.33 (core)	0.7 (core)
	Temperature [°C]	-5/65	3.76 (die)	3.76 (die)	1.8 (die)	9.48 (die)	2 (die)
	Supply Voltage [V]	0.16 - 1.15	25	0/75	0/70	25	–
	Memory Voltage [V]	0.16 - 1.15	0.25 - 1.2	0.29 - 1.2	0.19 - 0.6	0.2 - 1.1	0.4
	Frequency	2 Hz - 15 Hz	0.25 - 1.2	0.29 - 1.2	0.6	0.2 - 1.1	0.5 and 0.8
Min. Suspend Power		N/A	0.5 MHz - 297 MHz	12 kHz - 60 MHz	1 MHz - 50 MHz	1.6 kHz - 2.8 MHz	40 kHz - 80 MHz
		92.04 pJ/c.	550 nW	46 nW	N/A	595 pW	9408 nW
	MEP of SoC	@ 0.55 V/7 Hz	11.7 pJ/c. @ 0.39 V/688 kHz	6.3 pJ/c. @ 0.35 V/174 kHz	43.22 pJ/c. @ 0.37 V/13.7 MHz	33 pJ/c. @ 0.45 V/19 kHz	3.0 pJ/c. @ 0.4 V/48 kHz

Table 1.6: Ultra-Low-Power (ULP) System-on-Chip (SoC) and referencing of the low-power optimizations (2/2).

Chapter 2

Ultra-Low-Power System Definition

THIS chapter introduces the ULP SoC architecture developed during this work to reach maximum energy efficiency. Due to the limited availability of all the peripherals and sub-systems required to implement a full ULP SoC for IoT-oriented applications, an implementation framework is mandatory to keep track of optimizations from transistor to system-level. A generic SoC is examined in the context of energy limited resources. Therefore, considering the energy constraints of Figure 10 and possible applications, an ULP SoC functional description is proposed.

First, in Section 2.1 is analyzed the IoT nodes energy flow and the current general architecture adapted by this device. Emphasis is also performed on the clock generation and time keeping requirements along with realistic use cases scenarios. Consequently, an ULP SoC functional description is derived in Section 2.2. This organisation ensures an ideal system partition and low power modes which can be activated to reduce the power consumption during suspend sequences. A resulting implementation is then proposed in Section 2.3. This system configuration will be used for the following part of this work to demonstrate the technology, component and system optimizations performed in 28 nm FD-SOI (see Section 2.4).

2.1 Ultra-Low-Power System-on-Chip Architecture

In this section the requirements for ULP SoC platforms in the context of power constrained systems for IoT-oriented applications are detailed. Depending on the required use cases, such as miniaturized implantable biomedical sensor [153], industrial machine diagnosis application or general personal IoT [154], various architectures can be developed.

2.1.1 IoT Device Energy Flow

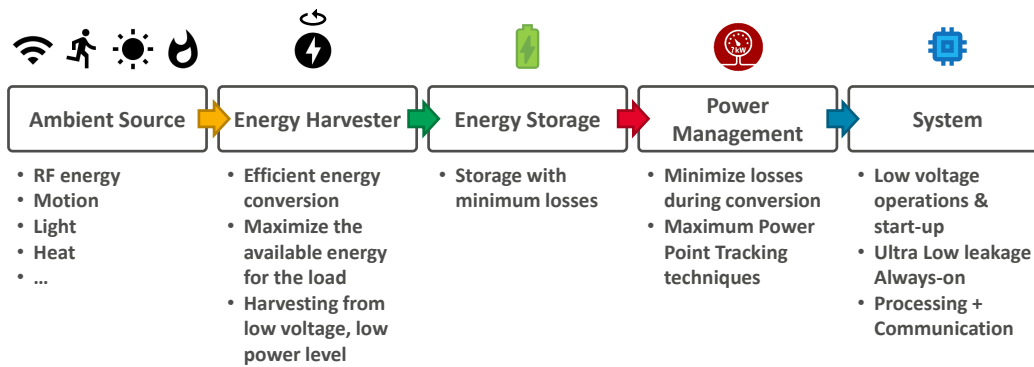


Figure 2.1: Energy flow for an ULP IoT device [155].

Figure 2.1 describes the general energy flow of an IoT device using an energy harvester. The energy is collected from one or multiple ambient sources as seen in Figure 12. An adapted harvester performs conversion while maximizing the available energy for the load. This is done using Maximum Power Point Tracking (MPPT) algorithms as in [156, 157]. Harvesting must be performed from very low power and voltage levels to extract every possible chunk of energy from the source. In fact, the challenge is to guarantee power conversion on a large scale and not only when the source is operating in the optimal conditions.

The harvested energy is then stored in a capacitor in the form of a voltage source with minimum conversion losses. If the energy harvester has not been designed to directly operate the complete system, the voltage available across the storage capacitor will define the amount of energy available to operate. The system will be able to power-up only if the storage reaches a pre-defined threshold level. This level indicates the minimum energy level of the IoT device.

Next, the stored energy is converted by a power management circuit to provide the various power supplies required by the next stage. Again, MPPT techniques can be applied at this stage to maximize the energy-efficiency of the power conversion.

Finally, at the heart of our IoT device are found communication and processing modules. To perform the desired functionality, low voltage operation and short start-up times are required to guarantee an extended lifetime. Moreover, as seen in Section 1.5.1, IoT devices spend a large percentage of their time in suspend modes. Hence, ultra-low leakage power consumption is needed.

In practice, the border between blocks is vague and adjacent modules are designed as a whole. For instance, the energy harvester is merged with the energy storage and the power management circuit, or the power management with the system, etc. However, significant im-

provements must be leveraged by optimizing this infrastructure as a whole. In the context of battery-operated systems, the ambient source and energy harvester are replaced by a battery, while the functionality of the remaining blocks stays unaltered.

2.1.2 System and Processing Elements

There is certainly not a unique solution to define a SoC, however as shown in Figure 2.2 these systems share some common features and a relatively general architecture [14, 158].

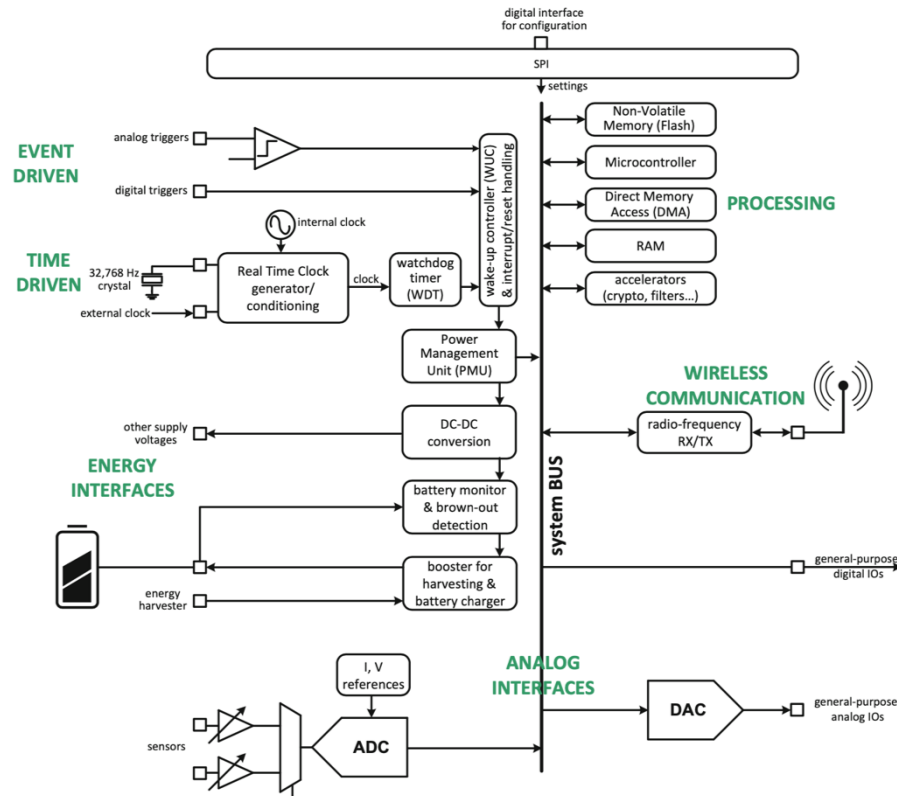


Figure 2.2: Relatively general architecture of IoT nodes with detailed sub-systems [14].

Through sensors/actuators combined with an analog front-end, these systems interact with the physical world. On the one side, programmable gain amplifiers and Analog-to-Digital Converters (ADCs) are used to convert the information to the digital world. On the other hand, Digital-to-Analog Converters (DACs) generate analog voltage for the actuation of General Purpose Input Outputs (GPIOs) or other external components. Digital I/Os are also available.

From these raw data, processing is performed with a micro-controller unit and necessary peripherals (Random Access Memory (RAM), Non-Volatile Memory (NVM), Hardware (HW) accelerator, etc...). RAM is needed for the execution of the program and to store some data in suspend modes. The NVM contains the instructions and data that need to be permanently stored even if powering down occurs. All the interactions between the system modules are performed using a unified communication bus.

Depending on the application and the configuration, the whole system interacts and responds according to specific events (time-driven) or periodic triggers (event-driven). Time-

driven operation requires accurate time stamping to keep track of the system evolution and guarantee inter-systems synchronization. Thus, an accurate clock source is required (see Section 2.1.3). Event-driven modes relies on monitoring digital or analog signals and detection of particular occurrences such as spikes, pulses, threshold values reached.

Communication with the external world is performed through standard digital interfaces (Universal Asynchronous Receiver Transmitter (UART), Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C)) or wirelessly using a radio interface. The radio is often split between a digital interface handling the radio protocol decryption and an analog front-end required to send the data on a physical layer.

The power sources are either a battery and/or an energy harvester yet, in both cases additional circuits monitor the state of the storage element and the energy available. Power conversion is performed to generate the various voltages needed by the components. The PMU rules the interaction between the energy source and the system depending on the selected power modes.

2.1.3 Clock Generation and Time Keeping

kHz VS. MHz Range

An SoC generally embeds different types of clock references. Low frequency sources are used for absolute accurate time generation. An on-board quartz is typically used, though on-chip integrated solutions are nowadays available (see Section 3.1). A standard 32.768 kHz is a trade-off between power consumption and ease of utilization¹ This reference has a twofold purpose. It is used by Real Time Clock (RTC) circuits to keep track of the time spent between two specific events. For instance, it helps to synchronize devices and transmit data at fixed intervals without power penalties due to missed data transmission [159]. At the same time, this low frequency signal can also be used as an input to generate the main system clock operating in the MHz range.

Then, a clock multiplier circuit is used to generate that second signal. It relies on an oscillator which frequency is tuned to match a multiple of the input reference. Common architectures are Phase Locked Loops (PLLs) yet other topologies can be found [71]. Moreover, for SoC embedding an RF transmitter, very high frequency local oscillators are needed (e.g., 2.4 GHz for Bluetooth). These types of clock sources are different from the digital ones since they depend on the Radio Frequency (RF) front-end and the power management strategies [160]. They are out of the scope of this chapter.

Clock Modulation for Duty-Cycled Systems

As previously explained, IoT devices follow duty-cycled operations to fit within a constrained power budget. When the system receives a wake-up order, it goes in active mode, where it starts collecting data using a panel of sensors. Then, a processing phase is performed leading to transmission of information generally using a wireless communication system. Thereafter, the system can go back to a suspend mode.

Consequently, the system clocks play different roles in timekeeping as shown in Fig-

¹It corresponds to 2^{15} Hz, making it trivial for a 15 bit digital counter to generate 1 second time intervals.

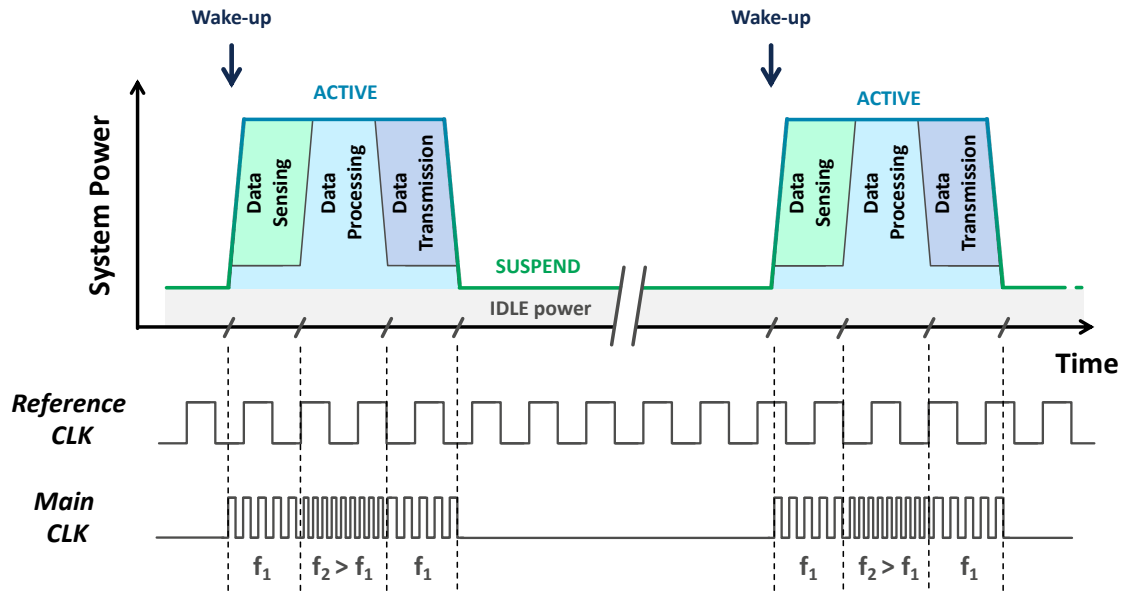


Figure 2.3: Clock modulation during the duty-cycled operation of an IoT device.

ure 2.3. Using a time frame from the milli second to tens of seconds, the low frequency reference clock defines the various phases of operation of the system while the main clock is used for cycle-to-cycle operation of the core.

Hence, the performance requirements and specifications are widely different for these two clocks. The reference clock should present a very low drift over a long period of time to stay synchronous with the outside world or base-station. It is also operating continuously, thus requiring a very low standby power. The main clock only needs a low cycle-to-cycle jitter but requires flexibility in the range of frequency generated. Indeed, it may require to be turned off or to operate at different output speeds if Adaptive Frequency Scaling (AFS) is used depending on the system workload (see f_1 and f_2 on Figure 2.3).

2.1.4 Use Cases Scenarios

The organization of a SoC is related to the application and the tasks that will run on the device. Similarly to the architecture presentation of Section 2.1.2, describing the whole panel of use cases scenarios might be tedious. However, by focusing on energy and activity driven systems, a relatively general application framework can be given. Thereafter, other use cases can be derived from these two main utilizations. In the following situations, an energy harvesting system, recharging the main energy source is assumed.

Energy Driven System

As shown in Figure 2.4, an energy driven use case maximizes the energy available for the system over the task or the application running on the system. Starting from a suspend phase, the system receives a wake-up signal. It performs sensing, processing, data transmission, which in return consumes the energy stored. Then, the system goes back to a suspend phase. Conse-

quently, during the inactivity sequence, the harvester recovers some power from the external world to refill the energy source.

However, the power consumption and the time lengths of the task performed are not necessarily constant. Some fluctuations might occur. Hence, for energy-oriented use cases, the system might decide to delay some tasks to the next active cycle. Unfinished tasks can be delayed and ordered using some modified Real Time Operating System (RTOS) or by directly employing round-robin scheduling. Minimum granularity on the tasks is also required.

This solution ensures a constant power consumption where the SoC activity and task execution are led by the power constraints. The energy budget to be harvested and the size of the storage capacitor are usually sized for a typical workload over one or more activity cycles, thus avoiding energy shortage.

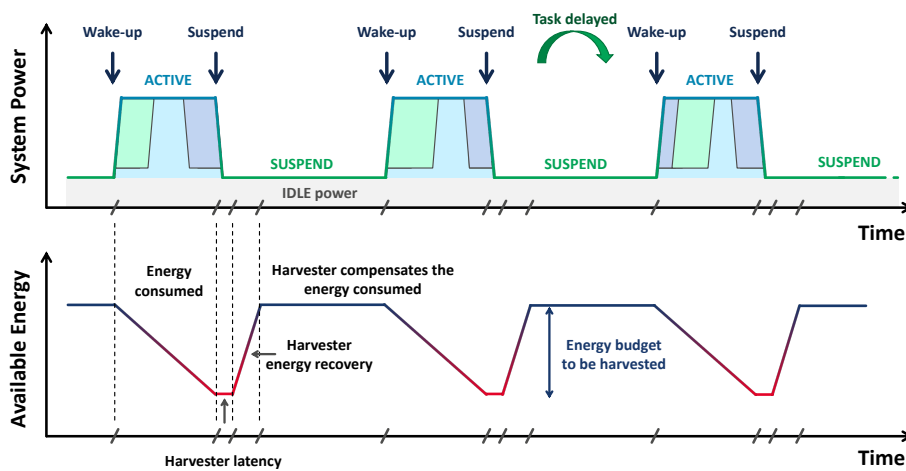


Figure 2.4: Energy driven use case.

Activity Driven System

Assuming an activity driven use case as in Figure 2.5, the task ordering is preserved no matter its duration. The system goes to sleep at the end of the whole execution or if not enough energy is available to perform the task. Therefore, the initial available energy for the system is not necessarily recovered in one suspend cycle.

This time the sizing of the energy harvester and storage unit is related to the application and a worst-case active phase (long sensing, processing and data transmission). This naturally increases the size required for the power elements, due to the extra margins taken.

Mixed-mode Solution

Using the previous use cases, a third-mixed mode can be chosen to operate the system. First, the SoC starts by following an activity-driven mode, thus preserving the tasks. Some energy is still recovered from the energy harvested, yet the longest active sequence is still limited by the storage capacity.

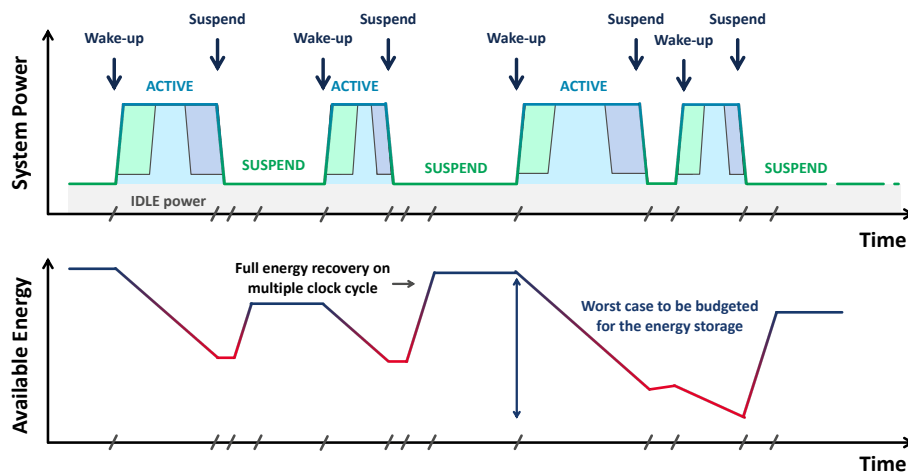


Figure 2.5: Activity driven use case.

Hence, when a given energy threshold is reached, the system switches to the energy-driven mode. Consequently, the task is limited by the energy harvester capabilities. It preserves the lifetime by constraining the tasks that might be executed.

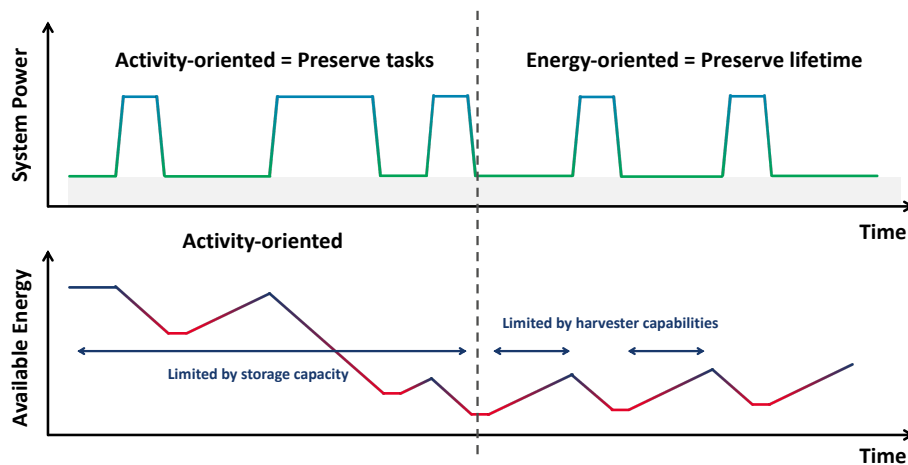


Figure 2.6: From activity to energy driven mode.

For Battery Operated only Devices

The previous assumptions and use cases remain applicable even though energy is not recovered during suspend sequences. Indeed, the idea of preserving the task or stopping the system when a given quantum of energy is consumed can be used. Be that as it may, it requires to increase the size of the energy storing element (i.e., the battery), to operate during the whole application lifetime.

2.1.5 Resulting Design Considerations

Following the previous sections analyses, several design considerations can be drawn to design an ULP SoC. Along its lifetime, the system alternates between active modes and suspend modes. To reduce the average power consumption over the life of the application, advanced modes – both in active and suspend – should be implemented to facilitate mode switching and energy efficiency.

The energy availability must be considered as the corner stone of the SoC. Hence, basic circuits are needed to monitor the state of the battery are the main energy source and detect potential shortage. This supply detector should also be programmable to switch from an energy driven mode to activity-oriented and conversely, while allowing modifications of the energy thresholds.

Date and time trackings are also necessary during the operating lifetime. An RTC associated with an accurate (low ppm deviation) and low-frequency (kHz range) oscillator is included to ensure the calendar functions. The core clock (MHz range) must guarantee a low cycle-to-cycle jitter but requires output flexibility. It requires fast on/off switching capabilities to operate between different frequencies or to be turned off.

Using an interrupt signal or an analog event the system should wake-up from the sleep-ing states on demand. At least one GPIO is required to receive this signal. Moreover, a Power Management Unit acting as a power and clock management system must be able to wake-up or switch-off sub-modules through programmable-interrupts.

Data should be saved into an embedded non-volatile memory to be propagated from one iteration of the active mode to the next one. However, the retention of the core state is not mandatory. In order to target the lowest sleep power and depending on the applications' contexts it might be omitted or bypassed (see Section 1.5.2). However, it leads to require system startup sequences to run at wake up which can lead to increase the response latency and energy consumed, as stated in [161].

Any system's elements not necessary to ensure the previous functions and/or specifications should be switched-off during the suspend states. The optional related voltage regulators must also be switched-off.

2.2 Ultra-Low-Power System-on-Chip Functional Description

2.2.1 SoC Description

Using Section 2.1 investigations, a ULP SoC functional description is given in Figure 2.7. This kind of representation highlights the minimum configuration required to build a generic IoT-oriented SoC with ULP system features. All the IPs used are supposed energy efficient yet, to decrease the power consumption during the lifetime of the SoC, functional domain partition is here used to define the logic relations between modules. As shown in Table 2.2.3, from this representation, power mode can be derived and then define PMU rules.

Starting from the energy storage, a first domain Power Down is used to monitor the available energy. Depending on internal or external events collected through the WIC, a start-up controller ensures the correct wake up of the rest of the system. Asynchronous digital modules are here selected to avoid the need for a clock reference and thus reduce the power

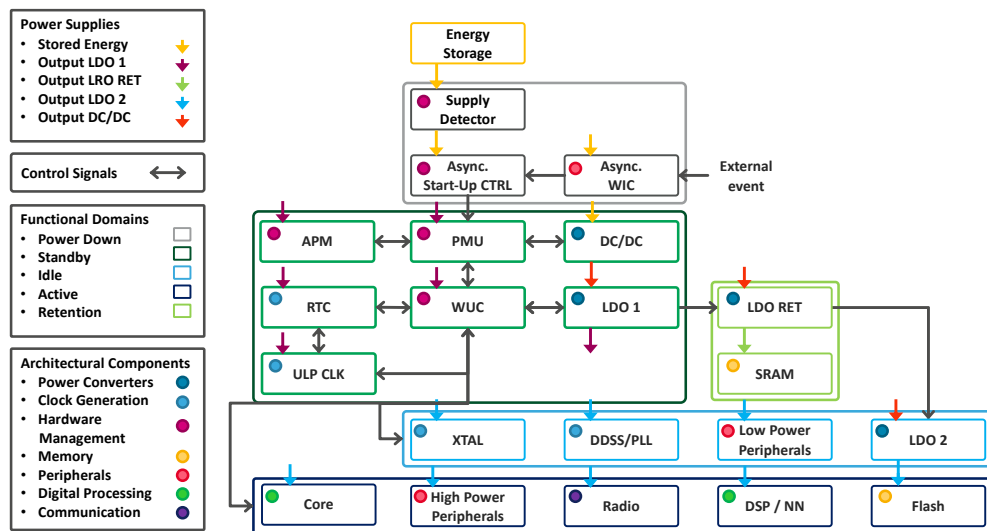


Figure 2.7: ULP SoC functional description.

consumption of this domain. Assuming enough energy is available, this functional domain is always-on and directly powered by the energy source.

The standby power domain built around the PMU ensures the minimum functions of the SoC. For correct operation, the ULP clock is first woken up by the asynchronous start-up controller of power down. Then, the same module communicates with the Wake-Up Controller (WUC) to restart the power converters and peripherals. When done, the start-up controller gives control of the system to the synchronous PMU which handles more system modes. An RTC is available for time tracking.

The retention domain contains the memory of the system. It embeds its own power generator. The control of the memory macros and its modes is done by the PMU depending on the required system modes.

The idle domain contains the peripherals needed for the core during idle phase, such as MHz clock generator, quartz references, low power peripherals and their necessary power supply. Lastly, the active domain contains the core itself, here an ARM Cortex M0+, high power peripherals, radio, hardware accelerators and memory. The peripherals of these two domains is controlled by the WUC which determined the current state following software order or the APM module.

2.2.2 Activity Timeline

Figure 2.8 gives an example timeline of the system activity and the activated domains. The SoC is first reset. Only the power down domain is activated. When sufficient energy is available, the system starts and activate the standby domain. Initialization of peripherals is performed, starting the retention and idle domain. Lastly, the system is fully woken up when the active domain is started. Thereafter, the system alternates between suspend mode and disable the corresponding domain. If an unexpected power shortage occurs, the whole SoC is disabled to preserve the last bits of energy.

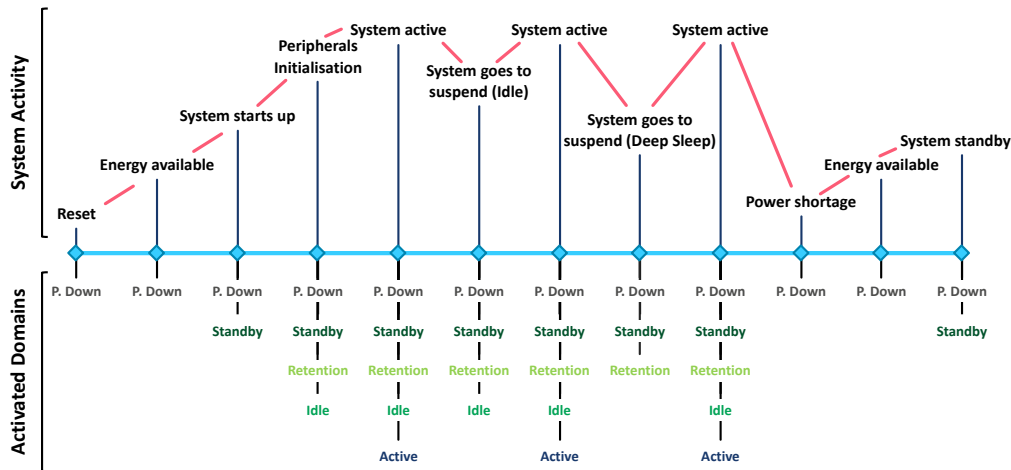


Figure 2.8: Timeline of the Figure 2.7 activated domains according to the activity.

2.2.3 Derivation of Power Modes

Using the functional description of Figure 2.7, a power modes derivation is adopted in Table 2.1. These modes are then available for the user in order to reduce the power consumption, without the need for internal understanding of the micro-architecture. The PMU and the WUC directly handle the micro-managing of the IPs.

	Peripherals	Active	Sleep	Ultra Sleep	Deep Sleep	Ultra Deep Sleep	Power off
Active	Core	ON	Gated	Gated	Retention	OFF	OFF
	High-P. Periph.	ON	Gated	Gated	Retention	OFF	OFF
	Radio	ON	Gated	Gated	Retention	OFF	OFF
	Flash	ON	Gated	Gated	Retention	OFF	OFF
Idle	DDSS	ON	ON	OFF	OFF	OFF	OFF
	Low-P. Periph.	ON	ON	OFF	OFF	OFF	OFF
	XTAL	ON	ON	ON	OFF	OFF	OFF
Standby	ULP CLK	OFF	OFF	OFF	ON	ON	OFF
	RTC	ON	ON	ON	-	-	OFF
	WUC	ON	ON	ON	ON	ON	OFF
	PMU	ON	ON	ON	ON	ON	OFF
	APM	ON	ON	ON	ON	ON	OFF
	LDO	ON	ON	ON	ON	ON	OFF
	DC/DC	ON	ON	ON	ON	ON	OFF
P. Down	Async. WIC	ON	ON	ON	ON	ON	ON
	Async. Start-up CTRL.	ON	ON	ON	ON	ON	ON
	Supply detector	ON	ON	ON	ON	ON	ON
Operating frequency		16 MHz	16 MHz	4 MHz	32 kHz	32 kHz	-

Table 2.1: SoC power modes and corresponding peripheral states.

2.3 Ultra-Low-Power System-on-Chip Implementation

Since all peripherals presented in Figure 2.7 were not all available for this work, a ULP SoC configuration has been derived. This system will be used in the following chapter of this work to evaluate the impact of actual low-power solutions in FD-SOI technologies as well as demonstrating novel system level optimizations. The digital core selected for this work is the ARM Cortex-M0+. An extended presentation of this IP is given in Appendix A.

2.3.1 System Partition

Table 2.1 is simplified by removing the power down functional domain and the corresponding modules. Therefore, a representation based on power domains can be made. These domains are defined as a collection of instances or hardware blocks with common power characteristics. This includes spatial constraints such as shared voltage supplies and temporal coherence. Indeed, the objects inside a power boundary must be able to operate in a similar logic state (ON/OFF, Powering DOWN/UP). From this definition, the following domains are formulated:

- Power-OFF (P.OFF) – This switchable OFF power domain contains the majority of the core logic. Any other peripheral that can be powered down along with the processor should be included. In the context of multi-core SoCs, several P.OFF domains including one or more Central Processing Unit (CPU) can be found. It corresponds to the former active functional domain;
- Memory (MEM) – This domain includes the memory macros of the system. Retention features can be implemented to power down parts of the power supplies while maintaining the stored information intact. Depending on the application, retention might be avoided as explained in Section 2.1.5. Since no flash memories are available in FD-SOI, this domain corresponds to the retention domain of Figure 2.7;
- Always-ON (A.ON) – This domain covers the components that must remain ON independently of the SoC state. Thus it regroups most IPs of the standby domain;
- Others – Any other peripheral or hardware block (e.g., power or clock generators) which can change their operating mode according to the power mode. It regroups the Idle and some of the standby domains IPs.

This partition model directly represents the physical implementation but contrary to Figure 2.7, it does not guarantee the logic relations between blocks. If a resource is required by a power domain component yet shared by other power domains, it must be in an accessible state to avoid logic dead-locks. For instance, this case happens when several masters – from different power domains – are connected to the system bus. In this case, the system bus must be placed in a separated power domain to be properly shared.

2.3.2 Resulting Memory Mapping

Based on the previous power domain representation, a generic power-oriented memory map is defined. It allows integration of the peripherals required for a specific implementation while guaranteeing easy integration of low power techniques and modularity of the Register-Transfer Level (RTL) description code. As an illustration, the Advanced High-performance Bus (AHB) to Advanced Peripheral Bus (APB) controllers required to address the A.ON and P.OFF can be separated in two different logic blocks and placed in independent power domains. Such orga-

nization results in power domains locations at fixed memory addresses, with no dependence to the power partitions. A global memory mapping adapted from ARM specifications is given in Figure 2.9.

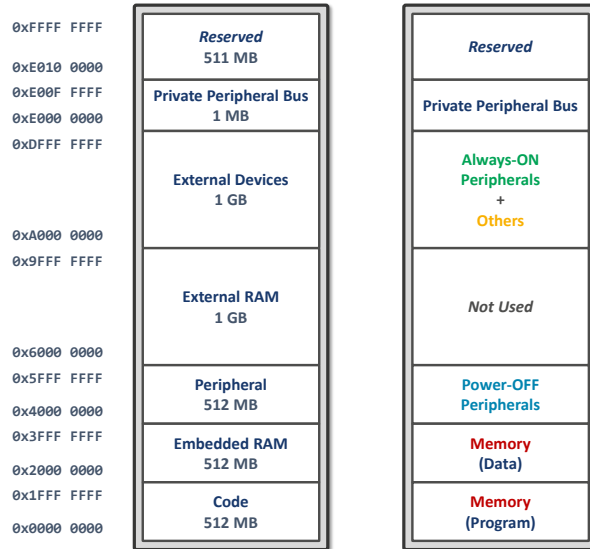


Figure 2.9: Architecturally defined memory map of the Cortex-M0+ (right) and corresponding implementation of this work (left).

2.3.3 Adopted Ultra-Low-Power Demonstrator Architecture

A physical architecture of the SoC is defined as presented in Figure 2.10. This architecture will be used as a template for the various implementations presented in this work (see Chapter 4).

The AHB-Lite system bus is added to the P.OFF to minimize the overall power consumption during suspend mode. The AHB to APB interfaces are split in two, considering the connection between the A.ON and P.OFF peripherals are also added to the P.OFF. This solution follows the memory map of the system shown in Figure 2.9, while minimizing the overall power in suspend sequences. The debug module of the core (see Figure A.1) is merged into the core logic and integrated in the P.OFF domain. However, the independent debug port (Debug Access Port (DAP)) is not switched-OFF during suspend modes as it offers access to the core and then it is kept in the A.ON to ensure a forced wake-up of the core and access to its registers.

The retention capabilities of FD-SOI memory macros is not used. Consequently, the program and data memories are both placed in the MEM power domain. Since no access to the information stored is presumed during the suspend modes of the core, the memory controller which interfaces the memories with the system bus is located into the P.OFF.

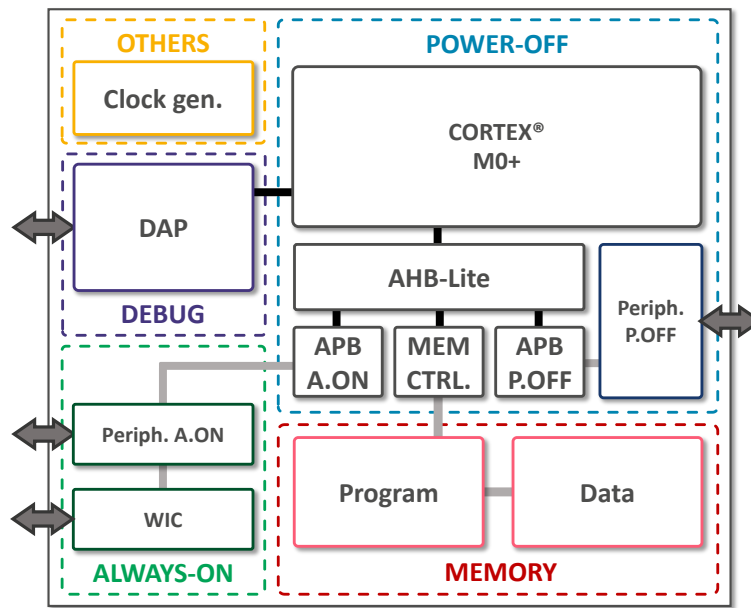


Figure 2.10: Adopted system partition for Ultra-Low-Power applications.

2.4 Conclusions

Starting from the energy flow of an ULP IoT device with their relatively general architecture, coupled with an analysis of the application use cases, several design considerations have been drawn. Thereupon, a SoC functional description is adopted to embed all the necessary sub-systems while allowing partitions and mode switching depending on the system activity and energy availability. Since all components were not available, a simplified implementation is proposed to demonstrate the technology, component and system optimizations that will be performed in the following chapters of this work.

Chapter 3

Clock Reference for Ultra-Low-Power Systems

LOW power systems require precise time and frequency sources to ensure correct operations. While improving the accuracy and stability of the clock, special cares should be placed on the reduction of the overall power to fit ULP applications' constrained budgets.

Since the clock reference remains active during most of the operations' phases of the SoC, it is a major contributor to the whole power consumption. Several solutions have been explored to decrease this impact without degrading the performances. A mixed analog and digital CMOS solution using a leakage-based Ring Oscillator (RO) appears as the best trade-off in terms of power, area and stability. This chapter presents the design of this module from specifications to silicon implementation in 28 nm FD-SOI.

Section 3.1 starts by highlighting the design considerations related to time references in low power digital systems, then focuses on the state-of-the-art CMOS-compatible timers. Consequently, a solution based on the re-locking of an CMOS ULP oscillator is proposed. The timer is based on a leakage-based ring oscillator – presented in Section 3.2 – combined with a digital compensation unit Control Logic Unit (CLU) – described in Section 3.3 – for PVT compensations. The 28 nm FD-SOI silicon implementation and the measurement performed are given in Section 3.4 before offering a summary of the whole system in Section 3.5.

3.1 Ultra-Low Power Clock in IoT nodes

3.1.1 Design considerations: Power, Area and Stability

In duty-cycled IoT devices, some IPs are switched-off during suspend sequences to save power. However, the timer cannot benefit from that technique. It must always remain ON for the entire lifetime utilization of the SoC. Therefore, the power consumption of the timer may easily dominate the overall power budget, even considering heavily duty-cycled high-power components such as the radio or the microprocessor [69, 162]. Hence, the power consumed by the timer¹ is the bottleneck for ULP systems integration.

The physical area of the clock generator is also a key metric. Size improvement reduces the production cost and enables higher volume of production but impacts the power dissipation and increases the variability of the device. PVTs also impact the timer and often appear as a constant offset. Using negligible overhead logic, they can be corrected through calibration [130]. Likewise, in low power systems, circuits operate at low voltages and suffer from voltage variations. For instance, a $\pm 10\%$ relative deviation results at 0.5 V in a ± 50 mV change on the supply voltage.

Ultimately, various random sources of noise, such as thermal noise, induce errors on the timer properties leading to a random spread of the frequency or period and phase [163]. If device synchronization is required between several timers, a random mismatch will appear. Then, timing uncertainties lead to the addition of energy expensive design margins [159]. The several types of variations and impacts on the timer stability and accuracy are proposed in Appendix B. The Allan deviation, which is the standard metric used to compare the frequency stability performance of timers [163], is also introduced.

3.1.2 Evaluation of Existing Low Power Time References and Clock Sources

Starting from common quartz oscillators to exotic chip-scale atomic time references [164], timers are classified using two intrinsic principles of operation (see Figure 3.1).

Linear oscillators, also known as harmonic oscillators, produce a sinusoidal signal. The output of a narrow-band electronic filter is amplified and fed back into the input of the former filter, closing the loop of amplification. On the contrary, relaxation oscillators use a non-linear component like transistors or switching devices to produce a non-sinusoidal output like saw tooth or square waves.

There are numerous ways to implement both types of oscillator to create proper time references. Recent ULP solid-state electronic devices reported into the literature can be sorted into three main categories: crystal-compatible oscillators, Micro Electro-Mechanical System (MEMS) devices and CMOS circuits.

¹A timer is defined as any system that can be used as a clock source, a RTC or a reference for a clock multiplier.

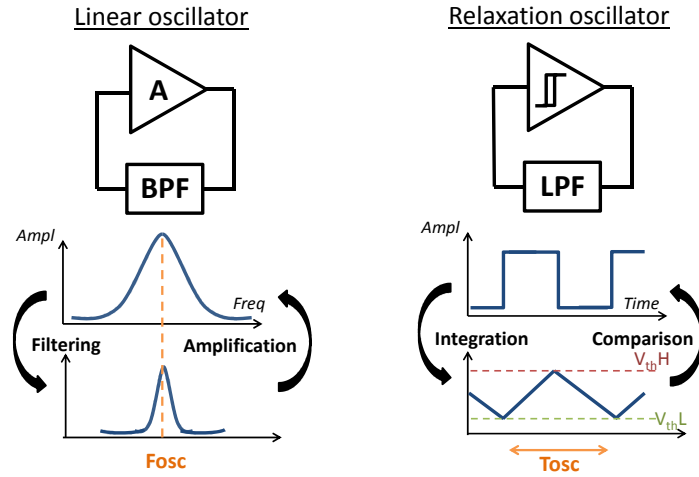


Figure 3.1: Oscillator classification based on their physical type of operation [71].

Crystal oscillators or XO

These solutions are widely known² and used to generate reference clock signals on Printed Circuit Boards (PCBs). They rely on the oscillation of a resonating piezoelectric element called quartz. It is the crystalline form of silicon dioxide SiO_2 . They are classified as linear oscillators with a narrow bandwidth due to the high-level Q-factor of the resonating element. Their oscillating frequency, which ranges from the kHz to the hundreds of MHz is directly related to the crystal physical dimensions. As a consequence, they can be easily one-time calibrated through their dimensions, but the mechanical behavior is limiting the run-time tuning range of a given unit.

With the proper calibration and control current, precision quartz oscillators, especially required in space applications, are achieving an impressive stability of $\sigma_y(\tau) \in [5.10^{-14}, 7.10^{-14}]$ for τ from 1 s to 100 s [165]. For common quartz crystals, lower values are expected ($\sigma_y(\tau) \approx 1.10^{-9}$) yet they are still considered as highly accurate and stable time references. However, despite the fact that such timers are power efficient – [166] reports a consumption of 1.5 nW for a 32.768 kHz crystal oscillator – quartz timers are difficult to integrate because they require off-chip components. Moreover, their start-up time is long, with a hundred of μs in the MHz range to a few seconds around 32 kHz [167].

Micro Electro-Mechanical System (MEMS)

Recently, new advances in micro-machining techniques have led to development of MEMS sensors and timers. Depending on the technology node chosen they can be integrated on chip as it is demonstrated in [168]. Due to the different physical phenomenon involved in their operation, MEMS-resonators present a better accuracy and stability than CMOS circuits. Their power consumption and area is currently still higher than that of other timers, yet new circuits techniques and materials are offering perspectives for future applications [169]. MEMSs could play a trade-off between quartz and CMOS time references [170].

²The year 2018 has even marked the 100th anniversary of the first quartz oscillator [163].

CMOS-compatible circuits

To reduce the power, size, Bill of Material (BoM) and achieve full-scale integration, CMOS-compatible solutions have been developed within the last few years as presented in [159]. Several techniques are available depending on the stability and the degree of integration required. These techniques are extensively described in Section 3.2.1.

Both linear or relaxation oscillators can be designed using CMOS circuits to produce a sinusoidal or a square output, that results in a trade-off between frequency, power and stability. Compared to the previous categories, CMOS-compatible oscillators are the best solution in term of energy efficiency. They also tend to present lower starting times, which is mandatory in compensated or close-looped systems. CMOS oscillators also show the advantage of tuning characteristics. Using variable components, current or voltage sources, their frequency can be trimmed or dynamically adjusted (e.g., voltage-controlled oscillators).

Their utilization is currently limited to their inherent PVTs sensitivity which affects their output frequency. Some component used in the circuit architecture might also appear as a limiting factor. Indeed, passive area-consuming components such as resistor, capacitor or inductor might limit the integration of these oscillators. In this case, a second trade-off between frequency range (or stability) and area might appear.

Summary

A qualitative study of the aforementioned source clock is given in Table 3.1.2. During the conception phase, the metrics and Figures of Merits (FoM) proposed can be defined to bound the design specifications. As a matter of fact, regardless of the clock source selected, the power efficiency tends to be reduced when the targeted operation frequency is increased. Inversely, the starting time is improved for devices operating at higher frequency. In any case, this results on a trade-off which impacts the final system's capabilities.

Metrics	XO	MEMS	CMOS
Type	Linear	Linear	Linear/Relax.
Output	Sinusoidal	Sinusoidal	Sinus./Square
Starting Time	$\mu\text{s} - \text{s}$	ns – ms	ps – ms
Freq. Range	kHz – MHz	Hz – MHz	Hz – GHz
Power	nW – mW	$\mu\text{W} - \text{mW}$	pW – μW
V_{dd} sensitivity	$\sim 1 \text{ ppm/V}$	10 ppm/V	100 ppm/V
Temp. stability	$\sim 0.01 \text{ ppm/}^\circ\text{C}$	$\sim 1 \text{ ppm/}^\circ\text{C}$	$\sim 1 \text{ ppm/}^\circ\text{C}$
Area [mm^2]	~ 1	~ 0.1	~ 0.01
Integration	off-chip	on-chip*	on-chip

* Limited to the technology node MEMS compatibility.

Table 3.1: Qualitative study of electronic oscillators.

3.1.3 Design Targets and Use Case

An ideal clock solution would combine all the benefits from the previous clock-sources: the accuracy of a quartz with the low power consumption of a CMOS-compatible solution. Moreover, as described previously in Section 1.5.1, IoT-oriented devices operate following high duty-cycled operation. Hence, clock-source power consumption must be imperatively reduced during the predominant power-consuming system's suspend sequences.

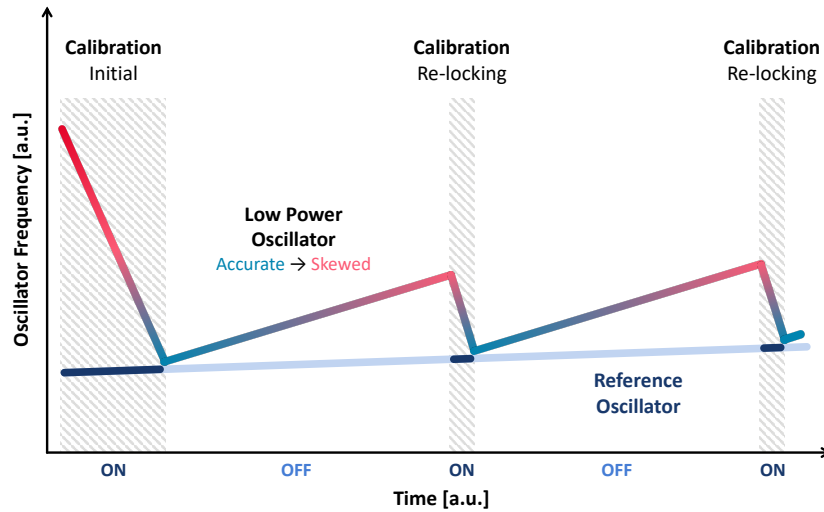


Figure 3.2: Relocking principle to achieve ULP operations.

A solution based on the re-locking of a CMOS ULP oscillator is thus proposed as shown in Figure 3.2. An accurate clock reference – for instance an XO – is activated periodically during the active sequences³ of the SoC to calibrate an ULP CMOS-solution. This second oscillator is then used during the suspend sequences to reduce the overall power.

First, an oscillator OSC_{REF} (in dark blue) is used as a fixed reference. It is selected for its property to accumulate a small amount of error over time. Then, a second oscillator OSC_{ULP} is chosen for its ULP properties but this time it presents inherent deviations over time due to external variations (represented using the red/blue color gradient).

Once in a while OSC_{ULP} is locked on OSC_{REF} to compensate from the accumulated error. The locking rate depends on the clock frequency variability. This way, the effective stability of the ULP oscillator stays within the stability bound of the reference. This technique takes advantage from the duty cycled operations of the system (see Section 1.5.1) and allows energy savings by powering down the reference between locking events. This relocking principle can also be scaled using multiple references as shown in [171] and employing an initial calibration before deployment of the SoC.

The starting time is an important metric for system sizing. Indeed, both clocking systems should present a fast settling time to reduce their power impact. Concerning the quartz, it narrows its selection to higher frequency range. Indeed, the starting time of an XO is inversely proportional to its operating frequency. kHz oscillators are reported with a ~ 100 's ms whereas

³If the long-term stability of the low-power oscillator is not sufficient for a given application, the power-hungry XO can be woken-up more often at the cost of extra power consumption.

MHz XO can reach a ~ 10 's μs starting time [172]. Consequently, during active mode the XO is used to directly clock the system, or it is fed to a clock-multiplier, such as PLLs, if a higher frequency is required. The CMOS-compatible clock source is on the contrary used to clock the digital Always-ON part(s) of the design. It operates in the kHz-range in order to guarantee a minimum system responsiveness as well as a low power consumption. The calibration/response time of this source is also a determining factor to reduce the required ON time of the XO.

Moreover, with an embedded frequency multiplier, once calibrated, the reference oscillator could be disabled as the low power oscillations could be fed to the multiplier and produce the MHz-range frequency required during active modes. This technique will be further explained in Chapter 4.

3.1.4 System Architecture

Figure 3.3 presents the HW module integration of the re-locking scheme based on an external reference. It is reduced to three mandatory building blocks. A CMOS-compatible oscillator (LRO) produces an output clock CLK_{LRO} . The output frequency is digitally controlled through several control bits and an enable signal. A digital solution is selected to ensure fast transitions between ON and OFF periods and re-calibration on the target frequency. It also facilitates the integration and portability across CMOS standard technology nodes.

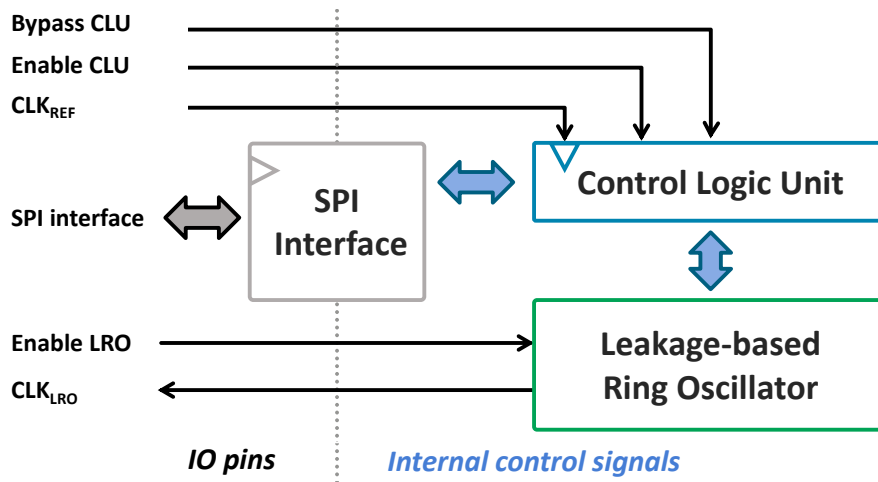


Figure 3.3: Block diagram of the clock generator architecture.

A Control Logic Unit (CLU) embeds a digital compensation circuit to calibrate the oscillator using an external known reference CLK_{REF} . After target configuration, it locks onto a desired frequency independently of the PVTs variations and within the design specifications. Other signals and their utilization will be later described in Section 3.3.

A SPI controller and configuration interface is added to communicate with the IP and ensure initial configuration. This block offers continuous monitoring on the designed modules. It collects data during the run-time operations. In a final system, it is not mandatory and can be removed. However, to be integrated in a complete SoC, a standard peripheral interface (e.g., APB) would be required to control and configure the final IP.

3.2 Clock Generation using a CMOS-compatible Oscillator

This section starts with an evaluation of the existing CMOS-compatible solutions. Then, using this analysis a solution relying on a leakage-based current controlled ring oscillator is detailed. The whole implementation is done in 28 nm FD-SOI

3.2.1 Existing Solutions

Linear Oscillators

Common linear CMOS solutions are using an LC resonator (a.k.a. tank) as the filtering element as shown in Figure 3.1. The oscillation frequency f_{osc} is given by;

$$f_{\text{osc}} \propto \frac{1}{2\pi\sqrt{LC}} \quad (3.1)$$

They currently achieve the best frequency stability of any on-chip timers [173]. The passive LC components and the transistors which provide the feedback amplification can be integrated on chip. However, to generate low output frequencies, a large LC couple is required. Passive elements can be used but increase the whole size of the implementation. Gyrator-based circuits which use active components reduce the area at the price of an increase in power consumption [174, 175]. For instance, [176] reports a $\times 3.9$ power increase (and a phase noise degradation) for the same LC oscillator with passive or active inductor.

Relaxation Oscillators

To overcome the inherent integration limitations and low frequency availability of the aforementioned solutions, relaxation circuits based on switching capacitors can be used. Without adding an impractical number of components and transistors, they often integrate compensation schemes or feedback loops as in [177, 178] to improve long-term stability and frequency stability over PVT variations.

Using similar elements, a linear version is the RC oscillator. They present an improved stability due to their incorporated feedback loops [179]. Offset cancellation techniques are also employed in [180] to reduce the power consumption while maintaining the temperature stability. However, it limits the long-term stability for intervals over 0.5 s.

Ring-Oscillators

From the well-known ULP Ring Oscillators (ROs) - a loop of an odd number of digital inverters which start to oscillate thank to the internal noise – several solutions have been derived. Those timers can be seen as a distributed version of the relaxation oscillator and can easily be integrated using digital standard cells. However, these types of oscillators tend to be highly sensitive to PVT variations as shown in [181–183]. To reach further low power consumption ($\sim \text{nW}$ range), on-chip oscillators using gate leakage currents have later been proposed in [184, 185]. However, the accuracy of such timer is not very well controlled across the fabrication process leading to calibration requirements [171].

Summary of Performances

Table 3.2 gives a qualitative summary of the performances of the CMOS-based oscillators described previously. Even though RO are highly sensitive to PVT variations, they offer room for trade-off between power and area in the context of ULP clock sources. Such solution has been selected for this work and will be described in the next section.

Oscillator type	Linear	Relaxation	Ring
Frequency Range	Hz – kHz	Hz – MHz	kHz – GHz
Power	μW – mW	nW – mW	pW – μW
PVT sensitivity	Small	Medium	Large
Area	Large	Medium	Small
Long-term Stability	Medium	Large	Medium

Table 3.2: CMOS-based oscillators summary of performances

3.2.2 CMOS Current Controlled Ring Oscillator

Ring Oscillator Architecture

From the previous design considerations, a 5 stages current controlled Ring Oscillator architecture has been designed (see Figure 3.4). This kind of topology offers a very low power consumption as extensively explained in [51]. nMOS devices are used to drive a current source I_{leak} to load an output capacitor C_L . Thus, the output frequency is directly given by the current flowing into the capacitor and V_{TH} the threshold voltage of the nMOS transistors, as shown in:

$$f_{\text{CLK}} \propto \frac{I_{\text{leak}}}{C_L \cdot V_{\text{TH}}} \quad (3.2)$$

Contrary to a standard approach using CMOS standard cell inverters, since only a PDN is used, no current spike is created when the inverter is switching (temporarily short-circuited), avoiding energy loss. C_L is the sum of the gate capacitance of the stage to load and the inherent parasitic capacitance of the node. This solution avoids the addition of bulky components yet, it requires specific effort during layout design to be properly sized.

A matching circuit based on standard cells is attached to the last transistor of the ring. It helps to maintain the dynamic swing of the signal between V_{dd} and gnd to drive the following stage composed of standard cells. It is designed for a reduced area impact, at the cost of extra short-circuits resulting in return to an increase of the PVT sensitivity. In [184], an alternative solution is to use a small pMOS transistor also known as keeper. This pull-up transistor must source enough current to overpower the nMOS logic stack leakage current in the slow-pFET/fast-nFET (SF) corner, while in the symmetric FS corner the keeper must be weak enough so that the nMOS can pull the dynamic node quickly enough to avoid excessive slowdown of the output frequency. These contradictory specifications result in maximum gate width and so area overhead.

Lastly, an AND gate is added into the feedback loop of the RO to act as an enable whereas a final output buffer ensures the correct driving of the output stages. The whole design is oper-

ating at a fixed $V_{dd} = 0.5\text{ V}$.

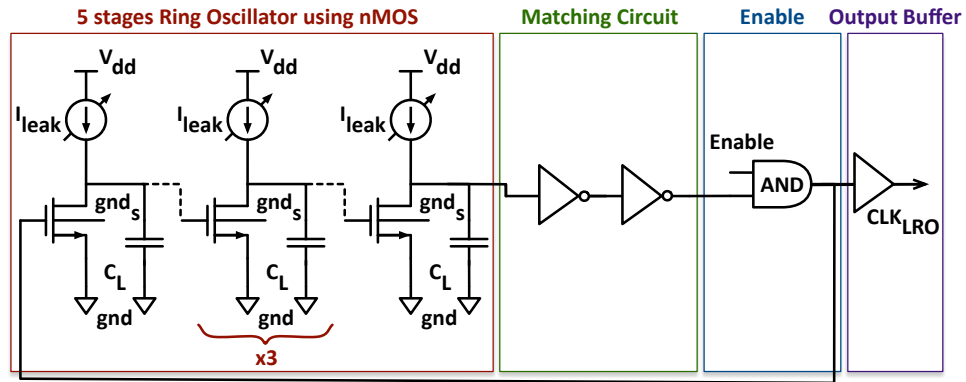


Figure 3.4: Schematic of the current controlled ring oscillator.

Digitally Controlled Leakage Sources

Gate leakage based current sources are implemented to produce I_{leak} with a minimum power consumption. Moreover, for frequency tuning and integration, a digital control is adopted. Combined with a digital compensation scheme – presented in Section 3.3 – it helps compensate the PVT sensitivity of the whole Leakage-based Ring Oscillator (LRO).

As presented in Figure 3.5, three control schemes of the gate leakage are possible. First, drain, source and bulk contacts can be tied up to the ground. It results in a reverse gate leakage current due to the discharge of the output capacitive node which is wasted due to the ground current path. A second alternative is to keep drain, source and bulk as open nodes yet, the capacitance of the gate becomes unpredictable affecting the total time constant of the device. The last option consists of applying an extra bias voltage to mitigate the previous phenomenon but requires extra external power.

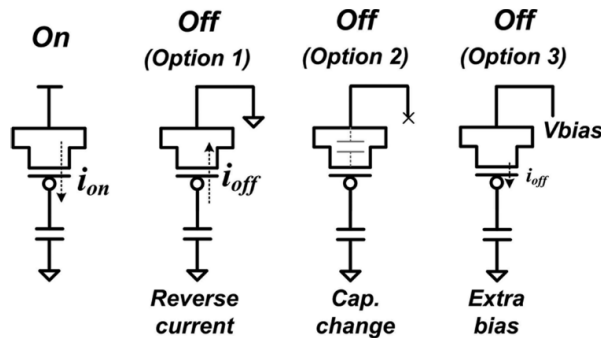


Figure 3.5: Current control schemes of a pMOS transistor for ON and OFF modes [185].

With these schemes, only the gate and junction currents are used to load the output nodes which results at ULV in small loading currents compared to the utilization of channel leakage [185]. Then, to produce an output frequency in the kHz range it would require to

greatly augment the number of sources or the effective size of the devices, thus increasing the system area.

Following these conclusions and adapted from [185], an all-digital control technique using the channel leakage is proposed as shown in Figure 3.6. This time, in ON mode, the source is connected to the supply voltage whereas the drain is tied to the load. Hence, all device leakage (channel, gate, junction) flow into the output load. In OFF mode, the source is connected to the ground. As the gate and bulk nodes remain connected to the supply voltage, the transistor is negatively biased. The channel leakage is negligible and only a small portion of current flows into the output load.

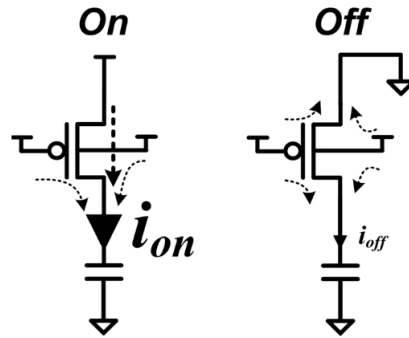


Figure 3.6: Current modulation of a blocked pMOS transistor using a source control scheme for ON and OFF modes [185].

In this circuit, the channel leakage is controlled by acting on the source through switching between supply and ground. This operation can be performed using a simple inverter cell, the input of the cell working as the digital input control of the current source.

Consequently, the digitally controlled leakage-based current sources have been implemented as shown in Figure 3.7. A set of three different arrays of transistors is used to generate the output current I_{leak} . An explanation of the source selection to produce a target output current is given in Figure 3.8.

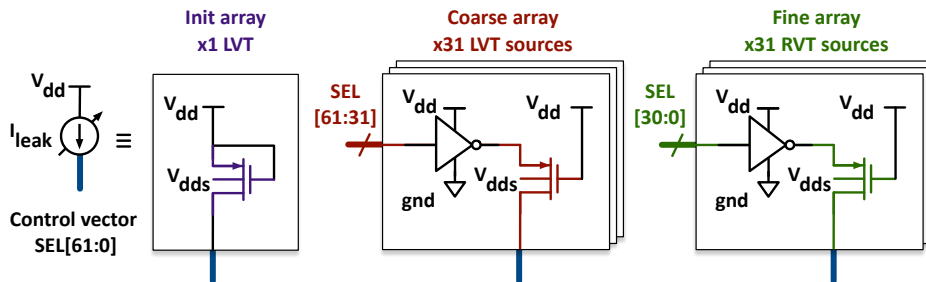


Figure 3.7: Schematic of the digitally controlled leakage sources.

By selecting N_x number of LVT pMOS and N_y number of RVT pMOS, a certain amount of current is produced, leading to a modulation of the whole LRO output frequency. The array based on LVT pMOS produces coarse steps whereas RVT pMOS are used for fine steps. Besides,

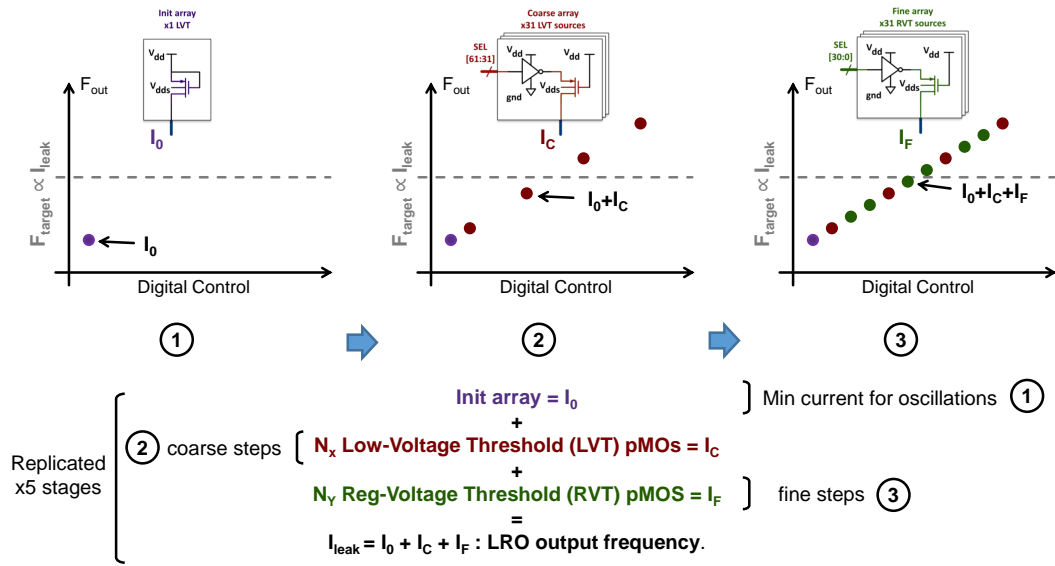


Figure 3.8: Current sources selection scheme to reach a given target frequency.

the Init array is added to guarantee oscillations' starting when no other source is activated and the RO is enabled. The 5 current sources of the 5 RO's stages are similar meaning Figure 3.7 is replicated 5 times. The control vector SEL is common to the 5 current sources.

To ensure high linearity when adding one more digitally controlled leakage source, a thermometer approach is chosen. As a consequence, two 31-bit input vectors are used to directly address the 2x31 sources. However, the CLU uses two 5-bit control vectors – L_c and R_c – to select the number of desired leakage sources (see Figure 3.21). Thus, a conversion stage converts the binary vector CTRL to a thermometer code SEL dispatched to the leakage sources. It offers a transparent interface to the user. An overview of two control modes and the LRO \leftrightarrow CLU interface is given in Table 3.3.

Control Vector	L_c – LVT array					R_c – RVT array					CTRL
	L_4	L_3	L_2	L_1	L_0	R_4	R_3	R_2	R_1	R_0	
Sources activated	0	0	0	0	0	0	0	0	0	0	0
Sources stopped	1	1	1	1	1	1	1	1	1	1	1023

Table 3.3: LRO \leftrightarrow CLU Interface

3.2.3 Current Sources Sizing: Process Variations Compensation

For PVT variations compensation, the coarse and fine arrays must offer adequate current to always ensure the operation at a target frequency for a given accuracy. For instance, the Bluetooth specification requires a ± 250 ppm accuracy at 32.768 kHz in standby mode [186]⁴.

⁴In nominal operation a ± 20 ppm for a 20 MHz clock reference to ensure correct channel spacing and thus communication.

Reaching such low stability value requires to decrease the minimum unit of leakage current available (downsizing the RVT sources) while increasing their number to ensure continuous steps with the LVT sources. Moreover, the number of LVT sources should also be increased to cover the process variation, and especially offer enough current in a slow corner or low temperature. In the end, the significant number of leakage sources increase the whole power consumption and impact the minimum unit achievable by adding leakage noise, thus limiting the ppm target reachable. Therefore, for this first implementation the ± 250 ppm has been relaxed.

A 2000 ppm uncertainty target is set for a typical corner corresponding to a direct trade-off between the number of leakage sources required, their size and their power consumption⁵. Assuming a 32.768 kHz frequency, Table 3.4 summarizes the resulting frequency and period variations resulting from the target accuracy.

Specification	Value
Nominal frequency [Hz]	32768
Targeted ppm	2000
Frequency variation $\pm \Delta f$ [Hz]	65.536
Minimum frequency [Hz]	32702.464
Maximum frequency [Hz]	32833.536
Period Variation [ns]	122.1

Table 3.4: Oscillator frequency specifications in TT.

The simulations were extended to verify the correct operations around 5 corner cases: Fast-Fast (FF), Fast-Slow (FS), Slow-Fast (SF), Slow-Slow (SS) and Typical-Typical (TT). The simulated output frequency of the LRO is reported in Figure 3.9 for the whole configuration range (the binary input code CTRL is converted in decimal).

Reaching the given target frequency requires a corresponding input control code in each corner cases. As a consequence, the Coarse array is used to compensate the large variation from process – in SS all sources are almost activated, in FF all sources are deactivated – and the Fine array handles small variations.

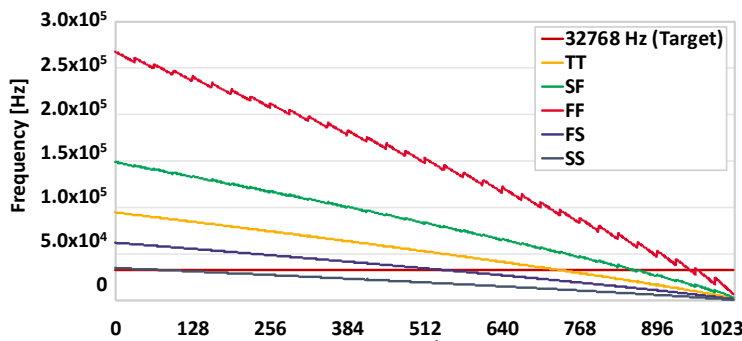


Figure 3.9: Simulated LRO output frequency over the whole configuration codes according to the process corner. Results obtained at 0.5 V/25 °C.

⁵This target is also directly related to the quality of the leakage models available in 28 nm FD-SOI

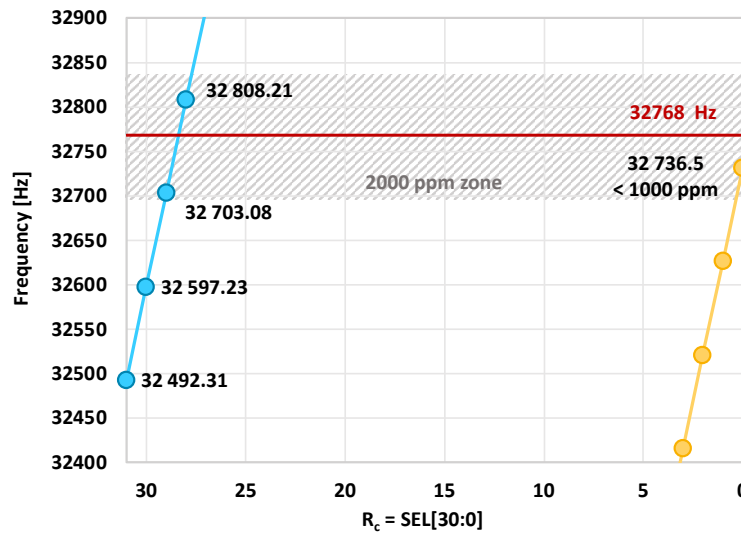


Figure 3.10: Evaluation of the LRO accuracy. Results obtained at 0.5 V/25 °C and a TT corner.

The oscillator evaluation accuracy around the target frequency is reported in Figure 3.10 for a TT corner operating a 0.5 V/25 °C. Only the R_c control vector is used, and each line is drawn at iso- L_c . The output frequency obtained within the 2000 ppm target zone confirms the oscillator compensation capabilities. Lastly, Table 3.5 gives the parameters of the devices used.

Stage	RO	Init	Coarse	Fine
Device type	RVT nMOS	LVT pMOS	LVT pMOS	RVT pMOS
Transistor size: W	80 nm	80 nm	280 nm	270 nm
L	50 nm	30 nm	30 nm	30 nm

Table 3.5: Design parameters of the devices used.

3.2.4 Leakage Sources Matching and Variability

Due to the high number of pMOS leakage sources, special cares are taken on the physical implementation to improve the linearity of the leakage current obtained between two sources. Fundamental layout techniques including the use of dummy cells and common-centroid for matching are mandatory [52]. However, in Silicon On Insulator (SOI) technology the silicon film thickness variations highly impact the characteristics of the MOSFETs.

Variation of the silicon film and impact on the device output current

For large areas, CMOS-technology experimental matching results exhibit a dependence on the transistor pairs layout [52]. Therefore, matching models in Computer Aided Design (CAD) highly depend on the device area (W and L). When using FD-SOI technology, other sources of variability are encountered (see Figure 3.11).

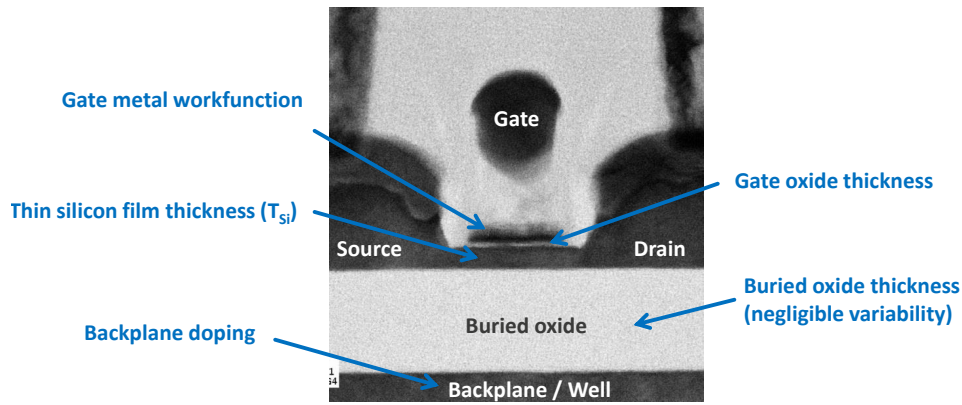


Figure 3.11: Main sources of variability for a FD-SOI transistor.

In particular, the thin silicon film thickness T_{Si} presents variations as shown in Figure 3.12a. These small perturbations on the surface define the roughness of the silicon film. As shown in Figure 3.12b, it affects the current delivered by the MOSFET.

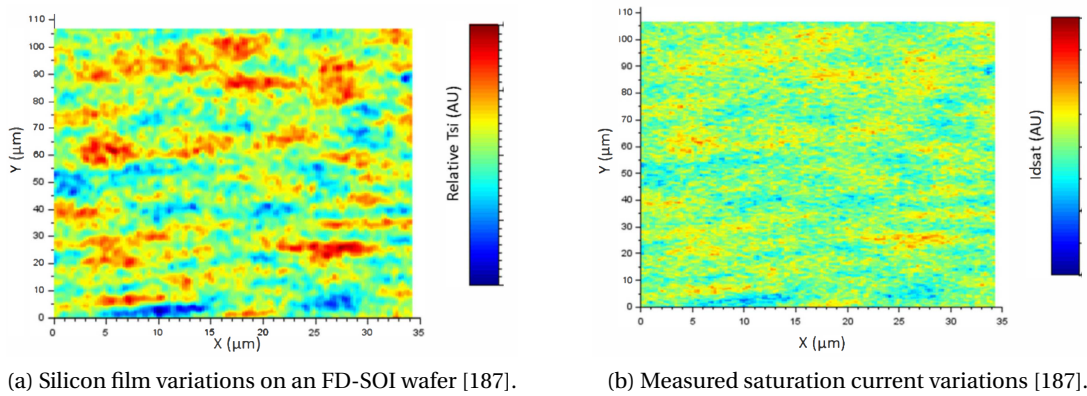


Figure 3.12: Silicon thickness evaluation of a wafer used in 28 nm FD-SOI technology [187].

Consequently, by exploiting the Power Spectral Density (PSD) of these variations, a complete statistical description of the surface is obtained⁶. As shown in Figure 3.13, it shows significant fluctuations with correlation lengths in the $1\mu m$ to $10\mu m$ with harmonics in the nm range (i.e., transistor scale).

⁶The PSD is the distribution of power into frequency components composing a signal. Here the T_{Si} variations.

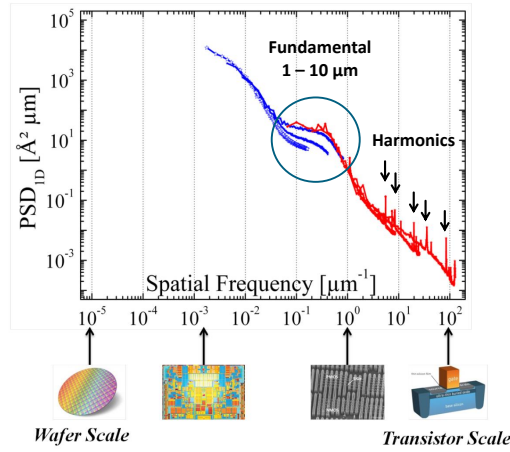


Figure 3.13: PSD functions describing the surface topological variations of an SOI wafer [188].

Resulting layout of the leakage current sources

In Figure 3.14 is given the resulting layout of the fine array composed of 31 unit leakage current structures, each containing 5 RVT pMOSs devices notes A,B,C,D and E for the 5 ROs' stages.

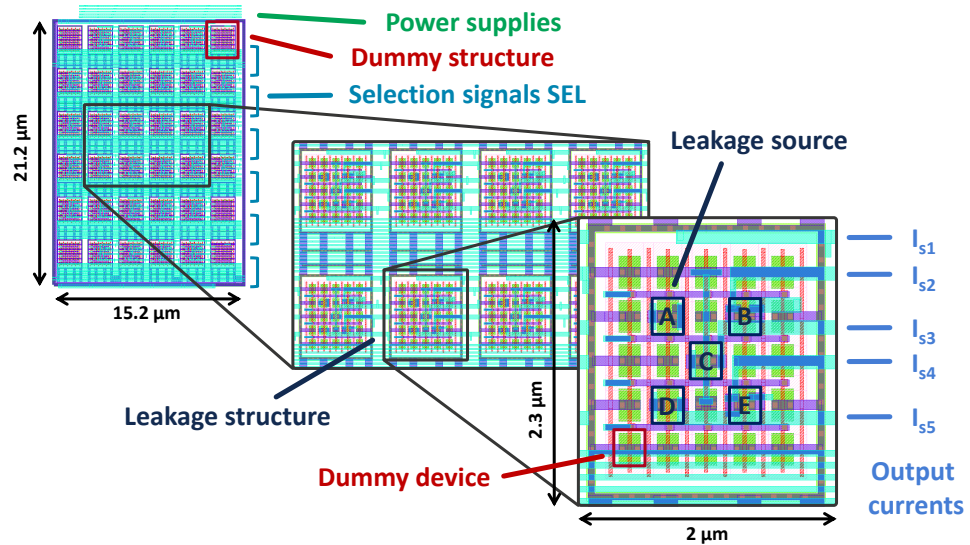


Figure 3.14: Layout of the fine array composed of 31 leakage structures, each containing 5 RVT pMOSs devices for the 5 stages of the ROs.

A common-centroid structure associated with dummy devices is adopted. Guard rings isolate the devices from external noise. These unitary macros are then integrated with a global array of 31 modules and 5 dummies. The SEL signals ensure the digital control while the output currents I_{sN} are connected to the stage N of the LRO. All power and biasing supplies are distributed within the array.

However, this whole implementation results in a size ($21.2 \mu\text{m} \times 15.2 \mu\text{m}$) within the order of magnitude of the T_{Si} variations. At the time of this study, the matching models were not

considering the distances between devices. Therefore, the silicon film oxide variations effect could not be evaluated properly and thus was difficult to be mitigated. However, to improve the linearity between two leakage sources, since enough sources are available, a random placement has been defined as shown in Figure 3.15. This solution easily ensures a cancellation of the unit current cell systematic mismatch effects.

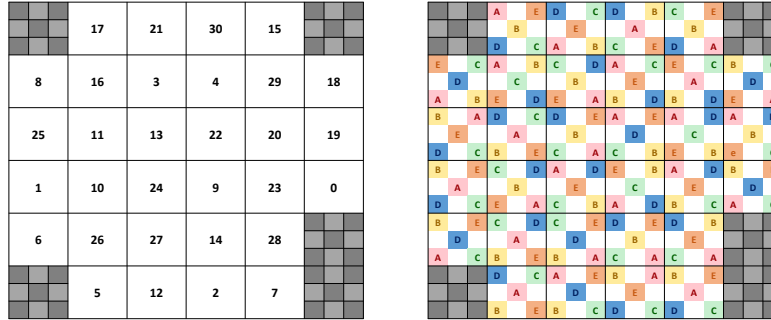


Figure 3.15: Chosen random distribution of the 31 leakage structure and associated leakage sources.

3.2.5 Complete Physical Implementation

The whole LRO physical implementation is given in Figure 3.16. The LVT and RVT leakage sources arrays are controlled using selection buffers. The init array which ensures oscillation at start up and the RO are also connected to the output of the arrays. A final area $1635 \mu\text{m}^2$ is obtained. Moreover, the full system is integrated into a SoC similar to [189] and fabricated in 28 nm FD-SOI technology. The compensation logic, which is purely digital, is synthesized and placed separately (see Section 3.3).

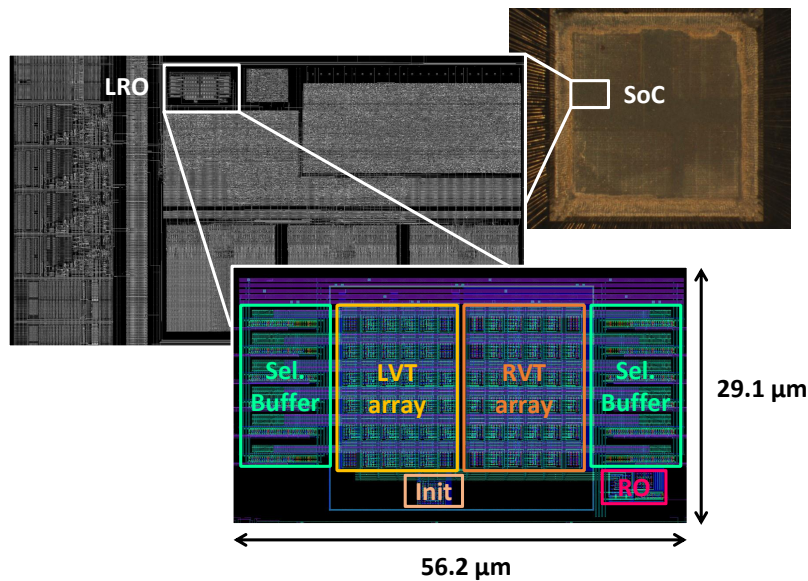


Figure 3.16: Testchip micrograph and view of LRO implementation. The LRO area is $1635 \mu\text{m}^2$.

3.3 Digital Locking Scheme

3.3.1 Evaluation of Various Digital Compensations

Due to the leakage sources utilization, the proposed LRO is power efficient yet highly sensitive to PVT variations. Calibration methods are required to improve the absolute frequency stability of the device. Hence, two approaches can be developed: reducing the sensitivity through analog circuit techniques or compensating the variations using an external digital feedback loop. In advance technology nodes, high noise immunity can be achieved using simple (i.e., negligible power) digital logic, whereas decreasing the sensitivity of the RO implies a more complex analog part. Therefore, this second solution has been chosen and this section explores several digital compensation techniques for re-calibration.

Successive Approximation Register

To achieve absolute frequency accuracy and improve the frequency stability with temperature changes, [184] and [171] propose a periodic re-calibration using a Successive Approximation Register (SAR) algorithm (see Figure 3.3.1).

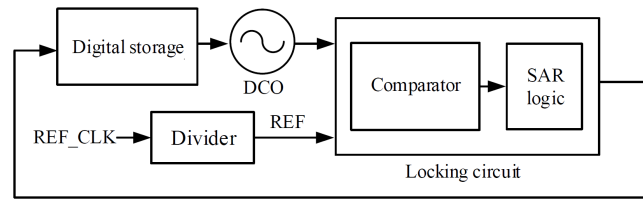


Figure 3.17: DCO architecture using a SAR based locking circuit [171, 184].

This technique, also used in some type of ADC [52], is a very smart and efficient method to calibrate digital tuning values. By construction, the maximum number of cycles required to calibrate the system is defined by the number of bits used. Moreover, all calibrations are equivalent to the initial calibration. In practice, it leads to a simple and very fast digital implementation, yet, a relaxing time is required for a correct settling time of the current sources.

However, as shown in Figure 3.18, this type of algorithm tends to have high jitter sensitivity degrading the overall accuracy (assuming a jitter over 1 LSB). Indeed, one false decision in one “bit-evaluation” cycle cannot be corrected during the next cycle. Actually, this characteristic is one reason for the speed and efficiency of the SAR. In typical ADC applications of the SAR implementation this characteristic is not observed as a drawback since the “bit-evaluation” can guarantee a correct output and no false decisions are made. Since the ULP oscillators tend to have high jitter the “bit evaluation” cannot guarantee a correct output in every cycle leading to false decision. This can significantly reduce the tuning accuracy of the SAR method for ULP oscillators. Increasing the sampling time and averaging over multiple sampling cycle could compensate the impact of the jitter, yet it would also increase the power consumed which is the main advantage of SARs.

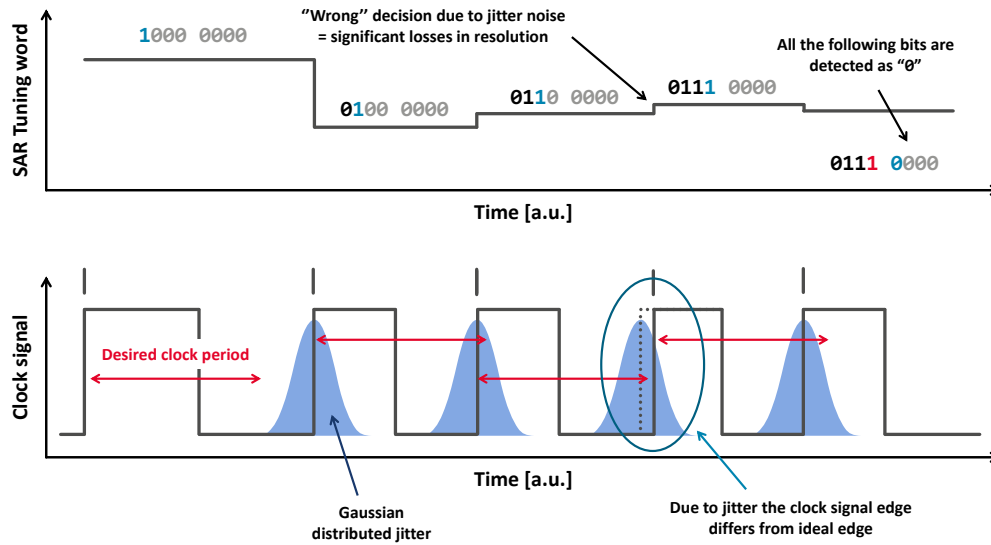


Figure 3.18: Evaluation of the tuning word in a SAR relocking scheme and impact of jitter [190].

Multi-level Bang-Bang

Other calibration techniques are proposed to compensate false "bit evaluations" although it might still affect your tuning accuracy. With a bangbang calibration scheme the tuning accuracy with a certain amount of false "bit-evaluation" can be improved by lowering the gain but it also leads to a longer settling time since the loop bandwidth is reduced.

To achieve high tuning accuracy and acceptable settling time at the same time, [191] implements a multilevel bangbang calibration scheme where the error signal from the digital phase detector is segmented into three levels in a nonlinear way. Therefore, large errors lead to a large feedback gain and a fast settling time whereas small errors (after settling of the tuning word) lead to a small feedback gain and a high tuning accuracy. This behavior can also be achieved through an IIR-filter implementation with an adaptive gain or bandwidth control but the multilevel bangbang calibration scheme leads to low gate count digital implementation allowing a very low power consumption.

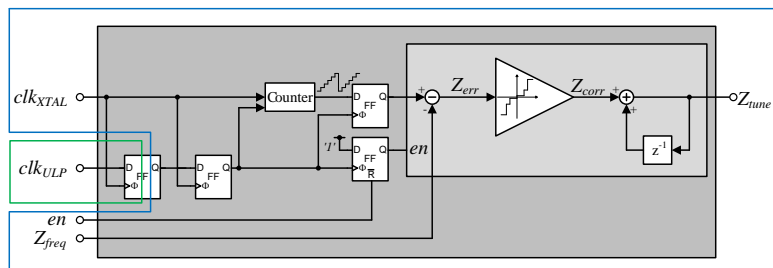


Figure 3.19: Multi-level bangbang calibration for high jitter tolerance and fast relocking capability [190].

Integral Control

The multi-level bangbang calibration can be extended by using an Integral control as shown in Figure 3.20. This correction integrates the control deviation ϵ as a function of time with the following transfer function:

$$C_i(z) = \frac{y^i(z)}{\epsilon(z)} = K_i \cdot \frac{1}{1 - z^{-1}} \quad (3.3)$$

With proper sizing, a stable high-speed feed-back loop is guaranteed in any case. The static error is non equal to zero yet, it can be compensated using a PI correction. This technique also shows high jitter tolerance, a low complexity and a fast relocking ability still limited by its bandpass.

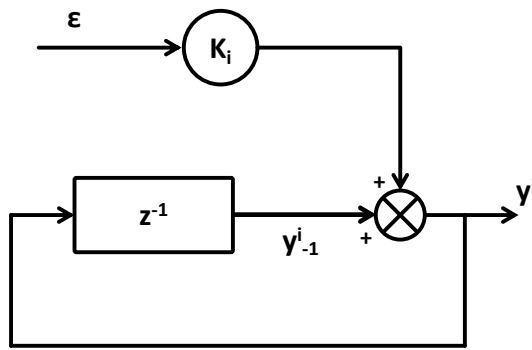


Figure 3.20: Structural diagram of an Integral corrector.

Summary

A summary of the aforementioned techniques is given in Table 3.3.1. Since circuits using SAR tend to have high jitter sensitivity, a Proportional Integral (PI) corrector solution has been selected. It benefits from the speed of the integral corrector thanks to its high bandpass and ensures a fast settling time of the compensation loop, a good accuracy and a simple implementation.

	Integral corrector	Multi-level bangbang	SAR
Constraint	Bandpass Depending on bandpass	Level number	Precision
Precision		1/2 LSB	1 LSB
Jitter insensitivity	✓	✓	✗
Tunability	✓	✓	✗
Speed	++	+	+++
Complexity	+++	++	+

Table 3.6: Qualitative study of digital compensations.

3.3.2 Proportional-Integral Corrector Analysis and Optimizations

Quartz Frequency Selection

Jitter tolerance and accuracy are required for a digital calibration method. It can be achieved with a low bandwidth of the calibration loop which leads to a large settling time. On the contrary, the system must observe a fast settling time to ensure a minimum power consumption. Indeed, during calibration, the XO is activated which consumes a lot of energy. Moreover, a XO wake-up time t_w depends on its quality factor Q . When Q increases ($f_{out} \nearrow$), $t_w \searrow$ decreases. An oscillator paired with a \sim MHz (resp. \sim kHz) crystal resonator would typically start up after a delay of a few ms (resp. \sim s). As long as the settling time of the digital calibration is negligible compared to the XO settling time, the digital calibration remains acceptable. For these reasons a 2^{22} Hz clock is used as a trade-off between start-up time, power and calibration time (see Section 3.4.2).

Control Logic Unit

Figure 3.21 describes the Control Logic Unit implementation. As a first step, a synchronization and counting stage is designed to produce the digital word $Z_{LRO/REF} = f_{REF}/f_{LRO}$. This frequency ratio is then fed to the compensation stage where it is compared with $Z_{target} = 2^{22}/2^{15} = 128$ to produce the error signal ϵ . Subsequently, ϵ is used into the PI stage presenting a programmable integral gain $K_i (\leq 0)$ to generate the binary control signal CTRL. However, to maximize the accuracy of the system, two modules have been added.

First, a low saturation module avoids oscillations due to the discrete error. On the one side, when a large frequency error is detected, a large gain ensures fast convergence. On the other side, when $\epsilon \rightarrow 0$, a smaller feedback gain is required to set the tuning word and target the correct output frequency. By detecting this low saturation, a second error signal ϵ' is produced to automatically tune the effective gain of the corrector to achieve accuracy and high jitter tolerance. In this implementation K_c is set to 0.1.

Secondly, an anti-windup stage avoids overshooting and continuous increasing of the accumulated error. Indeed, if ϵ remains positive (resp. negative) for a certain period of time, the control signal saturates at a min. value (resp. a max value) due to the limited number of leakage sources available. Yet, if the error stays positive (or negative) after saturation, the integrator continues to accumulate an error that will be difficult to compensate in a reasonable amount of time. This can lead to a significant error on the output or system instability. Therefore, a loop is added that uses a third error signal ϵ'' defined as the difference between the PI output and the effective output CTRL.

Finally, a conversion stage converts the binary code into a thermometer code to address the LVT and RVT array (see Section 3.2). Anyhow, with the bypass signal the user can directly send – via SPI – the CTRL vector and activate the leakage sources.

The CLU is using CLK_{REF} as the clocking element. As the error is updated every period of CLK_{LRO} , this solution helps relaxing the design timing constraints by allowing the K_i multiplication to be done over multiple CLK_{REF} periods. Moreover, when the relocking is done, disabling the reference clock leads to remove the dynamic power of the digital block and the XO power.

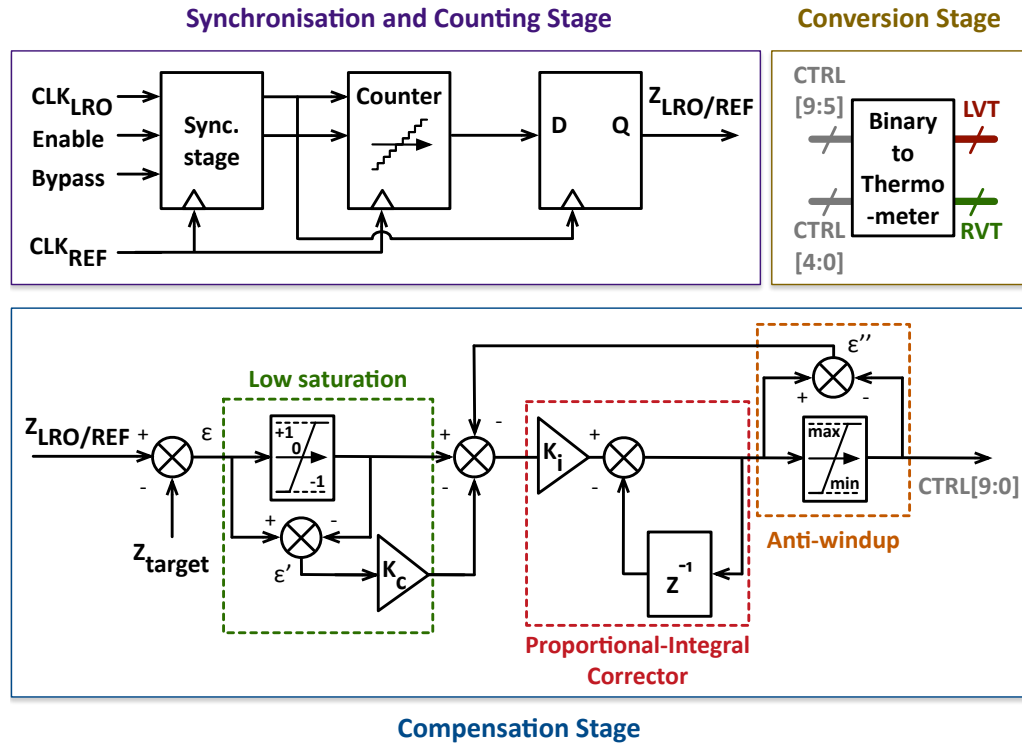


Figure 3.21: Control Logic Unit block diagram.

Stability Analysis

The previous CLU is implemented in VHDL. The convergence and the stability of the system is then evaluated according to several corrector gain values with the low saturation module deactivated. In Figure 3.22a is given the stability of the compensation stage for a starting code CTRL set to the maximum value (i.e., 1023). Here a 100 MHz frequency target is defined using a 4 MHz clock reference. For a correction factor between 0 and 1, no oscillations are observed on the output code, leading to a fixed output period. The fastest convergence is obtained for $K_c = -1$ with $16\mu s$. A similar analysis is performed with an initial control code set to the minimum value (CTRL = 0). This time no oscillations are observed on the output period. The convergence is also faster with $8\mu s$ for $K_c = -1$.

Consequently, to improve the convergence time of the CLU, the low-saturation module is activated as previously described (see Figure 3.23). Starting from the highest configuration with a gain $K_c = -5$, a $2\mu s$ convergence time is now obtained without any output oscillation. It improves the system by a factor $\times 8$. From the lowest configuration, the convergence time is less than $1\mu s$.

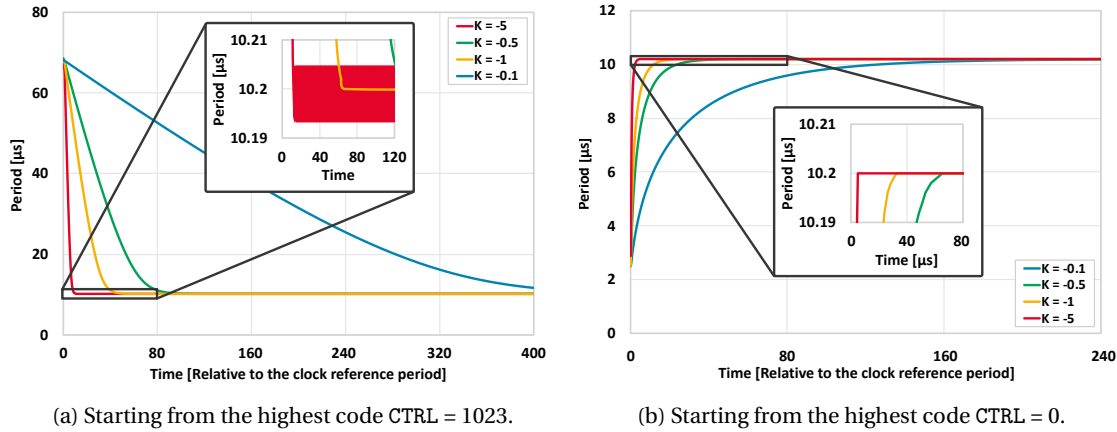


Figure 3.22: Stability analysis of the compensation stage according to the proportional coefficient with low saturation stage deactivated.

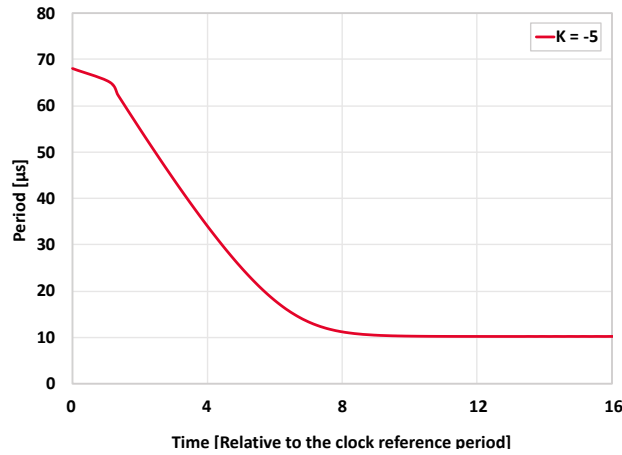


Figure 3.23: Stability analysis of the compensation stage with a corrector gain $K_c = -5$ and the low saturation stage activated.

Finally, the whole implementation works similarly to the bangbang implementation of [191]. Here levels are programmable through the PI correction gain. The non-linear evolution of K_c results in large feedback gain for large frequency error leading to a fast settling time and a very small feedback gain once the frequency tuning word has settled. Lastly, the programmable value of K_c is defined on 4 bits (and 1 bit for the sign). The positive value also avoids bit shifting operations, facilitating the implementation and thus requiring less hardware.

3.3.3 SPI Interface and Controls

For configuration and testing of the whole design, the LRO and CLU are modified, allowing access to the internal control vector as shown on Figure 3.24. For instance, a bypass signal has been added to directly control the input of the LRO. Moreover, an SPI slave interface from OpenCores [192] has been selected and instantiated.

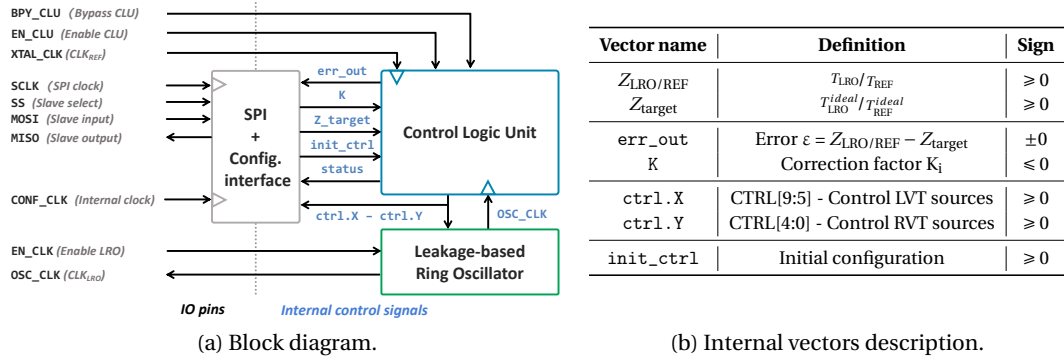


Figure 3.24: Block diagram of the complete system including the internal control signals and pin interfaces on the left. Description of the CLU configuration vectors on the right.

Using an external FPGA (see Appendix E) the configuration registers are programmed using the SPI configuration interface. A Synopsys DesignWare synchronization interface is also added to synchronize the various clock domains. The configuration protocol is shown in Figure 3.25. Two modes are available:

- **Instrumentation:** The BPV_CLU signal is set to high and the control bits for the LRO are sent through the SPI payload. The head flag is set to conf_init_ctrl to configure the control bits or conf_idle to read only.
- **Digital compensation active:** The CLU internal registers are configured through the SPI interface using the corresponding header code. The configuration must be done before the calibration. The EN_CLK signal enables the LRO when necessary as well as the EN_DCO signal enables the digital control logic. In this mode, the user should ensure the correct operation of the reference clock XTAL_CLK.

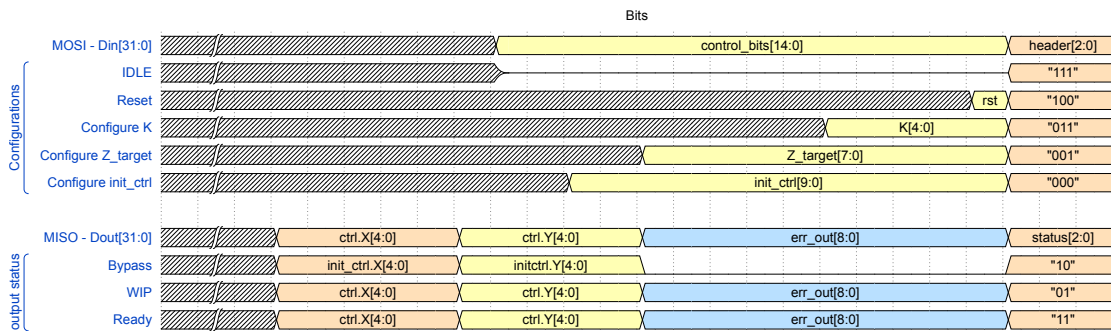


Figure 3.25: SPI and communication interface protocol with the system.

In both modes, the configuration interface clock, CONF_CLK must be defined as well as the SPI signals to communicate with the system. The IDLE configuration header is used to run the system even though no observations are performed. The output payload of the SPI bus delivers information on the control vectors X and Y, the error computed in the digital feedback loop and the status of the system. The system delivers the following information: bypassed, operating a computation (WIP) or ready.

3.4 Circuit Implementation and Measurements

The whole ULP oscillator and compensation unit is fabricated in 28 nm FD-SOI technology. A 32.768 kHz target is set and provides reference for a frequency multiplier or enables direct clocking of the A.ON domain of an ARM-based microprocessor during deep sleep operations. This section describes the results obtained on 42 packaged dice and measured using a custom development board as shown in Appendix E.

3.4.1 LRO Free Oscillations and Jitter

The output frequency of the oscillator according to the input control code is measured for 42 dice and reported in Figure 3.26. The grey area represents the whole set of measured values. For each sample the target frequency (dashed red line) can be reached showing proper process compensation capabilities. Moreover, the measured mean frequency is compared to the simulated value in TT, showing matching between CAD models and silicon results. From these results, 10 dice were randomly selected for further analysis.

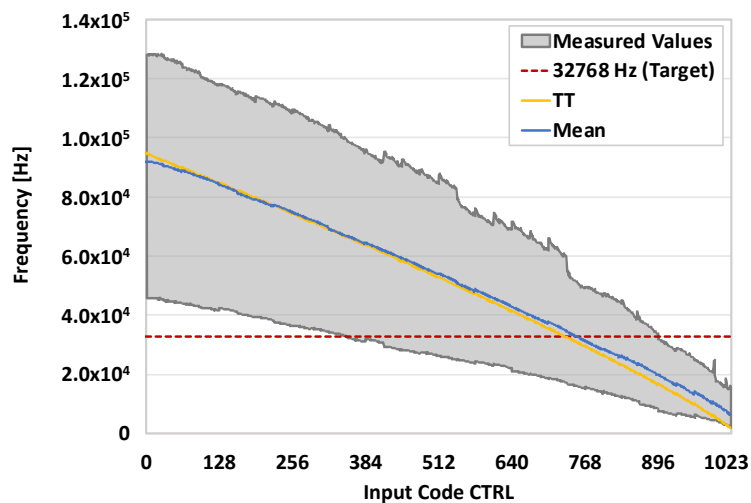
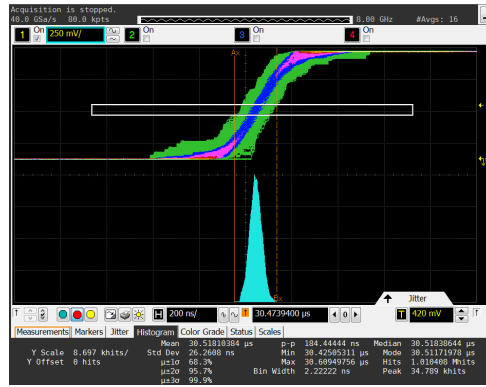
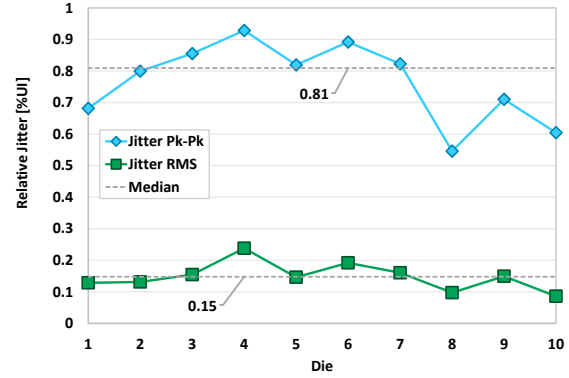


Figure 3.26: Comparison of simulated and measured LRO frequency ranges according to the input code for 42 dice at 0.5 V/25 °C.

For digital SoC clocking the relative peak-to-peak period jitter directly corresponds to a frequency penalty or extra margins in the clocked digital logic. Therefore, in order to use our clock in digital circuits, the peak-to-peak and Root Mean Square (RMS) jitter for 10 dice at 0.5 V/25 °C is reported in Figure 3.27. After an initial calibration, the output frequency of the tested die is centered around 32.768 kHz and the measurement done with a constant input code. The 10 dice median peak-to-peak jitter value is 247 ns and the RMS is 45 ns, i.e., 0.81% and 0.15% UI respectively.



(a) Jitter measurement.



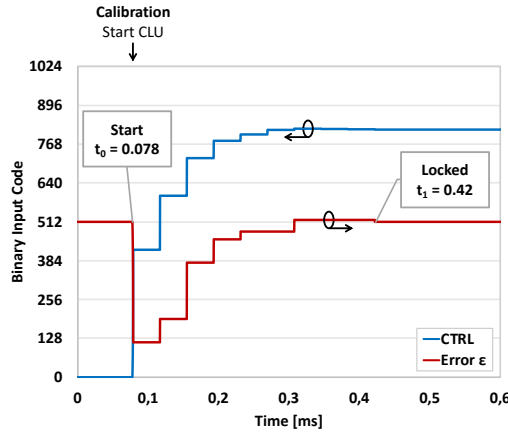
(b) Jitter measured on 10 dice.

Figure 3.27: Measured peak-to-peak and RMS jitter for 10 dice around 32.768 kHz at 0.5 V/25 °C.

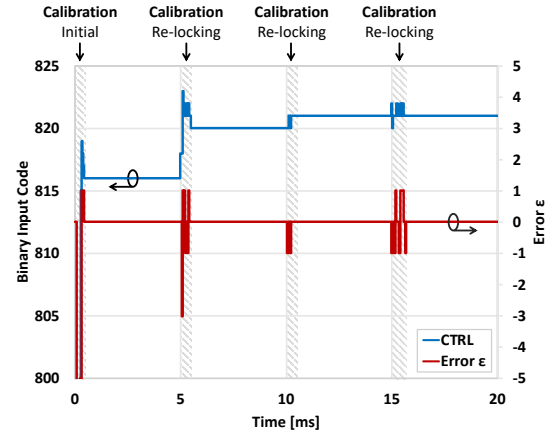
3.4.2 Digital Compensation Evaluation

Relocking Scheme Utilization

A measurement of a typical locking cycle is shown in Figure 3.28a. In blue, is represented the correction word CTRL controlling the leakage sources and in red, the error ϵ generated from the digital counter in the CLU. Starting from the fastest configuration (all sources activated), the system is able to lock in 0.342 ms showing a fast settling time. In Figure 3.28b, is applied the re-locking scheme presented in Figure 3.2.



(a) Typical initial calibration.



(b) Locking scheme applied every 5 ms.

Figure 3.28: Evaluation of the locking scheme at 0.5 V/25 °C.

The measurement is started with a default frequency tuning word (CTRL = 0) and a first initial calibration is done similarly to Figure 3.28a. Then, every 5 ms, periodical relocking is operated. The error evolution demonstrates the different feedback levels and fast locking capability. Between calibration loops, the CLU is disabled. A small frequency error occurs due to the inherent PVT sensitivity of the oscillator. Less than 1 ms is required to re-settle the frequency tuning word.

In this first implementation, the calibration is stopped when the error code reaches 0. However, it does not guarantee the optimal convergence. Since $f_{\text{REF}} = 128f_{\text{LRO}}$, an error of $1/128^{\text{th}}$ on f_{LRO} can be observed for a measure done only over one f_{LRO} cycle. Therefore, for an error less than 2000 ppm, the measured need to be done for at least 2000 f_{REF} cycles (i.e., 16 f_{REF} cycles which 0.5 ms)

Consequently, accuracy improvement can be obtained by implementing a counter that increment if the error is maintained at 0. When the counter reaches a value corresponding to the stability target the calibration is stopped.

Absolute Output Frequency

The absolute output frequency of the LRO when the system is locked is now reported for 10 dice in Figure 3.29 for nominal operating conditions (0.5 V/25 °C). The relative variation is obtained within the boundary defined during the design phase of the oscillator (see Section 3.2.3). Less than 2000 ppm variations are obtained with values ranging from 2 ppm up to 600 ppm. A median output frequency of 32 768.3687 Hz is calculated leading to a relative variation of 11 ppm.

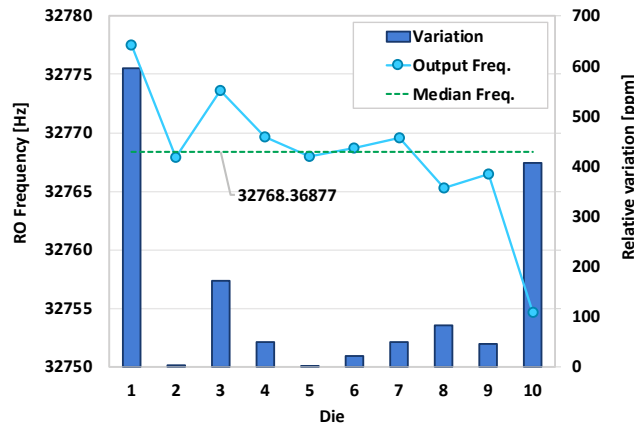


Figure 3.29: Evaluation of the output frequency when relocked, and corresponding relative error for 10 dice at 0.5 V/25 °C.

Voltage and Temperature Stability

The voltage stability of the CLU from 0.46 V to 0.54 V (i.e., $V_{\text{dd}} \pm 8\%$) is given in Figure 3.30a. Measured values are obtained for 10 dice when the system has locked. Across the whole voltage range, an average median frequency of 32 768.2870 Hz is measured leading to a ~ 90 ppm/V voltage stability. In addition, the temperature stability is characterized from 0 °C to 50 °C and results are reported in Figure 3.30b. An average median frequency of 32 771.0670 Hz is obtained leading to ~ 1.9 ppm/°C temperature stability.

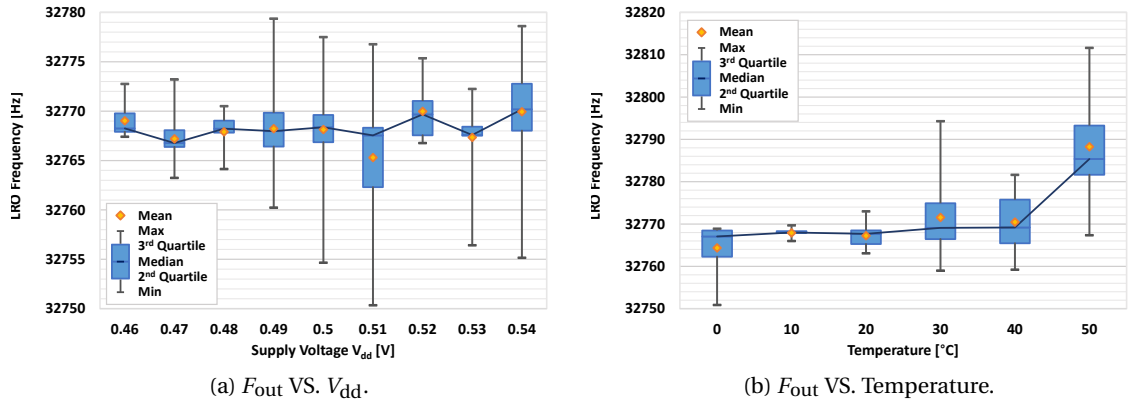


Figure 3.30: Measured output frequency when locked according to the voltage or temperature variations for 10 dice at 0.5 V.

Long-term Stability

The short-time stability of the oscillator is defined through jitter measurements and frequency stability over temperature and voltage. However, long-term stability is often described using the two-sample deviation $\sigma_y(\tau)$, also called Allan deviation (see Appendix B). As shown in [130], timer long-term stability is mandatory when used for SoC sleep modes. For instance, to initiate a communication synchronously and avoid energy-overhead due to missed radio transactions, the timer has to wake up the system at a precise time. In Figure 3.31, the Allan deviation is calculated for averaging periods τ up to 100 s with the max and min confidence interval. For intervals up to 20 s, $\sigma_y(\tau)$ is limited by white noise. Then, the Allan deviation is bounded by $1/f$ noise, which is reduced in advanced FD-SOI nodes [78]. This helps achieve a 0.4 ppm Allan deviation floor after 30 s.

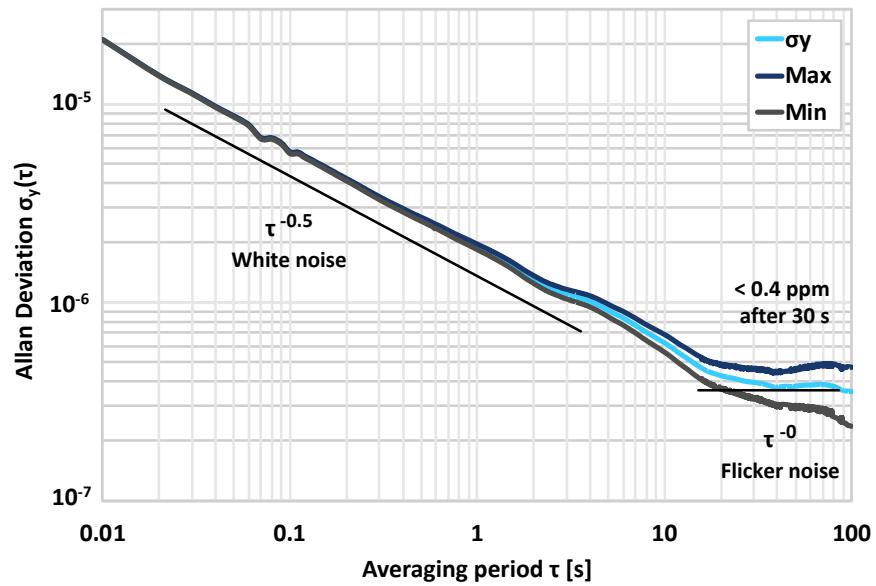


Figure 3.31: Measured LRO Allan deviation with min and max confidence intervals at 0.5 V/25 °C.

Power Consumption

The average power consumption of the LRO is 11 nW and 105 nW for the CLU, both measured at 0.5 V/25 °C with an output frequency of 32.768 kHz and a reference at 2²² Hz. Since the LRO and the digital logic of the CLU are embedded in the A.ON power domain of the SoC, the power consumption of the blocs are obtained by subtracting the power when the IPs are activated and disabled and the SPI/configuration interface clock gated. Consequently, the leakage power of these two sub-systems cannot be directly measured. However, using spice simulations, the leakage power is evaluated to 4 nW and 20 nW for the LRO and CLU respectively. It leads to a final power consumption of 15 nW and 125 nW.

3.4.3 Power and Drift Trade-off Resulting from the Relocking Scheme

Activity scenario

This section explores the resulting trade-off between drift and power based on the relocking scheme duty cycle. Since it is dependent of the activity of the system, a typical ULP wireless sensor node scenario is considered over one hour, as reported in Figure 3.32.

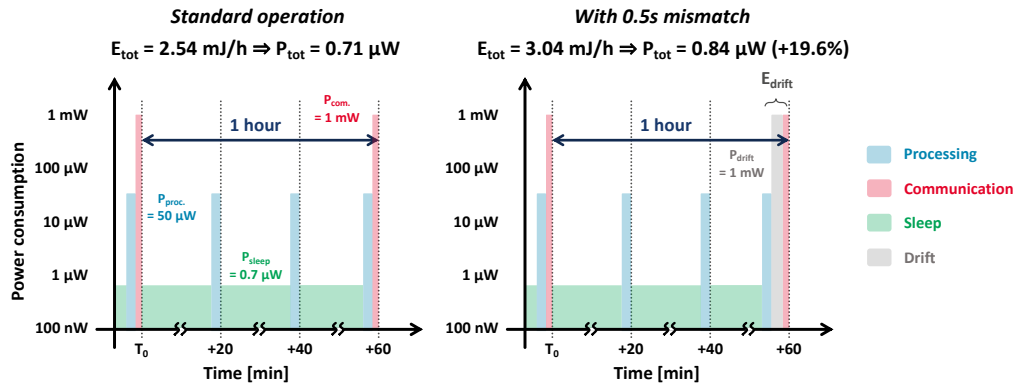


Figure 3.32: Example of a wireless sensor node operation scenario over a 1 hour period for a standard operation and considering a 0.5 ms communication mismatch.

The sensor is activated every 20 minutes to take measurements then process the data for 100 ms. Based, on the results of the SoCs presented in Chapter 4 (see Figure 4.51), this periodical task is assumed to consumes 50 μW. Then once in every hour, the collected data are transmitted by radio communication module. This second task is estimated to consume 1 mW during 1 ms.

The remaining time, the SoC is placed in a suspend mode (sleep) while a timer is assumed to run, thus keeping track of time changes and properly initiate the scheduled tasks. The whole power of background task is estimated to consume 700 nW. In total, the sensor node nominally consumes 2.54 mJ over a one hour activity, resulting in a 0.71 μW power consumption.

Since the sensors require to communicate with each other, they need to initiate communication synchronously. This synchronisation is performed using a timer. However, due the timer inherent drift a sensor node can turn on its radio earlier than others. In return, this

communications mismatch is translated in extra power consumption due to the waiting time between each devices. For instance, a 500 ms hourly drift result in 0.5 mJ extra energy, leading to a 3.04 mJ energy consumption per hour and thus 0.84 μ W of power consumption. Therefore, the benefit of the relocking scheme will reside in the trade-off between the power consumption of the whole clock system and the power overhead resulting from the low-power oscillator drift.

Mismatch according to the relocking scheme activity

As presented in Section 3.1.4 (see Figure 3.3) a relocking scheme is applied. Hence, the power of the XO reference accounts periodically for the total power consumption. As shown in Figure 3.33, an activity scenario over a period T_{tot} of 1 hour is targeted. A least one re-calibration is assumed during the communication step.

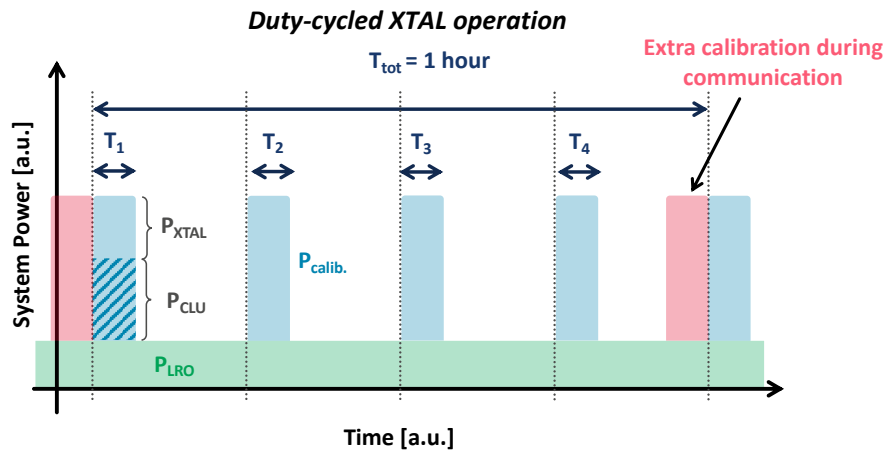


Figure 3.33: Evaluation of the total power and oscillator drift resulting from the duty-cycled utilization of the clock reference over a 1 hour period.

Accordingly, several metrics can be derived. First, the total calibration time $T_{calib.}$ is dependant of the unitary calibration time T_u and the number of activation N . In order to simplify the derivation an activity factor α can also be used.

$$T_{calib.} = \sum_i T_i = N \cdot T_u = \alpha \cdot T_{tot} \quad (3.4)$$

Using the re-locking scheme presented in Figure 3.2 and assuming a recalibration during the communication steps, the total drift D_r can be expressed as a function of the total deviation ppm_{tot} related to the number of activation N and total time T_{tot} and thus to the duty cycle α and unitary time T_u , as following:

$$D_r = \frac{ppm_{tot} \cdot T_{tot}}{N + 1} = \frac{(ppm_{tot} \cdot T_{tot}) \cdot T_u}{\alpha T_{tot} + T_u} \quad (3.5)$$

The total deviation of the system is the weighted average between the XO and the LRO

deviation both expressed in ppm:

$$\text{ppm}_{\text{tot}} = \alpha \cdot \text{ppm}_{\text{XO}} + (1 - \alpha) \cdot \text{ppm}_{\text{LRO}} \quad (3.6)$$

Lastly, the power consumed by the whole clock and re-calibration scheme is obtained according to the power consumption of the reference P_{XO} , the compensation unit P_{CLU} and the leakage-based RO P_{LRO} .

$$P_{\text{clk}} = \alpha \cdot (P_{\text{XO}} + P_{\text{CLU}}) + P_{\text{LRO}} \quad (3.7)$$

Table 3.4.3 reports the system deviation ppm_{tot} due to the relocking scheme along with the resulting mismatch over a 1 hour period. A commercially available XO with a $2^{22} \approx 4$ MHz frequency, a 100 nW power consumption and a 20 ppm is selected. The measured value of the digital compensation and the oscillator ($P_{\text{CLU}} = 125$ nW and $P_{\text{LRO}} = 15$ nW) are used to estimate the total power consumed by the system. The unitary calibration time T_u – the sum of the quartz starting-time and the CLU locking time – is set to 1 ms. A maximum 2000 ppm value is used for the LRO drift. Similar derivations can be done for several deviation values of the reference clock and oscillator.

α	$T_{\text{calib. [s]}}$	$T_{\text{c} \rightarrow \text{c [s]}}^*$	ppm_{tot}	Drift [s]	$P_{\text{Dr}} [\mu\text{W}]^{**}$	$P_{\text{clk [nJ]}}$
1	3600	10^{-3}	20	$2.00 \cdot 10^{-8}$	$5.56 \cdot 10^{-9}$	240
10^{-1}	360	10^{-2}	1802	$1.80 \cdot 10^{-5}$	$5.01 \cdot 10^{-6}$	37.5
10^{-2}	36	10^{-1}	1980.2	$1.98 \cdot 10^{-4}$	$5.50 \cdot 10^{-5}$	17.3
10^{-3}	3.6	10^0	1998.02	$2.00 \cdot 10^{-3}$	$5.55 \cdot 10^{-4}$	15.2
10^{-4}	0.36	10^1	1999.802	$1.99 \cdot 10^{-2}$	$5.54 \cdot 10^{-3}$	15.0
10^{-5}	0.036	10^2	1999.9802	$1.95 \cdot 10^{-1}$	$5.41 \cdot 10^{-2}$	15.0
10^{-6}	0.0036	10^3	1999.99802	1.57	$4.35 \cdot 10^{-1}$	15.0
0	0	0	2000	7.20	2.00	15.0

* $T_{\text{c} \rightarrow \text{c}}$: time between two calibrations

** E_{Dr} : mismatch power resulting from the drift over one hour. It is calculated assuming a 1 mW power consumption (i.e., communication power).

Table 3.7: Evaluation of the drift and corresponding mismatch power of the relocking scheme after 1 hour of activity, assuming a LRO with a 2000 ppm deviation.

Resulting power and drift trade-off

Depending on the activity scenario, a trade-off concerning the power resulting from the drift and the total power consumed by the clock emerges. This trade-off is evaluated relatively to the standard activity scenario of Figure 3.32. For a given LRO deviation, the power consumption resulting from the clock scheme P_{clk} and the resulting mismatch power P_{Dr} due to the drift are summed to the total power of the system. Finally, assuming a one hour scenario, Figure 3.34 reports the power overhead – compared to a standard operation – according to the duty cycle α .

Due to the inherent power consumption of the clock references, a minimum 34% power overhead is observed when the XO remains always-on (i.e., $\alpha=1$). Then, depending on the LRO deviation, by periodically switching-off the clock reference and using the low-power oscillator, some gain is observed on the overall power consumption. The optimal duty cycle α obtained

corresponds to the tipping point where the energies consumed by the reference and the low-power oscillator counter balance the power overhead resulting from the drift. Independently of the activation duty cycle, a minimum power overhead is observed due to power consumption of the clocking elements and the power resulting from the minimum drift.

This trade-off is highly dependent of the application power consumption and the expected activity scenario. With a drift power consumption of $P_{\text{drift}}=1 \text{ mW}$, α is located around 10^{-3} . During an hour, the compensation unit and the quartz are then powered-up during 36 s, meaning $36 \cdot 10^3$ activation sequences. Moreover, when the stability of the ULP clock is improved, the optimal duty-cycle value is decreased. The need for recalibration with the accurate clock source decreases.

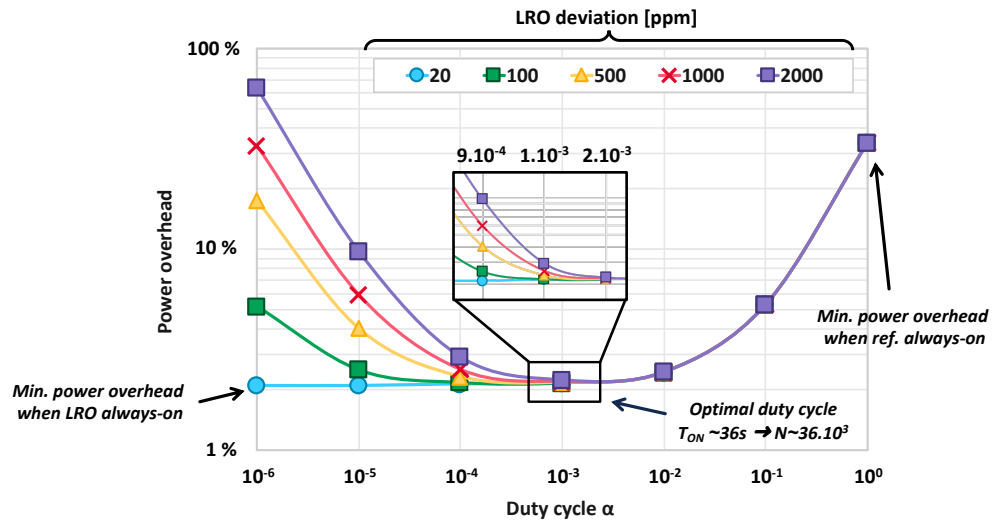


Figure 3.34: Power overhead resulting from the utilization of the relocking scheme. The power is given relatively to the activity scenario of Figure 3.32 where the power resulting from the oscillator drift and its power consumption are not considered.

3.4.4 Comparison with the State of the Art

CMOS-compatible oscillators

The system performances are compared in Table 3.8 with the latest State of the Art CMOS-compatible oscillators operating around 32.768 kHz. Compared to relaxation oscillators ([193] and VLSI'12 [194]) the utilization of a RO structure as the oscillating element helps reduce the total power consumption although the PVT sensitivity results in a reduced temperature range contrary to the aforementioned solutions. The RC oscillator of [195] offers traded-off performances. Indeed, for a similar power, it proposes reduced voltage and temperature stability for wider ranges.

[191] uses a charge-pump oscillator compensated with a multi-level bangbang which presents correct temperature stability. However, the bandgap required to supply a stable reference to the oscillator increases by 76% the total power budget. Consequently, solution using leakage-based oscillator like this work and [171] shows the best power efficiency. In term of

digital compensation, [171] presents a reduced power compared to this work. This is achieved using a SAR based compensation. However, it results in a voltage sensitivity far beyond other types of solutions.

To summarize, this work demonstrates a very low power consumption for simultaneously the oscillator and the digital circuit. Still, improvements on the power consumption of the digital logic could be expected. Moreover, the voltage stability presents great results relatively to other solutions, for a limited voltage range. Lastly, best-in-class temperature accuracy is obtained for temperature ranges suitable for IoT-oriented applications.

Feature	This work	JLPEA'16 [171]	ESSCIRC'16 [191]	ISSC'14 [195]	ESSCIRC'13 [193]	VLSI'12 [194]
Technology	28 nm FD-SOI	130 nm CMOS	130 nm CMOS	65 nm CMOS	180 nm CMOS	60 nm CMOS
Architecture	Ring Oscillator	Ring Oscillator	Charge-Pump	RC Oscillator	Relax. Oscillator	Relax. Oscillator
Osc. Area [mm²]	0.001635	0.269	0.014	0.015	0.105	0.048
Frequency [kHz]	32.768	12-150	32	33	32.55	32.768
Reference requirements	XTAL reference	XTAL reference @ initialization	XTAL reference	–	Process trimming	–
Power	15 nW (Osc.) 125 nW (Digital)*	20 nW (Osc.) 12 nW (Digital)*	80 nW (Osc.) 260 nW (Bandgap)	190 nW	472 nW	2.8 μ W
Voltage Stability [ppm/V]	90** [0.46 – 0.54] V	10000 [0.65 – 0.75] V	N/A	900 [1.15 – 1.45] V	11000 [1.0 – 1.8] V	6250 [1.6 – 3.2] V
Temperature Stability [ppm/°C]	1.9** [0 – 50] °C	7 [20 – 40] °C	10 [10 – 100] °C	38 [-20 – 90] °C	120 [-40 – 100] °C	32.4 [-20 – 100] °C
Allan Deviation floor	0.4 ppm** $\tau = 100$ s	N/A	60 ppm $\tau = 10$ s	4 ppm $\tau = 100$ s	N/A	N/A

* Do not included the clock reference power

** Measured when the system has locked after a voltage or temperature variation

Table 3.8: Summary of the achieved performances and state of the art comparison.

Allan Deviation

Lastly, in Figure 3.35 this work is compared with previously published systems through the Allan deviation with regard to the power consumption of the time reference. Compared to Griffith 2014 [195] (4th oscillators of Table 3.8), more than one decade is gained on the Allan deviation for a similar power consumption. The oscillator presented pushes the performances of CMOS-compatible oscillator towards XO implementations.

3.5 Summary

This chapter presents a fully-integrated Ultra-Low Power clock reference integrated in 28 nm FD-SOI. It relies on a programmable leakage-based RO for maximum power reduction. To ensure a stable output frequency, digitally controlled leakage sources are used. Special care during design and sizing is performed to maximize the matching of the source and ensure proper oscillations independently of the process corner.

Moreover, due to the high sensitivity of the LRO, a digital compensation unit CLU is added. Starting from a PI corrector, it integrates several optimizations for maximum accuracy with a reduced convergence time. Efforts are made to incorporate the previous module in a fully-integrated IP using an SPI interface. Next version could be replaced with a standard bus for utmost SoC integration.

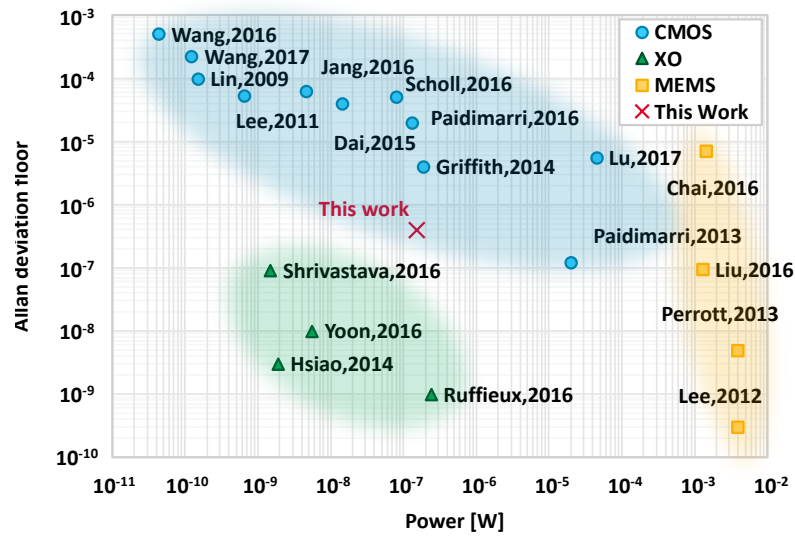


Figure 3.35: Comparison of Allan Deviation/Power with State of the Art clock generators.

The whole design is operating at 0.5 V. Forty-two dice were tested and 10 selected for digital compensation evaluation. Silicon results show a system offering high power efficiency combined with very low voltage and temperature variations. A power consumption of 15 nW for the oscillator and 125 nW for the digital are measured while the system ensures a 32.768 kHz with 90 ppm/V for $V_{dd} \pm 8\%$ and 1.9 ppm/°C from 0 to 50 °C. Lastly, the long-term stability capabilities of the whole system are characterized by a 0.1 ppm Allan deviation floor. In the end, by demonstrating an efficient low-cost clock, this work offers a versatile time reference and clock source for standard digital systems.

Future developments should be pursued toward improvement of the output frequency linearity and thus accuracy of the compensation. Even though an empirical approach has been used developed to compensate from the impact of FD-SOI, extended technology characterizations and layout techniques should be evaluated. According to the application accuracy requirement, the effect of the relocking scheme utilization period is also a metric that should be characterized for power evaluation in a complete SoC.

This work led to an A-SSCC 2018 publication, the reference is available in Appendix G.

Chapter 4

Ultra-Low-Power System Implementation

THE technical approach of this work aims to demonstrate the feasibility of an efficient ULP SoC using exclusively latest industrial guidelines or tools. In this chapter, several implementation techniques are explored to push down power consumption of the mass market Microcontroller Units (MCUs) to the hundreds of microwatt range.

Component's selection, based on adequately chosen FD-SOI technology features with architecture optimizations and design trade-offs, are bench-marked to improve the system energy efficiency. During implementation, the ULP techniques presented in Chapter 1 are adapted and enabled at ULV while respecting the constrained SoC power budget. Lastly, through an appropriate learning of the different power modes offered by the ARM Cortex-M0+, DPM is implemented to enhance the device lifetime and enable continuous functionality for IoT systems.

Section 4.1 presents the design methodology adopted for utmost energy efficiency at ULV in FD-SOI technologies. Techniques for static and dynamic power reduction performed during the implementation process are given respectively in Section 4.2 and Section 4.3. The following Section 4.4 extensively describes the application of self body-biasing at ULV. The resulting silicon implementations and measured performances are then evaluated in Section 4.5. A comparison of this work relative to the existing State-of-the-Art is finally provided in the last section, leading to conclusions and perspectives for future works (Section 4.6).

4.1 Ultra-Low-Power Design Methodology in FD-SOI

A design methodology applied to FD-SOI technologies has been developed to reach the utmost energy efficiency for an MCU core. It relies on a standard industrial EDA flow as described in Appendix C. This section presents the techniques explored during implementation to foster the best power performances of the final design while maintaining a tens of megahertz final operating frequency.

4.1.1 Minimum Energy Point Design Methodology

To reach the utmost energy efficiency offered by the technology, each design stage of the Cortex-M0+ core implementation has to be optimized: synthesis, place, route and verification. Even-though power optimizations can be added at several levels of a design – as shown in Chapter 1 – primary optimizations must be done to leverage the technology chosen. Hence, based on the measurements and knowledge gathered from previous implementations [81, 196] of ULV digital core in 28 nm FD-SOI, a technology-based methodology has been developed, improved during this work and later extended to 22 nm FD-SOI.

First, the trade-off between speed performances and power consumption resulting from the selection of a given transistor flavor (LVT or RVT) is not straight-forward. The optimal threshold voltage is defined following the results of bench-marking technique.

Then, at ULVs, special care has to be given during the clock design. Independently of the implementation technique chosen – H-tree VS. mesh network – the timing is less sensitive to resistance or capacitance parasitics [197]. On the contrary, it becomes strongly related to the gate delay. The cells used for the clock tree must be properly selected to mitigate the variability while ensuring speed performances at a reduced power cost.

Moreover, for digital MCU cores implemented in advanced technology node such as FD-SOI, the global activity factor is sparse while the leakage inherently increases due to feature size shrinking. It increases the MEP power supply $V_{dd,MEP}$ while the threshold voltage V_{TH} of the devices decreases (see Section 1.3). Therefore, the optimal operating voltage often appears to be above the threshold voltage in these designs. Compared to the minimum functional operating voltage, the margins used by EDA tools can be relaxed to match the reduced variability (i.e., reduced uncertainties and derating factors).

Lastly, EDA softwares are mostly set to reach timing constraints, then perform power and area optimizations. By limiting the set of standard cells to be used, the design performances can be significantly improved with a preliminary filtering of power-hungry cells. Moreover, optimizations through implementation trade-offs must be placed upon power consumption rather than gaining speed.

4.1.2 Transistor Features Selection

Problem Statement

As seen in Chapter 1, several transistor flavors are available in CMOS technologies. In 28 nm FD-SOI, Low Voltage Threshold and Regular Voltage Threshold are available. 22 nm extends the offer to Super Low Voltage Threshold (SLVT) and HVT.

An ideal implementation would use all flavor of transistor to: leverage speed in logic path, limit the power consumption, or recover leakage in non-constraining paths using RVT or HVT. However, due to physical implementation constraints, the SLVT/LVT are not mixable with RVT/HVT. Hence, at the early stage of the EDA flow, a particular technology type must be selected.

Moreover, FD-SOI technologies propose several gate lengths based on variations around the smallest gate size. They are referred as Polysilicon Biasing (PB) and offer a leverage on the performance of an implementation. In 28 nm FD-SOI, starting from the nominal length of 30 nm¹ corresponding to PB0, are obtained lengths of 34 nm – PB4, 40 nm – PB10 and 46 nm – PB16. In 22 nm are currently available PB0, PB4 and PB8. From a designer perspective, the PB is obtained during layout design, by adding a specific layer complying with the design rules, to increase the transistor gate length during manufacturing.

Body-biasing is also available in FD-SOI to cope with the timing and performance trade-offs. As it relies on an external bias generator, which increases the already constrained power budget, it has not been considered as a design tool in this analysis. Although some references and anticipations are given here, a full section (see Section 4.4) will be dedicated to its application for ULP designs operating at ULV.

Two techniques are here presented to properly select the transistor types for an implementation targeting the best energy efficiency: the evaluation through quasi Fan-Out of 4 (qFO4) structures or using direct synthesis loop. The qFO4 technique will be applied on the 22 nm FD-SOI to demonstrate the methodology on the wide range of transistor flavor offered by this technology. The same analysis can be conversely applied in 28 nm FD-SOI. The synthesis loop is applied in 28 nm as it requires low-voltage corners which were not available in 22 nm FD-SOI.

Evaluation through qFO4

To evaluate the impact of transistor flavor selection on the implementation of a digital core while greatly reducing the analysis time, the first approach is to define an elementary structure that will reflect the behavior of the core. A simple analysis consists to directly study the behavior of basic components such as transistors or standard cells. This approach is actually performed by silicon manufacturers to evaluate and characterize a technology and the standard-cells offer resulting.

However, the simulation study on these elementary structures presents a limitation in terms of precision as it does not necessarily emulate the datapaths behavior correctly. In fact, digital circuits are composed of paths made from logic gates based on networks of pMOS and nMOS transistors. Using an elementary structure, the effect resulting from the paths and architectures are masked. A solution is to use chained standard cell inverters to emulate the critical paths of a digital circuit yet, it also results to an approximate description of the system, as one cell type is used.

Consequently, to obtain results that take into account architectural effects and the diversity of networks encountered in critical paths, a specific logical path has been designed as seen in Figure 4.1. Each cell loads the equivalent of four times its own input capacity, which allows it to have a unitary Fan-Out of 4 (FO4) crossing time. The FO4 is actually a process-

¹The final 28 nm feature size is obtained during manufacturing by a $\times 0.9$ optical shrinking [80].

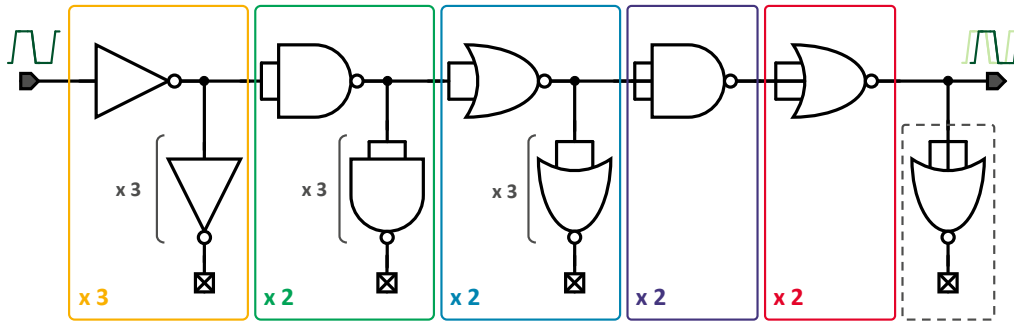


Figure 4.1: qFO4 logical path schematic based on 2 and 3 inputs NAND/NOR logic gates.

dependent delay metric used in digital CMOS technologies for characterization, but also by EDA tools to minimize the delay between gate stages [52]. The multi-inputs gates have all their inputs tied together as it corresponds to the maximum delay and leakage configuration [198]. This logical path allows to reflect the behavior of a circuit during one clock cycle which contains both high and low transitions. By design, this logical path eliminates the arbitrary aspect of sizing by a natural normalization of the output load of each stage. The whole path is composed of 11 stages and is called qFO4.

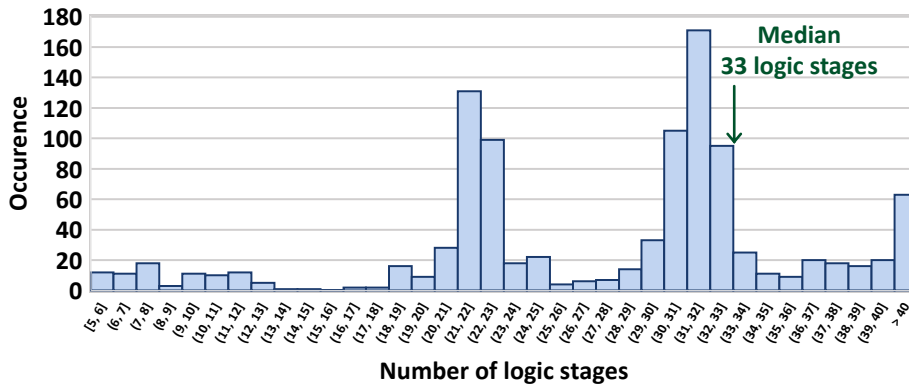


Figure 4.2: Evaluation of Cortex-M0+ critical paths lengths at 25 °C typical corner, nominal voltage in 22 nm FD-SOI LVT, targeting a 10 MHz operating frequency.

The accuracy of the qFO4 is improved by adapting the size of the path to the length of an average critical path of a real Cortex-M0+ implementation. Using the results of Figure 4.2 implementation, critical paths are extracted along with their number of stages. A median of 33 and an average of 35.5 logic stages is obtained. These relatively long paths for an MCU actually reflect the two-stage pipeline architecture of the Cortex-M0+.

Therefore, a test structure using 3 qFO4 and thus containing 33 logic stages has been derived and implemented in 22 nm FD-SOI. The path is then tested under several temperatures, corners, voltage conditions and cell types. First, the maximum frequency is reported in Figure 4.3 (left) according to the supply voltage from 0.3 V to 0.9 V and for the four available transistor types using simulations over five process corners at 25 °C. From this a minimum supply voltage at 0.4 V can be defined. Below – excepted for SLVT – the desired 10 MHz operating

frequency cannot be reached.

Then, the frequency according to the temperature for a supply voltage of 0.4 V is plotted in Figure 4.3 (middle). Assuming an IoT-oriented application and resulting temperature range between -20°C and 80°C , this second analysis reveals that HVT transistors are relatively slow. On the contrary, other types of transistors meet the operating frequency specifications. At this time, it is worth mentioning that in 22 nm FD-SOI, the naming convention of the transistor type and the threshold voltage associated (regular/low), does not provide a direct interpretation of their performances.

The frequency obtained depending on the process corners is reported in Figure 4.3 (right) at 0.4 V and 25°C . Following the voltage threshold reduction from HVT to SLVT, the variability of the device decreases. A $\times 4.7$ relative variation around the typical corner TT is observed with the higher V_{TH} down to a $\times 2.0$ for the lowest threshold. These variations must be taken into consideration at low voltage since it impacts the robustness.

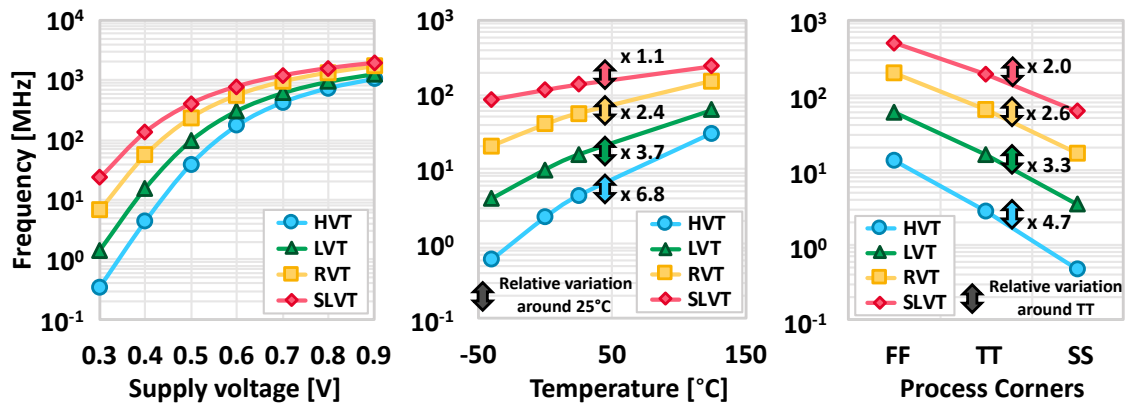


Figure 4.3: qFO4 frequency according to the Voltage (left), Temperature (middle) and Process (right) in 22 nm FD-SOI (0.4 V/TT/25 $^{\circ}\text{C}$).

The energy efficiency of the design is evaluated by reporting the energy per cycle – in that case the power-delay product of the path – according to the frequency for all flavors of transistor. Results are plotted in Figure 4.4 for a typical corner at 25°C . As systems operating most of their time in suspend modes are targeted as final applications, the static power consumption according to the transistor flavor must also be evaluated. In Figure 4.4 (right) is plotted the leakage power depending on the supply voltage for a typical corner at 25°C .

From all these results can be defined various trade-offs in term of power efficiency, operating frequency and leakage. Whereas HVT devices appear to be the most efficient, they do not meet the target frequency and present high variability. An LVT-flavor implementation is then optimal for energy efficiency with limited PVT sensibility. This first analysis gives a rough approximation of the performances of the final structure and it is flexible enough to evaluate a digital core under a wide range of conditions in a small amount of time.

FD-SOI Technology Comparisons

Using the qFO4 path with similar driving strength cells, technology node comparison is done in Figure 4.5. Relative variation from 28 nm RVT to 22 nm LVT/HVT and 28 nm LVT to 22 nm RVT/SLVT are given while considering the same 0.5 V supply voltage. The energy performances

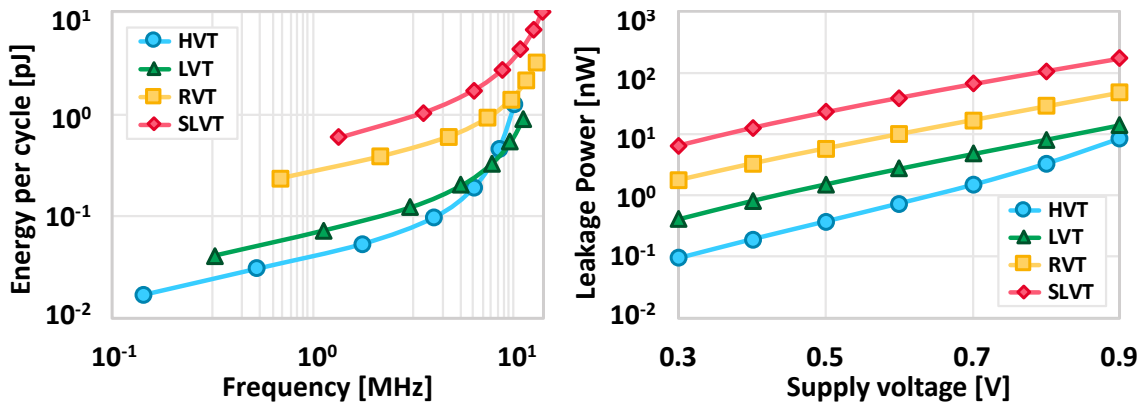


Figure 4.4: Energy performance according to the frequency (left) and leakage power according to the supply voltage (right). Results obtained for a qFO4-based path in 22 nm FD-SOI (TT/25 °C) and 0.1 V supply voltage steps.

and leakage power are also reported according to the qFO4 using several gate lengths PBs. These standard cell gate length PB variations help to leverage the performance/leakage trade-off during the implementation.

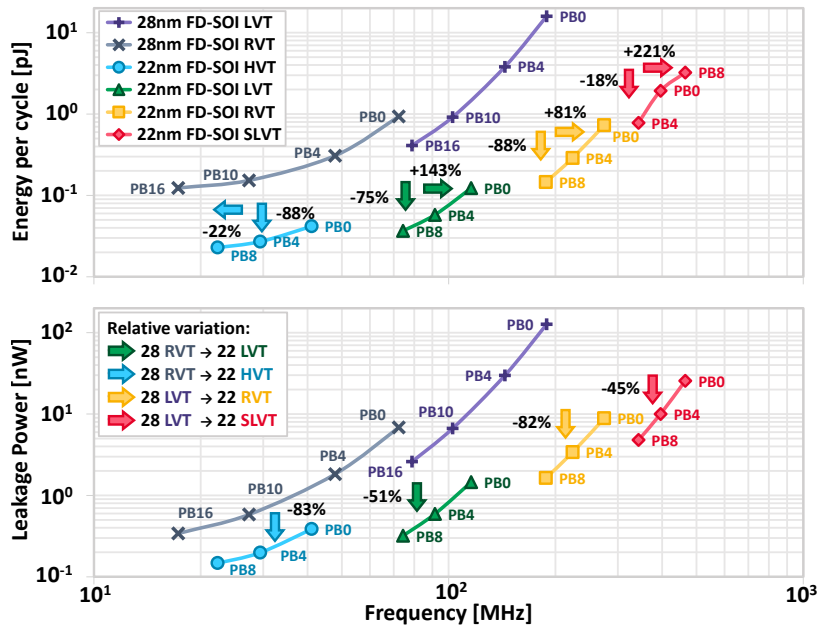


Figure 4.5: Comparison of in 22 nm and 28 nm FD-SOI technology nodes using qFO4 paths. Results based on energy performance (top) and leakage power (bottom) according to the frequency (0.5 V/TT/25 °C). Colored arrows show the relative variations from 28 to 22 nm using median values.

Evaluations through Synthesis Loops

A second method to evaluate the performance of the core relies on the available corners and cells characterizations proposed in the liberty files. Using synthesis loop the maximum operating frequency of a design can be determined. The Cortex[®]-M0+ MEP is extracted and reported in Figure 4.6.

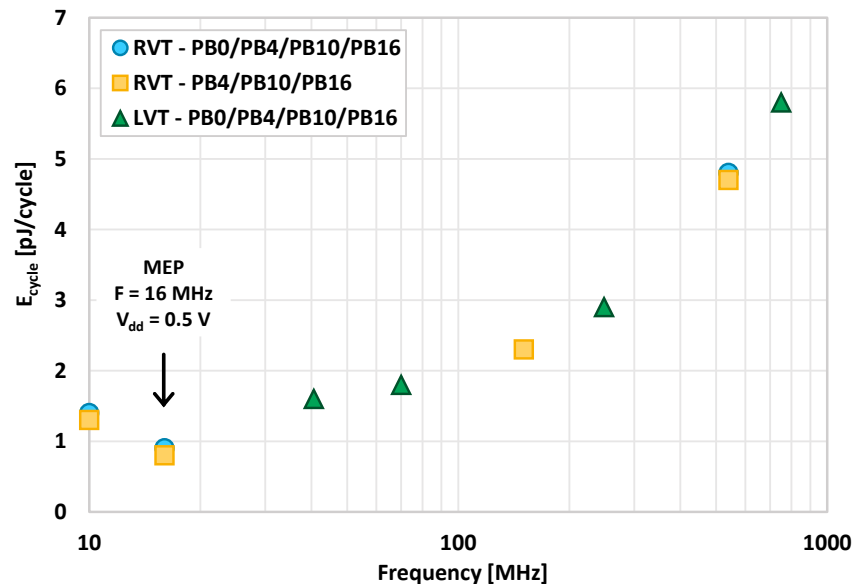


Figure 4.6: Subset of Cortex-M0+ Synthesis loops performances. To define the optimal set of parameters, several operating conditions, PBs and standard cell libraries are benchmarked for a supply voltage ranging between 0.45 V and 0.9 V, with typical corners at 25 °C.

Through PB restrictions, the gate-length of the cells used has also been modulated to help the EDA software to improve the final implementation. Removing minimal size gate length PB0 allows a gain in energy. The MEP was achieved at the 0.5 V/16 MHz operating condition, using RVTs and limiting the standard cell target-libraries from 34 nm to 46 nm gate-length.

In 28 nm FD-SOI, RVTs are also preferred over LVTs for their lower static power consumption. As explained hereafter in Section 4.4, the leakage of RVT flavor transistor can be further improved in sleep modes using the wide range RBB capability. Currently, most State-of-the-Art 28 nm FD-SOI works are designed using LVT to benefit of the speed-up obtained from the wide range of FBB. Indeed, higher frequencies can be achieved with very-low voltage at the expense of higher leakage [199]. For an RVT design, FBB is still possible but limited as previously explained in Section 1.2.3.

The evaluation through synthesis loop relies on the actual design of the core and the timing and power evaluation comes from the information available in the libraries used. Then, low voltage corners must be provided to the EDA tools. If they are not provided by the Silicon manufacturer, characterization must be performed which require time. More accurate results are expected yet the technique is not foolproof. In fact, a global activity around 10% is generally used for power evaluation. Results can be refined using a testing pattern of the activity resulting from a specific software.

4.1.3 Clock Tree and Standard Cell Selections

The clock tree synthesis was performed using a limited collection of cells composed of only two balanced and high strength RVT inverters of 34 nm gate-length. Limiting the type of cells enables the mitigation of the clock tree variability: the timing is mostly dependent of the gate delay, thus the cumulative variability across the tree can be controlled.

The utilization of the lowest threshold voltage is recommended for clock tree synthesis to reduce the variability. However, with FD-SOI technologies, the inherent flip-well architecture of the LVT cells prevents mixing with RVT cells. The selection of RVT inverters with 34 nm gate lengths is therefore a trade-off enabling:

- A reduced variability when using longer gate length;
- Steep transitions to limit the amount of buffering stages;
- Secure timings; the slowdown is lower in the clock tree than in the datapaths (which are based on 34 nm to 46 nm gate lengths).

Finally, the use of high strength is the result of clock tree synthesis benchmarking. In fact, when the strength is lowered, the number of gate stages quickly rises in order to recover from the transition time degradation.

4.1.4 Electronic Design Automation Flow Tuning

In 28 nm FD-SOI, the nominal voltage is defined at 0.9 V, with a 0.4 V threshold voltage for RVT-transistors. However, the full system was implemented with ultra-low voltage corners centered at 0.5 V, allowing near-threshold operations [66]. The clock and data derating along with setup and hold uncertainties were defined using process variability measurements data at this voltage.

To improve the variability tolerance, the hold timing violations were checked in all corners, (i.e., 3σ fast/slow corners, both using $-40^\circ\text{C}/125^\circ\text{C}$ temperatures, and all min/max Resistor/Capacitance configurations). It has been observed during Static Timing Analysis (STA) that most of the fixed violations were generated in the slow process/low temperature corner. For this condition, the clock tree timing is degraded while the data paths timings depend on their gate composition and depth. It results in unpredictable races. The full set of corners for hold violations analysis and fixes was only applied after the Place & Route completion.

During implementation, only a reduced set of corners, the fastest ones, were used for hold checks to prevent the tool from convergence issues and over-buffering² With this simplified procedure, the inserted buffers do not create challenging datapaths in terms of setup due to the PVT variations impact on timings, thus avoiding hold/setup fix loops from the tools. The full set of corners was then checked in sign-off using a dedicated STA, and remaining violations were fixed using ECO loops. This approach offers a limited buffering cost of 2% while ensuring functionality across corners.

²Since uncertainties are not fixed and scale with corners, at low voltage, the worst hold corners is the slow one at low temperature.

4.2 Static Power Reduction

From the previous section, technology perspectives were explored for optimal energy efficiency and reduced leakage during suspend mode. Consequently, this section introduces the techniques implemented using these technology nodes to mitigate static power. The solutions proposed have a limited power overhead with a maximum gain on leakage. Power gating strategies are first examined in two ways; a global power switch and a distributed implementation. State retention with power gating is later investigated as well as a double power gating scheme.

4.2.1 Power Gating

Most efficient power switch designs are offering stable power supply delivery, fast switching and voltage level monitoring [200]. Such features are resulting in complex and large designs, with a power imprint that does not fit the targeted Cortex-M0+ Core characteristics.

Hence, two solutions have been analyzed. In 28 nm FD-SOI, a global power switch using LVT pMOSs transistor is used whereas, in 22 nm FD-SOI, distributed power switches are employed. The implementations, sizing and cell selections of these techniques are reviewed in the following section, whereas the corresponding results and their impact on system power consumption will be presented in Section 4.5.

Global Power Switch Implementation

A global implementation using distributed LVT pMOS transistors is presented in Figure 4.7 to reduce complexity of the power switches floor plan integration. Their gates are tied to the body and controlled using a command signal CMD generated externally by a digital Power Management Unit (PMU) (see Section 4.3).

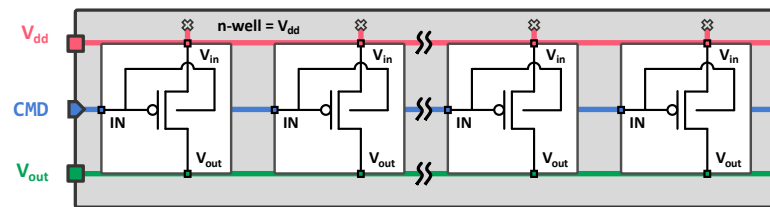


Figure 4.7: Global power switch design using LVT flavor pMOS with gates tied to the body.

As introduced in [104], this solution is based on DT-CMOS. In FD-SOI, it ensures fast swing recovery and super shut-off performances without compromising the system body-biasing capabilities [201]. Applied to LVT this technique extends leakage-reduction capability of the transistor when off by applying an RBB scheme. The I_{ON} current corresponds to the nominal biasing conditions of the device as shown in Figure 1.9 (see Section 1.2.3).

With an RVT implementation, a similar RBB scheme would require an over- V_{dd} supply voltage thus, needing an extra generator. However, using the same input/body connection, it results in an FBB mode when ON and nominal operation when OFF. This scheme can also be applied in standard bulk technology but is limited to 0.6 V due to the pn-well diodes.

A comparison of the R_{ON} and R_{OFF} for unitary LVT and RVT pMOS devices using a stan-

standard connection and the DT-CMOS technique (gate-body connection) is reported in Table 4.1. Since the biasing conditions are the same for both modes when the switch is ON, no differences are observed on R_{ON} . For the same given area and operating conditions, the LVT device presents the best ON and OFF resistance ratio. It results in the maximum ON current while offering the best leakage current reduction during OFF state. Considering the same type of device, a 26% gain is obtained on R_{OFF} compared to a standard implementation.

Mode	DT-CMOS mode		Standard	
	RVT ¹	LVT ²	RVT	LVT
R_{ON} ($V_{CMD} = 0\text{ V}$)	5.84 k Ω	4.24 k Ω	5.84 k Ω	4.24 k Ω
R_{OFF} ($V_{CMD} = 0.5\text{ V}$)	798 M Ω	770 M Ω	916 M Ω	570 M Ω
$R_{ON}/R_{OFF} (\times 10^{-5})$	1.37	1.34	1.57	1.81

¹ For RVT pMOS in 28 nm FD-SOI, no bias translates by $V_{PW} = V_{dd}$.

² For LVT pMOS in 28 nm FD-SOI, no bias translates by $V_{PW} = 0\text{ V}$.

Table 4.1: Shut-off performance analysis for a single pMOS transistor in RVT and LVT flavors using a standard connection and the DT-CMOS technique (0.5 V/25 °C).

The layout of the power switch is shown in Figure 4.8. Unitary pMOS were designed and assembled together to form a global switch. Due to the FD-SOI cross section and the DT-CMOS scheme adopted, the p-well of the LVT device must be isolated from the p-type substrate, which is connected to the ground. Therefore, deep n-well layers (represented in orange) are added to each unitary device and connected to the supply voltage V_{dd} . This guarantee an easy integration into the final floorplan while avoiding well shortcuts or pn-well diode leakage during operation. The area of a single switch is 20.4 μm^2 which is mostly defined by the deep n-well Design Rules Check (DRC) constraints. The global area is 797.1 μm^2 . A total of 25 unitary pMOS switches is used to ensure sufficient I_{ON} current.

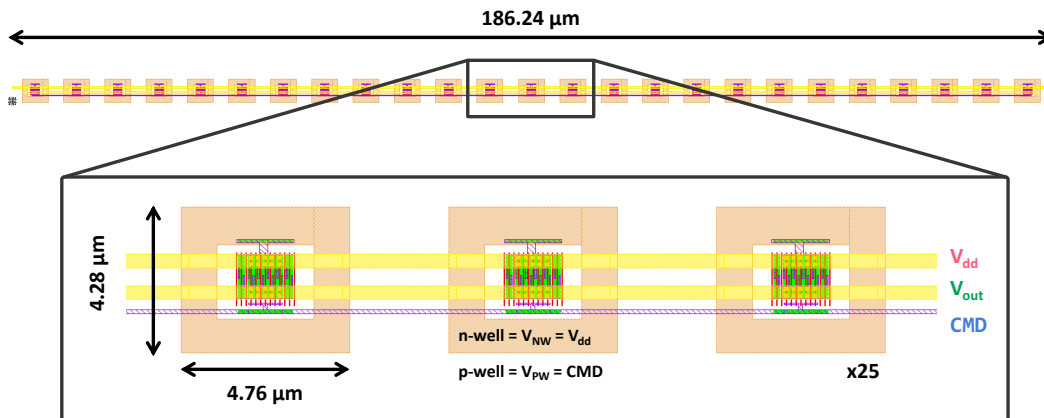


Figure 4.8: Layout view of the global pMOS power switch using 25 unitary switches. The area of a single switch is 20.4 μm^2 and the global area is 797.1 μm^2 .

The power switch was designed after the floor-planning and power grid. Free spaces

and area margins were left on top of the core power grid to properly insert the power switches into the design. As shown in Figure 4.9 a supplementary set of power stripes named V_{dd_MCU} is required to connect the core power grid to the power switch. The V_{dd} stripes are also drawn over the core to supply other power domains situated below. Then, the V_{dd} capacitance C_{LOAD} of the core is directly extracted from the layout using standard EDA tools. This implementation method has been chosen because the power switches sizing does not impact the digital synthesis and place & route.

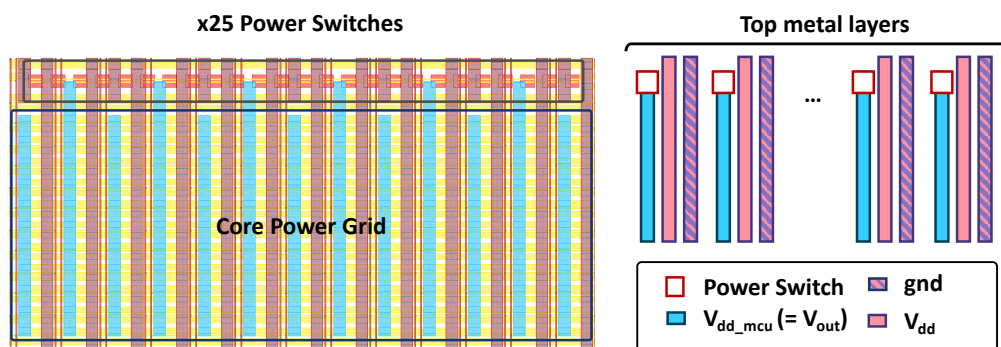


Figure 4.9: Integration of the global power switch into the core power grid.

Furthermore, the power switch aims to minimize the static current consumption while conserving subsequent dynamic current driving capabilities. The Cortex-M0+ was modeled using a current load ranging from $1\mu A$ to $50\mu A$. It reflects the simulated current profile of the MCU for a given set of programs. Since a unique power generator is used, a deliberately constrained -2% power impact is targeted at 0.5 V. Using qFO4 simulations, a 2% switch drop compensation on the gated domain results in a 1% extra power consumption on other power domains.

Therefore, the resistance of the core R_{LOAD} is defined between $490k\Omega$ and $9.8k\Omega$ for a typical corner TT 25 °C. For fast FF and slow SS both at 25 °C, using technology performances simulations, R_{LOAD} has been set respectively between $[200k\Omega; 4.1k\Omega]$ and $[1M\Omega; 21k\Omega]$, thus taking into account the core leakage variations according to the process.

The simulated performances of the power switch across 3 PVT corners are given in Table 4.10b. Figure 4.10a offers an overview of the methodology adopted for these simulations. A 2 mV worst case voltage drop is reported in typical corner TT 25 °C for a $9.8k\Omega$ load. For FF and SS, a maximum voltage drop of respectively 4 mV and 1 mV is observed for the unfavorable R_{LOAD} , yet fitting the power margin.

The restoration time has been characterized to remove the need for a feedback controller with acknowledgment, replaced by a hard-coded wait operation. These specifications ensure the minimum power consumption overhead due to the controller area and activity power savings, and the inherent switching time when the SoC goes into power off modes. The Unified Power Format (UPF) descriptions to instantiate power switches in a standard EDA flow are given in Appendix D.

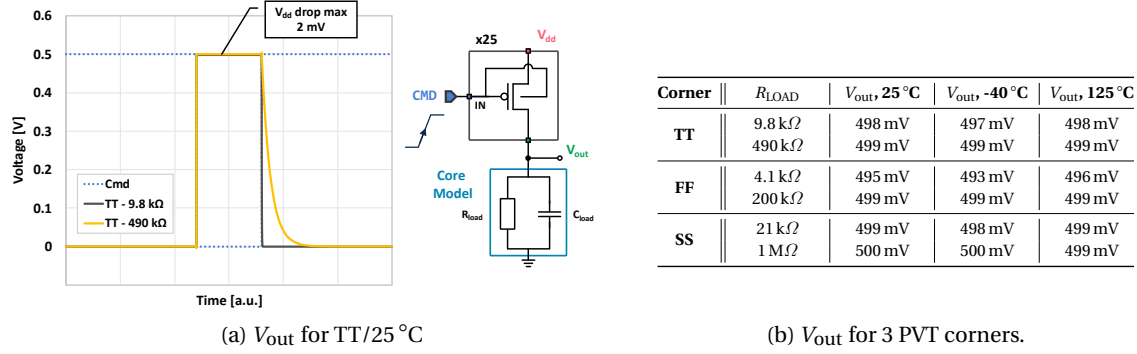


Figure 4.10: Performances evaluation of the global power switch for a 0.5 V supply voltage.

Distributed Power Switches

A second approach for power gating relies on distributed power switches which might be available in standard cell libraries (see Figure 4.11). The incorporation of the power switches is done at the early stages – place step – of the EDA flow. Logic cells are then added around. The control signal routing is performed by the tools as described in the power intent (see Appendix D).

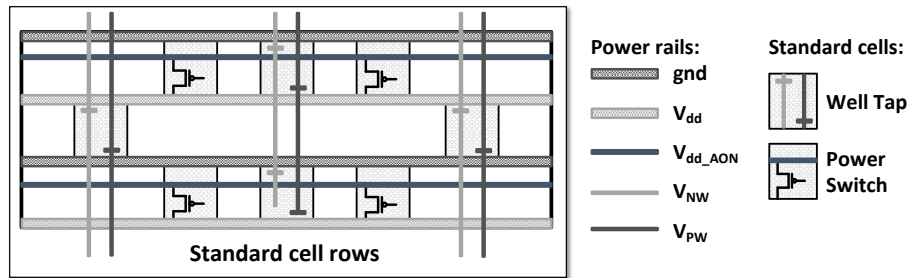


Figure 4.11: Schematic floor plan for distributed power switches insertion in standard cell rows. The power gates are directly incorporated into the core design along with other physical cells.

Figure 4.12 highlights the standard cell abstraction of a distributed header power switch. The power gates share a common power and ground row architecture with the standard cells. The output supply $V_{out} = V_{dd}$ is turned ON or OFF by using the input control signal CMD. The global V_{dd_AON} input supply of the design (see Figure 4.11) is connected to the input supply port of the cell using vias and an auxiliary power grid. Unitary, a power gate is laid out as multiple fingers of pMOS switches.

Whereas bigger cells ensure best driving capabilities, they also result in higher leakage current during OFF states. Bigger areas might also lead to placement constraints related to the power grid. Similarly, it might also impact the placement of other cells leading to cell congestion, hence timing constraints. Then, the number of instances used over the whole power domain can also be used as a weighting factor. By using smaller cells in larger number, similar current capabilities can be obtained yet relaxing the placement constraints. Lastly, PB can be applied to mitigate the leakage when the power switch is OFF.

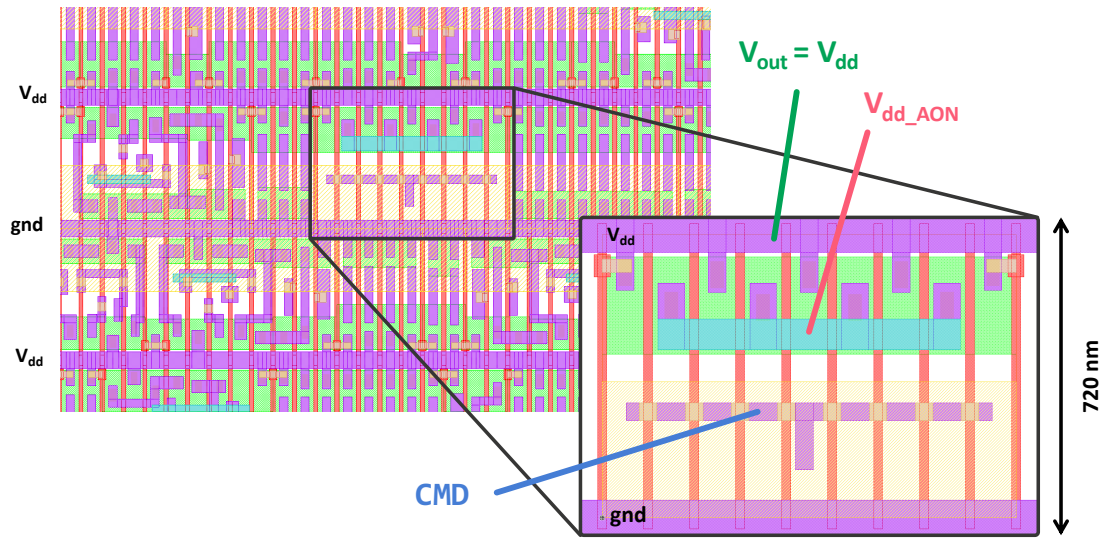


Figure 4.12: Header power gating standard cell layout and integration into the core standard cell rows in 22 nm FD-SOI.

As LVT and RVT transistor flavors are not mixable, the core implementation results in the selection of the transistor flavor used in the cells. Moreover, since the body-biasing option is applied on the whole island gated through the well taps, the utilization of a DT-CMOS scheme is here discarded.

Similarly to global power switch simulations, the distributed power switches are benchmarked for two types of cell available in the 22 nm FD-SOI standard cell libraries (see Figure 4.13). RVT devices are used for compatibility with the RVT-based core implementation. The impact of the number of instance as well as the PB used is evaluated. The output voltage during the active mode of the core (power switch ON) is reported in Figure 4.13a, while the current when the power switch is OFF is shown in Figure 4.13b.

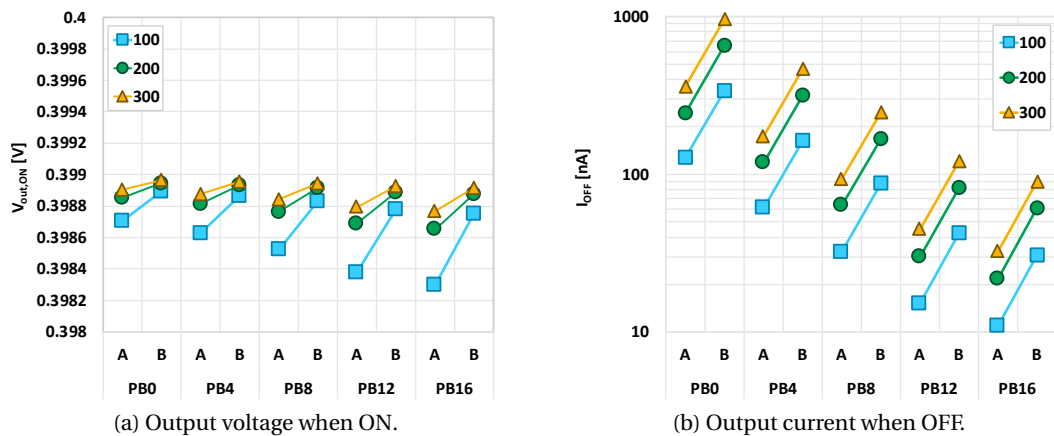


Figure 4.13: Performance evaluation for two type of RVT header cells in 22 nm FD-SOI according to the number of cell instantiated. Cell A is a power switch with a $\sim 200 \mu\text{m}^2$ area and $\times 20$ unitary driver strength. Cell B is a power switch with $\sim 400 \mu\text{m}^2$ area and $\times 60$ driver strength.

4.2.2 State Retention with Double Power Gating

Problem Statement

The power gating strategies developed in the previous section are based on power switching to provide leakage power saving. Hence, if no retention strategy is anticipated, any state is lost in the power gated domain. For state-less systems like fixed function accelerators or system targeting tasks without a need for past remembrance, this approach is acceptable.

When data have to be propagated from one active mode iteration to the following, state restoration can be applied by architectural state saving into a memory before and after powering OFF and ON. However, it results in significant timing overhead and thus energy cost. To mitigate this impact, the state of the system can be preserved using retention registers (see Section 1.5.2). Consequently, this has an energy cost due to the leakage of the always supplied registers. To offer flexibility, this solution can be combined with a second power gating strategy. It allows to retain the information for certain periods while offering a fully off mode for leakage reduction. This solution is called double-gated State Retention with Power Gating (SRPG) as shown in Figure 4.14.

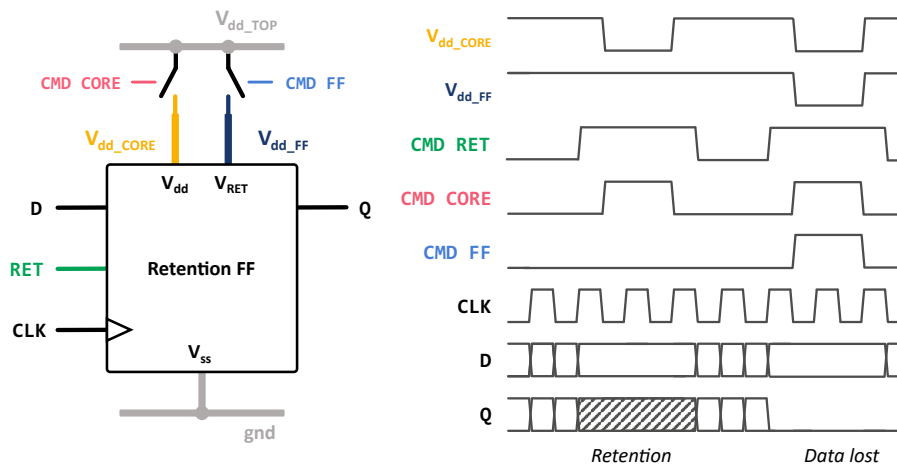


Figure 4.14: Schematic description for double-gated SRPG implementation with corresponding control signals and sequencing. CMD RET ensures the retention control, CMD FF for power gating of the V_{dd_FF} supply and CMD CORE for power gating of the V_{dd_CORE} supply.

Physical Implementation

While SRPG is a known solution for low-power [161], the physical implementation of double-gating remains complex in a standard digital floorplan region.

Above the version 2.0 of the IEEE Std. 1801, an arbitrary number of extra supplies can be defined into a power domain to connect the retention flip-flops, power gates or always-on buffer. Nevertheless, for a given power domain, a single primary default supply is still expected to be routed as the standard cell main rail. Whereas it does not have impact during place and route step, it is used by power aware EDA tools during verification. Hence, a solution fully compatible with the UPF standard has been developed by using a triple power grid connected to two power domains as shown in Figure 4.16.

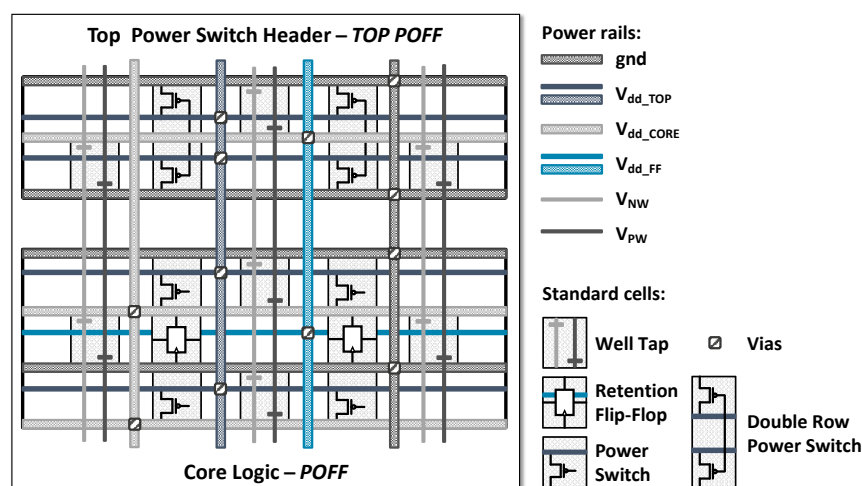


Figure 4.15: Triple power grid description for double-gated SRPG implementation in 22 nm FD-SOI.

In a first power domain *POFF*, composed of the core logic, header power switches are distributed throughout the floorplan. They connect the always-on power supply V_{dd_TOP} to the standard cell main rails V_{dd_CORE} , using a first power grid. Retention registers are also added to this power domain. They are both connected to the standard V_{dd_CORE} and an auxiliary switchable power supply V_{dd_FF} .

To ensure the extra shutdown of V_{dd_FF} , a second power domain *TOP POFF* is added above or below. This domain contains power switches (in this case double row cells) that connect the main power V_{dd_TOP} to V_{dd_FF} (standard cell main rail of this domain). At the end of the implementation, this power domain is filled with filler cells along the already placed power switches and eventual always-on cells. This solution allows to use of the same library than power gate cells but without particular confinement to dedicated rows to prevent shortcuts between V_{dd_TOP} and V_{dd_FF} .

Lastly, V_{dd_CORE} and gnd supply stripes – third power grid – are added on the two domains to reduce the power grid voltage drop all along the design. Well taps are normally connected to the bias supplies. By carefully placing the low-level metal stripes for V_{dd_FF} and V_{dd_TOP} , and playing with the double row flip-flop and power switches placement, this solution does not increase the routing complexity or the pin access to the regular cells. Moreover, the always-on cells needed for buffering power gate, clamp or retention controls can be placed inside the cell rows without particular precautions. Their power routing is ensured directly by the EDA tools.

A layout abstraction of the implementation is proposed in Figure 4.16. The same control signals than Figure 4.14 are used. Their utilizations for the SoC power modes will be explained in Section 4.3.1. The corresponding UPF description of this implementation is extensively described in Appendix D.

An alternate solution to the top power switch header is to use a custom power gate cell which does not drive the standard cell main rail but a secondary retention rail. This allows distributed placement within the logic gates but results in placement conflicts with the always-on cells.

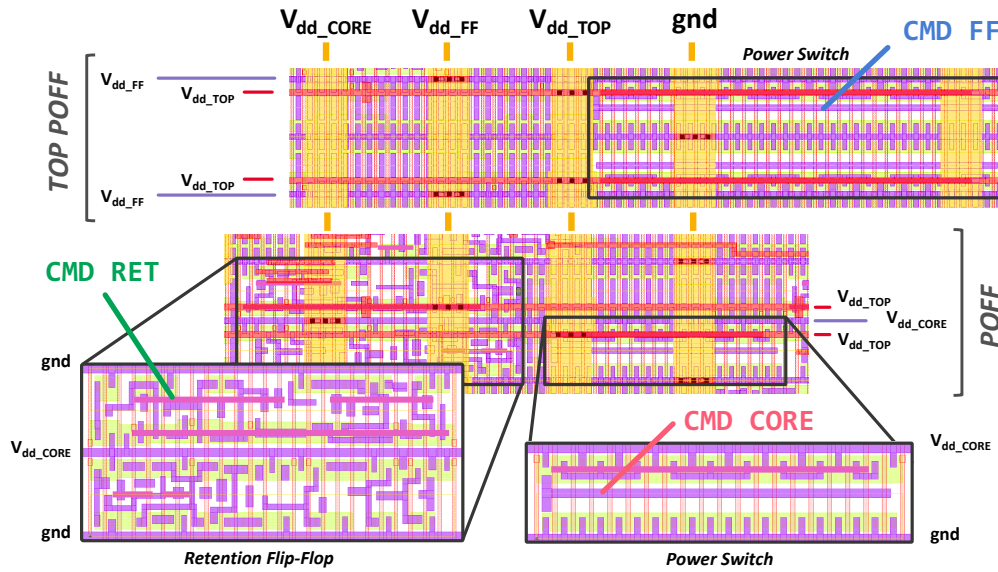


Figure 4.16: Layout abstraction of the triple power grid for double-gated SRPG implementation in 22 nm FD-SOI.

Application of Double-Gated SRPG

The selection of registers that need retention Flip-Flops is done during synthesis through the UPF power intent. By default, all the registers of a given power domain are replaced. For a Cortex-M0+ core, depending on the peripherals included in the retention domain, it results in instantiating ~3000 cells. For known micro-architectures, refinement can be done by selecting only a set of essential registers (e.g., status register) that needs to retain information. Hence, a gain in term of area and power consumption is obtained.

Double power-gated SRPG extends the available SoC power mode by completely switching-off a power domain. However, since information is lost (see Figure 4.14), it works complementary to a non-volatile memory (e.g., flash memories). For short suspend sequences, standard retention is used in the FFs, whereas long-term sequence uses the long term storage. If such memories are not available in the system, hot reset (a.k.a. context restoration) needs to be applied to reinitialize the system after powering on (see Section 4.3.1).

4.3 Dynamic Power Reduction

The previous section has exposed techniques to reduce the static power consumption. Consequently, this section presents the techniques developed at systems level to dynamically reduce power during the active and suspend states. Using the Cortex-M0+ features are derived ULP modes as well as system portions and control generations for Dynamic Power Management (DPM). Dynamic Frequency Scaling (DFS) is also investigated for extra power reduction during mode switching.

4.3.1 Dynamic Power Management

Power Management Strategy

The Cortex-M0+ includes in its micro-architecture the support of two suspend modes – (normal) sleep and deep sleep – to reduce the power consumption when no power processing is required (see Appendix A). From these modes, a power strategy is defined along with ULP modes and power domains. The detail of the peripherals included in each power domain and their status depending on the selected ULP mode is given in Table 4.2 relying on the SoC of Section 2.3.

In Active mode, all the peripherals of the SoC are powered-on. During sleep mode, clock-gating of the P.OFF and MEM domain are activated. Further power saving is obtained in deep sleep mode by activation of retention in the power-off and switching off the MHz clock generator. The ultra deep sleep mode leverages the double-gated SRPG for ultimate power reduction. The A.ON domain contains peripherals that remain powered on independently of the mode. These modes can be entered using a Wait For Event (WFE) or Wait For Interrupt (WFI) instruction, or with the Sleep-On-Exit feature of the Cortex-M0+ [202].

Power domain	Peripherals	ACTIVE	SLEEP	DEEP SLEEP ¹	ULTRA DEEP ²
Power-OFF	Core High-power peripherals Debug	ON	Gated	Retention	OFF
Always-ON	WIC PMU RTC Low-power periph. Low-power clock	ON	ON	ON	ON
Others	MHz clock generator	ON	ON	OFF	OFF
Memory	SRAM	ON	Gated	Gated	Gated

¹ Only available when SRPG is implemented.

² Only available for a double-gated SRPG system or implementation without state retention.

Table 4.2: System partitions and proposed ULP modes.

A hardware control module or PMU implements the Table 4.2 modes. It is embedded into the A.ON domain. The PMU contains memory mapped control registers for configuration through the software executed on the core. With this SoC architecture, the user has the opportunity – depending on his application requirements – to select the desired suspend mode and wake-up on certain hardware events. Some implementations of this work do not include state retention (see Section 4.5), so the deep sleep mode is not available. A proposed context restoration using a back-up memory is discussed in Section 4.3.1. Frequency switching depending on the mode are discussed in Section 4.3.2.

The retention mode of the memory macros is not used. Indeed, the system is designed to operate at ULV which is already beyond the nominal mode of the memories. Reducing further the power supply during suspend mode will not ensure information retention. Clock gating strategies are thus sufficient. Some memory macros propose embedded power switches on the periphery for supplementary power reduction. This solution can be used as an extra suspend mode.

Sequencing of the Power Mode Control Signals

The controls signal required to drive the power gates, retention registers, isolation cells and other peripherals during a specific mode are sequenced using the PMU. As shown in [139], for complex or re-programmable solutions, a second small MCU (like the Cortex-M0+) can be used. In our specific context, doubling the core is not a viable solution. Therefore, the PMU is implemented as a programmable Finite State Machine (FSM) for minimum area and energy overhead.

A WIC is also added to the A.ON. This small interrupt detection module mirrors the interrupt masking function of the Nested Vectored Interrupt Controller (NVIC). Adding this block allows to stop all the clock signals going to the core or switching it down while keeping the interruption handling functionality. The WIC integration is completely transparent from the software perspective. The interrupt mask information is transferred from the core to the block before entering suspend modes.

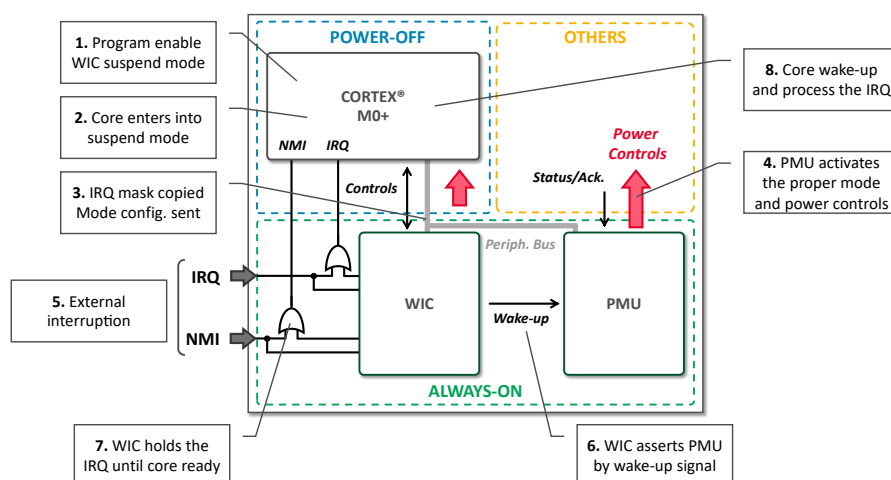


Figure 4.17: Interruption handling during suspend modes and PMU control signals.

The complete suspend mode operation and the sequencing of the SoC when entering in suspend mode is given in Figure 4.17 whereas Figure 4.18 shows the corresponding timing sequencing.

Before going to a suspend mode, the WIC is loaded ①. Then signals are asserted by the core to the PMU ② and the WIC ③. Assuming the deep or ultra deep modes requests, a sleep signal is raised along with a deepsleep signal. Consequently, the clock is stopped, then isolation clamping activated, followed by the retention signal assertion. Lastly, the power-gating network is turned off ④. When an external interruption is detected ⑤, the WIC sends a request to the PMU to restore the power, the retention and the clock signals to the necessary hardware blocks ⑥, while holding the interrupt until the core is ready ⑦. The processor wakes up, resumes operation, process the interrupt and clears the request ⑧. If the sleep mode is requested by the software, only the sleeping signal is asserted ② so only the clock signal is stopped ④. When an external event is received, the core resumes its operation as in the previous modes.

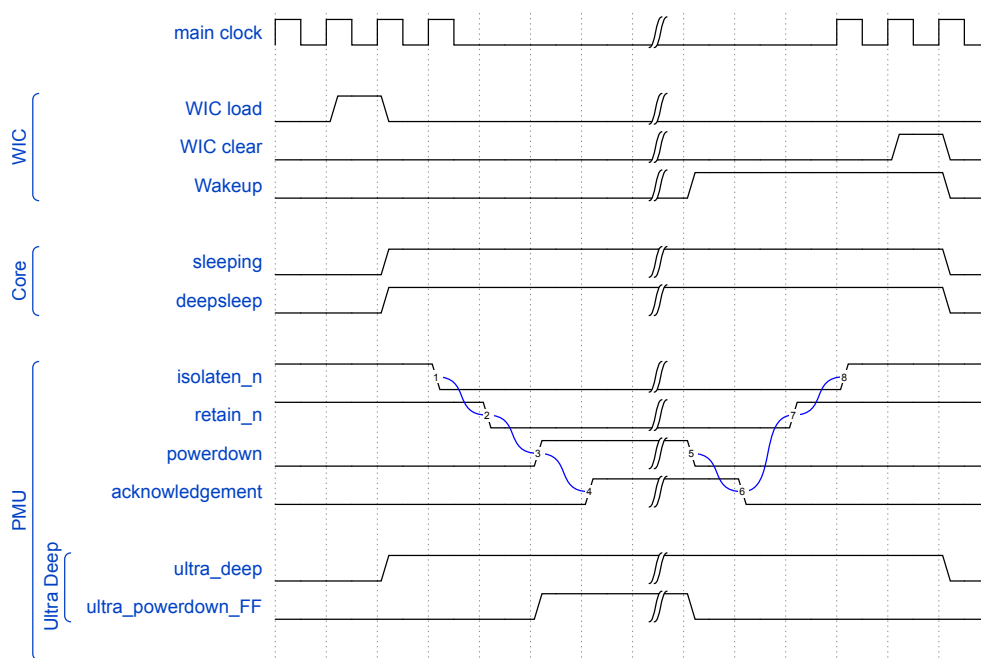


Figure 4.18: PMU control sequence and signal generation during suspend mode transitions.

An acknowledgement signal is expected by the PMU to synchronize with other peripherals (e.g., clock generator). As mentioned in Section 4.2.1, the acknowledgement signal from the power switch has been removed by characterizing the restoration time of the power gate and adding limited logic (e.g., synchronous counter or delay line). Consequently, external conditions that might result from a wake-up signal occurrence just after turning on or off the power gates must be held. Wake-up requests during state switching are delayed. To allow the cancellation of suspend mode sequences (i.e., wake-up during powering down or up), acknowledgement signals are mandatory. They allow splitting of the PMU sequences into atomic transitions. To do so, analog solutions will rely on comparators or Schmitt's triggers to monitor the voltage rail charging. Digital approaches will prefer power-gate daisy chaining [69] or frequency monitoring of RO connected to the supply voltage.

The ultra deep sleep mode is activated by programming registers inside the PMU³. By activating it, it generates extra control to switch off the retention register and turn the whole power off domain down. This control sequence is also effective when no retention is performed into the core.

Context Restorations

When the core is totally gated and no retention is implemented, the current core state is lost after wake-up. Such solution is acceptable yet requires system startup sequence to run at start up. Then, this solution deals between ultra low power consumption and increases response latency (so extra energy consumed) during initialization.

Extra state retention memories (e.g., Static Random Access Memory (SRAM)) and firmware help shortening the restart process. Application Programming Interfaces (APIs) are added to store the processor registers and states into the SRAM before power down (context_save() function). When the processor resumes, the bootloader restores the information automatically from where the system was in the application code (context_restore() a.k.a. hot reset function). However, some processor states like exception status (i.e., IPSR) might not be restored limiting the “no retention” feature to the thread mode (see [161]).

To overcome such limitations, the interruptions must be handled. In Figure 4.19 is given a C-type code to deal properly with the Interrupt ReQuests (IRQs) and the context restoration when no retention is implemented. The processor only wakes up using the highest priority interruption (reset) or by “start logic” feature on the IO ports. The boot loader uses a counter value stored in an A.ON memory to differentiate between the initial or resumes starts-up. Accordingly, it returns to a given program entry point.

```

1  // Boot loader first entry point
2  ...
3  // Before entering suspend mode
4  get_primask();           // Store PRIMASK register
5  __disable_irq();         // Stop the interrupts
6  context_save();          // Store program context
7  // System shutdown
8  SetSleepMode(1);         // Select Ultra Deep Sleep
9  __WFI();
10 ...                     // System powered down
11 ...                     // System wake-up using reset
12 // Boot loader re-entry point
13 // Context restoration
14 context_restore();        // hot reset
15 set_primask();           // Restore PRIMASK
16 handle_irq();            // Handle pending IRQs
17 // Resume
18 ...

```

Figure 4.19: Context restoration and IRQ handling during suspend mode with no retention.

³This solution is required to avoid proprietary ARM Core code modifications.

Simulations and Comments

Using the flow described in Appendix C, power aware verifications are performed on the RTL description associated to an UPF power intent. The timing diagram of a gate level simulation is shown in Figure 4.20. Due to power gating, corruption of the signal SLEEP and SLEEPDEEP occurs. However, thanks to isolation strategies, the system recovers when a Non-Maskable Interrupt (NMI) is received.

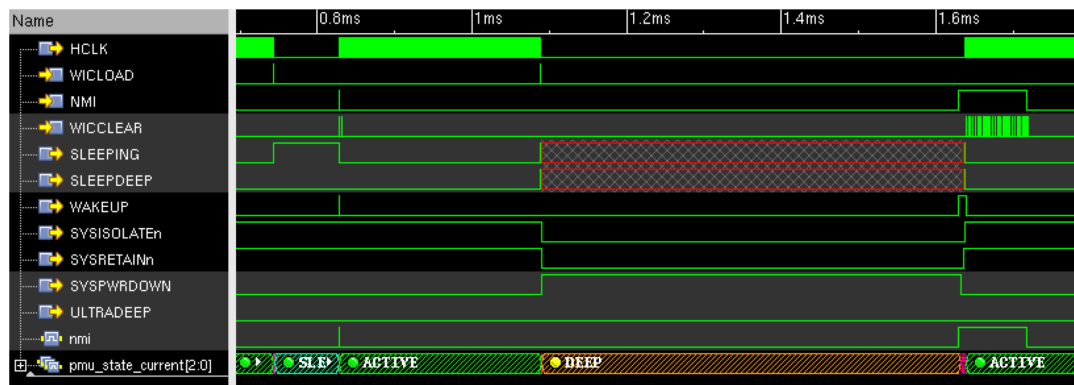


Figure 4.20: Simulated behavior of the PMU using standard EDA simulator and UPF power intent description.

Extra verifications are also supported by EDA tools for sign-off, such as: power grid integrity, voltage drops, or in-rush currents and ground bounce during powering up. In large industrial SoCs, these steps help sizing of the power gates and distribution networks and are complementary to the design methods proposed in the previous sections. Depending on the issue observed, mitigation techniques such as power stripes sizing, power gate duplication and staggering, etc... are available. These steps benefit in yield increasing after manufacturability.

4.3.2 Clock Gating and Frequency Scaling

This section explores the techniques applied on the clock signal and generators to reduce the power consumption at system-level. The techniques presented are combined with the previously described power modes.

Dynamic Frequency Scaling

DVFS is an ULP technique to decrease the system power consumption by reaching the minimum voltage and frequency for the proper system operation (see Section 1.5.2). To save subsequent power, only one voltage ULV source is used to power the whole SoC. Consequently, only Dynamic Frequency Scaling (DFS) can be implemented. However, during active mode, DFS only is not worthwhile to conserve switching power. Reaching the best energy efficiency requires Dynamic Voltage Scaling (DVS) too [203].

To implement a relevant alternative, this work fosters the utilization of DFS during suspend modes to reduce the overall power consumption. Indeed, during sleep sequences only the consumed power (in W) impacts the energy sources, while the best energy efficiency (in

pJ/cycle) is crucial when the system performs computations. As stated in Section 1.3.2, by switching to lower operating frequency, automatic power gains are observed.

Thereupon, several clocking solutions are integrated into the SoC as seen in Table 2.2. The ULP low frequency clock generator LRO (see Chapter 3) and the clock multiplier DDSS (presented in the following section) are integrated in the 28 nm FD-SOI implementations of this work. External 4 MHz and 32 kHz are also available for reference or testing purposes. To benefit from frequency scaling or clock gating during suspend modes a clock/power architecture is later derived.

MHz-range Frequency Synthesizer

In [204] is proposed clock generator which trades-off phase-locking for lower power and area. This custom digital alternative to PLLs is added in the 28 nm SoCs demonstrator to provide a flexible MHz-range clock source (see Section 4.5).

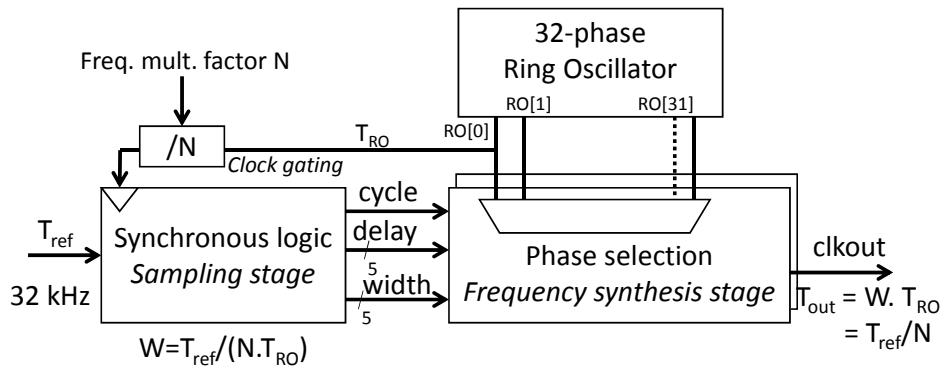


Figure 4.21: Block diagram of the Direct Digital Sampling and Synthesis (DDSS) clock generator [205].

As shown in Figure 4.21, the circuit provides a frequency output equal to a programmable multiple N of a low frequency reference. It operates on a free-running unregulated internal 32-phases oscillator which unregulated PVT-dependent period T_{RO} is first compared with a known fixed 32 kHz reference T_{ref} through a sampling stage via a simple counter. This stage generates a command output W inversely proportional to T_{RO} . Then, the synthesis stage preforms a frequency division of the same oscillator by the factor W provided by the first stage. This way, the output period is $T_{out} = W \cdot T_{RO} = T_{ref} / N$, which is independent of the un-calibrated T_{RO} period. The sampling stage provides an updated value of W every full period of T_{ref} (i.e., every 31.25 μ s).

To offer a fine time granularity of the output, the second stage frequency division is implemented by a fractional phase-selection divider, providing output periods with $1/32^{nd}$ increments (frequency synthesis stage). For instance, for a value of $W = 33/32$, the first output rising edge is generated by the phase 0 of the oscillator. Then, in the next cycle, the phase 1 is used, providing the desired period of $33/32 \cdot T_{RO}$. The oscillator is custom designed, while the rest of the circuit is automatically placed and routed using standard cells. The total clock generator area is 981 μ m².

Clock Gating and Frequency Switching

The complete clock gating and switching available in the SoCs is presented in the block diagram of Figure 4.22. For clock source selection at the top level of the SoC, several clock multiplexers are placed along the low frequency (kHz range) and high frequency (MHz range) clock signal paths. For instance, when `CLK1_SELECT = '1'`, the external clock CLK ref 1 is directly fed to the MHz clock input of the core. This also allows the selection between an external clock reference for the clock multiplier or the ULP clock. In such a way, these designs support high-level architectural clock gating. Moreover, the designer can determine which clock segments can be explicitly individually gated at system level. The necessary programming and initialization of the clock generators is performed using external dedicated interfaces.

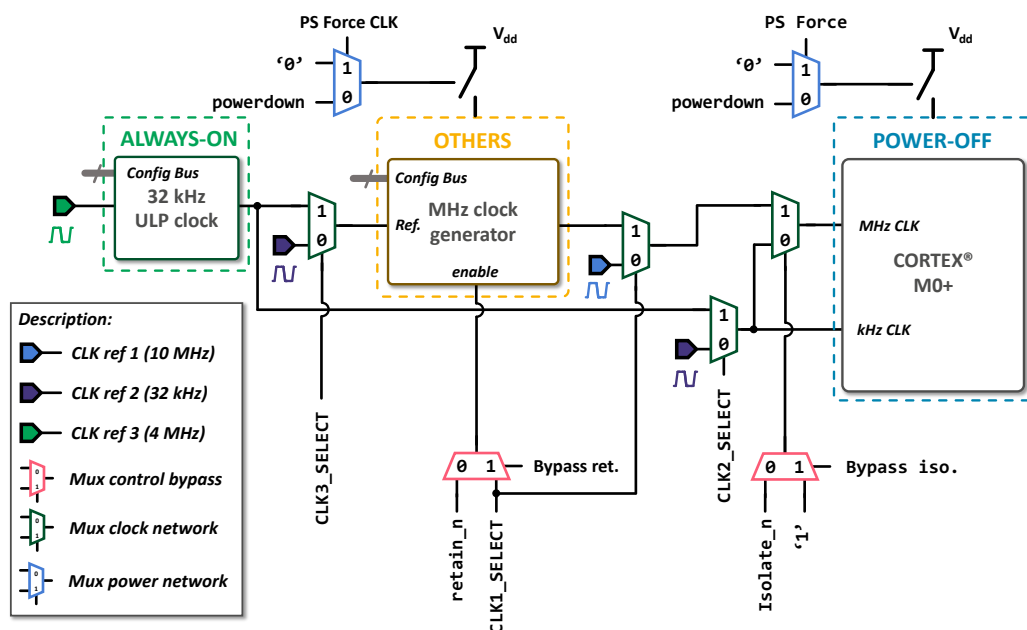


Figure 4.22: Block diagram of the DFS implementation at system level.

Using the signal generated by the PMU (see Figure 4.18), mode-related frequency switching and power gating are available. The blocks and clocks selected according to the mode are given in Table 4.3. In low power modes, the fastest clock frequency is no longer necessary. Consequently, by switching the main clock from a MHz range to a kHz, the power consumption of the A.ON is reduced. For further power saving, the main clock generator is disabled, and power gated in deep/ultra sleep mode and the low frequency reference clock is used. These extra modes, available using external signals will be used in Section 4.5.2.

Mode	ACTIVE	SLEEP	DEEP & ULTRA
Clock Range ¹	MHz	MHz	kHz
LRO	ON	ON	ON
DDSS	ON	ON	OFF
Extra Clock mode ²	f_1 (MHz-range)	$f_1/2$ (MHz-range) f_2 (kHz-range)	f_2 (kHz-range)

¹ As defined by the PMU.

² Available using external controls.

Table 4.3: DFS and clock generator modes.

Standard multiplexers are available to bypass the control signals from the PMU and thus disable some isolation, retention or power off functions of P.OFF. Currently, these cells are for instrumentation purpose. In an improvement perspective, the bypass signals could either be integrated into the PMU and programmed depending on the application requirements or simply be removed. All the multiplexers are instantiated on top level of the SoC and placed in the A.ON domain. Clock gating is also added to the design by the EDA synthesis tools. Assuming a synthesizable RTL description, transparent clock gating is inferred for common sub-expressions in the enable terms of synchronous logic.

4.3.3 Ultra-Low Voltage Memories

The systems presented in this work features 8 KB for instruction and program memories and 4 KB for data storage, both sized to enable decent program execution capabilities. At the time of the design, flash memories were not available in both FD-SOI technology nodes.

In 22 nm FD-SOI, the memory macros provided by Global Foundry, with low-voltage capabilities were used, while the 28 nm FD-SOI implementations integrate standard STMicroelectronics SRAMs with ultra-low operating voltage capabilities [206]. In both cases, the same supply voltage than the core is used.

The latter memory macros are based on 6-transistors (6T) bitcells, thus offering a very low static current in the pA/bit range compared to specific 8T or 10T architectures. A high-density area of $0.120 \mu\text{m}^2$ per bitcell and 0.0097 mm^2 for the 4 KB macro is obtained. To enable ultra-low voltage operation at 0.5 V and retention at lower supply values, read and write assist mechanisms are implemented [206]. Both are tunable using 4 margin bits. These margin bits can be set through dedicated registers, the code being extracted during wafer level test engineering. Although, these margins act as a process compensation feature to enable very low ppm: all the tested circuits in this work are fully functional with the default value.

4.4 Ultra-Low-Voltage Dynamic Body Biasing

In the previous sections, the application of aggressive voltage scaling, combined with technological to system-level techniques has been introduced. To be deployed on a mass market scale, ULP SoCs using ULV still require a minimum operating frequency as well as robustness over a reasonable temperature range.

In FD-SOI, the performance/power trade-offs and the temperature range can be leveraged using body-biasing [136]. However, these solutions rely on a specific body bias generator which automatically increases the SoC power budget and area [207]. Hence, this section explores the concept of body-biasing at ULV by adapting the SWBB technique on FD-SOI technologies (see Section 1.3.4). The bias generator is removed and replaced by discrete biasing steps to fit the strict power budget of $\sim 100 \mu\text{W}$. First, the technique is presented and evaluated on 28 nm and 22 nm nodes for frequency boost or temperature compensation. A resulting implementation is then described.

4.4.1 Concept of Back-Biasing at Ultra-Low-Voltage

From Bulk Swapped-Body Biasing to FD-SOI Body-Biasing

As seen in Section 1.3.4, swapped body biasing techniques applied at low voltage have demonstrated effective speed-up and performances improvements for static CMOS schemes [105] or dynamic implementations [137]. In this work, an alternative is adapted to the specificity of FD-SOI to enable energy and frequency trades-off.

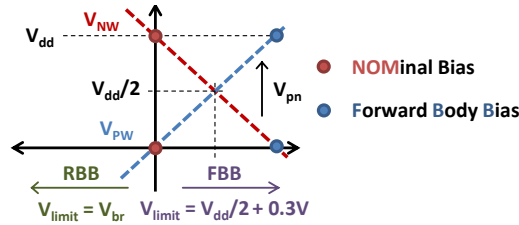


Figure 4.23: Proposed 28 nm FD-SOI body-biasing modes. The NOM condition is given by $V_{NW} = V_{dd}$ and $V_{PW} = gnd$, with V_{NW} and V_{PW} the body-bias voltage for the n and p wells (pMOS and nMOS). The second FBB mode supposes $V_{NW} = gnd$ and $V_{PW} = V_{dd}$.

Using the 28 nm FD-SOI RVT MOSFET triple well structure presented in Figure 1.9a, two discrete bias modes NOM and FBB are derived as shown in Figure 4.23. With this configuration the p and n wells are respectively connected between V_{dd} and gnd in NOM mode and “swapped” with the FBB mode. Since the main supply voltages are used to bias the wells, no external bias generator is required. The technique is applied to RVT flavor transistor instead of LVT to avoid excessive diode current while maximizing the forward range up to $V_{limit} = V_{dd}/2 + 300mV^4$.

The maximum supply voltage acceptable before leakage occurs is plotted in Figure 4.24 along with the voltage applied using the FBB mode. At 0.6V, the forward diode voltage is reached, which sets a theoretical bound to the technique. The leakage current observed in

⁴The forward diode voltage is limited by $V_{pn} \approx 0.6V$ while the reverse diode voltage accepts a few volts before voltage breakdown V_{br} becomes an issue (see Section 1.2.3).

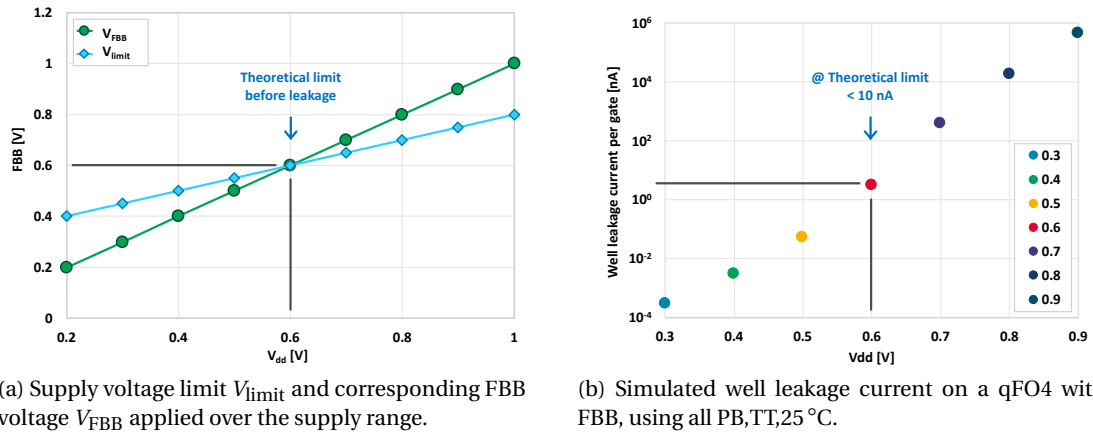


Figure 4.24: Evaluation of the supply voltage before well junction leakage appears using the FBB mode.

the well using the FBB is also reported in Figure 4.24b for a qFO4 path test structure (average over all the PBs in TT corner at 25 °C). Beyond 0.5 V, currents over the nW range appears, with a maximum of tens of nW at V_{limit}. Similar evaluation where done using fast corners and high temperature and a maximum of hundreds of nW at V_{limit} is observed. In any case, it confirms the limitation of the technique at higher voltage due to excessive leakage.

Optimizing the Energy/Frequency Trade-off

Body bias modulation is a known strategy to dynamically ensure digital circuits operations at a targeted frequency while lowering down the overall power consumption [137]. To reduce the current leakage in a suspend mode, RBB is applied to the wells while FBB is chosen in an active operating mode, increasing the circuit speed.

Using a generator, continuous body bias steps are available, yet the solution of this work offers only two discrete modes. Consequently, the frequency boost obtained by the technique must be significant to be used. This also presumes a dynamic utilization of the technique where the FBB is applied during active mode, while the system switches back to NOM in suspend sequences to reduce the leakage.

Temperature Compensation

ULV operations in FD-SOI help reaching the barrier of Sub-10-pJ/cycle energy consumption while maintaining reasonable operating frequency[208, 209]. However, the SoCs operation are restricted to limited temperature ranges.

Then, instead of optimizing the energy/frequency trade-off, SWBB can be used dynamically for temperature compensation. FBB boosts the performance while the temperature is decreasing yet maintaining the operating frequency. At high temperatures, switching from FBB to NOM operation minimizes the leakage.

Process Compensation

The swapped body biasing can also compensate process variations. To recenter slow or fast devices, static swapped body biasing is applied as shown in [210]. Since this last utilization is not compatible with energy/frequency or temperature compensation, it is not explored in this work.

4.4.2 Evaluation in 28 nm FD-SOI

The swapped body biasing technique presented in Section 4.4.1 is first applied to 28 nm FD-SOI. For characterization purpose, qFO4 structures presented in Section 4.1.2 have been configured with the full PB offer, resulting in several data points for given PVT configurations.

Energy/Frequency Trade-off

Figure 4.25 reports the evaluation of the frequency on a qFO4 test structure with and without the FBB applied according to the supply voltage. As shown in Figure 4.25b, by considering the frequency variation relatively to the NOM mode ($V_{NW} = V_{dd}$ and $V_{PW} = gnd$), a maximum appears at 0.4 V. This confirms the interest of SWBB at low voltage where FBB has more impact.

However, as ~ 10 MHz frequency is targeted, the 0.3 V operating supply can still not be considered, as only a maximum MHz frequency is observed after FBB application. Furthermore, the 0.3 V configuration is not compatible with the single supply voltage strategy and the memories minimum voltage. Thereupon, the analysis now focuses on a 0.4 V to 0.5 V supply range, leading to a relative frequency gain between $\times 1.93$ and $\times 1.75$.

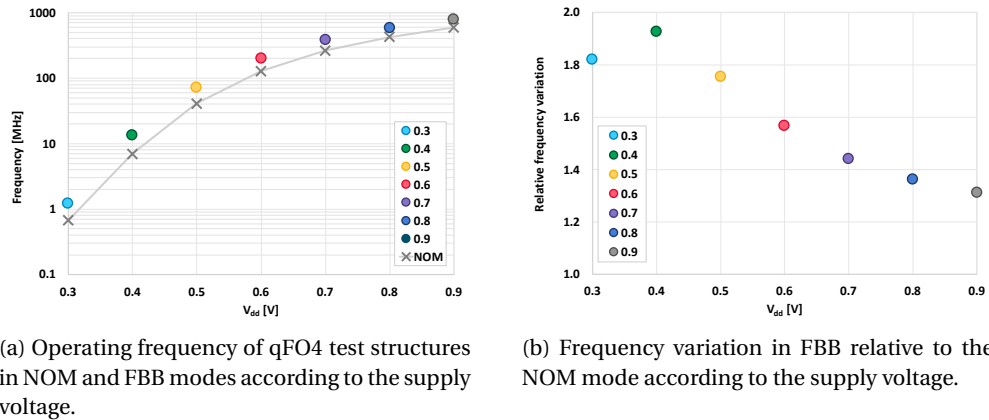


Figure 4.25: qFO4 frequency using NOM and FBB modes according to the supply voltage at TT/25 °C in 28 nm FD-SOI.

The energy/frequency trade-off is plotted in Figure 4.25 for 0.4 V and 0.5 V in TT corner at 25 °C. At 0.4 V, an averaged $\times 2.1$ frequency gain is observed on the whole PB offer at the cost of $\times 1.7$ extra energy, leading to a > 1 frequency/energy ratio. The extra frequency obtained is more significant than its energy cost. Moreover, assuming an implementation discarding the PB0 the FBB mode becomes less energy consuming for a given target frequency. At 0.5 V, the energy ratio become < 1 , with $\times 1.8$ frequency gain and $\times 2.0$ energy cost. Still, if a high

frequency is required for the system to operate, the FBB shows the best trade-off. The NOM presents the minimal energy consumption. This analysis confirms the performance and power trade-off using SWBB between 0.4 V and 0.5 V.

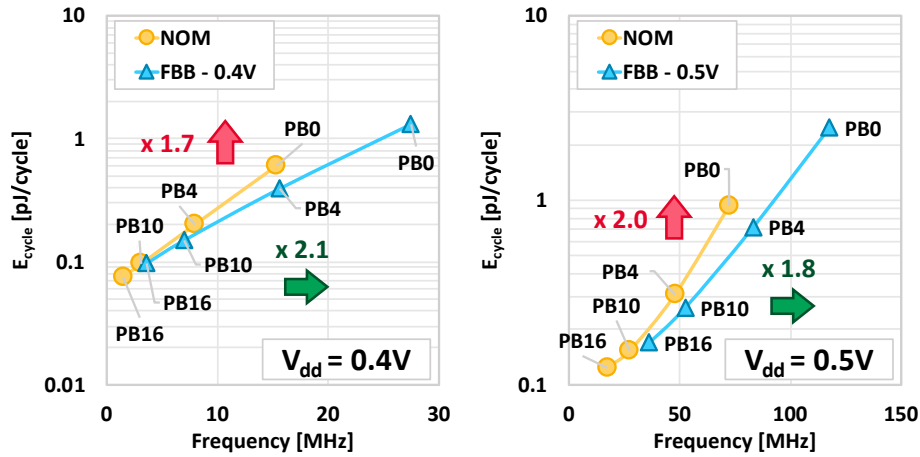


Figure 4.26: Evaluation of the energy per cycle according to the frequency for 0.4 V and 0.5 V in TT corner at 25 °C in 28 nm FD-SOI.

The FBB extra energy cost is compensated by the frequency gain during active mode of the system. However, the resulting increase in leakage power is detrimental during suspend modes. Figure 4.27 plots the leakage power of the qFO4 test structure using several PBs according to the supply voltage using a TT corner at 25 °C. A $\times 2.1$ and $\times 2.5$ increase consumption are observed at respectively 0.4 V and 0.5 V.

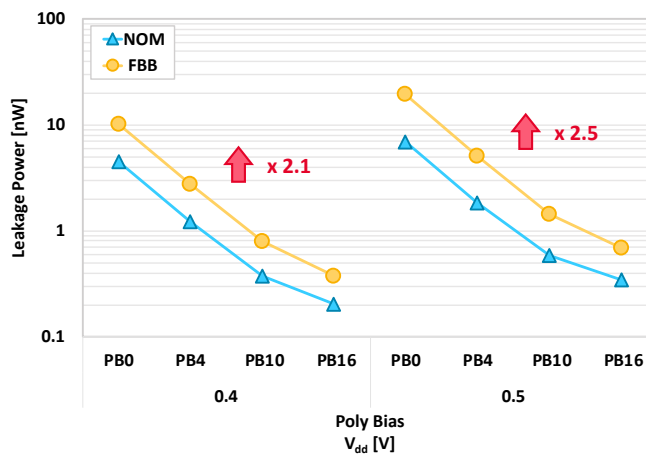


Figure 4.27: Leakage power of a qFO4 according to the PB in TT corner at 25 °C in 28 nm FD-SOI.

Consequently, body bias modulation is used as a strategy to dynamically ensure digital circuits operations at a targeted frequency while lowering down the overall power consumption. To reduce the current leakage in a suspend mode, NOM is applied while FBB is chosen in an active operating mode, increasing the circuit speed.

This approach assumes timing constrainting on the circuit. Let's assume a SoC operating at a given frequency F_{FBB} in FBB mode for maximum frequency and energy efficiency. During suspend mode, the system is switched to NOM in order to reduce the leakage power. Parts of the circuit are powered down or clock gated (i.e., insensitive to the frequency change) yet, some remaining blocks in the A.ON are still clocked by F_{FBB} . Thus, precautionary measures must be taken.

On the one hand, the A.ON should be able to work with F_{FBB} in both FBB and NOM. This is only acceptable if a small portion of logic is involved or rapid hardware is in the domain (e.g., memories). On the other hand, if the operating frequency cannot be maintained, the system should switch to a slower frequency F_{NOM} .

This second solution is adopted in this work by using the Direct Digital Sampling and Synthesis (DDSS) in two clock modes (16 MHz and 8 MHz). The 32 kHz clock reference – provided externally or using the LRO – is also used to reduce the dynamic power consumption that is involved in the suspend modes.

Temperature Compensation

As FD-SOI has proven a tens of MHz operations capabilities at ultra-low voltage, the boost offered by the FBB can also be used to maintain a SoC operating frequency over a wide temperature range. Using the same qFO4 test structure, the simulated frequency obtained is given in Figure 4.28 for different voltage supply and from -40°C to 125°C using a typical corner.

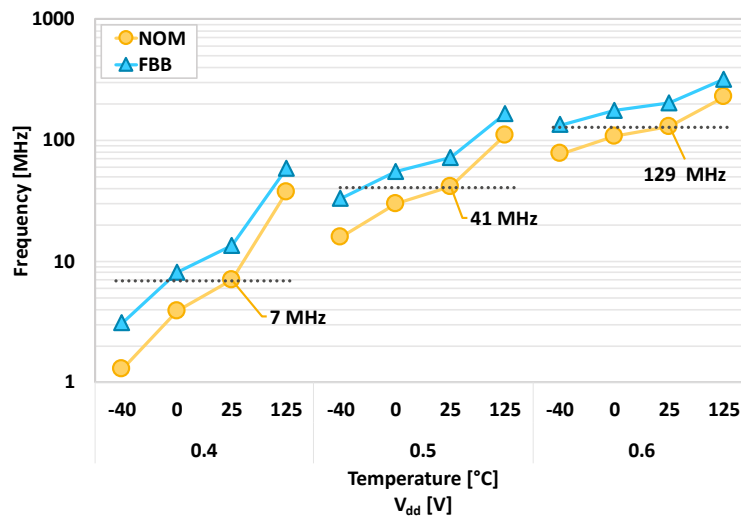


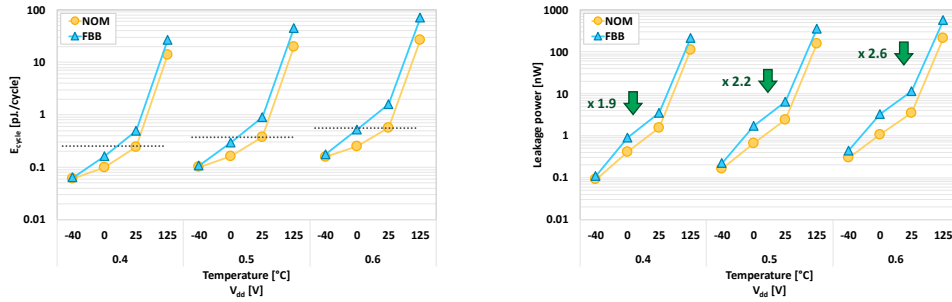
Figure 4.28: Frequency of a qFO4 according to the temperature and the supply voltage using a TT corner, over all PBs in 28 nm FD-SOI.

The nominal operating frequency at 25°C is highlighted using a dash line for three different supply voltages. At 0.4 V, the 7 MHz F_{NOM} can be recovered with the FBB modes at 0°C . To increase the range of temperature accessible while maintaining the frequency, the supply voltage should also be increase. At 0.5 V, F_{NOM} is still guaranteed below -20°C , while only a sufficient 0.6 V supply voltage can guarantee the frequency recovery with biasing at -40°C .

With IoT-oriented applications, devices are expected to work in limited temperature

conditions such as -20°C to 80°C , or even -0°C to 60°C in controlled environments like buildings or home-related applications [33]. In this case, the operating supply voltage can be dealt with the operating frequency and the extra power needed for a specific temperature range. This validates the trade-off offered by SWBB applied for temperature compensation.

In addition, switching between NOM and FBB mode impacts the power budget. In Figure 4.29 are reported the energy per cycle and the leakage power on each mode according to the supply voltage and the temperature. Both energy (see Figure 4.29a) and leakage (see Figure 4.29b) scale with the temperature. Thus, by switching to FBB mode, the power consumption relatively to the NOM at 25°C does not increase.



(a) Energy per cycle according to the temperature.

(b) Leakage power consumption according to the temperature

Figure 4.29: Power consumption of a qFO4 according to the temperature and the supply voltage using a TT corner over all PBs in 28 nm FD-SOI.

4.4.3 Evaluation in 22 nm FD-SOI

From the previous learnings, the SWBB technique is similarly applied to the 22 nm FD-SOI node. However, due to different operating conditions of this technology node, some mandatory adjustments on the body bias must be made. First, this section explores the modifications made to the SWBB technique. Then, the energy/frequency trade-off and temperature techniques are discussed.

Adjustments for 22 nm FD-SOI

In 22 nm FD-SOI, considering an RVT flavor MOSFET, the well structure is the same than 28 nm FD-SOI as shown in Figure 1.9a [83]. Be that as it may, the nominal biasing condition NOM is given by $V_{PW} = V_{NW} = gnd = 0\text{ V}$, with respectively V_{PW} and V_{NW} the body-bias voltage for the pMOS and nMOS devices (see Figure 4.30, left side). Hence, in NOM mode, to avoid excessive diode current, the forward diode voltage results in $V_{limit} = 0.3\text{ V}$.

Contrary to the 28 nm FD-SOI RVT, the pMOS transistor is naturally boosted. This is needed due to the significant hole-mobility degradation in advanced technology node. Generally, strain doping is used to restore the performance of p-type devices. In this case, it is also coupled with static body biasing.

Adapted from the 28 nm FD-SOI, is derived an asymmetrical reverse body biasing mode RBB with $V_{NW} = V_{dd}$ and $V_{PW} = gnd = 0\text{ V}$, and an asymmetrical forward body biasing mode

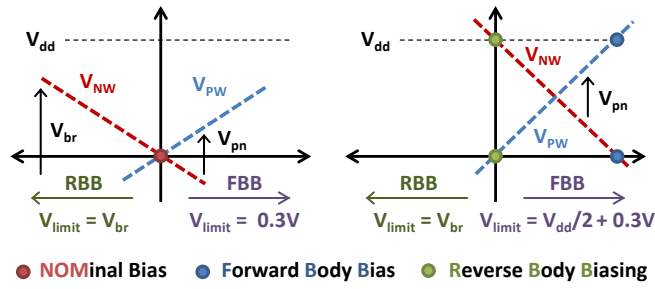


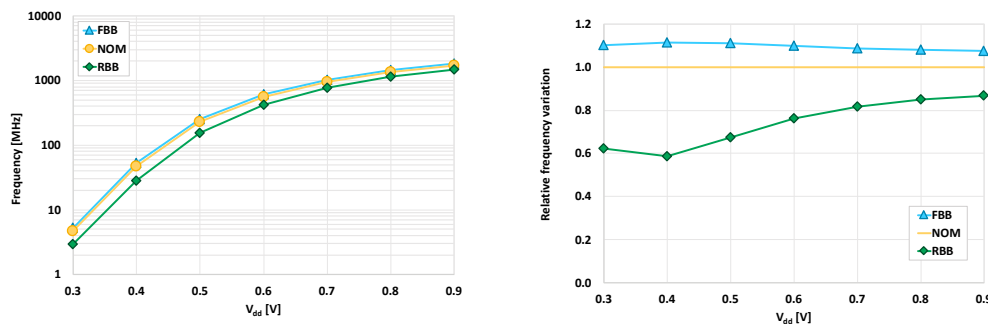
Figure 4.30: Proposed 22 nm FD-SOI body-biasing modes.

FBB with $V_{NW} = gnd = 0V$ and $V_{PW} = V_{dd}$. For forward body biasing, the diode results in $V_{limit} \approx V_{dd}/2 + 0.3V$ before the forward p-well/n-well diode current becomes prohibitive.

These implementations are restrictions of the full body bias possible range. They results in a lowered efficiency of the FBB and RBB modes with regard to the SWBB 28 nm FD-SOI application. In return, they still do not rely on negative or over V_{dd} supply voltages.

Energy/Frequency Trade-off

Figure 4.31 reports the evaluation of the frequency on a qFO4 test structure in NOM mode and with RBB or FBB applied according to the supply voltage using a TT corner at 25 °C. By considering the frequency variation relatively to the NOM mode (see Figure 4.31b), the gain brought by FBB is flattened along the supply range. However, a maximum is reached at 0.4 V, with a $\times 1.1$ relative deviation. It also corresponds to the maximum deviation in RBB, with a $\times 0.59$ factor. The 0.4 V supply voltage point is thus chosen since it offers the maximum deviation between modes.



(a) Operating frequency of a qFO4 test structures in RBB, NOM and FBB modes according to the supply voltage.

(b) Frequency variation in RBB and FBB relative to the NOM mode according to the supply voltage.

Figure 4.31: Evaluation of a qFO4 frequency using NOM and FBB modes according to the supply voltage at TT/25 °C in 22 nm FD-SOI.

The energy per cycle according to the frequency for 0.4 V and 0.5 V in TT corner at 25 °C in 22 nm FD-SOI for the three available PBs is reported in Figure 4.32a. Switching from NOM

to RBB, the energy consumption is reduced by $\times 0.86$, yet it scales the operating frequency by a factor $\times 0.58$. From NOM to FBB, the operating frequency is this time increased by $\times 1.1$ and the energy increased by $\times 1.8$.

According to these results, the NOM mode in 22 nm FD-SOI already appears as a proper technology trade-off between frequency and leakage. This hypothesis is also confirmed by the evaluation of the leakage power of a qFO4 according to the PB in TT corner at 25 °C in 22 nm FD-SOI that is plotted in Figure 4.32b. The utilization of the NOM mode only impacts the leakage power by a factor $\times 1.17$ ($= 1/0.85$).

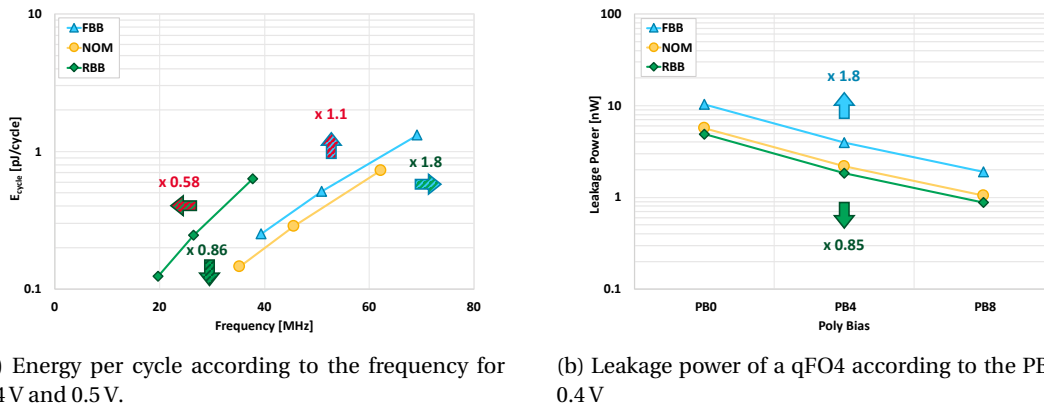


Figure 4.32: Performance analysis of a qFO4 according to the SWBB mode in 22 nm FD-SOI. Results are obtained for all PB in TT corner at 25 °C.

This phenomenon and the prevalence of the NOM mode over the two others can be explained by the asymmetrical nature of the body-biasing schemes. As stated in Section 4.4.3, it is due to the already available nominal operating condition of the 22 nm FD-SOI technology. In RBB mode, whereas the pMOS transistors are effectively biased in reverse (i.e., the threshold voltages of the device increases), the nMOS transistors remain in their nominal operating conditions. On the contrary, in FBB mode, the pMOS transistors conserve their nominal operating conditions whereas the nMOS transistors are this time biased in forward (i.e., the threshold voltage of the devices decreases). Therefore, as only a part of the nMOS and pMOS network is successively biased, the impact of the RBB and FBB modes compared to the FBB/NOM implementation in 28 nm FD-SOI is tenuous.

Figure 4.33 shows the propagation of a rising edge along a qFO4 path at 25 °C using a typical corner at 0.4 V. Each node of the path is plotted, node N corresponding to the output of stage N along the path. On the one hand, when RBB is applied, the slope is rapidly degraded at each node due to the slowdown of the pMOS transistors. At this low voltage operation, due to a limited driving strength the following stage cannot restore the slope, which is progressively degraded. On the other hand, when FBB is applied, the slope along the path increases. However, it rapidly reaches the maximum driving capabilities of the various stages (composed of different cells).

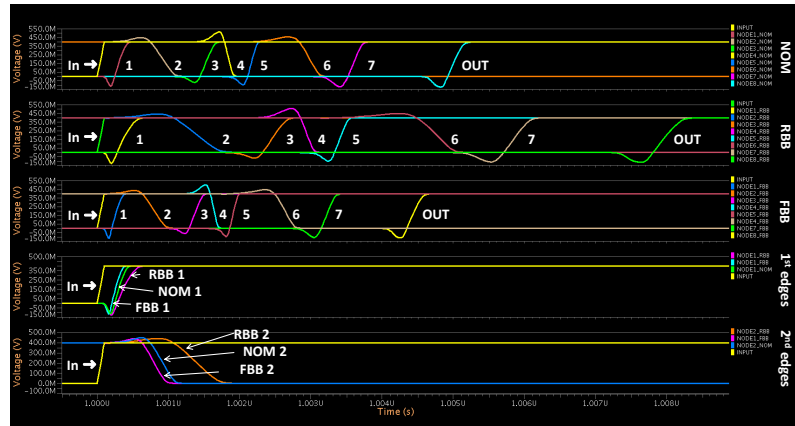


Figure 4.33: Rising edge propagation according to the bias mode along a qFO4 using a TT corner at 0.4 V/25 °C in 22 nm FD-SOI.

Temperature Compensation

The simulated frequency obtained for qFO4-based critical paths are given in Figure 4.34 for different voltage supplies and from -40 °C to 25 °C. From 0.4 V, the 28 MHz F_{MAX}^{RBB} can be recovered both with the NOM and FBB modes at 0 °C. Only a sufficient 0.5 V supply voltage can guarantee the frequency recovery with biasing at -40 °C.

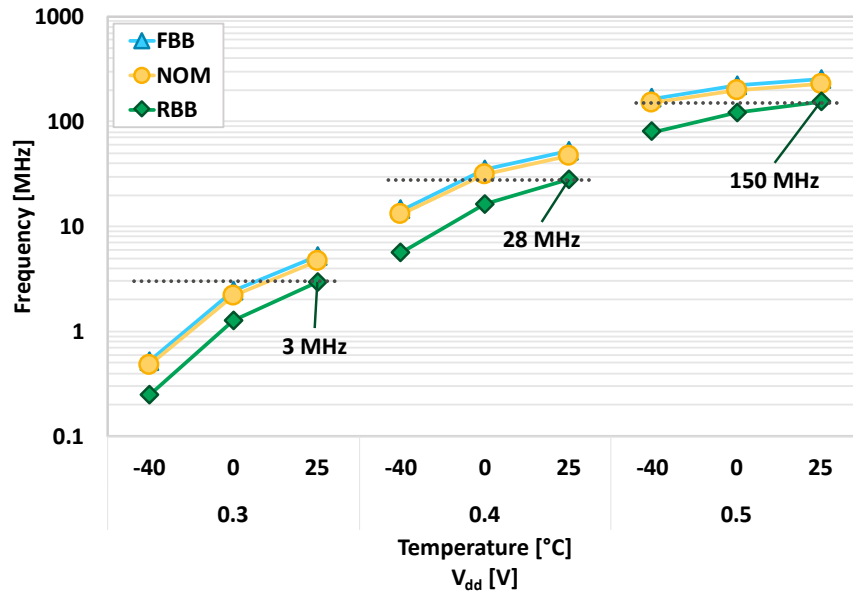


Figure 4.34: Frequency of a qFO4 according to the temperature and the supply voltage using a TT corner over all PBs in 22 nm FD-SOI.

The impact on the power budget of mode switching is reported in Figure 4.35. Due to the limited results obtained by comparing the extended bias to the nominal mode, a second approach is to consider only FBB and RBB. Hence, a minimum energy consumption and leakage power would be guaranteed by the RBB mode, while the FBB help maintaining the frequency

along the temperature range. However, this solution comes at the cost of the maximum frequency which is now reduced. Moreover, leakage recovery is no longer a practicable solution at higher temperatures.

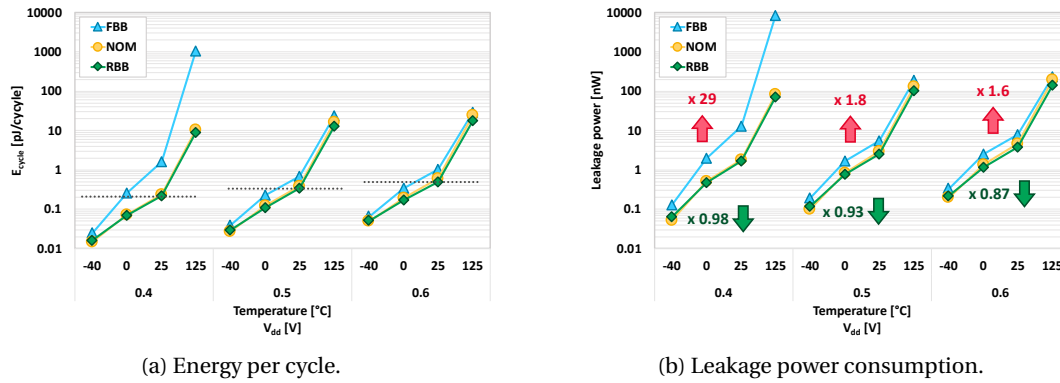


Figure 4.35: Power consumption of a qFO4 according to the temperature and the supply voltage using a TT corner and over all PBs in 22 nm FD-SOI.

4.4.4 System Integration

In this section the implementation and integration of the SWBB technique is detailed. It is based on a complete digital standard flow enhanced with a self body biasing design methodology. A temperature compensation scheme is developed and combined with various power modes for a minimum Active and Sleep energy consumption. By enabling body biasing the frequency/power consumption can also be traded-off. The system integration is shown using the 22 nm FD-SOI technology. All the discussions are fully compatible with the 28 nm node.

Self Body-Biasing using Inverter Standard Cells

For a tiny ULP SoC, the area and power impact of a body bias generator is significant and could undermine the benefit of body bias. Currently, the best State-of-the-Art bias generator power consumption is in the μ W range [207]. Considering an ULP system with a hundreds of μ W constrained power budget in active mode, these figures would represent $\sim 10\%$ of the SoC power and $\sim 10\%$ of the overall area. In advanced suspend mode for which the power is even further reduced ($\sim 1 \mu$ W), it even leads to a 90% overhead. Then, a solution demonstrating a low-overhead, fully synthesizable and zero-trim all digital biasing scheme is proposed for the FD-SOI technology.

As shown in Figure 4.36, it uses the V_{dd} and gnd power supplies available through pairs of inverters (i.e., bias switches) to produce the bias powers V_{NW} and V_{NPW} supply voltages. The digital control signals CTRL p-well and CTRL n-well are then adjusted to set the NOM, RBB or FBB modes. The inverters are selected among the standard cells offer. To properly size the bias switches, the well diodes and the MOSFET, capacitances have to be modelled accordingly to the floor plan which host the SWBB technique.

The diodes parameters – which limit the bias voltage to be applied – are extracted directly from the core layout preliminary completed with fillers. A model correlation is performed

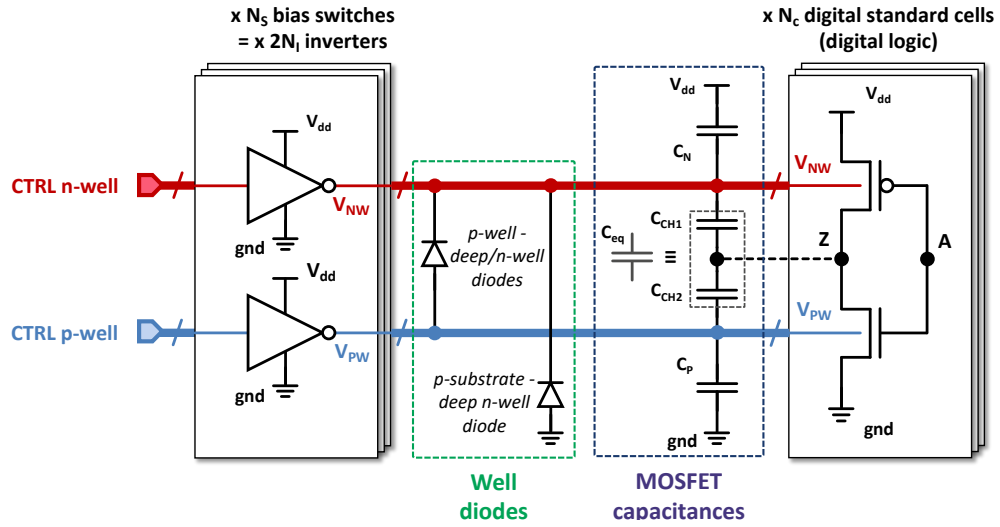


Figure 4.36: Self body bias generator using standard inverters. The well diodes model the p-well/n-well and deep n-well/p-substrate diodes under the BOx. The MOSFET capacitance model reproduces the standard cells capacitance (like source-box-well transistor capacitances) and the spread capacitances along the V_{NW} and V_{NPW} supplies. C_{eq} is the equivalent model of C_{CH1} and C_{CH2} , assuming an averaged activity on Z. The large triple well / p-substrate capacitance under the body-biased area is omitted since it lies in series with a large substrate resistance.

with previous measurements of power consumption in the wells. The same technique is used to size a body bias generator in FD-SOI.

The capacitive loads are based on standard models which result from technology characterization⁵. Using a generic MCU implementation, the standard cell with the most occurrence has been extracted. Then, a test structure composed of thousands of these cells is made. By doing AC measurements on the test vehicle while ensuring a stimulation on its inputs, a general transistor capacitance per area is obtained.

A $\sim 1 \text{ nF/mm}^2$ order of magnitude is obtained for C_N , C_P and C_{eq} (the equivalent model of C_{CH1} and C_{CH2} , assuming an averaged activity on Z). The capacitance models are required for dynamic adjustments of body biasing and stability evaluation of potential control loops. In this case, only two bias steps are required with no intermediary point. Then, the effect of the capacitance is limited, and only an inverter with enough drive strength is required.

Finally, 7 pairs of 24 nm gate-length, RVT-flavor inverters with a strength equivalent to $\times 16$ unitary-finger inverters were chosen. The power switches sizing and integration do not impact the digital synthesis and place & route; they easily integrate in a standard digital flow compatible with most EDA tools as shown in the next section.

⁵The standard cells MOSFET capacitances can also be extracted from Alternative Current (AC) simulation of the cell using post-layout netlist without the wells' diodes. As the channel capacitance C_{CH1} and C_{CH2} are dependent of the polarization of the output node Z, the extraction requires an average activity on system studied.

Floor-planning

Figure 4.37 sketches the floorplan modifications required for the integration of the SWBB technique. The implementation relies on a top bias header power domain and the utilization of main external bias power stripes V_{PW_ext} and V_{NW_ext} ⁶.

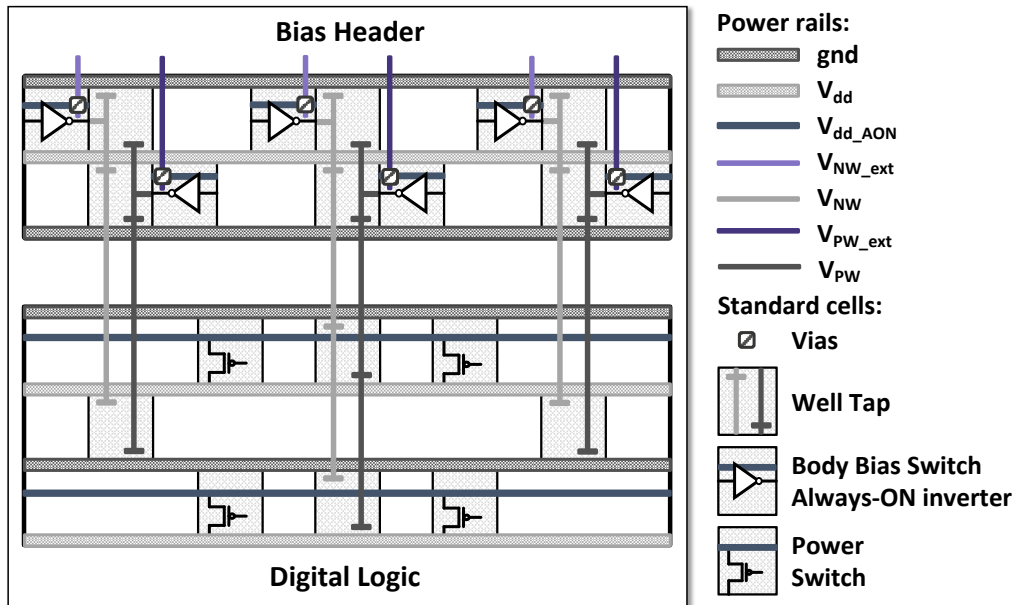


Figure 4.37: Overview of the system floorplan with integrated self body-biasing.

Always-on type inverters are distributed by pairs for V_{PW}/V_{NW} in the Bias Header. Their power supply is done through additional stripes V_{PW_ext} and V_{NW_ext} , connected to their always-on power input. The cells outputs are connected with the bias stripes together along with the well taps. In the main power domain filled with digital logic and distributed power switches, the bias is distributed through well taps.

A layout abstraction of this implementation is proposed in Figure 4.38. The power controls are ensured by CMD PW and CMD NW on the always-on inverter. They are automatically routed by the tool. Standard well taps are used, and fillers added to the bias header domain for density requirements. All this implementation can be described using the IEEE 1801-2009 UPF standard.

⁶An implementation using the main power stripes V_{dd} and gnd can be also derived.

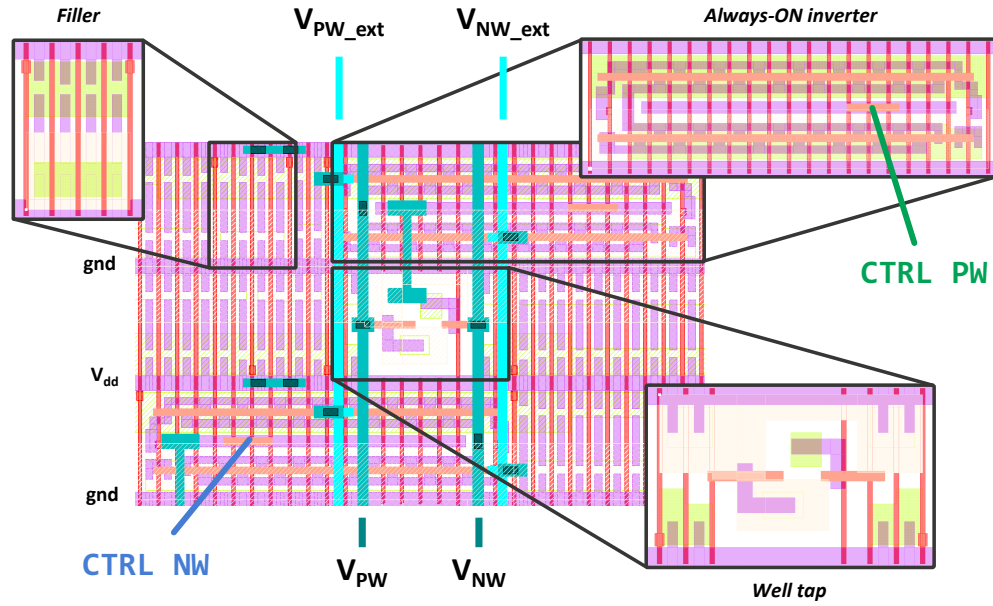


Figure 4.38: Layout abstraction of the SWBB implementation with external access to the bias stripes in 22 nm FD-SOI

Sequencing with the PMU

SWBB commands are controlled using the SoC PMU through a 2-bits configuration vector and the powerdown signal as shown in Figure 4.39. The first three configurations 00, 10 and 01 are used for static selection of the bias mode and generation of CTRL n-well and p-well signals. This helps for instrumentation and evaluation of the mode. Moreover, combined with an external temperature sensor and pre-configured threshold value stored in a backup memory, discrete dynamic compensation can be performed.

An automatic mode is finally available with the 11 configuration. The powerdown signal from the PMU is then fed to the control signals of the wells. During the deep sleep mode, the RBB mode is applied, whereas the system runs using the FBB mode in active state. This adaptive scheme helps minimizing the leakage power while ensuring a high operating frequency.

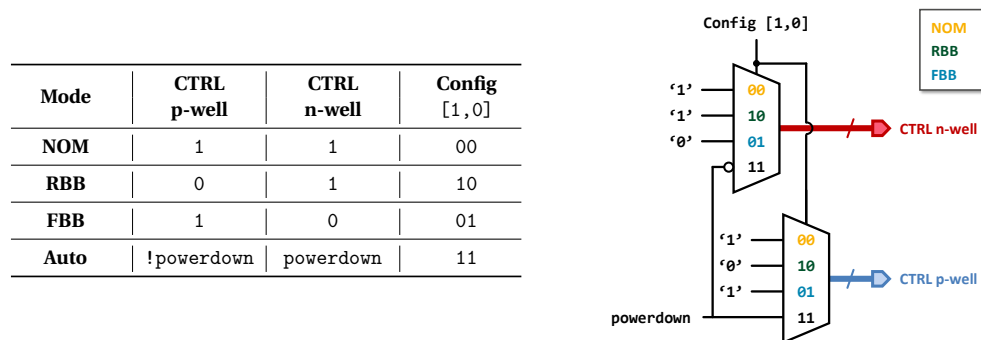


Figure 4.39: Self body bias generator mode selection through PMU controls.

4.5 Silicon Implementations and Measured Performances

This section introduces the silicon realizations made during this work to evaluate and characterize the low power techniques implemented all along this chapter. All the SoCs rely on the FD-SOI technology either in the 28 nm or 22 nm nodes, with ULV operations.

In Section 4.5.1 are presented the various ULV SoCs taped-out to demonstrate ULP techniques from technology to circuit level. The measurement setup created to evaluate their performances is then provided in Section 4.5.1. Next, in Section 4.5.2 is given the measurements of a primary circuit in 28 nm FD-SOI. It will be used as a reference for performance evaluation of the technology and the ULP techniques. The impact of SRPG is consequently evaluated to assess the trade-off between state retention and power consumption. The technology scaling impact is then estimated through a realization in 22 nm FD-SOI. Lastly, the self-body bias technique is measured and analyzed in Section 4.5.5.

4.5.1 Testchips Realizations

The NZP SoCs are integrated into several testchips and presented in the following subsections. A summary of all the SoCs is given in Section 4.6, Table 4.9. A complete test chip gallery is also available in Appendix G, Table G.1. All the SoCs are packaged using a standard 304 pins SBGA package. A test and instrumentation setup built around a Kintex-7 Field-Programmable Gate Array (FPGA) board is used as described in Appendix E.

NZP28 – V1.0

The first NZP28 V1.0 (see Figure 4.40a) was fabricated using RVT 28 nm FD-SOI technology from STM. The system area is 0.073 mm^2 . It is composed of three ULV power domains as shown in Figure 4.40b.

An A.ON integrates a custom PMU, a one-cycle switching frequency synthesizer DDSS (see Section 4.3.2), and event triggered peripherals. A second P.OFF domain includes the ARM Cortex-M0+ core, the system and peripheral buses and interfaces, a custom power switch and some peripheral. Lastly, a Retention domain combining 3 SRAM banks of 4 KB each. One for data memory (DMEM) and two for program/instructions memory (PMEM)⁷. All these power domains can be measured separately.

The PMU is implemented as an FSM to enable active and suspend modes at very low power cost. as no retention is implemented in the P.OFF, only the sleep and ultra deep modes are available. The peripherals included are; a timer controller and 10 GPIO ports in P.OFF, an SPI, an RTC, anWIC and the ARM DAP in the A.ON. They are memory-mapped and connected to the core through the APB bus. Currently, the DAP is not switched-off during suspend modes as it offers access to the core. For debug reasons it has been kept in the Always-On to ensure the correct wake-up of the core.

⁷Due to an error in address decoding in the memory controller to select the second PMEM memory cut, only 4 KB are accessible. It has been fixed in the later versions.

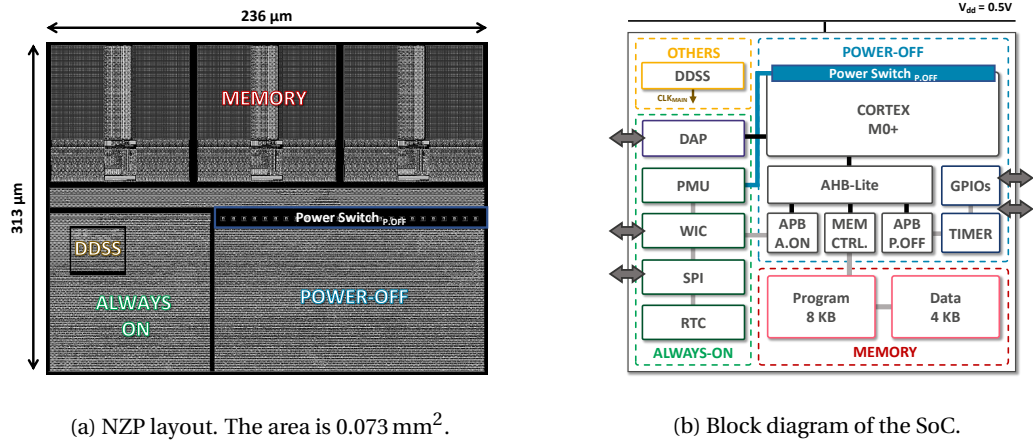


Figure 4.40: NZP28 V1.0 – 28 nm FD-SOI implementation.

NZP28 – V2.0

The second NZP28 – V2.0 (see Figure 4.41) is also fabricated using RVT 28 nm FD-SOI technology from STM. The system area is 0.082 mm². It shares a similar architecture with the NZP28 V1.0 yet, it includes a supplementary ULP 32 kHz clock reference called the LRO (see Chapter 3). Moreover, a second version of the PMU is included for instrumentation of the clock generators during the various power modes of the system. Only one power pad is available for power measurements.

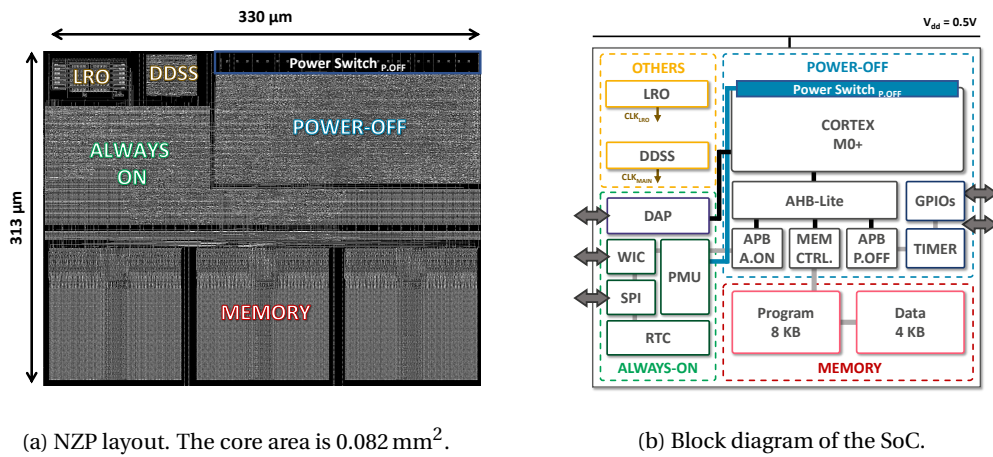


Figure 4.41: NZP28 V2.0 – 28 nm FD-SOI implementation.

NZP28 – V3.0

The last implementation in 28 nm FD-SOI technology is also fabricated using RVT from STM (see Figure 4.42). The system area is 0.082 mm^2 . Contrarily to NZP28 V1.0 and V2.0, this SoC embeds retention registers into the P.OFF domain, allowing SRPG. A power switch is also added on top of the DDSS for power reduction in suspend mode as shown in Section 4.3.2. Only one power pad is available for power measurements. A third version of the PMU is introduced here for automatic clock switching and IP power gating depending on the SoC suspend modes.

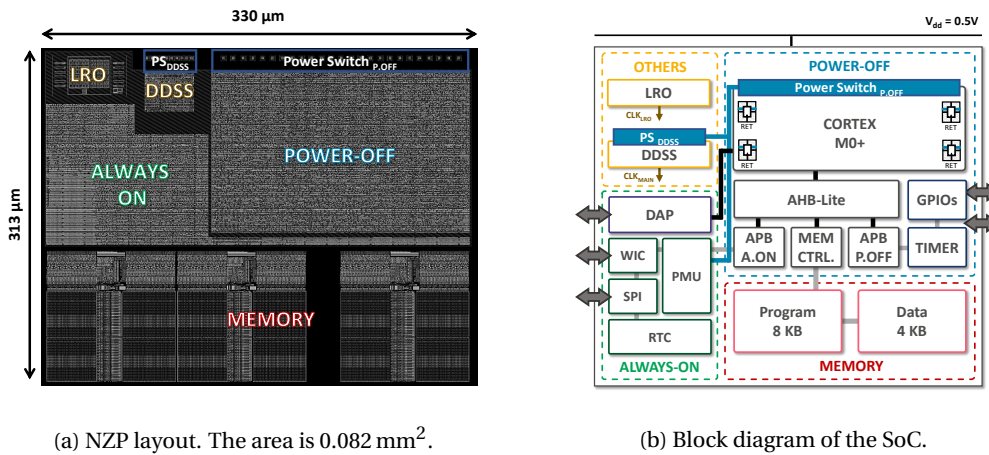


Figure 4.42: NZP28 V3.0 – 28 nm FD-SOI implementation.

NZP22 – V1.0

In 22 nm FD-SOI, an SoC has been designed to demonstrate the adaptive self body biasing techniques on an RVT implementation and evaluate the performances of the technology at ULV (see Figure 4.43).

It shares a similar architecture with the previous NZP SoCs in 28 nm FD-SOI. It embeds 4 power domains operating at the same fixed voltage V_{dd} . A P.OFF domain includes an ARM Cortex-M0+ core, an AHB controller connected to APB buses/memory controller, the ARM DAP and switchable peripherals. The whole domain can be powered down using distributed standard cells power switches. A MEM domain is integrated with 3x4 KB SRAM banks for program and data memory. An A.ON domain integrates event triggered peripherals, a WIC, the DAP, a 16 B memory (for temperature sensor calibration storage) and a custom PMU.

The PMU is also designed as an FSM which depending on the SoC modes – active and suspend Modes (sleep and ultra deep) – enables clock or power gating for P.OFF, or controls the bias power switches, located in a separated power domain (Bias Power Switch). The A.ON, P.OFF and MEM domains share the same biasing rails, but only the interface logic in the MEM can be biased. The memory macros bias rails are always kept in nominal biasing conditions. Dedicated power pads are available for each power domains.

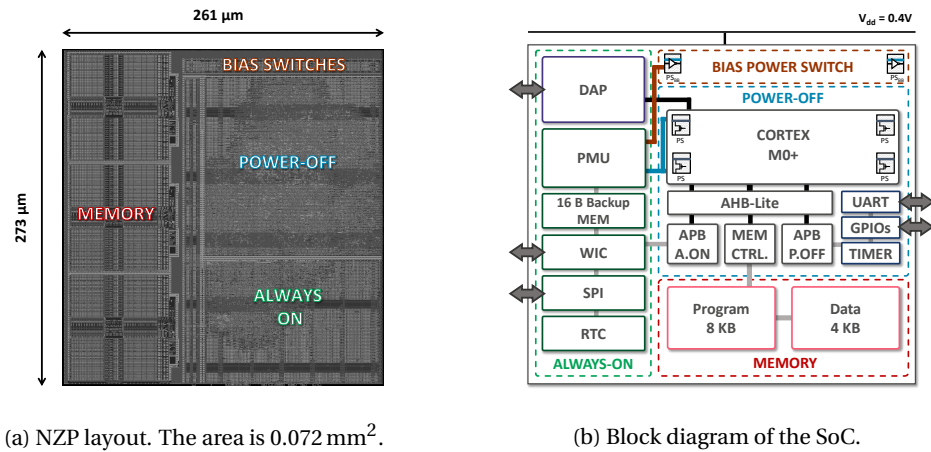


Figure 4.43: NZP22 V1.0 – 22 nm FD-SOI implementation.

NZP22 – V2.0

The last implementation is based on LVT 22 nm FD-SOI as shown in 4.44. Double power-gating is combined with retention register – as presented in Section 4.2.2 – allowing double power gating SRPG in the P.OFF domain.

The peripheral and memory integration is the same than NZP22 V1.0. However, the hardware power management of the SoC has been reorganized between a PMU, a WUC and an APM hardware module (see Chapter 5, Section 5.4). PMU and WUC are integrated in the A.ON whereas the APM is placed in a separated switchable power domain. Power mode change during the system activity can thus be defined by the software or given by the APM module. Dedicated power pads are used for power measurements on P.OFF, MEM and A.ON (the APM power grid is shared with the A.ON).

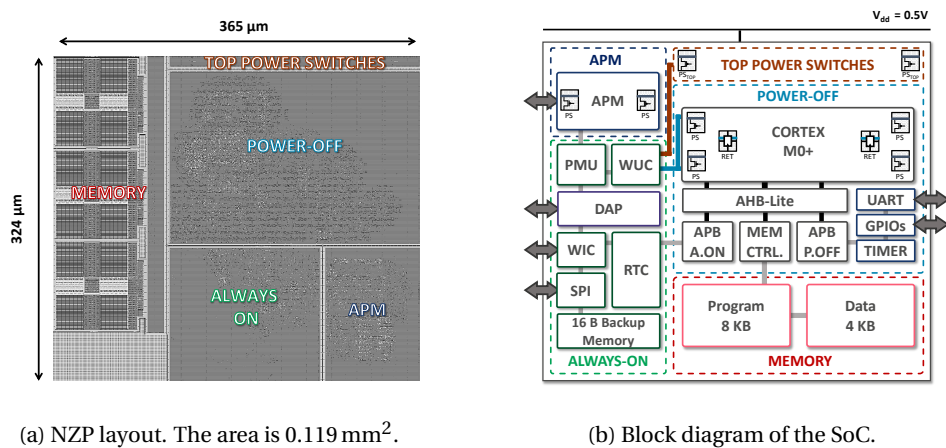


Figure 4.44: NZP22 V2.0 – 22 nm FD-SOI implementation.

4.5.2 Power Consumption Evaluation using ULP Modes and Techniques

This subsection presents the results obtained on NZP – Space Qualification (SQUAL) V1.0. This first 28 nm FD-SOI demonstrator gives baseline results to analyze the impact of ULP design choices and techniques in this technology. Moreover, the utilization of the available low power modes is evaluated for adaptive power reduction and efficient interaction between the SoC and the running application

Benchmarking and Measured Performances

A C-code program running the Dhrystone benchmark enhanced with sleep and ultra deep operations is used as software for yield measurements on 74 dies at wafer level, and performances measurements on 5 packaged dies, at 25 °C for a TT corner lot. The cumulative percentage of working parts from 0.45 V to 0.5 V is reported in Figure 4.45, as extracted from the wafer. These silicon results show the reference processor can be supplied down to 0.48 V, within a parametric loss of less than 2% as opposed to the 100% yield at 0.5 V.

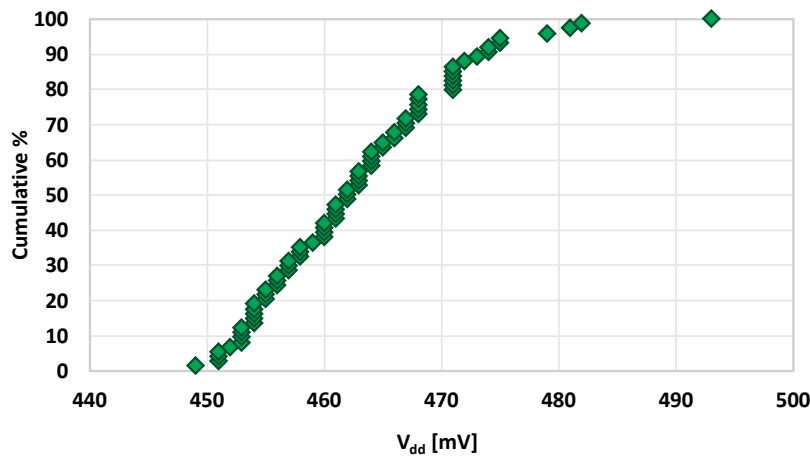


Figure 4.45: Cumulative percentage of NZP28 V1.0 working parts (74 dies) according to the supply voltage at 16 MHz/25 °C with no bias.

Speed performances obtained are given in Figure 4.46 for $V_{dd} \in [0.45\text{ V}; 0.65\text{ V}]$. The mean maximum clock frequency goes from 8 MHz at 0.47 V to 150 MHz at 0.65 V. The targeted 16 MHz is reached at 0.5 V with a mean at 16.7 MHz, a minimum value 16 MHz and maximum value 19 MHz. The 100% yield limit covering all biasing conditions is also reported at 0.5 V. The origin of the failure is mostly due to the fixed operating frequency since more devices are still running at lower frequencies (at 0.5 V). However, some limitation are still observed, even at low frequency, due to other failure mechanisms (memories, hold, etc...) but the design was not validated for these conditions.

In the context of energy harvesting, the system needs to target the lowest leakage current. In 28 nm FD-SOI, when RVT transistors are used, RBB can be used for drastic reduction of the transistor current (see Section 1.2.3). However, this technique impacts the operating frequency of the system if used in active parts. In Figure 4.46, the evolution of the mean F_{max} for

a given RBB voltage is plotted. Bias ranges from No Bias ($V_{\text{bias}} = 0$ mV) to a value $V_{\text{bias}} = 500$ mV with 100 mV increments. A frequency drop of 52% is observed at $V_{\text{dd}} = 500$ mV between No Bias and 500 mV bias. Below the 0.5 V supply voltage, crossover happens between bias conditions due to measurement failures leading to limited statistical relevance. This phenomenon also highlights the impact of RBB on the minimum supply voltage.

The bias voltage is used symmetrically on both pMOS and nMOS transistors of the SoC through an external generator; $V_{\text{PW}} = -V_{\text{bias}}$ and $V_{\text{NW}} = V_{\text{dd}} + V_{\text{bias}}$. Currently, bias can be applied over all SoC domains, yet it does not affect the memory macros and the powerswitch. In fact, these two components are self-biased and isolated using Deep n-well. RBB utilization in sleep and ultra deep modes is scheduled for static power consumption reduction (Section 4.5.2 & Section 4.5.2).

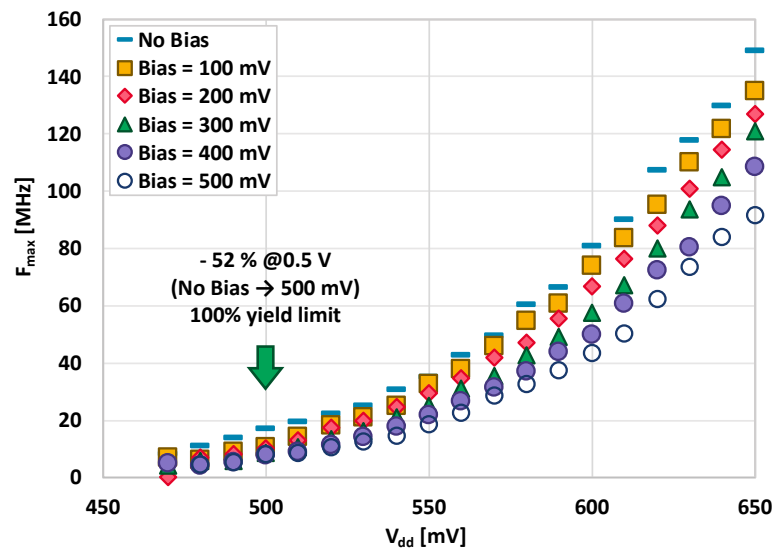


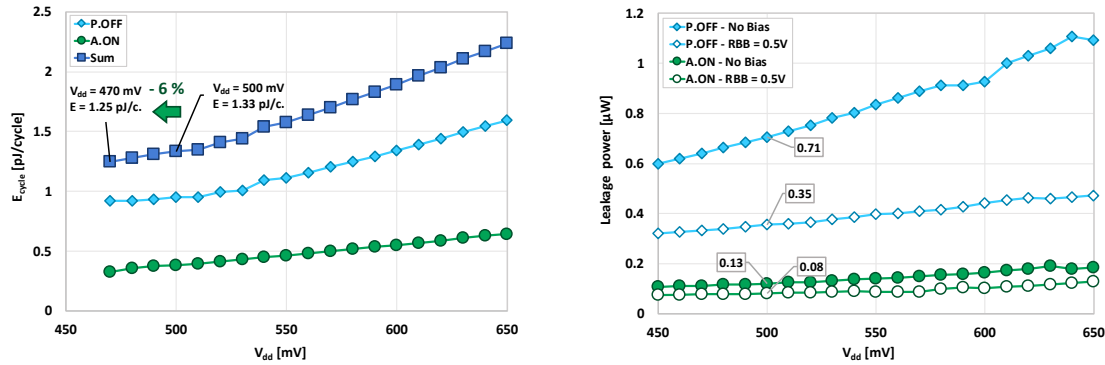
Figure 4.46: Measured mean core frequency of NZP28 V1.0 as function of the supply voltage, for multiple RBB values at 25 °C.

Power Evaluation

The MEP of the digital logic – P.OFF, A.ON and the sum of these two blocks – is given in Figure 4.47a for a wide voltage range. The MEP is reported at 0.47 V and 0.5 V with respective values of 1.25 pJ/cycle and 1.33 pJ/cycle, leading to a 6% difference.

The effective operating point of the SoC is set to 0.5 V/16 MHz, to benefit from the frequency gain between 0.5 V/16.7 MHz and 0.47 V/8 MHz. This pragmatic trade-off between operating frequency and power consumption is justified by the context of IoT applications working in the MHz range. Moreover, with respect to Figure 4.46, the 0.5 V/16 MHz condition ensures a 100% yield at 25 °C.

To characterize the benefit of using RBB for reducing the SoC power consumption, the mean leakage power reduction according to the supply voltage for the P.OFF and A.ON power domains are reported in Figure 4.47b. On average, a leakage reduction of respectively 52% and 59% for the P.OFF and the A.ON is measured. It leads to an overall average gain of 49% across the whole voltage range. At our operating voltage of 0.5 V, it is translated into a leakage power diminution of 0.36 μ W for the P.OFF and 0.05 μ W for the A.ON.



(a) Mean energy/cycle versus the supply voltage for the A.ON and the P.OFF without bias.

(b) Mean leakage power versus the supply voltage.

Figure 4.47: SoC power evaluation according to the supply voltage. The Dhrystone program is used at 16 MHz in Figure 4.47a. In Figure 4.47b, the 0.5 V RBB impact is also evaluated. All results are given at 25 °C).

System Power Breakdown

The power contribution of each part of the system (including the memories) was measured separately at 0.5 V/16 MHz/25 °C/no bias, plotted in Figure 4.48 and the energy/cycle figures associated reported in the first row of active in Table 4.4. The total energy consumption when the system is running is 2.67 pJ/cycle. The static power consumption due to the leakage current is 1.5 μ W, dominated by the P.OFF.

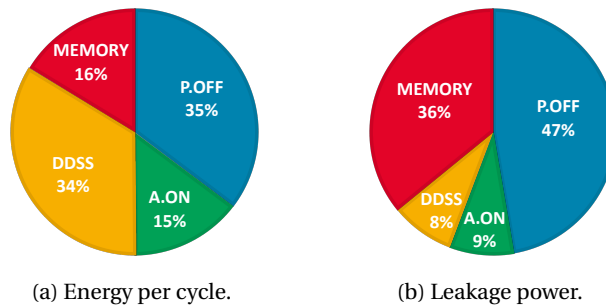


Figure 4.48: NZP28 V1.0 SoC power breakdown at 0.5 V/16 MHz/25 °C.

Resulting Power Savings from the Available ULP Modes

Architecturally, two low power modes are proposed: normal sleep and ultra deep. In our implementation, the sleep behavior is defined by the clock gating of the M0+ core whereas the ultra deep enables the use of the power switch on the P.OFF domain. Based on these sleep mode features, the PMU is designed and merged with the WIC as additional hardware level to enable mode transitions. With this architecture, the SoC user has the opportunity – depending on his application requirements – to clock gate or power down the microcontroller and wake up on certain hardware events.

Table 4.4 gives the baseline results which demonstrate the leverage offered by the M0+ power modes combined with the PMU to reduce the total power consumption of the system. For each state, the first row refers to the energy per cycle. The second row is the resulting total power (i.e. the sum of dynamic and leakage power). All measurements were performed at 0.5 V/16 MHz/25 °C and no bias. In active mode, the workload is the Dhrystone program executed from program memory. Switching from active to sleep power mode leads to a reduction of the core power consumption from 15.1 μ W to 1.96 μ W, reaching its lowest value of 4.3 nW in ultra deep.

State	P.OFF	A.ON	DDSS	Memory	Units
Active	0.94	0.39	0.89	0.44	pJ/c.
	15.1	6.23	14.5	6.96	μ W
Idle ¹	0.86	0.38	0.89	0.25	pJ/c.
	13.7	6.23	14.5	4.02	μ W
Sleep	0.12	0.22	0.89	0.03	pJ/c.
	1.96	3.59	14.5	0.53	μ W
Ultra Deep	2.7×10^{-4}	0.21	0.89	0.03	pJ/c.
	4.3 nW	3.49	14.5	0.54	μ W

¹ Idle given as the worst waiting option. It corresponds to active polling.

Table 4.4: NZP28 V1.0 SoC energy/cycle and total power breakdown at 0.5 V/16 MHz/25 °C.

In low power modes, the fastest clock frequency is no longer necessary, and a frequency adjustment strategy can be applied. Consequently, by switching in one cycle the DDSS clock from 16 MHz to 8 MHz, the power consumption of the A.ON and the SoC is respectively reduced by 48% and 13%. For further power saving, the DDSS is disabled in ultra deep and the 32 kHz reference clock is used. This reference is generated off-chip using a quartz source. As reported in Figure 4.49, this last technique decreases the A.ON/SoC power consumption by 95%.

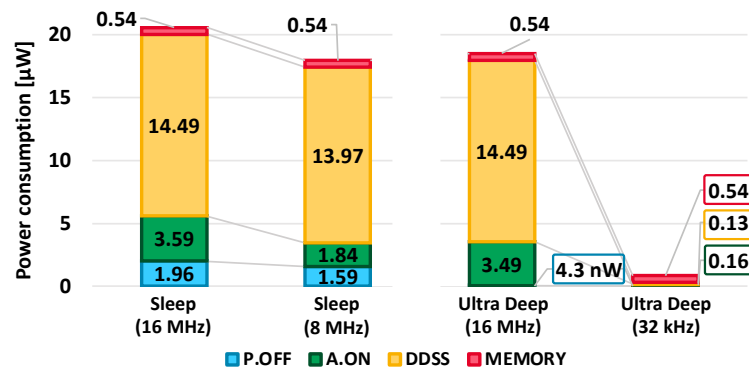


Figure 4.49: Influence of frequency scaling on the power consumption in Sleep (left) and Ultra Deep (right) modes at 0.5 V/25 °C.

Subsequently, trimming the SoC frequency offers room for RBB usage. An external $V_{\text{bias}} = 0.5$ V is applied leading to a nMOS backgate voltage $V_{\text{PW}} = -0.5$ V and a pMOS backgate voltage $V_{\text{NW}} = 1.0$ V. As a consequence, a SoC static current consumption reduction of 31% and 12.5% in respectively sleep and ultra deep is observed in Figure 4.50.

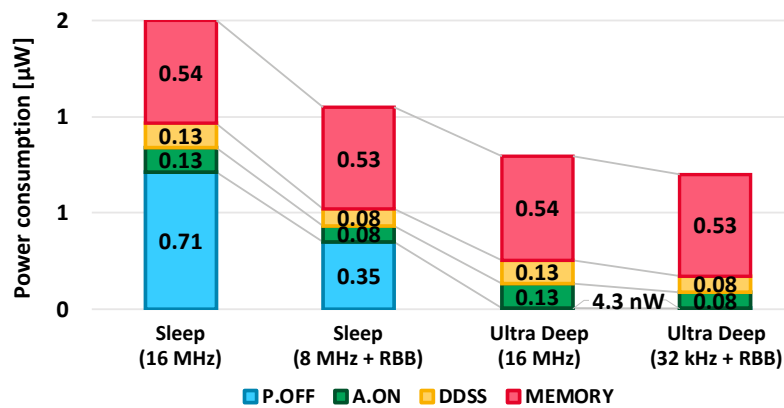


Figure 4.50: Reduction of the leakage power using RBB at 0.5 V/25 °C.

By combining together all these power saving approaches, we induce a final SoC power breakdown from active to sleep with frequency scaling and RBB of 53%, and from active to ultra deep with power gating of 98%. This leads to a total power consumption of the SoC of 0.7 μW in ultra deep mode (see Figure 4.51). Lastly, the power modes selected and the associated actions are summarized in Table 4.5.2.

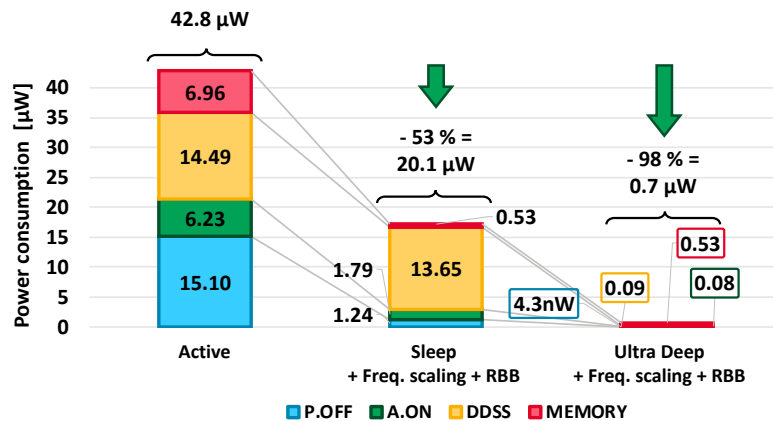


Figure 4.51: Total power breakdown of the NZP28 – V1.0 SoC at 0.5 V/25 °C.

State option	Active	Sleep +Freq. scaling +RBB	Ultra Deep +Freq. scaling +RBB
A.ON clock	16 MHz	8 MHz	32 kHz
P.OFF clock	16 MHz	Gated	Gated
DDSS	ON	ON	Disabled
Power switch	ON	ON	OFF
Body bias	0 V	RBB = 0.5 V	RBB = 0.5 V

Table 4.5: Summary of the NZP V1.0 power modes and configuration.

4.5.3 Impact of State Retention Power Gating

Estimation of SRPG using CAD results.

Figure 4.52 reports the impact of SRPG estimated on the core (P.OFF) during the synthesis flow step between NZP28 V2.0 and V3.0, respectively without and with state retention. Data are obtained for a typical corner, at 25 °C and 0.7 V.

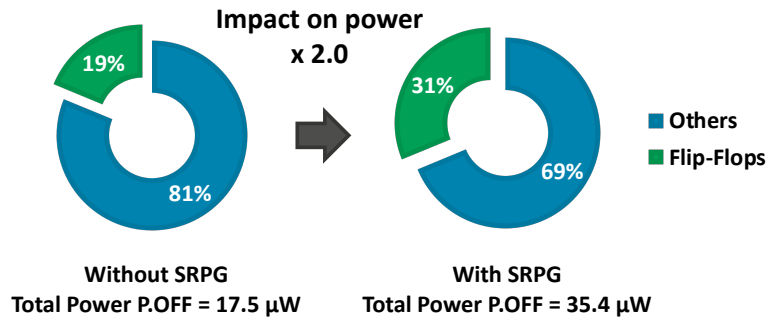


Figure 4.52: Power impact of SRPG in 28 nm on the P.OFF domain using synthesis data at 0.7 V, TT, 25 °C.

The retention register power consumption is $\times 3.36$ higher compared to a standard FFs implementation. They account for 31% of the block power consumption. This is a direct result of the increased complexity of retention cells that need more control logic to save the information during retention times. In return it increases the power consumption of such cells. Moreover, for a given power consumption retention FFs tends to be slower, and so the datapaths tends to be broadened to maintain the target operating frequency. Therefore, the system power remain dominated by the combinational logic (69% of the total power). Finally, at block level the SRPG implementation impacts the power consumption by $\times 2$.

At SoC level (see Table 4.5.3), either register and combinational logic power consumption increase after insertion of SRPG and dedicated control logic. This results in a $\times 1.4$ leakage and $\times 1.15$ total power consumption inflation. Again, the impact of SRPG is mitigated by other power consuming elements in other power domains of the SoC.

Groups	Internal ¹	Switching ²	Leakage	Total ³
Register	$\times 1.12$	$\times 1.10$	$\times 2.00$	$\times 1.12$
Combinational	$\times 1.25$	$\times 1.22$	$\times 1.92$	$\times 1.24$
Sequential	$\times 0.91$	$\times 4.57$	$\times 1.05$	$\times 1.00$
Memory	$\times 1.94$	$\times 0.73$	$\times 1.23$	$\times 1.00$
Total	$\times 1.15$	$\times 1.14$	$\times 1.40$	$\times 1.15$

¹ Sum of short circuit power and dissipation due to the charging of the internal loads.

² Power dissipated by charging the output load.

³ Weighted mean of all power contributions.

Table 4.6: Estimation of the SRPG impact on the power consumption at SoC level. Results are based on synthesis data on a 28 nm FD-SOI based core at 25 °C, 0.7 V between NZP28 V2.0 and V3.0.

Lastly, the area impact of SRPG is reported in Figure 4.53. The $\times 1.4$ area increase on the registers (which only impacts the P.OFF power domain), results in a $\times 1.1$ increase at SoC level.

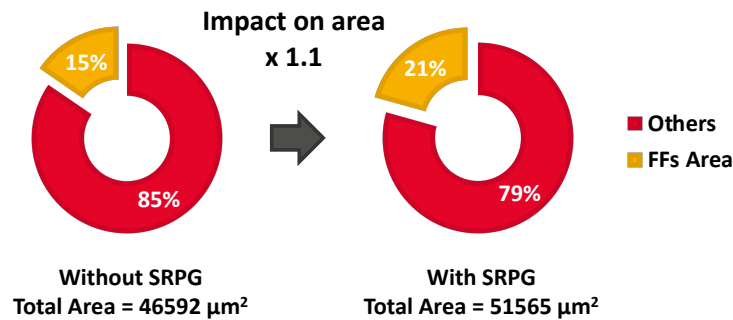


Figure 4.53: Area impact of SRPG in 28 nm on the whole SoC using synthesis data at 0.7 V, TT, 25 °C.

Measured Performances on Silicon Implementations

The impact of SRPG is now evaluated through measurements performed on 28 nm testchips operating at 16 MHz/0.5 V/TT/25 °C. Due to the limited number of power pads, the power consumption is evaluated at SoC level.

Figure 4.54 highlights the power consumption as well as the leakage power measured between NZP28 V2.0 and V3.0. From a design without retention to a design including SRPG in the P.OFF domain, a $\times 1.1$ and a $\times 1.7$ increase is observed respectively on total and leakage powers. Moreover, the area impact of SRPG on the physical implementations is given in Table 4.5.3, confirming the $\times 1.4$ increase on the P.OFF domain.

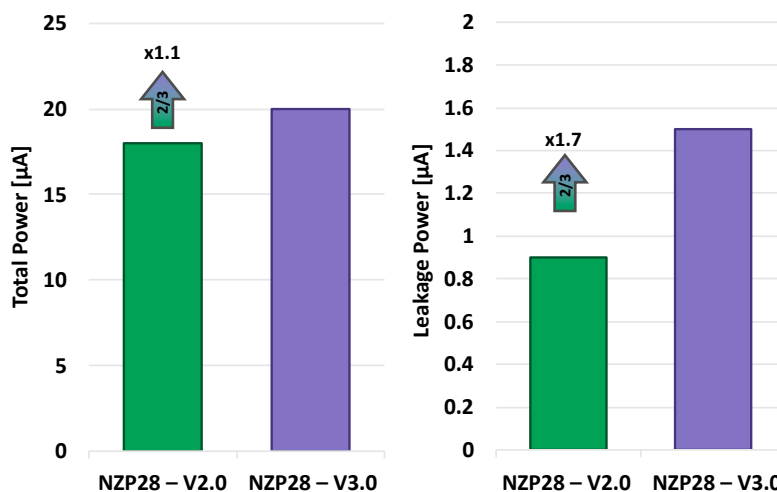


Figure 4.54: SoCs NZP V2.0 and V3.0 power consumption in 28 nm FD-SOI using experimental measurements (0.5 V/TT/25 °C). LRO and DDSS disabled.

The interest of an SRPG implementation depends on the use cases. For reduced suspend sequences, the possibility of completely switching down the system is limited, while re-

Group	Area [μm^2]	Ratio
P.OFF – NZP28 V2.0	16440	$\times 1.4$
P.OFF – NZP28 V3.0	23880	

Table 4.7: SRPG impact on the area of the P.OFF domain.

tention FF can be quickly activated. In that case, SRPG also extends the hardware capabilities to use retention only for a given set of tasks.

4.5.4 Technology Scaling Evaluation

Using the several testchips realized in 28 nm and 22 nm FD-SOI, an evaluation of the performances resulting from technology scaling is proposed. These silicon measurements also ensure a validation of the qFO4 methodology presented in Section 4.1.2. No body bias is used during these measurements. From all these results can be defined various trade-offs in term of power efficiency, operating frequency and leakage.

The energy/frequency trade-off observed on experimental measurements on the core and qFO4 simulations, both performed at 25 °C, is given in Figure 4.55. The measurements are obtained on the P.OFF domain directly accessible through dedicated power pins on NZP28 V1.0 and NZPs22 V1.0. Simulations results are obtained by weighting up the qFO4 paths according to the PB distributions of the system P.OFF.

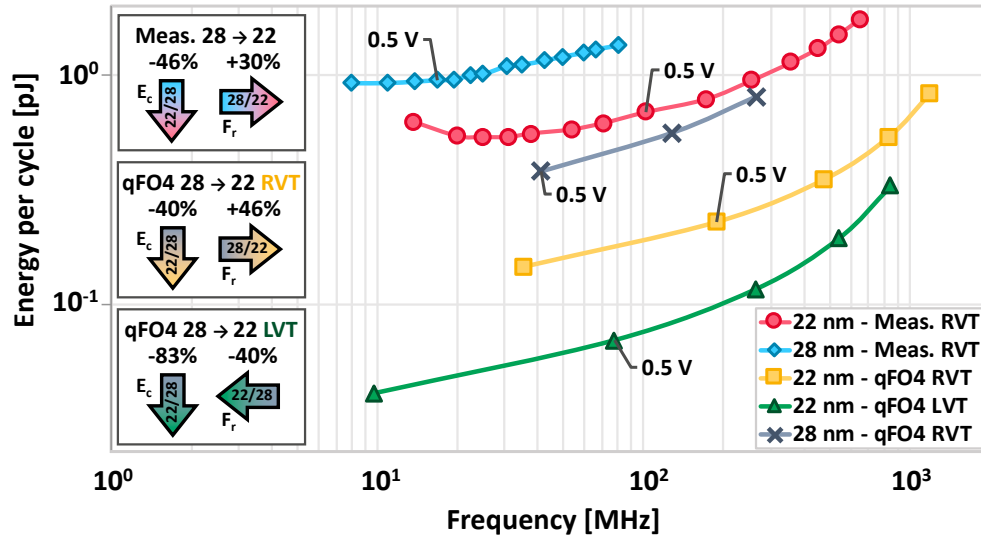


Figure 4.55: Energy/Frequency Trade-off analysis in 28 nm and 22 nm FD-SOI using experimental measurements on the core (25 °C) and qFO4 simulations (TT/25 °C). Relative variations are given using median values at ULV operation (i.e., $V_{dd} \in [0.4; 0.6] V$).

The qFO4 energy gains estimated give realistic trends on the final silicon behavior, with a 6 percentage-point difference. However, the frequency gain estimations are mitigated by speed limiting memories, leading to a rough prediction (16 percentage points). Similar estimations are done on leakage power in Figure 4.56, showing accurate trends of the core leakage power with only a 3 percentage-points difference.

Therefore, the qFO4 analysis – validated by silicon measurements – gives a correct approximation of the final design performances. Then, projections to 22 nm LVT devices are possible allowing prediction of the NZP22 V2.0 performances. A maximum 83% energy and 90% leakage reduction is estimated, accompanied with a 40% frequency degradation. Using data from the NZP28 V1.0 core, the 22 nm LVT utilization would bring a similar 0.5 V core (i.e., P.OFF) implementation in the 0.15 pJ/cycle range with a 10 MHz operating frequency. The μW power consumption for 32-bit processor in active mode would thus be reached.

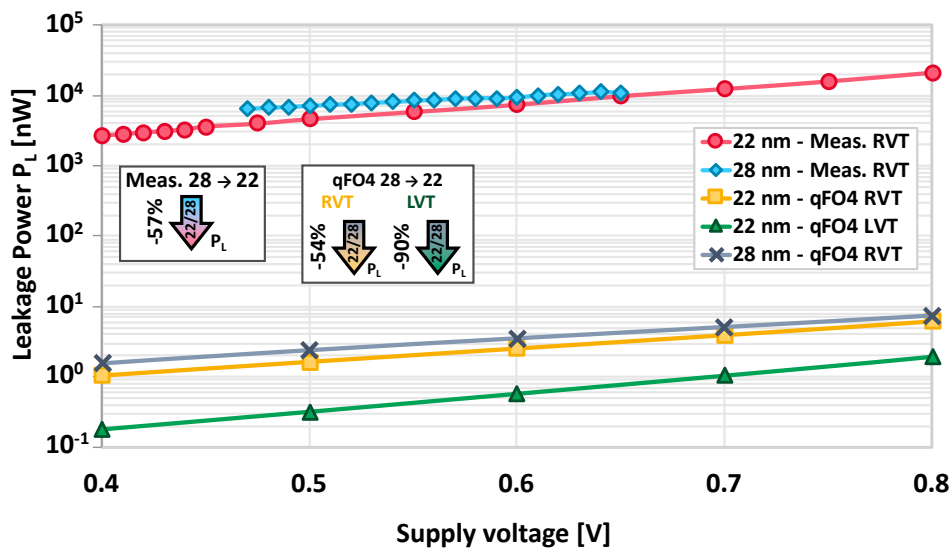


Figure 4.56: Leakage Power according to the supply voltage in 28 nm and 22 nm FD-SOI. Results obtained using experimental measurements on the core (25 °C) and qFO4 simulations (TT/25 °C). The relative variations are given using median values at ULV operation (i.e., $V_{dd} \in [0.4; 0.6]\text{V}$).

4.5.5 Utilization of Self-Body Bias

This section explores the concept of self-body biasing presented in Section 4.4. Results are first given on a fully integrated system based on a 22 nm - RVT implementation. Then experimental measurements are proposed on a 28 nm SoC using an external bias generator.

Measured Performances on 22 nm FD-SOI

The system tested in this subsection is the NZP22 V1.0 SoC. For validation and measurements, the Dhrystone benchmark was used on 6 packaged dice, at lab temperature (i.e., 22.5 °C). Due to the FBB operating conditions in 22 nm FD-SOI aforementioned in Section 4.4, all FBB measurements reported are limited up to $V_{dd} = 0.6\text{V}$ to avoid excessive leakage. The speed performances obtained are first given in Figure 4.57.

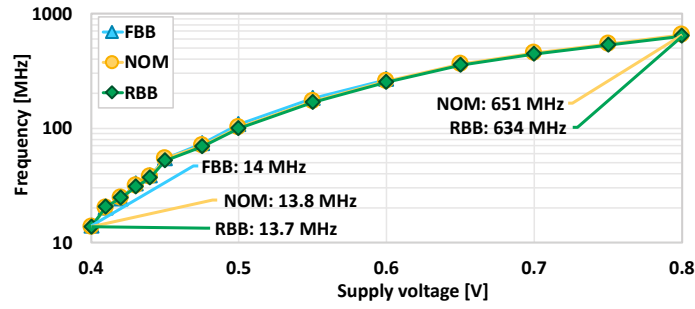


Figure 4.57: Measured NZP22 V1.0 SoC maximum frequency according to the supply voltage and the bias modes at 22.5 °C.

The SoC's power performances are reported in Figure 4.58. The power contribution of each part of the system was measured separately and reported at the MEP obtained for RBB/0.42 V/20 MHz. The total energy consumption when the system is running is 1.13 pJ/cycle and the total power consumption due to the static current is 3.15 μ W, dominated by the M0+ core in both cases. Wells' currents are included in the A.ON, with a limited impact; \sim 100 nW are measured in the worst case 0.6 V, FBB.

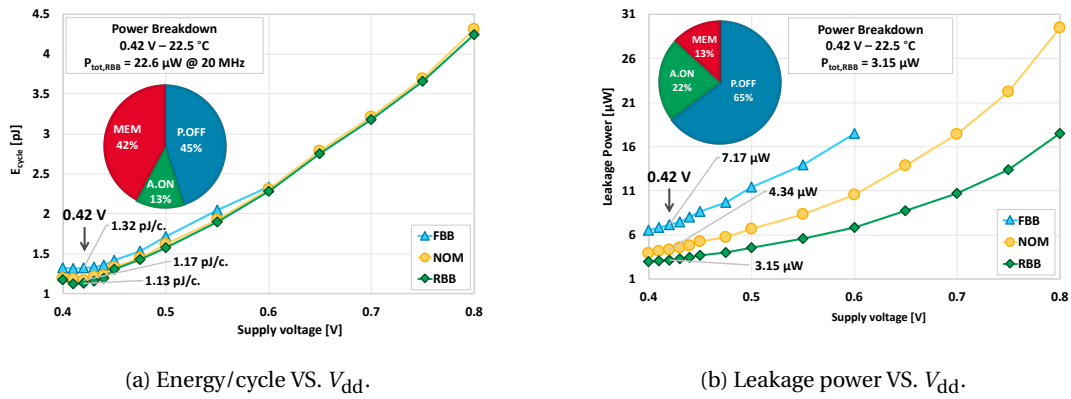


Figure 4.58: NZP22 V1.0 measured SoC performances: Energy/cycle (right) and leakage power (left) at 22.5 °C.

Switching from the standard nominal mode to the proposed RBB mode leads to a 3% energy gain, and 27% leakage reduction. The reduced leakage power is due to an increased threshold voltage of the pMOS transistors in RBB, while FBB decreases the threshold voltage of the nMOS transistors improving the maximum frequency. Independently of self-body bias, standby modes are available to reduce the system power consumption by 69% and 77% in sleep and ultra deep respectively (see Table 4.8).

Consequently, to the previous results, the SoC adaptive self-biasing technique is tested for temperature compensation. The maximum achievable frequency according to the body-biasing mode has been measured over a -10 °C to 60 °C temperature range for various supply voltages. Results are reported on Figure 4.59 for one die randomly selected. For $V_{dd,MEP} = 0.42$ V the 20 MHz SoC target frequency at 22.5 °C is reached.

State	P.OFF	A.ON	MEM	Units
Active	10.2	2.9	9.5	μW
Sleep	872	2474	344	nW
Ultra Deep	7.56	2428	344	nW

Table 4.8: NZP22 – V1.0 power breakdown according to the ULP modes 0.42 V, 20 MHz, 22.5 °C, RBB mode.

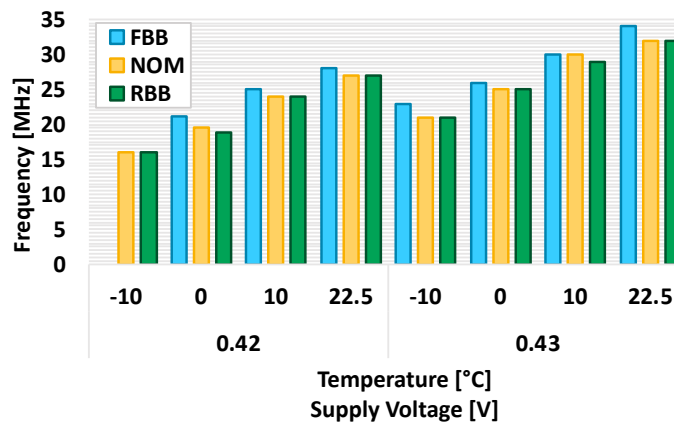


Figure 4.59: NZP22 V1.0 measured maximum frequency depending on the supply voltage, temperature and bias mode for one die.

When the temperature reaches 0 °C, the system controls the bias swapping to reach FBB mode and ensures the continuous functionality of the 20 MHz target frequency. A wider temperature range can be supported by the system for higher V_{dd} values at the cost of an increased energy consumption. For instance, at 0.43 V the 20 MHz frequency is still guaranteed at -10 °C yet, this section is currently focusing on performing at the MEP.

Considering the Dhrystone test, the 100 worst critical paths are involving the non biased memories, which represent $\frac{3}{4}$ of the period delay. Therefore, only 25% of the period is impacted, lowering the frequency gain of using FBB/RBB compared to CAD results. Several solutions can improve the self-bias efficiency. To avoid the memories in the critical paths, their size can be reduced by splitting them in smaller cuts, and placing them closer to the logic areas. De-skewing can also be used to improve the path timing, within the available hold margin. We can also improve the system FBB/RBB efficiency, using memories supporting Body-Biasing on the periphery, which are later available in the NZP22 V2.0.

Figure 4.60 reports the SoC's consumption with the temperature compensation scheme enabled at 0.42 V/20 MHz. At 0 °C using the FBB mode, a 1.05 pJ/cycle is obtained while maintaining the operating frequency. At higher temperatures, the system enables the swapping to reach RBB mode. At 60 °C, the system energy consumption is reduced by 15% – from 1.92 pJ/cycle in NOM mode, down to 1.63 pJ/cycle – while the leakage is reduced by 24%. Body-bias Swapping commands are controlled using the SoC PMU. With an external temperature sensor and pre-configured threshold value stored in the backup memory, switching from RBB/NOM/FBB modes can be dynamically and efficiently performed.

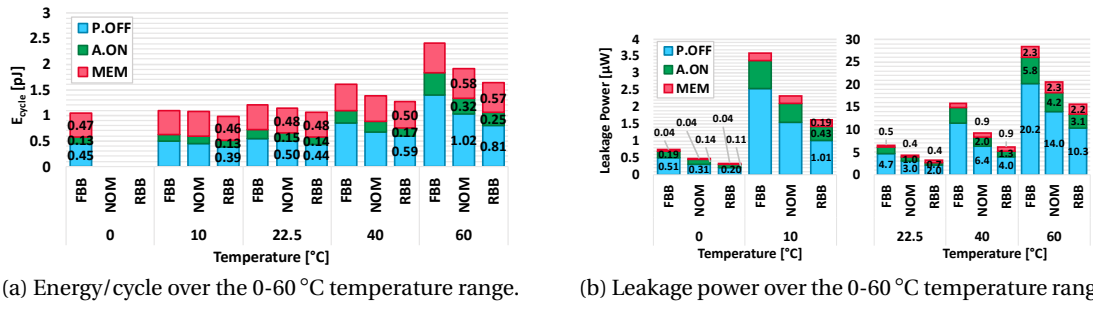


Figure 4.60: NZP22 V1.0 SoC's power with temperature compensation scheme enabled: Energy/cycle (left) and leakage power (right) at 0.42 V/20 MHz.

Measured performances on 28 nm FD-SOI

As demonstrated in Section 4.4.2, a standard 28 nm FD-SOI implementation further increases the swapping efficiency. Indeed, it enables both pMOS and nMOS networks to be forward or reverse biased (the pMOS NOM body-biasing voltage is effectively $V_{NW} = V_{dd}$). Using an external bias generator, the self-body biasing technique is tested on the NZP28 – V3.0 for a 0.5 V supply voltage. The energy efficiency and leakage power according to the NOM and FBB modes are reported in Figure 4.61.

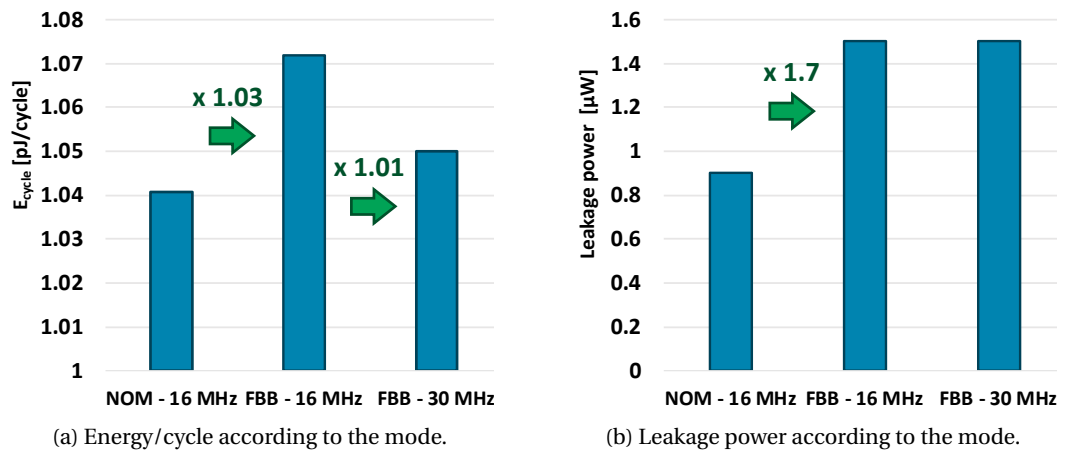


Figure 4.61: Evaluation of self-body bias scheme on NZP28 V3.0 for energy/frequency trade-off. Energy/cycle (left) and leakage power (right). Results are obtained at 0.5 V/25 $^{\circ}\text{C}$.

Considering the 16 MHz nominal operating, an extra $\times 1.03$ energy per cycle is observed when the FBB is activated (see Figure 4.61a). This also results in a $\times 1.7$ increase in leakage power (Figure 4.61). However, thanks to the FBB mode, a $\times 1.9$ frequency boost is given. The SoC can operate at 30 MHz with a minimized energy impact ($\times 1.01$). Therefore, the technique can be used as a tuning knob for energy/frequency trade-off.

Temperature compensation is also explored in Figure 4.62. Starting from 25 $^{\circ}\text{C}$ in FBB mode at 16 MHz, the temperature is progressively decreased to -10 $^{\circ}\text{C}$. While the NOM cannot operate below the initial temperature, the FBB mode allows continuous operation. Moreover, in this situation, it does not come at the cost of extra power consumption.

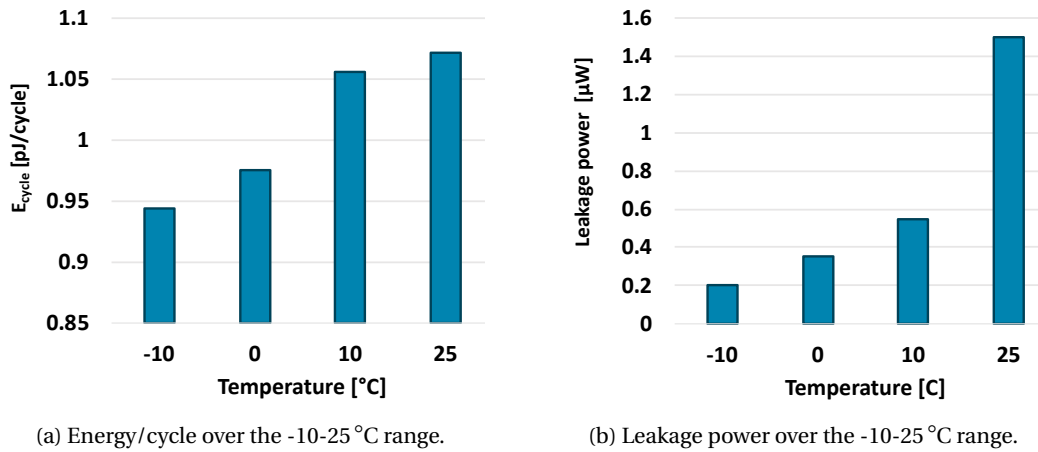


Figure 4.62: NXP28 V3.0 SoC's power with FBB scheme enabled for temperature: Energy/cycle (left) and leakage power (right). Results are obtained using the FBB mode at 0.5 V/16 MHz.

These interesting results must be mitigated by the power consumption observed in the wells. The previous results only consider the power on the main power supplies. In fact, beyond a 0.3 V FBB, an excessive 10 mW is measured whereas in NOM mode only 100 nW are reported. Currently no root cause has been found to explain this situation. It is not related to the pn-well diodes because no symmetric currents are measured on the V_{NW} and V_{PW} supplies. Additionally, the leakage current boom occurs long before the diode threshold voltage.

Therefore, even though the NXP28 V3.0 is fully isolated from the other IPs of the testchip, a dedicated design incorporating the bias switches in 28 nm FD-SOI is required to identify the leaky components. In the end, it would validate or discard the solution in this technology node.

4.5.6 Towards Adaptive Power Management

The time required to switch the processor between states results in a longer response latency and consumes a fair amount of energy. The most energy-efficient SoC suspend state depends on these power and timing penalties. Hence, to select the lowest power mode for a given timeout, it is necessary to evaluate the energy overhead due to these transitions. Even though the ultra deep mode offers the best energy performance, the most efficient power mode depends on the application suspend sequences. To determine an optimum for a given inactivity time, Figure 4.63 quantifies the overhead of each mode.

The idle state is given as the worst power saving mode. It ideally starts in 0 clock cycles and corresponds to an active polling mode of the core. The switching time, to enter and leave modes, from active to sleep or ultra deep (respectively t_1 and t_2) and the associated energy are defined by the hardware design. They have been validated using RTL and Prime Time Power simulations. This time results in a minimum number of cycle and energy reported in Figure 4.63 using the green square ■ and yellow circle ●, respectively for sleep and ultra deep modes. The slopes are given by the energy/cycle associated to each mode. Finally, the blue diamond ◆ define the breakeven time t_i when mode switching is beneficial in term of energy. No retention is used in the core for ultra deep. As a consequence, the system startup sequence and reset must be considered. Aware of this constraint, a timing overhead was taken into account

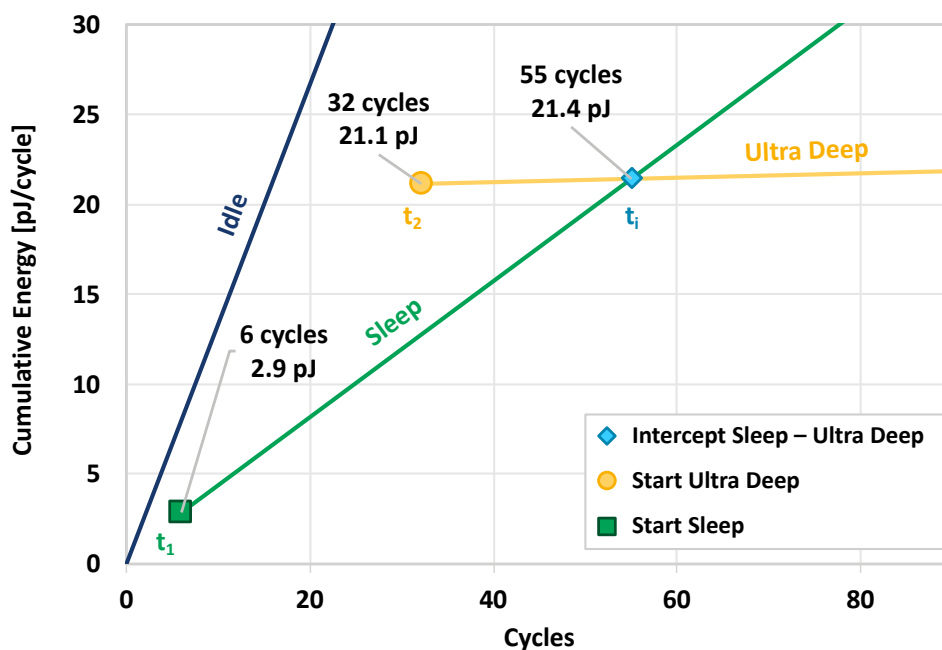


Figure 4.63: Evolution of the energy consumption depending on the selected power state at 0.5 V/25 °C. Active, Sleep and Ultra Deep modes use Table 4.5.2 results on the NZP28 V1.0. For Idle, figures are extracted from Table 4.4

to define the 32 cycles to enter and leave ultra deep mode.

The time t_2 is also application dependent and corresponds to the context that should be restored. It is defined between the entrance into the active mode (a deepsleep order is asserted to the PMU) and the fetching of the first instruction from the program memory. In our case, no specific variable restoration is required; it leads to a small amount of cycle which corresponds mostly to the SoC reset execution. This sequence also includes the time to switch the clock frequency and enable RBB. For sleep mode, the frequency synthesizer is still active, while it is disabled in ultra deep. The DDSS can switch frequency or be disabled in one cycle, therefore the impact on the 32 cycles overhead is minor but considered.

For energy driven applications, the gains of entering the power-off modes is benchmarked: the sleep state can directly be selected as it consumes less than this ideal Idle mode. For sleep time over 55 cycles the ultra deep should be triggered because it leads to the lowest SoC energy mode. However, this presupposes an accurate description of the SoC power consumption, which is actually dependent of PVT variations.

In the case of constant duty cycled core operations, the sum of the time-periods spent in active mode T_{ON} and Suspend mode T_{OFF} is constant. Knowing T_{ON} through a simple counter gives us T_{OFF} and consequently the best suspend mode can be chosen. In this context of known active and suspend times, the scaling of the mode selection can be analyzed to deal with larger SoCs. However, it presumes a known activity and an accurate estimation of the power consumption in each power mode. Moreover, the running application is susceptible to evolve along the lifetime of the system, hence smarter techniques are required.

4.6 Summary of Performances and Conclusions

4.6.1 Summary of Performances and Techniques Used

Table 4.9 summarized all the performances and techniques used in this work to achieve ULP consumption.

Intrinsic Performances

Through five silicon demonstrators, we have demonstrated the feasibility of an ULV and ULP SoCs using exclusively the latest industrial components: industry catalog CMOS standard cells and memories, standard FD-SOI technology features, and standard EDA flows⁸. Core MEPs are obtained at 0.5 V and 0.42 V, reaching the sub-pJ/cycle barrier while maintaining a ~10 of MHz operating frequency with a 100% yield. Improved results are expected with the last 22 nm FD-SOI LVT implementation.

ULP techniques for static and dynamic power reduction

Several power switches designs are tested and optimized. The integration is simplified by removing the power level detection and enabling hardware/software latency. Clock gating and frequency switching is also available to reduce the dynamic power consumption. Moreover, retention is integrated and evaluated. To mitigate its impact, a double power gated SRPG implementation is also proposed. In 28 nm FD-SOI, the frequency generation is done through the design of a purely digital architecture enabling fast frequency switching and stable reference.

Impact of ULP modes

The utilization of the suspend modes available are evaluated for dynamic power reduction and efficient SoC design/applications interactions. ULP modes act as enabler of autonomous system and energy saver in such advanced technology nodes. The inherent leakage can be mitigated and μ W consumption reached in suspend modes.

Utilization of Self-Body Bias

The proposed self-body bias technique available at ULV extends the utilization on the SoCs in term of frequency or temperature ranges. It exploits the FD-SOI without the need for supplementary body bias generators. Even though its interest is limited in 22 nm, it shows promising results in the 28 nm node.

Technology Scaling

22 nm follows the path of the 28 nm FD-SOI nodes. It improves the frequency performances while maintaining a highly competitive energy efficiency. Moreover, at ULV no extra leakage cost is observed. With these two technology offers, a large panel of applications can be targeted,

⁸This concept of using the technology “as it is” for ULV operations, has been later validated by [209].

and performances trade-off can reach the utmost energy efficiency at ULV. For system spending most of their time in suspend mode, the 22 nm LVT will guarantee the lowest leakage while the 22 nm RVT version helps reaching higher operating frequencies.

4.6.2 Conclusions

The work presented in this chapter differs from the existing publications by a technical approach relying on simple yet highly efficient techniques. To be adopted, mass market requires cost effective and robust solutions.

Therefore, former ULP techniques and new innovative concepts are explored from technology to system level. A SoC with active power consumption that fits a $\sim 100\mu\text{W}$ power consumption is demonstrated thanks to: an adequate utilization and bench-marking of the technology features, specific tuning of the implementation flow, a careful selection of the design components and low-power techniques, the optimization of the system architecture coupled with a learning of the different power modes offered by the Cortex-M0+ for power mode selection, an ultra-efficient SoCs. Moreover, room is still available in the whole power budget to expand the system with communication capabilities and sensing, leading the way towards fully autonomous systems.

However, the best ULP modes do not guarantee an efficient power consumption during the system activity. Due to the alternation between active and suspend sequences, adaptive power management is required for optimum suspend mode selection. Due to PVT variations, a complete characterization of the system power consumption does not guarantee the most favorable solution. Thus, efforts should be pursued towards innovative and flexible power management techniques relying on learning the SoC activity (see Chapter 5).

This work led to several publications in conference and journal papers (ESSCIRC 2017, A-SSCC 2017, JSCC 2018 and SSC-L 2019). The complete reference list is available in Appendix G.

Reference	NZP28 – V1.0 [208]	NZP28 – V2.0	NZP28 – V3.0	NZP22 – V1.0 [211]	NZP22 – V2.0
Features					
Technology	28 nm	28 nm	28 nm	22 nm	22 nm
CPU	FD-SOI RVT 32-bit ARM Cortex-M0+ 4 + 8 KB SRAM (data + inst.)	FD-SOI RVT 32-bit ARM Cortex-M0+ 4 + 8 KB SRAM (data + inst.)	FD-SOI RVT 32-bit ARM Cortex-M0+ 4 + 8 KB SRAM (data + inst.)	FD-SOI RVT 32-bit ARM Cortex-M0+ 4 + 8 KB SRAM (data + inst.)	FD-SOI LVT 32-bit ARM Cortex-M0+ 4 + 8 KB SRAM (data + inst.)
Memory	PMU, Clock gen., RTC, SPI, GPIOs, WIC, Timer, DAP	PMU, Clock gen., RTC, SPI, GPIOs, WIC, Timer, DAP	PMU, Clock gen., RTC, SPI, GPIOs, WIC, Timer, DAP	PMU, RTC, SPI, UART, GPIOs, WIC, Timer 16 KB back-up mem., DAP	PMU, WUC, APM, RTC, SPI, UART, GPIOs, WIC, Timer 16 KB back-up mem., DAP
Peripherals					
Optimizations					
Device	Sub/Near-Threshold Advanced CMOS Techno. Multi- V_{TH} implem.	Sub/Near-Threshold Advanced CMOS Techno. Multi- V_{TH} implem.	Sub/Near-Threshold Advanced CMOS Techno. Multi- V_{TH} implem.	Sub/Near-Threshold Advanced CMOS Techno. Self Body Biasing Clock Gating Power Gating	Sub/Near-Threshold Advanced CMOS Techno. – Clock Gating Power Gating
Circuit	Clock Gating Power Gating	Clock Gating Power Gating	Clock Gating Power Gating	Clock Gating Power Gating	Clock Gating Power Gating
Gate	ULP Clock Generator	ULP Clock Reference ULP Clock Generator	ULP Clock Reference ULP Clock Generator	Self Body Bias Generator	–
Modules	AFS, DPM	AFS, DPM	AFS, DPM, SRPG	DPM	DPM, APM
System				Dynamic Self-Body Biasing	Double-gated SRPG
Performances					
Area [mm²]	0.073	0.082	0.082	0.072	0.119
Temperature [°C]	25	25	25	0 - 60	TBM
Supply Voltage [V]	0.5	0.5	0.5	0.42	TBM
Memory Voltage [V]	0.5	0.5	0.5	0.42	TBM
Frequency [MHz]	16	16	16	20	TBM
Suspend Power [nW]	704	900	1500	2780	TBM
MEP of core [pJ]	0.94	1.12	1.25	0.51	TBM
MEP of SoC [pJ]	2.67	2.71	2.85	1.13	TBM

TBM: to be measured.

Table 4.9: Ultra-Low-Power System-on-Chip and referencing of the low-power optimizations performed.

Chapter 5

Adaptive Power Management

IN the previous Chapter of this thesis, a general ULP SoC has been designed allowing several power modes. These modes – available at the system level – are employed to reduce the overall power consumption during the inactivity of the system. However, following the conclusion presented in Section 4.5.6, due to timing overhead resulting from mode transitions, the lowest power mode may not be the best power mode for the system activity and wake-up sequences.

Therefore, this chapter investigates Adaptive Power Management (APM) for the optimal selection of ULP modes. Using Machine Learning (ML) techniques based on Reinforcement Learning (RL), a complete HW IP is presented. This block is integrated into a SoC architecture to guide the PMU for selection of the best ULP mode depending on the system activity.

In Section 5.1 is first introduced the activity-oriented power management and the necessary frameworks of the QL algorithms required to understand the APM module. A first evaluation is done on a generic architecture, allowing validation of the idea in Section 5.2. In Section 5.3, the optimized QL implementation is then applied on the SoC presented in Chapter 2. Consequently, Section 5.4 offers an insight on the HW implementation behind the APM module. Lastly, the functional validation of the block and the measured results are presented in Section 5.5. It leads to the conclusions and perspectives of Section 5.6.

5.1 Activity-oriented Power Management

5.1.1 Problem Statement

Contrary to existing SoC power simulators [212], APM techniques must provide the optimal suspend mode without requiring any costly power bench-marking. Moreover, except from receiving a suspend order from the core, the APM module will perform the task without the need for explicit instructions. On the contrary, it relies on activity patterns and inferences. Hence, it leads to consider ML solutions, which are one study field of Artificial Intelligence (AI).

To greatly reduce the energy consumption and fit the constrained power budget, the dedicated system should also be simple in term of implementation and modelling of components. Similarly, ULP features must be integrated and a fully digital implementation encouraged for flexibility and re-usability on several systems. Robustness to power consumption fluctuations that might occur due to PVT variations or variability between chips is also mandatory.

Whereas a software implementation would be available, a separate HW implementation of the APM is required to decrease the computation load on the core. Lastly, depending on the system activity, the HW module will consider the power and activity states of the SoC to determine the best power mode. This information might be used by the PMU to set the SoC in the given mode. However, the application will be able to bypass the automatic mode selection. A standard software-determined power mode is then available for the user.

5.1.2 Existing Solutions

Review of APM Utilizations

Chapter 4 presents DPM techniques that rely on dynamic reconfiguration of an electronic system. It guarantees the SoC running application the targeted performance levels, with a minimum number of active components, or a minimum load on such components. The exploration of this concept has resulted in the development of DPM schemes (a.k.a. policies) to comply with the system workload over runtime [138] and reduce the overall power consumption by using dynamic reconfigurations [140].

At system level, DPM has proven to be very effective for power reduction [213]. It takes wise decisions and efficiently determines the system components that are idle or underutilized and place them in ULP modes. However, the power manager requires application and architecture specific information.

Moreover, the system application, the environment and the hardware itself suffer from uncertainty and variability. For instance, the workload of a complex system depends on the nature of the application, the input data and the user context. As it impacts the system speed and power consumption during its lifetime, robust power management requires awareness on the system variations.

Consequently, the APM concept (a.k.a. Adaptive DPM [214]) has been developed. APM uses an algorithm to dynamically select the best DPM policies from a set of candidates. The workload of a system and the transition between the system modes tends to be optimized relatively to the application running in a variable environment. It was successfully employed on Hard Disk Drive (HDD) of personal computers. In these architectures, the HDD requests are clustered into sessions given by the underlying application. Then by predicting the session

lengths, the system can shutdown components between sessions to save power [215].

Then, the APM utilization has been applied to IoT-oriented devices where conventional low-power design techniques and hardware architectures have only provided partial solutions to reduce the power consumption. Indeed, to reach utmost energy efficiency, Software (SW)/HW became mandatory to determine the best power mode [216].

Traditionally, adaptive power management solutions heavily rely on heuristics, yet, new researches have been carried out to use intelligent and machine learning techniques [217]. For instance, it allows sustainable operation of energy harvesting WSN in a variable environment [218]. Moreover, the development and the application of RL frameworks on SoCs open new challenges. Indeed, this area of ML does not require extensive models to be applied on complex systems, like hybrid electric vehicles [219] or networks of devices [220, 221].

Thus, the application of APM for SoCs power management has rapidly evolved over the last twenty years yet, no complete SoC HW implementation is presently demonstrated. The reviving trend around AI has restarted the interest for this subject. Indeed, the model-free framework provided by RL algorithms gives further research opportunities to reduce the power consumption of SoCs depending on their application and workload. Furthermore, deep RL implementation using Neural-Networks (NN) can also be used on wider heterogeneous SoCs where the determination of the system parameters is sensible [222].

Algorithms and System Representation

When referring to APM combined with RL techniques, the model-free QL [223–225] and Time Difference (TD)-learning [226, 227] algorithms are especially used.

On the one hand, QL (see Section 5.1.5) is applied to find a sequence of actions (mode transitions) associated with states and learns the optimum succession of actions. On the other hand, TD algorithms optimize a selected state before mode transition occurs. It uses less variables to describe the problem and tends to be faster. However, it relies on random functions similar to Monte-Carlo methods, which can be costly in ULP devices. Moreover, when the system-description that needs to be optimized becomes too complex, TD algorithms tends to show a lack of accuracy in contrast with Q-Learning. For these two reasons, a QL implementation has been chosen [227].

Learning algorithms rely on Markov Decision Process (MDP)-based diagrams (see Section 5.1.4). Alternatives such as Semi Markov Decision Process (SMDP) [227] or Time-indexed Semi Markov Decision Process (TiSMDP) [138] are available. Both system representations add a quantum of time to the execution of the algorithm. The SMDP allows actions to be taken with a variable time of execution through exponential functions requiring complex calculations and time stamping. The TiSMDP model uses time index in each state requiring more memory to store the states information. This last solution has been evaluated in Section 5.1.4 yet, to comply with the low-power budget the non-modified MDP diagram would be favorable.

5.1.3 System Definition

Application on Suspend Sequences

APM techniques tend to minimize the energy consumption during active and suspend modes [213]. While it offers significant energy reduction for complex system composed of one

or multiple devices, it naturally increases the implementation complexity. To mitigate this HW impact, APM utilization is here considered for suspend sequences only. It will explore ideal trade-offs in the power-performance design space to determine better suspend mode power management policies.

An activity driven model is also selected. Wake-up signals are expected to happen at a quasi-periodic rate as shown in Figure 5.1. This reflects an energy-harvester powered system where the battery is reloaded periodically, or a system where software events such as watchdog timer (or keep alive signals) are waking-up the system.

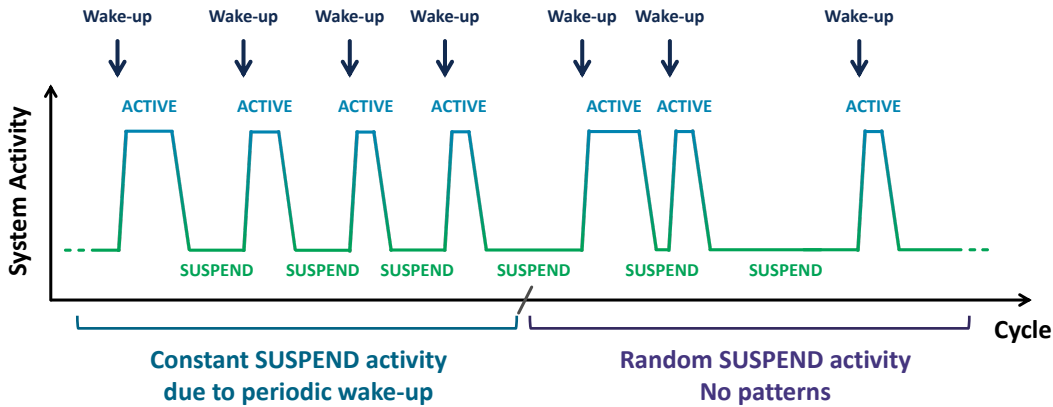


Figure 5.1: System activity describing various suspend scenarios resulting of periodic or random wake-up.

This constraint is due to the learning capabilities of the algorithm. If a random suspend activity happens between multiple algorithm executions (or trial), no pattern would be found to determine a best power mode. In that case, a heuristic solution would be preferential. However, a fully deterministic activity is not expected. The algorithm should be robust over potential variations during the workload activity. It must be able to converge to new solutions when variations on the activity are occurring (see Section 5.2.3).

Moreover, a representation using cycles instead of time units is used for analysis consistency. As shown in Chapter 4, the power modes selected by the APM can differ by their operating frequency. Consequently, the corresponding time for one cycle unit might differ if frequency switching occurs. The number of cycles guarantees a time consistent unit over several trials. Cycles-to-time conversion in standard time unit is possible by considering the number of cycles for each frequency. The infimum of the time will consider that the system was operating at the maximum frequency for all cycle. The supremum uses the minimum frequency.

Heuristic Solutions

To evaluate the benefit of the APM implementation, heuristic solutions are defined. The objective of a heuristic is to provide in a reasonable time a solution that is good enough to solve the problem. Generally, it is not the best of all solutions for the problem yet, it approximates the exact solution.

Examples are given in Figure 5.2 for a system composed of active, sleep and deep sleep modes. The first solution is the sleep heuristic, which selects the sleep mode as soon as the

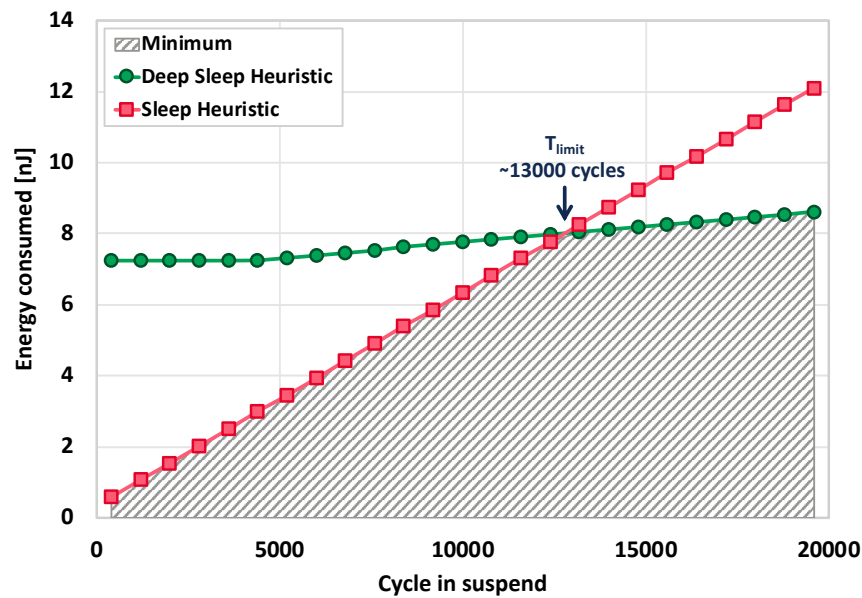


Figure 5.2: Energy consumed by the system according to the heuristic solution selected. The shaded grey area corresponds to the minimum energy that can be reached.

system goes into suspend mode. The second one is the deep sleep heuristic. Both solutions approximate the optimum power consumption defined by the grey area yet, they work only for a certain amount of cycle.

From these solutions is defined an energy color-map corresponding to the energy consumed according to the number of trials (i.e., executions of a given algorithm) and a number of cycles spend in suspend mode. It is relative to the minimum energy achievable by the system as shown in Figure 5.3 for the deep sleep heuristic. The power mode selected is always the same, so the evolution of the energy is constant at every trial. An optimal solution would consume less energy for every amount of cycle in suspend.

Algorithm Lifespan

According to the analysis presented in Section 4.5.6, the transition time between modes impact the power consumption of the system. It also determines the number of cycles before the heuristic solution, corresponding to the lowest power mode, becomes the optimal solution. Considering the Figure 5.2 metrics, for a workload spending more than 13000 cycles in suspend mode, the deep sleep heuristic reaches the minimum energy.

This time limit is defined as the algorithm lifetime given by T_{limit} . It depends on the number of system modes, their power consumption and transition time. It is also sensitive to PVT variations which might affect the power consumption. Using nominal operating conditions, it gives a rough approximation of the number of cycles before a simple counter would be able to determine the best solution (as explained in Section 4.5.6).

As shown in Table 5.1, the algorithm lifetime helps to define a system where APM optimizations will be preferential. The higher it is, the more room it leaves for optimal mode selection. Using the data from NZP28 – V1.0 (see Section 4.6.1), a T_{limit} of 55 cycles is obtained,

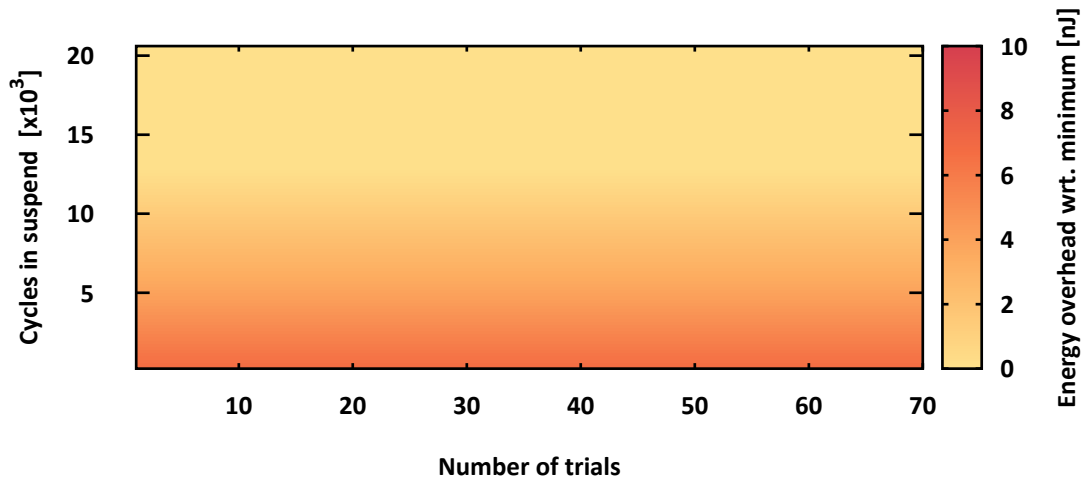


Figure 5.3: Color map representation of the deep sleep heuristic solution given in Figure 5.2. The energy consumed is given according to the number of cycles spent in suspend mode and the number of trial. The energy is given relatively to the minimum energy that can be reached for a given architecture.

leading to $5.5\mu\text{s}$ at 10 MHz. In this situation, the deep sleep heuristic is almost optimal. Using the same SoC architecture, a second estimation is done with a longer transition time of $450\mu\text{s}$, corresponding to context saving into a memory as it will be detailed in Section 5.1.6. This time a 1.3 ms algorithm lifetime is obtained corresponding to the intercept of Figure 5.2.

The same exercise is reproduced using the power modes and transition times of several IoT-oriented and general computing devices. For consistency, a 100 MHz operating frequency is chosen for all systems. Due to their small transition time, tiny devices leave a small margin to apply APM algorithm. When the complexity of the SoC increases, it leaves more rooms for power mode optimizations.

	System	States	State Power	Transition times	T_{limit}
NXP	NXP28 – V1.0	Idle / Sleep / Deep Sleep	$40\mu\text{W}$ / $20\mu\text{W}$ / 700nW	600ns / $3.2\mu\text{s}$	$5.5\mu\text{s}$
	Extended transition	Idle / Sleep / Deep Sleep	$40\mu\text{W}$ / $20\mu\text{W}$ / 700nW	600ns / $450\mu\text{s}$	1.3ms
IoT	ST BlueNRG-2 [228]	Idle / Sleep	3mW / $6.3\mu\text{W}$	$200\mu\text{s}$	$200\mu\text{s}$
	Intel Edison [229]	Idle / Sleep	100mW / 5mW	400ms	400ms
General	Network Card [219]	Busy / Idle / Sleep	1.6W / 0.9W / 0.1W	0.3s / 0.7s	0.8s
	IBM Hard Drive [230]	Idle / Idle LP / Standby / Sleep	1W / 0.8W / 0.3W / 0.1W	0.7s / 19s / 96s	163s
	Toshiba Hard Drive [230]	Idle / Standby / Sleep	0.9W / 0.3W / 0.1W	10s / 70s	105s

Table 5.1: Evaluation of different systems in term of power states, power consumption and transition times. The T_{limit} before best mode selection becomes trivial is estimated for a 10 MHz operating frequency for the NXP and 100 MHz for other devices.

5.1.4 System Representation

Presentation of the Markov Decision Process Framework

A mathematical framework has to be defined to describe the decision making of the system and apply optimization algorithms. Hence, this subsection introduces the concept of MDP¹.

This discrete time stochastic control process models the action selection among several alternative possibilities where the outcomes are partly random (the SoC must wake-up) and partly under control of a decision maker (the SoC goes into a suspend mode). MDP are an extension of the well-known Markov chains [231, 232]. They introduce actions and rewards to the decision process hence allowing the concepts of choice and motivation, in order to perform an action under given circumstances. A finite MDP is defined using 4 sets (S , A , P_a and R_a), with:

- S : a finite set of state s ;
- A : a finite set of action a and $A_s \subset A$ the finite set of action available from state s_t ;
- $P_a(s, s') = Pr(s_{t+1} = s' | s = s, a = a)$: the probability that action a in state s at a given time t will lead to state s' at a time $t + 1$;
- $R_a(s, s')$: the immediate reward received after transitioning from state s to state s' , following the action a .

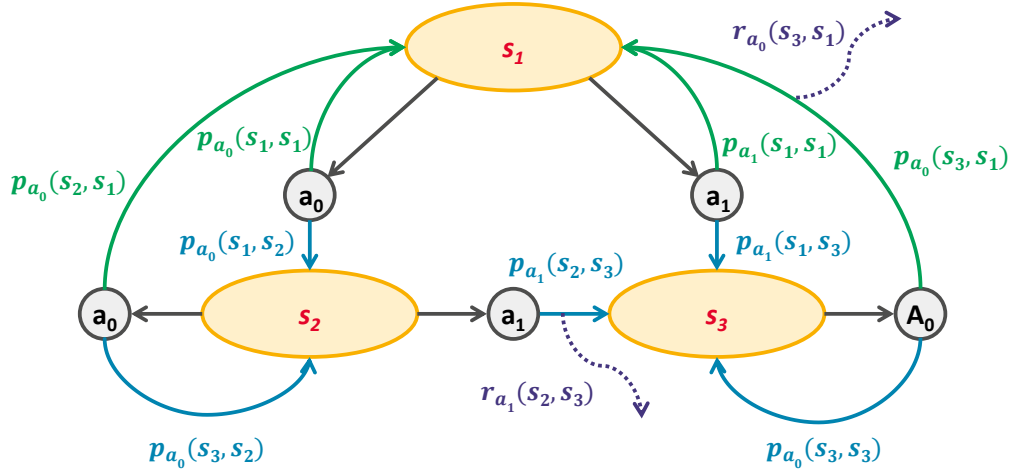


Figure 5.4: Example of simple finite MDP with 3 states (s_1, s_2, s_3), 2 actions (a_0, a_1) and 2 rewards ($(r_{a_1}(s_2, s_3), r_{a_0}(s_3, s_2))$).

As shown in Figure 5.4, for a given time t , the process is in a state s . An action a available in the state ($a \in A_s$) is selected by the decision maker. Then, the process responds at the next time step by switching into a new state s' , allowing the decision maker to collect a reward $R_a(s, s')$. The probability the process moves into a new state is given by the state transition function $P_a(s, s')$, depending on s' , s and a . However, for a given s and a , it is independent of all previous states and actions, thus satisfying the Markov property (i.e., the memoryless property of a stochastic process) [231]. A finite MDP, (i.e., finite number of states and actions) can be seen as a stochastic generalization of FSMs.

¹The name of MDPs is a reference to the Russian mathematician Andrey Markov.

Introduction to the Time-indexed Semi Markov Decision Process Framework

The Time-indexed Semi Markov Decision Process (TiSMDP) model extends the MDP by adding information on the time spent in each mode. As shown in Figure 5.5, within this diagram, an action can only occur from a state on a time index t_i to a state on the next time index t_{i+1} . This representation allows two consecutive decision instances to present a different MDP representation as well as variation in execution timing. TiSMDP multiplies the number of states by the number of time step, then increasing the time of learning in a linear way. It also enlarged the number of parameters to be stored to execute the model. This solution system representation has been explored in Section 5.2.2 for performance evaluation.

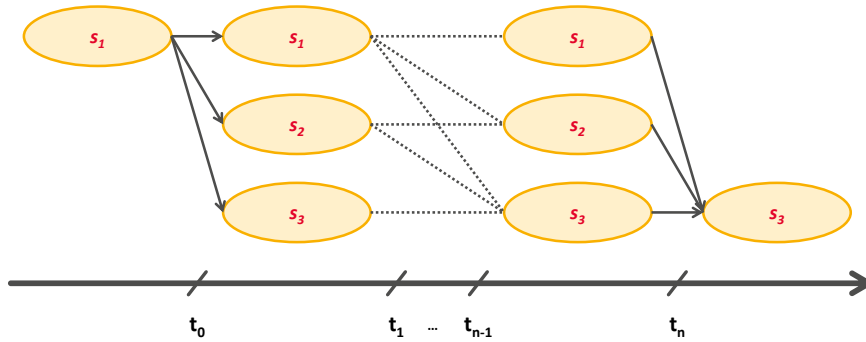


Figure 5.5: Time-indexed Semi Markov Decision Process (TiSMDP) representation using 3 states.

5.1.5 Reinforcement-Learning through Q-learning

Utilization of Machine Learning

Machine Learning (ML) regroups several types of algorithms and statistical models that systems can use to perform a specific task without a need for explicit instructions.

The first category of ML are supervised algorithms [233]. They build mathematical model based on a set of data that contains inputs and the desired outputs. Supervised Learning addresses a lot of interesting problems from classifying images to translating text. However, it requires a set of labelled data to perform training and optimize its accuracy [233]. Consequently, a second category of algorithms, called unsupervised learning, consist of defining patterns in data set without pre-existing labels. This type of self-organized learning only uses input data.

In both cases data sets are required to test the system, however they do not necessarily exist for all domains. Creating such data bases might thus be expensive or unfeasible. Consequently, it leads to a third main concept of ML: Reinforcement Learning, where actions are learned by “trials and errors”. Indeed, RL consists in learning the actions to be taken from experiences, in order to optimize a quantitative reward over time. Through iterative practices, the agent seeks an optimal decision-making behavior by maximizing its rewards over time. RL does not assume knowledge on the mathematical model of the MDP, so it can be used when exact models are not available. In that sense, RL algorithms are model-free; They only depend on actions and rewards to infer the best actions.

Reinforcement Learning Framework

From the system definition and the given environment of our system, RL algorithms appears to be a preferable solution. As shown in Figure 5.6, the reinforcement learning model is formally given using an agent which evolves in a given environment that influences its behavior [234]. An agent is an autonomous entity which acts directing its activity towards achieving goals. For our application, a SoC (the agent) sees its power consumption evolve according to the mode selected and the actions performed in each mode. The environment represents the battery of the system as well as the physical components which drain energy from this source.

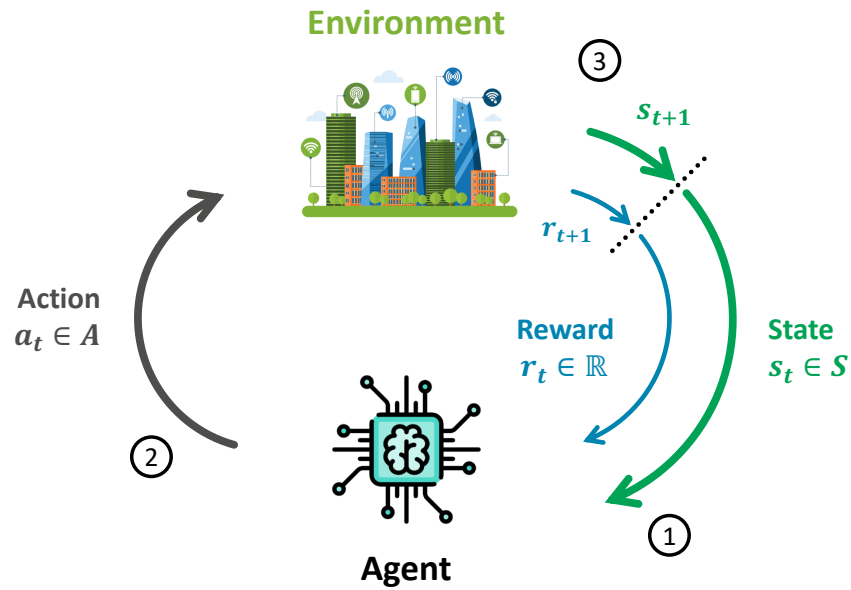


Figure 5.6: Basic agent and environment interactions used in RL algorithms.

RL algorithms are based on the general MDP framework which uses the states, actions and rewards to determine the quality of a state-action combination over the following function domain:

$$RL: S \times A \rightarrow \mathbb{R} \quad (5.1)$$

At each time step t of the algorithm ①, the agent perceives its state s_t and all possible actions A_s . Then ②, it selects an action $a_t \in A_s$. Accordingly ③, it receives from the environment a new state s_{t+1} and a corresponding reward r_{t+1} .

Based on these interactions, the reinforcement learning algorithm helps the agent to develop a policy that leads to maximize the number of rewards. The rewards can also be replaced by a malus (or penalty). In this situation, the amount of maluses should be minimized.

Reinforcement Learning Implementations

Several techniques are available to implement RL algorithms. First, Monte Carlo (MC) methods are employed but require random functions which can be expensive in terms of HW [233]. Time Difference (TD) learning extends MC methods by allowing adjustments during the algorithm execution, according to the current state of its prediction [226]. In return, it leads to increase the number of computations. This technique is called bootstrapping.

Lastly, State-Action-Reward-State-Action (SARSA) and QL algorithms tend to optimize a function representing the state of the system, its possible actions and potential rewards that might be collected selecting an action [234]. These two algorithms slightly differ by their on and off policy characters.

Indeed, every action performed by the agent yields a reward but the decision of which action to choose is made by the policy. Algorithms concerned about the policy which yielded past state-action decisions are referred as on-policy algorithms (SARSA). Those ignoring it are known as off-policy (QL). This character of the algorithm helps determine absolute rules to be followed however it might be restrictive in an unknown environment [234].

Lastly, deep RL techniques have also been developed by using NN architecture and the aforementioned methods. This solution does not require to explicitly design the state space. Moreover, it is capable of scaling RL techniques in problems that were previously unsolvable except at the price of expensive HW resources². A summary of RL algorithms is given in Table 5.2.

Algorithm	Model	Policy	Operator	Bootstrapping
MC	Model-free	Off-policy	Mean	No
TD	Model-free	Off-policy	Mean	Yes
SARSA	Model-free	On-policy	Q-function	No
QL	Model-free	Off-policy	Q-function	No

Table 5.2: Comparison of Reinforcement Learning algorithms implementations.

The Q-learning Algorithm

Q-learning is a model-free reinforcement learning algorithm³. It does not require a model of the environment and of its reward and next-state probability distributions. Moreover, it handles problems with stochastic transitions and rewards. It is also considered as an explicit trial-and-error algorithm [234]. For any finite MDP, the Q-learning algorithm leads to a policy which maximizes the expected value of the total reward over any and all successive steps. Thus, it is optimal.

The algorithm itself evaluates a quality function Q on a state-action combination:

$$Q : S \times A \rightarrow \mathbb{R} \quad (5.2)$$

²Recently, Alpha Star by Google Deep Mind can master the real-time strategy game StarCraft II and beat human players [235].

³The "Q" names the function that returns the reward used to provide the reinforcement, which can be seen as the "quality" of an action chosen for a given state.

Each time t the agent selects an action a_t , observes a reward r_t and enters a new state s_t , the Q function is calculated using a value iteration update (aka. backward induction [236]). Formerly, $Q_t(s_t, a_t)$ is evaluated ($\forall t \in \mathbb{N}^*$) by a weighted average between the old value and the new information from a state s_t to s_{t+1} as shown in (5.3):

$$Q_{t+1}(s_t, a_t) \leftarrow (1 - \alpha) \cdot \underbrace{Q_t(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a(Q_t(s_{t+1}, a))}_{\text{estimate of future value}} \right)}^{\text{learned value}} \quad (5.3)$$

The α coefficient is the learning rate ($0 \leq \alpha \leq 1$). It determines to what extent the newly learned value overrides the old information. On the one side, if α is set to 0, the algorithm only exploits prior knowledge and learns nothing new. On the other side, when the α factor is set to 1, it makes the agent ignore prior knowledge and explore new possibilities, thus considering the most recent information. This would correspond to a fully deterministic environment. Therefore, under stochastic conditions, α is decreased to be closer to 0. The learning rate is generally selected as a constant for all time steps t .

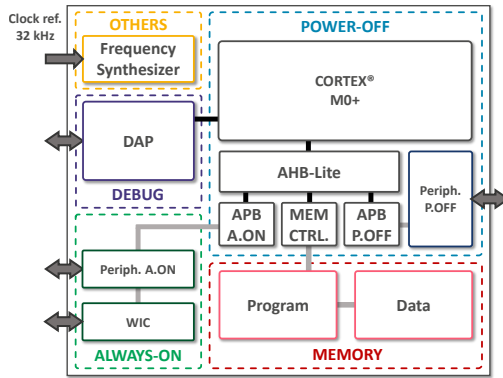
The discount factor γ expresses the capacity of the agent to foresight future rewards. When set to 0, the system becomes myopic as it only considers the current rewards r_t at a time step t . While γ approaches 1, the agent starts to strive for long-term high rewards, and when $\gamma \geq 1$, the action values may diverge, which could lead to propagation of errors and instabilities in the system [234]. To accelerate learning, the agent can start with a lower discount factor and increase its value towards a final value closer to 1 [233].

Lastly, Q-learning is an iterative algorithm which relies on an initial condition before the first update occurs. Optimistic initial conditions represented by high initial value, will encourage exploration of other states [237]. Indeed, no matter the action selected by the agent, the update of the Q function will cause lower values than the other alternative, hence increasing their future selection by the agent. In this work implementation, the first reward is used to reset the initial conditions. Therefore, the first time an action is chosen, the reward sets the value of the Q-function. Assuming preferential states for a given application, the Q-function values could be set to predetermine initial conditions. Nevertheless, this would require a form of training of the system over that particular application and access to the Q coefficients. This kind of application could be seen as a form of model training, which is available for neural networks.

5.1.6 Application on an ULP SoC

Algorithm Application

The APM technique using RL is first applied on the generic SoC architecture of Figure 5.7a. Eight system states are derived as shown in Figure 5.7b. They combine frequency switching from 16 MHz to 8 MHz or 32 kHz and power modes relying on techniques like clock and power gating of the A.ON, P.OFF and MEM domains.



(a) SoC architecture.

	Frequency Mode		
	16 MHz	8 MHz	32 kHz
Power modes	Idle	Active polling Active Active Active	Active polling Active Active Active
	Sleep	Clock gated Clock gated Active Active	Clock gated Clock gated Active Disabled
	Deep Sleep	Power gated Clock gated Active Active	Power gated Clock gated Active Power Gated

(b) SoC power modes.

Figure 5.7: Generic SoC architecture used as a template for the APM implementation along with the power modes and states available. The power domains are represented using the following color: P.OFF in blue, MEM in red, A.ON in green, and Others in yellow.

Using these system states, an MDP representation is obtained as shown in Figure 5.8. The actions are defined as a possible transition between the system's power states. They present a probability p to be fully accomplished (blue arrow) and a probability $1 - p$ to be interrupted (green arrow) by a wake-up signal. This signal also results in a return to the Idle-16 MHz state until the next suspend sequence. Five different actions are thus obtained:

- a_0 : Remain in the state;
- a_1 : Switching frequency to 8 MHz;
- a_2 : Switching frequency to 32 kHz;
- a_3 : Switch from the current state to Sleep;
- a_4 : Switch from the current state to Deep Sleep.

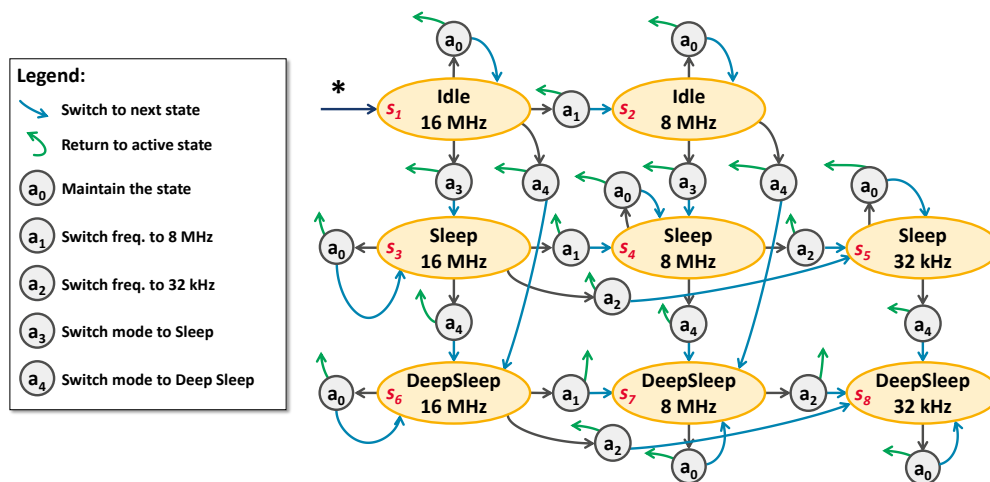


Figure 5.8: Generic system MDP with 8 states and 5 actions. Rewards and transition probabilities are not represented.

The number of available actions is a trade-off between complexity and flexibility of the system. Indeed, only mode or frequency transitions are available whereas simultaneous frequency/mode switching are excluded. Moreover, only transitions to less consuming power modes are evaluated. This design choice helps relax the mode transition constraints on the WUC in charge of hardware control. It also reduces the amount of Q-coefficients to be stored to perform the algorithm application.

All available actions are achieved after a known number of system cycles depending on the system implementation. For instance, a frequency transition from 16 MHz to 8 MHz is faster than mode transition from Idle to Deep Sleep. However, the action a_0 does not rely on a system-defined number of cycles. Instead, a C_{loop} parameter corresponding to the number of hardware wait cycles is used.

Definition of the Q-matrix

The Q-Learning algorithm is then applied on the system MDP of Figure 5.8 in order to select the best action a_t at an algorithm step t when the system is in a state s_t .

As each action is associated to a unique state transition, the MDP diagram can be described using unique coefficient $Q(s_t, a_t)$ as shown in Figure 5.9a. A Q-matrix is then associated (see Figure 5.9b) which contains the $Q_{i,j}$ with i the index of the starting state and j the final's state index after a transition. These coefficients are updated after any suspend sequence depending on the system power consumption.

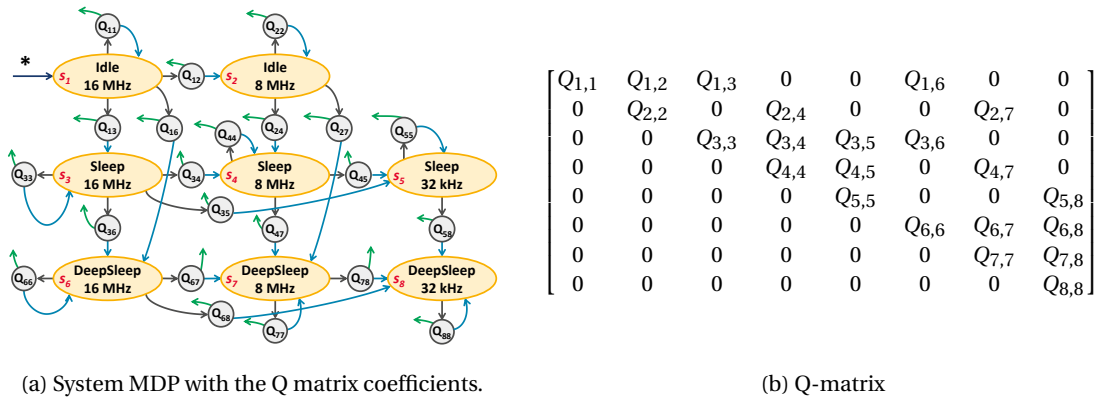


Figure 5.9: MDP diagram of the system and Q-matrix associated.

This representation also helps understand how the algorithm is applied by clarifying how Q-matrix coefficient are used. When the system reaches a specific state, it evaluates the coefficients corresponding to nearby actions. Then, it follows the path with the lowest $Q_{i,j}$ coefficient to select the next state. After a complete transition to the selected state, the $Q_{i,j}$ values are updated.

SoC Power Consumption and Transition Time

To describe the system and evaluate the power consumption depending on the activity, an energy E and transition time matrices T_P and T_F are necessary. As shown in (5.4), E describes

the energy/cycle associated to each state of Figure 5.8. For this first architecture, results are obtained on the NZP28 - V1.0 presented in Chapter 4, Section 4.5 are used. As a first order model, the energy consumed during a transition between two different modes is derived using the power consumption of the source and destination mode multiplied by the transition time. The wake-up energy is calculated in the same way.

$$E = \begin{bmatrix} E_{S1} & E_{S2} & 0 \\ E_{S3} & E_{S4} & E_{S5} \\ E_{S8} & E_{S7} & E_{S8} \end{bmatrix} = \begin{bmatrix} 2.594 & 3.33 & 1.52 \\ 1.288 & 2.15 & 0.60 \\ 1.162 & 1.79 & 0.08 \end{bmatrix} \quad (5.4)$$

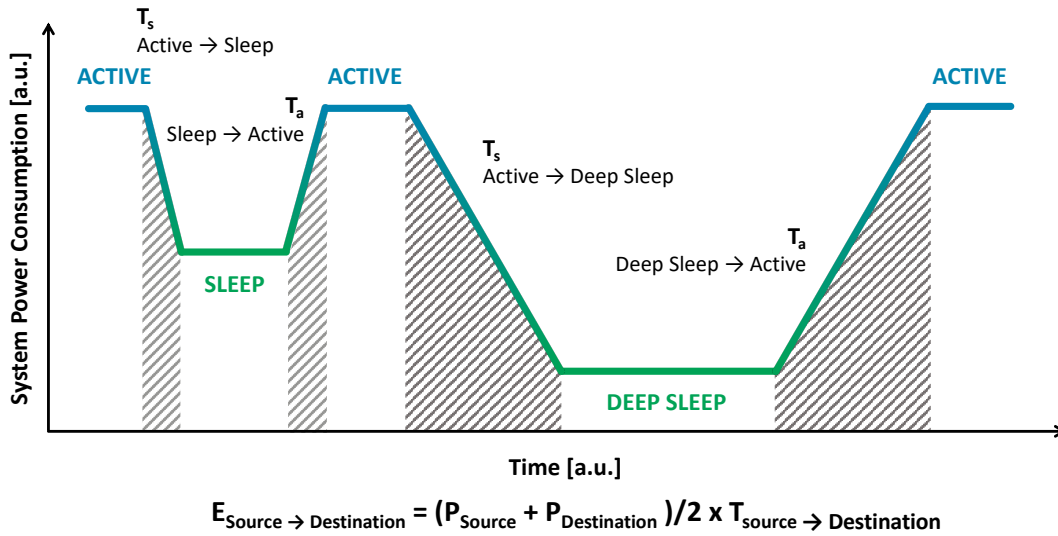


Figure 5.10: Definition of the energy consumed during the system transitions.

As shown in (5.5), the values stored T_P represent the transition between the power modes of the SoC, whereas T_F expresses the transition time to switch between frequencies. Values are given in cycles. To simulate the proper behavior of a generic SoC, transition value from active to idle and active to sleep are based on NZP28 - V1.0 measured value.

The deep sleep transition is extended (if not it would limit the algorithm lifespan to 5.5 μ s as previously explained in Section 5.1.3). Consequently, the deep sleep transition is estimated to 4500 cycles. It is given has a worst case for which a full data writing (in burst mode) is done in a 4 KB external memory. This reflects the storage of the core state and application information for data retention. The energy cost of the transfer is accounted thru the Active mode power consumption

The value given in Table 5.1 is the corresponding time assuming a 10 MHz frequency.

$$T_P = \begin{bmatrix} T_{\text{Idle} \rightarrow \text{Idle}} & T_{\text{Idle} \rightarrow \text{Sleep}} & T_{\text{Idle} \rightarrow \text{DeepSleep}} \\ - & T_{\text{Sleep} \rightarrow \text{Sleep}} & T_{\text{Sleep} \rightarrow \text{DeepSleep}} \\ - & - & T_{\text{DeepSleep} \rightarrow \text{Sleep}} \end{bmatrix} = \begin{bmatrix} 1 & 50 & 4500 \\ 0 & 1 & 4500 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.5)$$

For T_F , margin have been added to consider the programming time and settlement of the various clock generators of the system. T_F is given in (5.6)

$$T_P = \begin{bmatrix} T_{16\text{MHz} \rightarrow 16\text{MHz}} & T_{16\text{MHz} \rightarrow 8\text{MHz}} & T_{16\text{MHz} \rightarrow 32\text{kHz}} \\ - & T_{8\text{MHz} \rightarrow 8\text{MHz}} & T_{8\text{MHz} \rightarrow 32\text{kHz}} \\ - & - & T_{32\text{MHz} \rightarrow 32\text{kHz}} \end{bmatrix} = \begin{bmatrix} 1 & 20 & 100 \\ 0 & 1 & 100 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.6)$$

Q-learning Operations

To minimize the power consumption of the system and match with the goal of the APM, a penalty-based variation of the Q-learning is used [224]. For each algorithm step, an action a_t is selected according to a specific policy. The usual default policy is to select the action from the state s_t with the lowest Q coefficient:

$$a_{t+1} = \underset{a}{\operatorname{argmin}} Q_t(s_t, a) \quad (5.7)$$

Then, the coefficient associated to the selected action is updated as shown in (5.8):

$$Q_{t+1}(s_t, a_t) \leftarrow (1 - \alpha) \cdot Q_t(s_t, a_t) + \alpha \cdot \left(M_t + \gamma \cdot \min_a (Q_t(s_{t+1}, a)) \right) \quad (5.8)$$

Here, a penalty coefficient $M_t = \delta E$ is used, corresponding to the amount of energy consumed by the system during the state transition. With this implementation, the algorithm tends to minimize the whole energy consumption of the system by seeking minimum energy penalties. The Q-learning algorithm application is summarized in the following pseudo-code.

Algorithm 1 Q-learning

1: **Initialize:**

$$\forall (s, a) \in (S, A) : Q_0(s, a) = 0$$

$$s_0 = s_{init}, a_0 = \emptyset$$

2: **for** $t \in [0, N]$ **do**

3: Choose an action a_t in the current states based on current Q-matrix estimates

4: Observe the outcomes s_{t+1} and M_t , resulting state and penalty of the action a_t

5: **Update:**

$$Q_{t+1}(s_t, a_t) = (1 - \alpha) \cdot Q_t(s_t, a_t) + \alpha \left(M_t + \gamma \cdot \min_a (Q_t(s_{t+1}, a)) \right)$$

$$Q_{t+1}(s, a) = Q_{t+1}(a, s) \quad (\forall (s, a) \neq (s_t, a_t))$$

6: **end for**

5.2 Q-learning for Power Management: Generic SoC Application

This section describes the implementation of the APM HW module on a generic SoC presented in Section 5.1.6, Figure 5.7. First, the Python/Verilog co-simulation workflow is presented. It ensures continuous integration from the model definitions up to the physical APM gate netlist. Several Q-learning solutions are then benchmarked and validated for maximum robustness, convergence and accuracy. A Q-learning coefficient methodology is also proposed to maximize the energy gain. Lastly, conclusions are drawn on the APM application for generic ULP SoCs.

5.2.1 Simulation Environment and Methodology

Whereas Hardware Description Languages (HDLs) are convenient for HW description and implementation, they remain tedious for system modeling and evaluation. On the contrary, Python is a high-level general-purpose programming language allowing modeling, simulations and validation of algorithms. It is scalable and offers high modularity, leading to fast update of test benches and components. Moreover, the Python's large standard library provides numerous tools to test and evaluate the performance of a module, while maintaining an integrated python environment. Since current industrial synthesis tools do not support Python HW descriptions⁴, a Python/Verilog co-simulation workflow has been developed (see Figure 5.11).

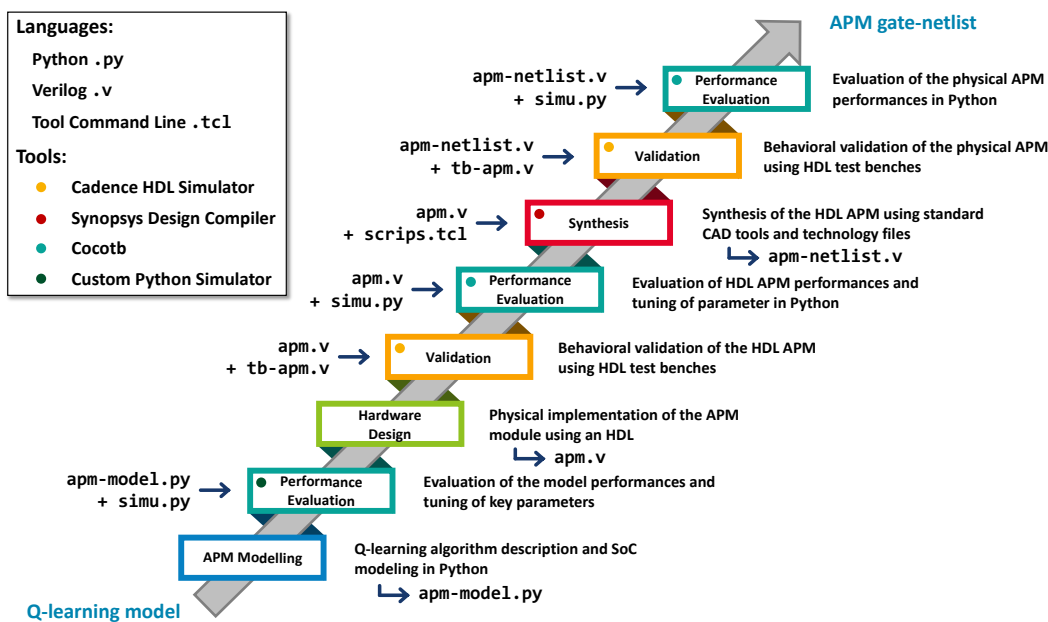


Figure 5.11: Mixed HDL–Python conception and validation flow developed for co-validation of the APM HW module.

Python is selected to build an APM model integrating a proposal for an RL algorithm implementation known as `apm-model.py`. Several testbenches are then derived to test the model for a given task and potential perturbations, all based on the generic architecture of

⁴The free opensource MyHDL package, which uses Python as a hardware description and verification language still requires hazardous conversions to Verilog or VHDL [238].

Section 5.1.6. These tests are realized during a performance evaluation step with a `simu.py` module operated in a custom Python simulator.

Consequently, an implementation of the algorithm is done in Verilog leading to `apm.v`. A behavioral validation of the HW module is done using a test bench written in Verilog (`tb-apm.v`). For this task, a Cadence HDL simulator is used. Then the performances of the module are checked using the first Python test `simu.py`.

To operate the test and verify the Verilog RTL using Python, the COroutine based COsimulation TestBench (`cocotb`) environment is used [239]. No additional RTL code is required, the `cocotb` testbench uses `simu.py` to drive stimulus onto the inputs of `apm.v` and monitors the outputs directly in Python. The same Cadence simulator is used to simulate the RTL and observe the signals from the Device Under Test (DUT).

After testing and validation of the `apm.v` module, a physical synthesis is performed using standard CAD tools and `.tcl` scripts. It produces a physical implementation (i.e., gate-netlist) of the module (`apm-netlist.v`). Again, the same behavioral Verilog tests and python modules are used to check and evaluate the performances.

5.2.2 Benchmarking of Several Reinforcement-Learning Solutions

Using the aforementioned Python environment, several RL algorithms are tested. The QL is evaluated on two system representations (MDP and TiSMDP), whereas the TD uses a TiSMDP. The convergences of these implementations are given in Figure 5.12. The deep sleep heuristic and the minimum achievable energy are also plotted as references. The TD solution is discarded since it shows a lack of accuracy in such variable environment. The heuristic solution appears stable however, it only works for a precise number of cycles.

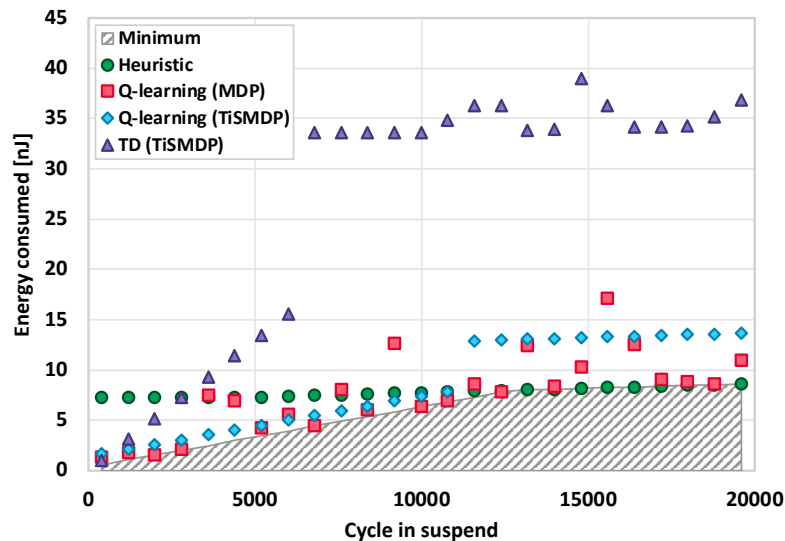
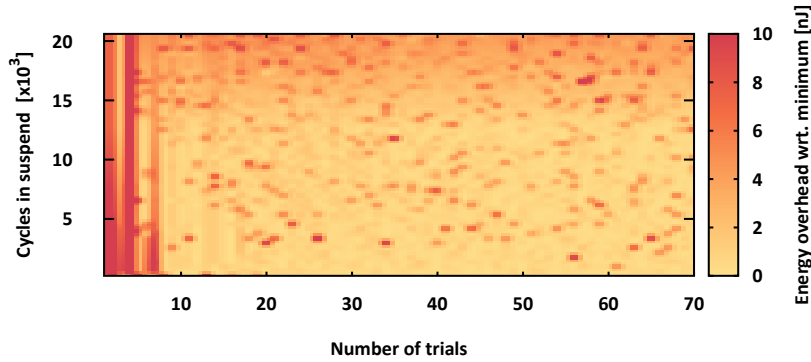
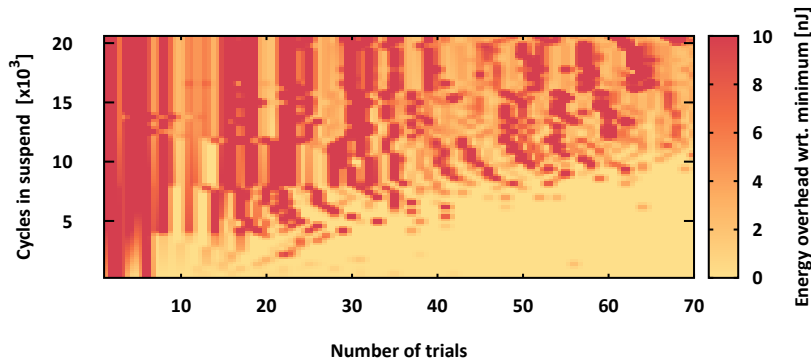


Figure 5.12: Evaluation of different RL-algorithms. Q-Learning (QL) is applied on a Markov Decision Process (MDP) and Time-indexed Semi Markov Decision Process (TiSMDP) representation. Time Difference uses a TiSMDP. A deep sleep solution is selected as a reference heuristic. The shaded grey area corresponds to the minimum energy that can be reached.

The remaining QL models are evaluated using a color map representation as shown in Figure 5.13 (for a number of trials up to 70). The energy consumed is given relatively to the minimum solution. On the one hand, the MDP representation ensures a fast and agile learning. After a few numbers of cycles, a low energy consumption is reached. On the other hand, the TiSMDP offers the best convergence for short periods of suspend. Moreover, the learning is slower; it requires more learning sequences to converge towards an optimal solution. Thus, this behavior is detrimental to preserve the energy consumed for a low number of cycles and trials.



(a) Energy color map of the Q-learning solution using an MDP representation.



(b) Energy color map of the Q-learning solution using an TiSMDP representation.

Figure 5.13: Energy color map of a the Q-learning algorithms applied on a MDP (top) and TiSMDP (bottom).

Performance metrics of the previous solutions are given in Table 5.3. As expected, the deep sleep heuristic does not require learning to be applied. Considering the first 10 trials, the QL on MDP cost 44.6 nJ compared to the minimum solution. The TiSMDP gives a higher learning over-cost of 293 nJ. After convergence, the energy consumption averaged for trial between 10 and 70 cycles in suspend mode up to $20 \cdot 10^3$, are 2.04 nJ/trial, 1.83 nJ/trial and 1.75 nJ/trial for respectively the deep sleep heuristic, the QL on MDP and TiSMDP. Following these results, the QL solution applied on MDP appears preferential. It reaches a low energy consumption after a short amount of trials with a limited learning cost.

Algorithms metrics	Deep Sleep Heuristic	Q-learning on MDP	Q-learning on TiSMDP
Learning cost ¹	0 nJ	44.6 nJ	293 nJ
Energy consumption ²	2.04 nJ/trial	1.83 nJ/trial	1.75 nJ/trial

¹ Obtained on the first 10 trials.

² Averaged over 10 to 70 and cycles and suspend cycles up to 20×10^3 .

Table 5.3: Performance evaluation of the APM according to the minimum achievable energy.

5.2.3 Algorithm Robustness and Validations

This subsection now focuses on the QL algorithm based on an MDP representation of the system. In Figure 5.14 or any color map representation previously presented, the suspend activity of the system is assumed constant over the time. However, over the lifetime of the SoC perturbations or variations in the suspend sequences profile occur.

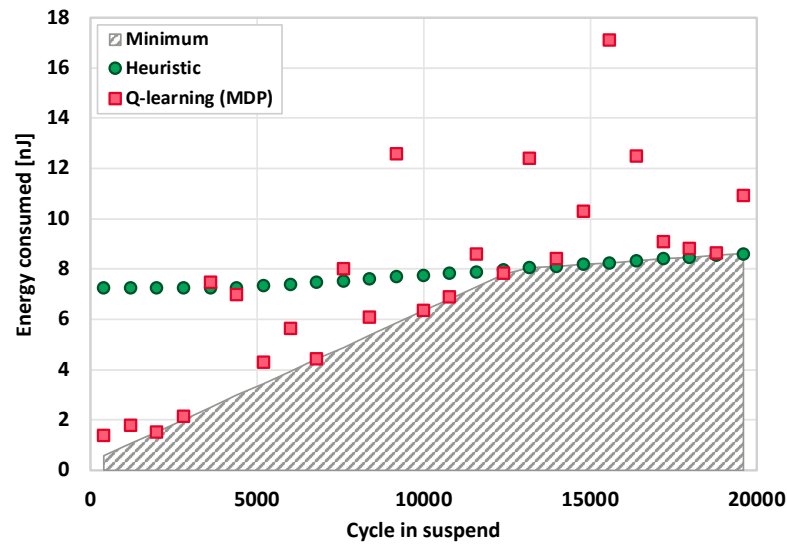


Figure 5.14: Comparison of the heuristic solution with the Q-learning algorithm.

First, the system can move from short suspend sequences to longer patterns, and conversely. This behavior is characterized by a step in the activity scenario. A continuous modification of the suspend activity can also be observed. In this case, the system extends or shortens its suspend sequences. Lastly, sudden variations (or impulsions) can result from external perturbation on a short amount of time. This last scenario can be seen as a quick occurrence of two consecutive step variations. All these behavior results are due to modification of the system activity workload, such as switching from high to low performances to handle a specific task or interruption. From these scenarios and thanks to the linearity of the system, can be derived other types of variations, assuming the system activity and constraints given in Section 5.1.3.

Therefore, these three scenarios are tested and reported in Figure 5.15. In this case, variations from 500 cycles in suspend mode to a 20000 cycles profile are given. For all type of situations, the system is able to converge to an optimal solution after a given number of trials.

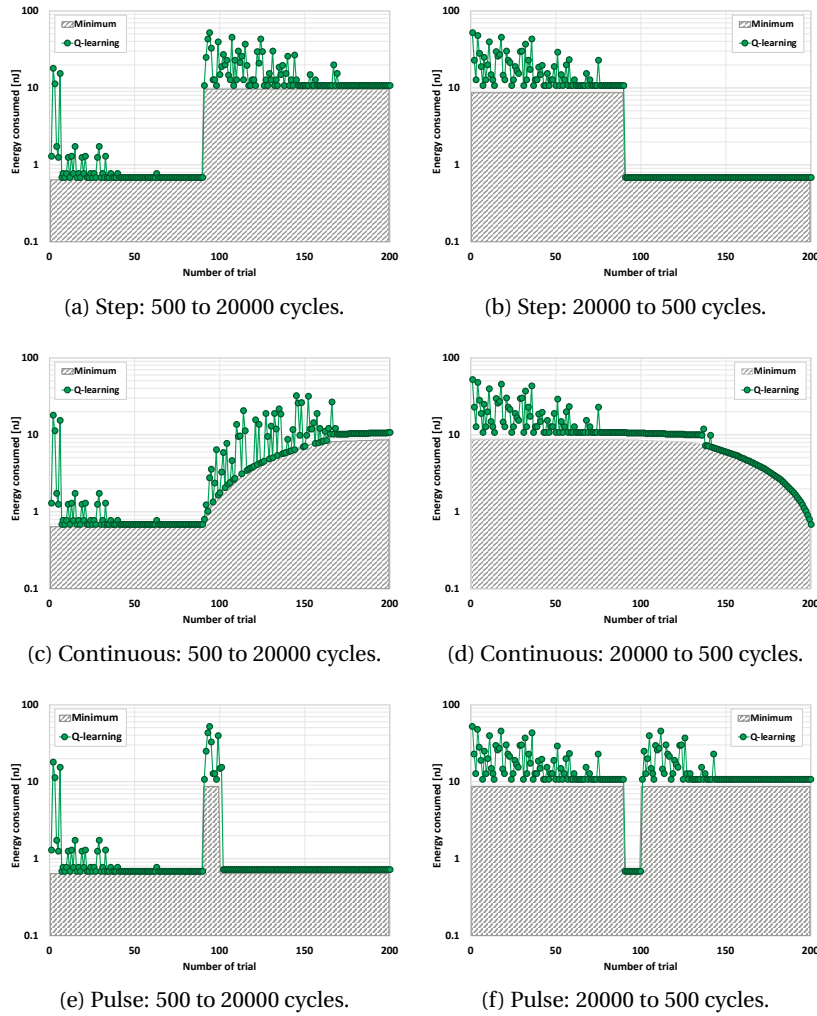


Figure 5.15: Evaluation of the robustness and stability of the Q-learning solution depending on several activity scenarios (step, continuous, impulsion) from/to 500 cycles in suspend from/to 20000 cycles.

When the number of cycles in suspend mode is small, the system is faster to find a solution close to the minimum. In fact, around 30 trials are necessary for a 500 suspend cycle activity, whereas more than 90 trials are required with the 20000 suspend cycle activity. Hence, the learning time of the APM is activity-related. This phenomenon also appears when the activity changes from 500 to 20000 cycles (see Figure 5.15a, Figure 5.15c, Figure 5.15e).

This behavior results from the initialization of the Q_{matrix} with 0 coefficients and the paths followed by the QL on the system MDP. Indeed, when the QL algorithm starts all states are treated with the same weight. There is no better initial mode than others. All states are then explored sequentially in the MDP starting from 0 to 8 (cf. Figure 5.8). However, for short suspend sequences, the modes reached first (mostly idle S_1/S_2 and sleep $S_3/S_4/S_5$) are profitable. They consume more energy yet present a shortened timing overhead to be entered and left. Consequently, an optimal solution is found with less trials. For longer suspend sequences, the algorithm has to reach the deep sleep modes ($S_6/S_7/S_8$) to sense if they are optimal for the

longer activity. By tuning the Q_{matrix} initialization with higher coefficient for the idle and sleep states, better convergence can be observed for longer suspend sequences. On the other hand, it is detrimental for short suspend sequences. This concept will be extensively discussed in Section 5.4.1.

By observing the convergence of the QL implementation after a scenario transition from 20000 to 500 cycles, it comes out the QL is faster to find an optimal solution (in the idle or sleep states). This time, the Q_{matrix} coefficient are already initialized and, the QL starts its operations with the best state (i.e.; the lowest coefficient). However, when the transition occurs, the timing penalties resulting from the deep sleep transition (see (5.5)) are way above the 500 cycles. Even though the deep sleep states are extremely low power, the transition time of 4500 cycles results to 10 nJ, whereas 0.8 nJ can be reached in sleep states. Therefore, the corresponding Q coefficient gets a malus so important, that the mode is discarded for the next trial⁵. Consequently, the robustness of the QL is related to transition time between mode. The better the states are spread in term of power overhead due to the transition time, the better the convergence will be improved.

From all this observation can be drawn two major conclusions. First, the QL can recover from variation in the activity. However, due to a required convergence time, the new activity must be constant for at least the learning time. Assuming a known activity, the Q_{matrix} could be tuned to make some states beneficial. Secondly, the transition time between mode impacts the convergence. Then, a system with a large amount of modes is not necessary to reduce the power consumption. A few modes with sufficient spreading in term of transition time and power consumption should be preferred. It would also decrease the HW implementation in term of data and coefficient that need to be stored.

5.2.4 Algorithm Optimizations

Exploration

As shown in Figure 5.16 with the green line (no exploration), in some specific activity scenarios, the algorithm does not reach the minimum achievable energy. In that case, the algorithm pursues its learning in a wrong direction and stays locked onto a local minimum. The QL algorithm supposes that some actions to reach another state are not interesting, because the action's coefficient is never the minimum in the Q_{matrix} .

As any trial-and-error system, RL requires testing of the available actions. Therefore, an exploration policy is added to force the algorithm to select an action supposed to be non-optimal and check its status (i.e., evaluated if a neighbor action is potentially beneficial).

Various exploration policies have been studied in the literature. Ideal exploration policies such as ϵ -greedy and Boltzmann (softmax) policy relies on random testing [234, 240]. During the action selection process, a random probability is added which results in the selection of a random action instead of choosing the action associated to the best coefficient. This solution is simple and enables the algorithm to search for better solutions and avoid local minimum. However, it relies on a true random generator that might be complex to integrate in a first version.

⁵There is actually a one trial delay after the transition edge, where the QL continues to select the previous optimal state. However, this choice results in an important malus, leading to select a mid-power range state in the sleep power modes.

Improved solutions tend to reduce the exploration rate during the system's activity to decrease the energy consumed by the random selection of an energy-hungry action. Such techniques are: SA-Q-Learning and ERE-Q-Learning from [241] and pursuit from [242]. However, for system assuming modification in the activity, they cannot be easily adapted. Lastly, Upper Coefficient Bound (UCB) based solutions are interesting but lead to complex implementations which increase the hardware cost [242].

Therefore, a counter-based exploration has been developed where the algorithm performs exploration at a regular rate and simplify the integration. The solution relies on a first counter C_s incremented every time a new state is selected. Secondary counters C_{a_n} are incremented when the actions are selected, thus tracking the actions picking by the algorithm. When C_s reaches a programmable threshold value, a new action is selected corresponding the highest value of C_{a_n} .

This solution can be easily implemented in HW and only relies on secondary counters variables. As shown in Figure 5.16, this solution leads to similar result than the ϵ -greedy policy. In fact, a counter-based exploration every N cycles reflects an ϵ -policy with a probability $1/N$.

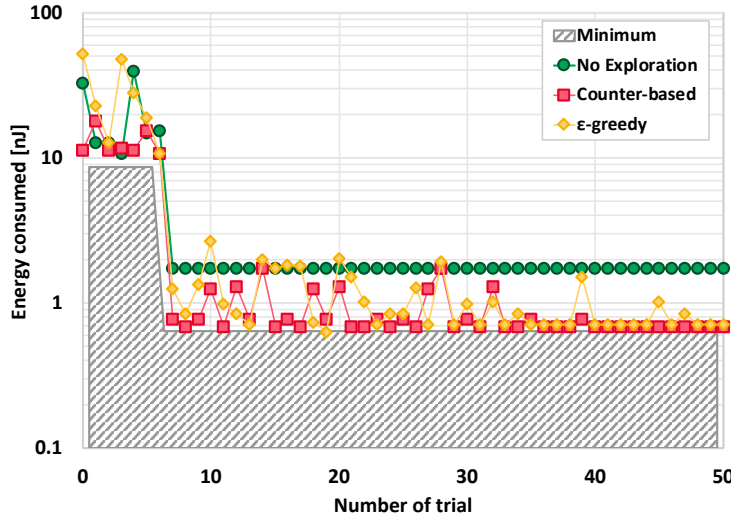


Figure 5.16: Evaluation of three different exploration policies: no exploration, ϵ -greedy and the counter-based version.

Computations using Integer Representation

The learning formula requires floating variables. To reduce the hardware cost, it was simplified to use integers only.

Assuming $Q = Q(s_t, a_t)$, let's define $\alpha_0 = \lfloor \alpha \cdot 2^{size(\alpha)} \rfloor$ and $\gamma_0 = \lfloor \gamma \cdot 2^{size(\gamma)} \rfloor$ as the integer approximation of respectively α and γ . The learning formula can then be adapted as following:

$$Q = (1 - \alpha) \cdot Q + \alpha \cdot (M + \gamma \cdot Q_{next}) \quad (5.9)$$

By using the integer approximation in (5.9), it can be derived:

$$Q = C_1 \cdot Q + C_2 \cdot M + C_3 \cdot Q_{next} \gg size(\alpha) \quad (5.10)$$

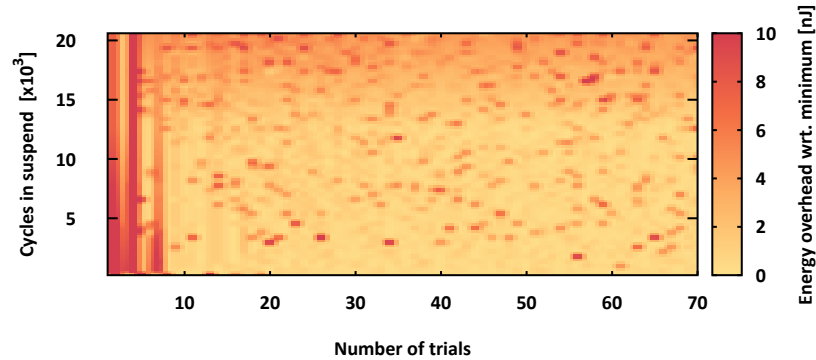
This simplification reduces the computation complexity of this formula. Accordingly, the systems will work with 2 additions and 3 multiplications of the variables with the constants:

$$\begin{cases} C_1 = 2^{\text{size}(\alpha)} - \alpha_0 \\ C_2 = \alpha_0 \\ C_3 = \frac{\alpha_0 \cdot \gamma_0}{2^{\text{size}(\gamma)}} \end{cases} \quad (5.11)$$

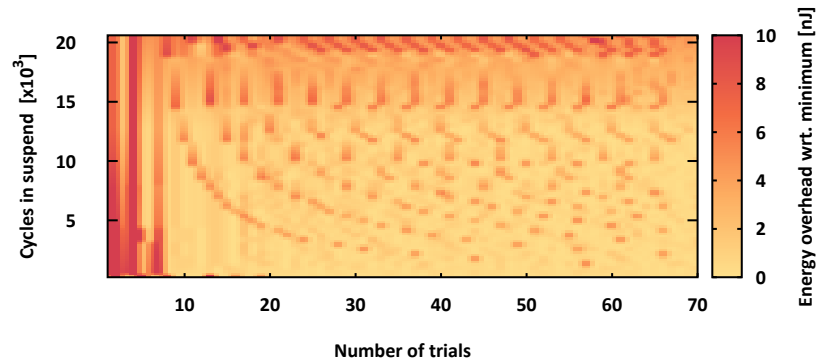
With large number of bits, the approximation made previously becomes negligible. Then, the size of every constant and coefficient has been set to 24 bits. This approximation is validated with python simulations. No degradation is observed on the final result. The HW implementation `apm.v` using this integer representation is presented in Section 5.2.6.

Resulting Implementation

The color map of the resulting QL implementation using counter-based exploration and integer representation is given in Figure 5.17b and compared to the reference using floating point number and no exploration of Figure 5.17a. These results are obtained on `apm.v` model. Regular over consumption artefacts appear, corresponding to the exploration trials. Even though it added some extra power consumption at a given number of trials, it avoids to be stuck in a local minimum, thus globally reducing the energy consumed by the system. This implementation will now be used in the following sections of this work.



(a) Q-learning solution using floating point and no exploration.



(b) Q-learning solution using integers and a counter-based exploration.

Figure 5.17: Energy colormap of the implemented Q-learning solution.

5.2.5 Selection of Q-learning Coefficients

As stated in Section 5.1.5 and the presentation of QL formula, the learning rate α and the discount factor γ directly affects the learning speed and accuracy. The section analyses the selection of these two parameters in order to maximize the energy while minimizing the energy cost due to the learning time.

Figure 5.18 reports the learning cost and the energy gain of the APM according to the α and γ values, both ranging from 0 to 1. The learning cost is the energy consumed by the algorithm during an initial learning phase while, the energy gain is the energy saved relatively to the heuristic solution after learning. Extra explanations on the computation of these two metrics are given in Section 5.2.6. The configuration where $\alpha = 0$ and $\gamma = 0$ leads to a myopic system evolving in a deterministic environment. In other words, the algorithm exploits prior knowledge (i.e., learns nothing new) and only considers the current reward/penalty. All metrics are normalized to this solution which acts as a reference.

Using the results from Figure 5.18a, a γ value closer to 1 should be selected in order to maximize the energy gain. With such high value, the algorithm will look for high-reward configuration (in our case low-maluses). The α is then determined using Figure 5.18b. With a low coefficient ranging from 0 to 0.4, the system is prone to minimize the learning cost.

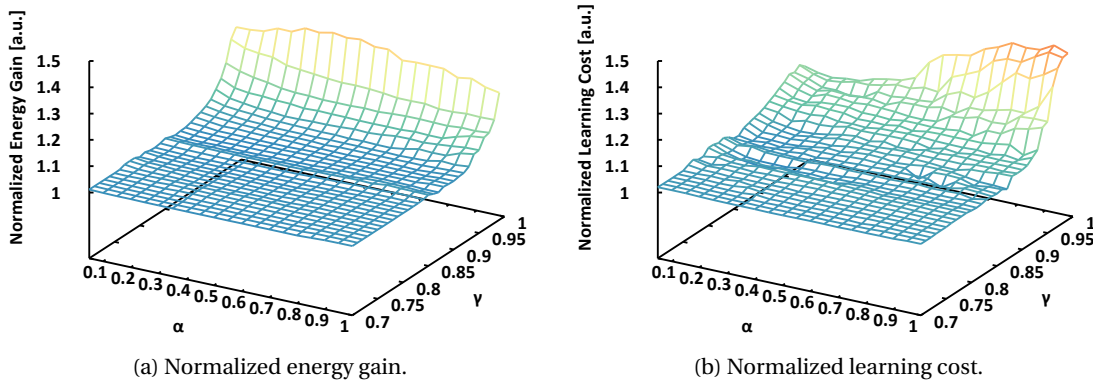


Figure 5.18: Estimation of the APM performances according to the learning rate α and discount factor γ . Plotted values represent the absolute difference with the $\alpha = 0 / \gamma = 0$ algorithm configuration.

However, α also impacts the stability of the QL operation. In Figure 5.19 are plotted the color map for several learning rate values, assuming a discount factor of 0.95. This time the reported energy consumption is the difference with the myopic/determinist solution ($\alpha = 0$ and $\gamma = 0$). On the one hand, when α is closer to 0 (see Figure 5.19a), the learning time increases. Energy spikes also appears due to non-optimal selections of the system states. Due to the slow learning, it takes more cycles to reach a better solution. On the other hand, for α closer to high values (see Figure 5.19d), instability occurs during the execution. These bad convergences into non-optimal solution impact the final energy gain.

In conclusion, the learning rate α should be small to minimize the learning cost, while ensuring stability. However, it could lead to increase the learning time. The discount factor γ should be close to 1, to maximize the energy gain obtained by the QL state selection. Finally, a couple $\alpha = 0.3$ and $\gamma = 0.95$ has been selected. These values are now used in the next sections.

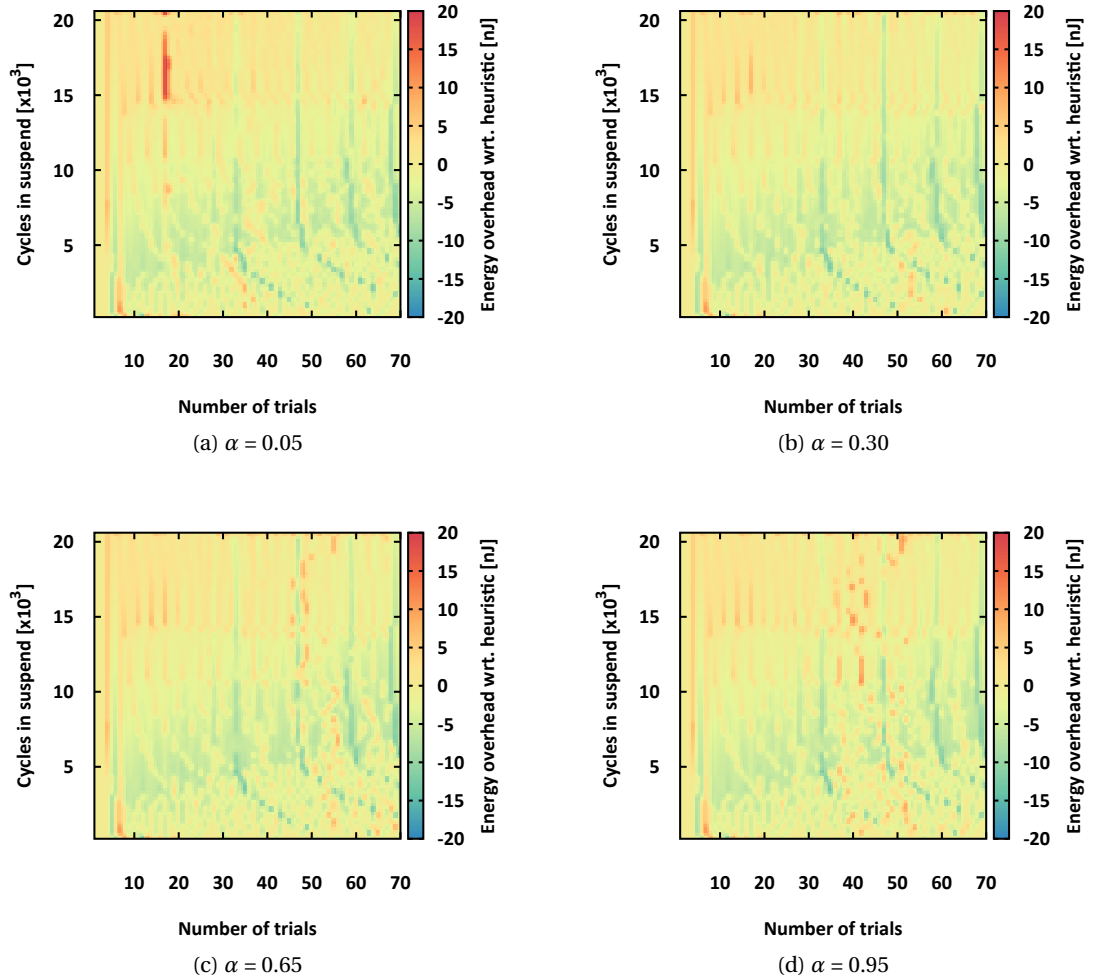


Figure 5.19: Impact of the learning rate α on the APM performances with a discount factor $\gamma = 0.95$.

5.2.6 Partial Conclusions

Using the environment presented in Section 5.2.1, the QL has been applied to create an APM model. The exploration technique, the optimization and tuning of the QL parameters of the previous section, are now fully considered in the physical HW solution `apm.v`. This section gives the results obtained on the final gate netlist `apm-netlist.v`.

Performance Evaluation

The performances of the physical APM are evaluated by comparing the energy gain and losses relatively to the deep sleep heuristic for several trials and cycles spent in suspend mode (see Figure 5.20). A negative energy consumption corresponds to a positive gain compared to the heuristic. Respectively, a positive energy is a loss. The energy penalty due to the learning phase of the system is clearly visible for a number of trials ≤ 10 . Beyond, the algorithm offers performance gains or penalties.

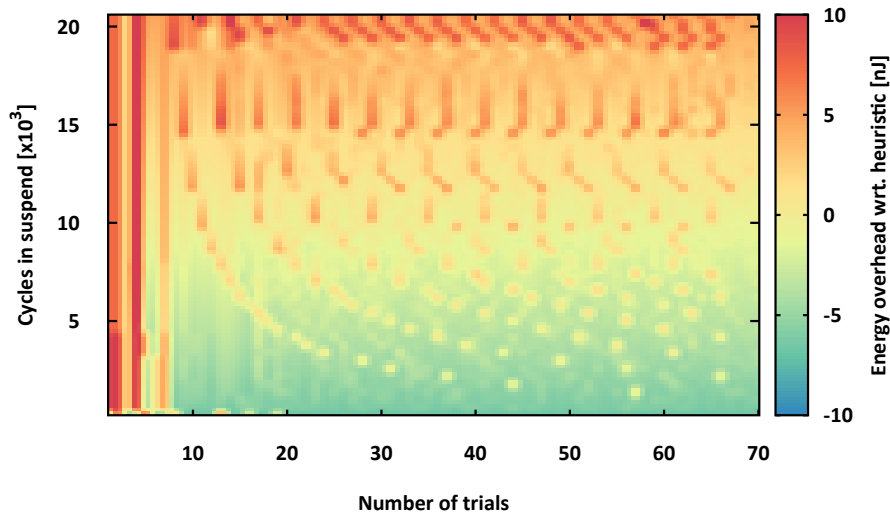


Figure 5.20: Comparison of the energy consumption gains and losses of the generic APM compared to a deep sleep heuristic solution. Energy consumed [nJ] according to the number of trials and the number of cycles the system spent in suspend mode.

Several metrics presented in Table 5.4 help to analyze the APM performances.

- Learning time L_t : maximum number of trials to get an energy ≤ 0 , and thus observe a gain at iso-cycle;
- Learning cost L_c : sum of average energy per cycle consumed for each trial during the learning time.
- Energy gain E_g : energy per cycle consumed for trial executed after the learning time. Here, it is the averaged energy per cycle for trial between 10 and 70, and a number of cycles up to $22 \cdot 10^3$. The energy saving (%) is obtained relatively to the energy consumption of the heuristic solution.
- Sustainability time S_t : number of cycles required to gain energy and thus recover from the learning cost. 66 trials are estimated, to balance the energy lost during the learning time.
- Algorithm lifespan A_{life} : maximum number of cycles to get an energy ≤ 0 , and so observe a gain at iso-trial.

Q-learning Metrics	Absolute Performances
Learning time L_t	≤ 10 trials
Learning Cost L_c	87 nJ
Energy gain E_g	1.3 nJ/trial \rightarrow 17%
Sustainability time S_t	≥ 66 trials
Algorithm lifespan A_{life}	$\leq 10^3$ cycles

Table 5.4: Performance evaluation of the APM relatively to the Deep Sleep heuristic solution.

Similarly to the learning time, the algorithm lifespan characterizes the number of maximum cycles before the APM implementation does not offer energy gain anymore. Even though T_{limit} (see Section 5.1.3) gives the time when the deep sleep heuristic is optimal, due to the exploration and fails in optimal mode selection, extra energy is consumed. Therefore, $A_{\text{life}} \leq T_{\text{limit}}$. This metric defines a hard deadline for which the system should be set to deep sleep in order to save energy.

Hardware Evaluation

Beyond the absolute energy gain proposed by the APM, the HW implementation results in extra components active in the system which consume power and area. An evaluation of the HW APM module is proposed in Table 5.5 comparatively to the NZP28 V3.0 architecture presented in Section 4.5.1. The results are obtained for the typical low-voltage implementation corner available for NZP28 – V3.0 (0.7 V, 25 °C in 28 nm FD-SOI – RVT) with a 200 MHz operating frequency.

Metrics	Power [μW]	Energy [pJ/c.]	Cells	Area [μm^2]
Generic APM	250	1.25	7078	11144
NZP28 V3.0	584	2.92	18542	60519
APM Impact	+43%	+43%	+38%	+18%

Table 5.5: Evaluation of the APM HW module on a complete SoC based on the NZP in SQUAL V3.0 architecture. Results obtained at 0.7 V, 25 °C in 28 nm FD-SOI – RVT at 200 MHz.

Due to the reduced power consumption of the whole SoC, implementing the APM would result in a +43% power and energy impact. In term of cells and area, an impact of respectively +38% and +18% is calculated. These figures are explained by the high number of coefficients that need to be stored and accessed to implement the QL algorithm. Moreover, all these values are currently stored in simple registers. Power optimization are expected by using a small memory macro.

Status

The QL implementation to perform APM depending on the activity shows promising results to save energy during the system operation. However, the resulting HW implementation highly impacts the power budget of our power constrained SoC. Optimization should thus be done to decrease the number of coefficients to be stored by reducing the number of available states. Moreover, all the ULP techniques presented in the previous chapter should be added to reduce the power impact in the SoC active modes where the APM is not useful. Lastly, assuming a more power-hungry system – resulting from the in memory size or the number of available IPs – the APM impact could become negligible and thus highly beneficial.

5.3.2 Simulated Performances

Using the co-simulation environment, an HW APM implementation is obtained and tested. This section gives the results obtained on the final gate netlist `apm-netlist.v`. Again, the deep sleep heuristic is used as a reference for performance evaluation. The energy consumption of the physical APM are given in Figure 5.22 and the metric associated in Table 5.6

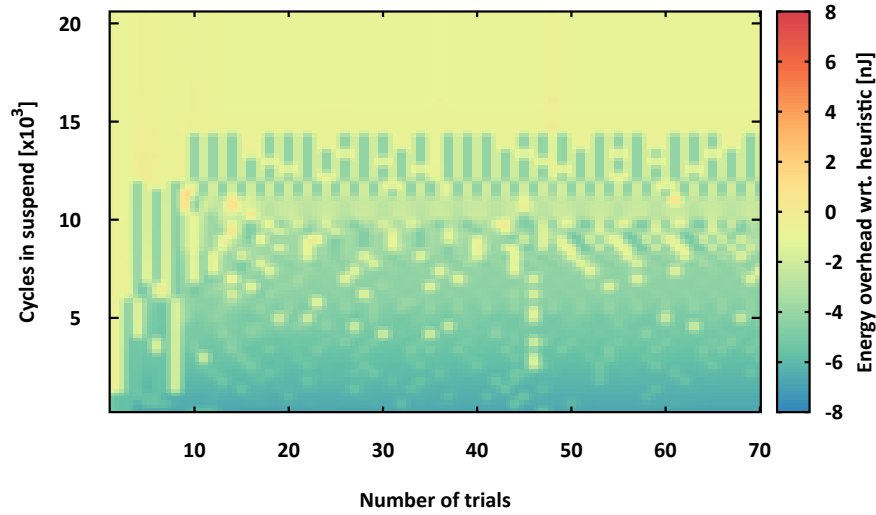


Figure 5.22: Comparison of the energy consumption gains and losses of the NZP-oriented APM compared to a deep sleep heuristic solution. Energy consumed [nJ] according to the number of trials and the number of cycles the system spent in suspend mode.

Thanks to the reduction of the states' number, better performances are obtained. Indeed, the QL algorithm scans less power modes and efficiently converges towards the optimal power modes. Hence, the learning time L_t is reduced to 5 trials with an associated learning cost L_c of 75 nJ. The energy gain E_g now reaches 12 nJ per trial offering 150% power reduction compared to the deep sleep heuristic. Since L_c decreases and E_g increases, the sustainability time S_t is closer to L_t with 6 trials needed. The exploration over-cost resulting from wrong decisions also diminishes. Hence, the algorithm lifespan \mathcal{A}_{life} is equal to T_{limit} with 13×10^3 cycles.

Q-learning Metrics	Absolute Performances
Learning time L_t	≤ 5 trials
Learning Cost L_c	75 nJ
Energy gain E_g	12 nJ/trial \rightarrow 150%
Sustainability time S_t	≥ 6 trials
Algorithm lifespan \mathcal{A}_{life}	$\leq 13 \times 10^3$ cycles

Table 5.6: Performance evaluation of the NZP-oriented APM relatively to the Deep Sleep heuristic solution.

5.3.3 Area and Power Evaluation

Reducing the complexity of the system MDP results in better performances yet, it also impacts the HW implementation. Using the available low-voltage corners in 22 nm FD-SOI – LVT, the APM V1.0 relying on the generic SoC is compared to the second NZP-oriented APM in Table 5.7.

Metrics	Power [μ W]		Cells		Area [μm^2]
	Registers	Total	Comb.	Seq.	
APM V1.0	149	270	5915	926	6539
APM V2.0	94	214	4902	623	5506
Relative variation [%]	-37%	-21%	-17%	-33%	-16%

Table 5.7: Performance comparisons of two versions of the APM. Results obtained at 0.65 V, 25 °C in 22 nm FD-SOI – LVT with a 200 MHz operating frequency.

A 37% power reduction is observed on the register power consumption as 5 states associated to 4 actions are now stored (contrary to the 8 states with 5 actions of the V1.0). However, the power saved on the final design is limited by the control logic of the design, leading to a 21% power gain on the whole module. Similarly, the number of sequential cells is reduced, corresponding mostly to the size reduction of the Q_{matrix} . The impact on the combinatory elements is less (-17%), since some cells are still required to address the Q_{matrix} and perform the QL computation. Finally, a 16% area reduction is observed.

The implementation impact of the APM is given in Table 5.8. The NZP22 V2.0 architecture is used. Results are obtained at 0.65 V, 25 °C in 22 nm FD-SOI – LVT with a 200 MHz operating frequency. The complete module integration is given in Section 5.4. Consequently, the NZP metrics include the APM module in the power, cell and area calculations. The resulting figures still reveal a significant APM impact on the final design. The extra module results in 42% of the total power, 19% of the cell utilization and 13% of the whole system area. Moreover, this first evaluation does not consider the accurate energy monitoring required and its impact on the system power consumption.

Metrics	Power [μ W]	Cells	Area [μm^2]
APM V2.0	214	5542	5506
NZP*	505	28460	43462
APM impact	42%	19%	13%

* Including the APM IP

Table 5.8: Performance evaluation of the APM relatively to the NZP22 – V2.0 SoC. Results obtained at 0.65 V, 25 °C in 22 nm FD-SOI – LVT with a 200 MHz operating frequency.

Conclusions

With this simplified architecture, better absolute performance results are obtained. However, they are mitigated by the physical implementation impact on the final design. Indeed, the whole system is still limited in term of global power consumption, leading the APM to contribute to a significant part of the total power.

Assuming these results, two options are available to reduce this impact. First, the APM could be integrated in mobile-oriented SoC where the power consumption would deal with the ~ 10 s of milliwatt range. In such design, the APM power consumption would be hidden in the total power.

Secondly, in the context of our SoC, consuming ~ 100 s of microwatts, the APM should be used periodically. During the first suspend sequences of the system, the APM is activated to perform proper learning of the activity. Then, after a time corresponding to the learning time, the power modes found by the APM are directly used by the PMU. The APM is thus disabled or power gated to save power. Periodically, the APM should be switched on again to check if the system activity did not evolve and requires to find a better power mode. By duty-cycling the APM utilization, extra power can be saved. This desired behavior results in physical implementation choices which are discussed in the next section.

Lastly, the current energy monitor required by the APM to perform the algorithm operation is not included. Similarly, this extra module should not impact the limited power budget.

5.4 APM Hardware Description and SoC Implementation

This section presents the HW implementation of the APM. Verilog code parametrization efforts have been done to allow a generic module which can be reconfigured depending on the number of desired states and actions. The APM is first described in Section 5.4.1. The SoC integration and the power optimizations performed are then given in Section 5.4.2 and 5.4.3.

5.4.1 APM Design

APM/PMU Interactions

The APM module is designed to determine the optimal power mode of the system. Its operation are closely related to the PMU of the system. A minimal interaction is defined to ensure scalability of the solution to other SoC architecture. Such model is presented in Figure 5.23.

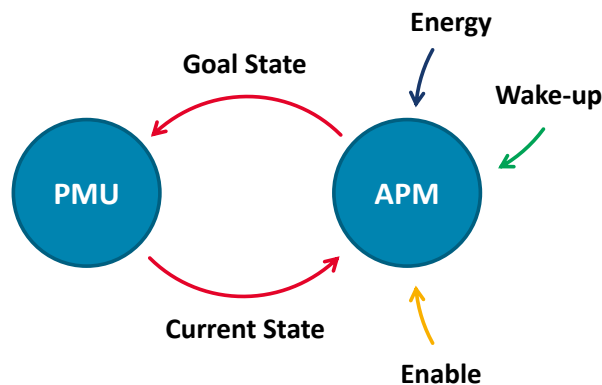


Figure 5.23: Interaction model between the APM and the PMU of the SoC.

When the APM HW is activated with the enable signal, it sends a goal state to the PMU which acknowledges it by sending its current states. The states sent by the PMU are the ones available in the MDP description. An extra state CHANGING helps the APM track the transition between mode. During a suspend sequence of the SoC, several goal states can be sent by the APM, corresponding to the exploration required to determine the optimal mode. For a given state, the energy is continuously evaluated using an external digital vector.

Then, when a wake-up signal is received by the SoC in the form of an interruption, the APM stops its computations. It requests an active mode to the PMU to resume the normal operation of the SoC.

APM FSM

The implementation of the APM is performed using a re-synchronized Mealy state machine. The corresponding state diagram is given in Figure 5.24.

Assuming a proper reset of the state to start in IDLE mode, the APM is activated by receiving an ENABLE signal. Then, the module sends a new goal state to the PMU, and goes into COUNTING_i or WAITING_i depending on the selected goal and PMU status.

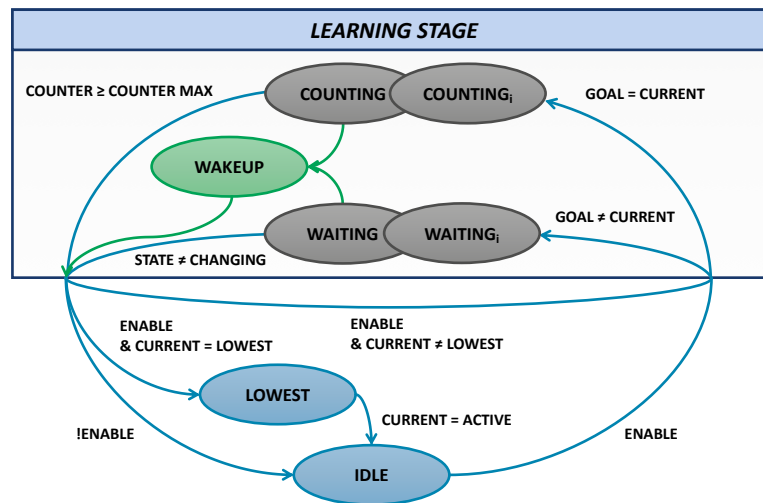


Figure 5.24: FSM of the APM HW module.

After one cycle, the intermediary states “ i ” are directly updated. This solution allows to determine the beginning of a new goal and trigger an update of the previous learning coefficient and starting a new calculation of the energy penalty. When the goal is reached, the APM either returns to the IDLE state or defines a new goal state depending on the ENABLE signal.

The APM shows two main behaviors with this implementation. In WAITING, the state goal is different than the current power state. The system waits for the complete execution of the current mode transition. In COUNTING, the goal state imposes to stay in the same power state. The APM counts until it reaches a threshold value to search for a new action. This solution avoids excessive mode transition through a programmable counter value C_{loop} .

Figure 5.25 details the beginning of a suspend sequence. ENABLE is set to 1 then the APM sends a new goal state. At the beginning of each new goal state, a NEW_GOAL signal is generated to determine the beginning of a learning session and store the current value of the energy. Similarly, the NEW_REWARD signal indicates when the mode transition is over and the resulting penalty can be calculated. When NEW_REWARD is set to one, it activates a secondary reward module which computes the subtraction of the current energy with the one obtained at the previous NEW_GOAL signal assertion.

Figure 5.26 shows the end of a suspend sequence. When the WAKEUP signal is triggered, the APM goes into the WAKEUP state and waits until a complete wake up of the system. Then, it performs a final learning to compute the wake-up penalty of the last selected mode.

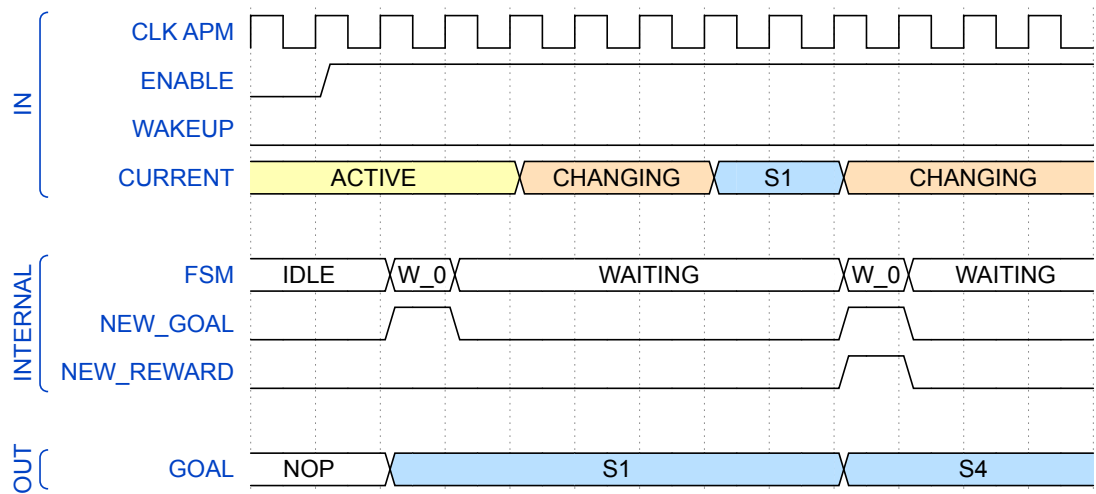


Figure 5.25: Suspend sequence timing diagram of the APM – Beginning.

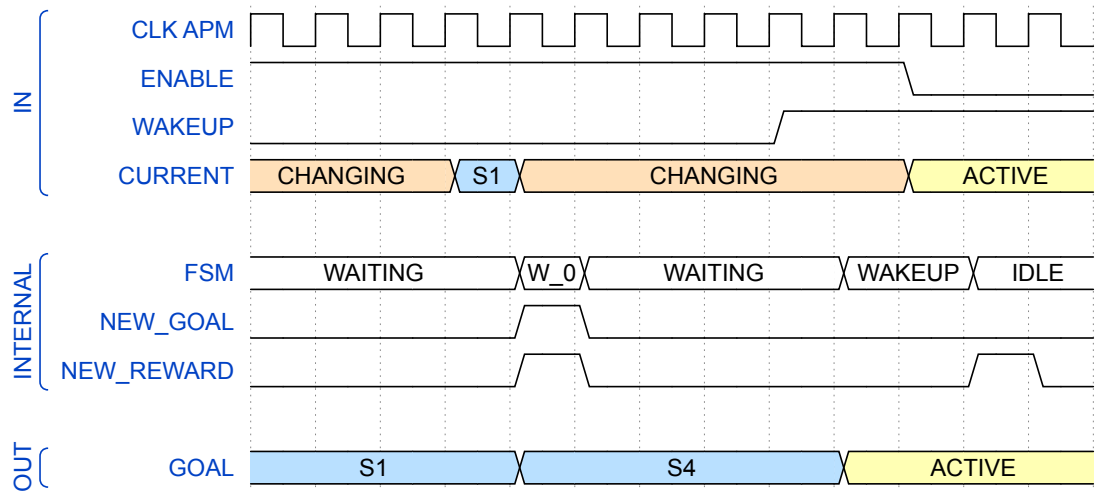


Figure 5.26: Suspend sequence timing diagram of the APM – End.

APM block diagram

Figure 5.27 shows the internal structure of the APM module. The Q_{matrix} block contains the array of coefficients. The reward block derives the penalty using the sum of three multiplications according to the formula presented in (5.11) of Section 5.2.4. Two policies are defined for the path selection. The selection of the path with the minimum Q coefficient (see Section 5.1.6) and the exploration policy as previously described in Section 5.2.4. At last, a control block FSM manages the sequencing of the APM tasks, whereas a power state module interfaces with the PMU.

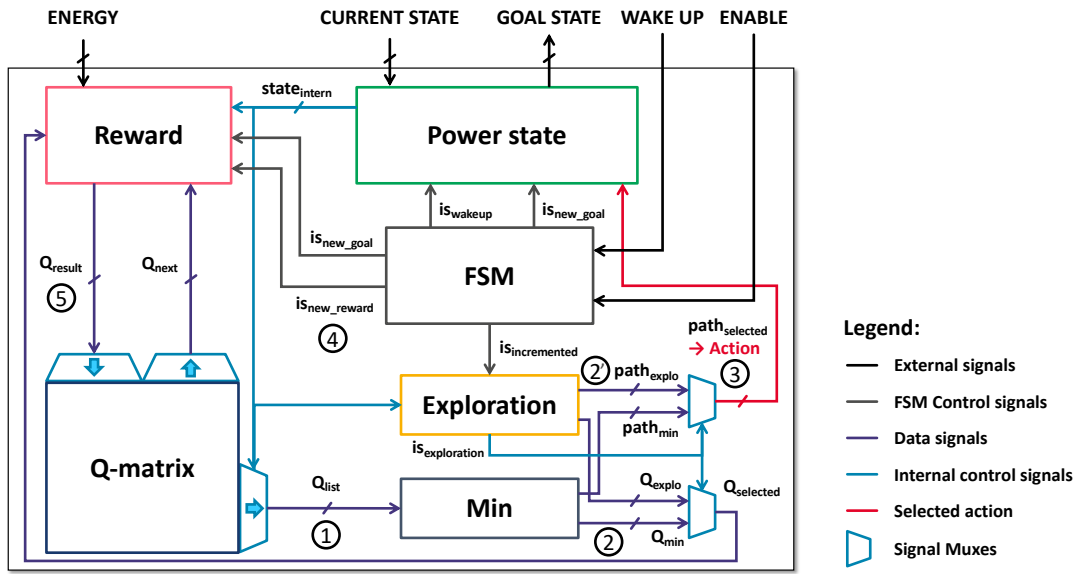


Figure 5.27: Block diagram of APM HW module.

In standard operation phases, the current state is used to select the list of possible coefficients from the current state (Q_{list}) ①. A minimum function is executed on this list and the best action is extracted ②. During the exploration phase, the selected action from the exploration block is chosen and sent as a goal ②'. In both cases, this action is finally transformed into a goal state according to the MDP diagram ③.

Each time a state transition of the MDP diagram is terminated, a penalty computation request is sent to the reward block ④. The new coefficient is stored in the Q_{matrix} as soon as the computation has finished ⑤.

Q-matrix definition and future optimizations

To allow a system update according to the SoC power modes and states, the whole APM implementation uses Verilog parameters and constant definitions. In particular, depending on the states and actions available in the system MDP, the number of coefficients of the Q_{matrix} or the number of possible paths is updated.

```

1 generate
2     genvar var_st, var_path;
3     for (var_st=0; var_st<`NB_ST; var_st = var_st+1)
4         begin : q_matrix_st_loop
5             for (var_path=0; var_path<`NB_PATH; var_path = var_path+1)
6                 begin : q_matrix_path_loop
7                     if (PATHS[`GET_ITEM(`BIT_ST, var_path, `NB_PATH, var_st)] == `ST_NOP)
8                         begin : q_matrix_dummy
9                             // when available paths, we disable the Q coefficient
10                            assign q_matrix_output[var_st][var_path] = {`BIT_Q{1'b1}};
11                        end
12                    else
13                        begin : q_matrix_coefficients
14                            always @(posedge clk_apm or negedge RESETn)
15                                if (!RESETn)
16                                    q_matrix[var_st][var_path] <= 0;
17                                else if (reward_st[`NB_CYCLES_REWARD-1] == var_st
18                                    & reward_path[`NB_CYCLES_REWARD-1] == var_path
19                                    & prod_sum_en)
20                                    q_matrix[var_st][var_path] <= reward_result;
21                            assign q_matrix_output[var_st][var_path]
22                                = q_matrix[var_st][var_path];
23                        end
24                    end
25                end
26            endgenerate

```

Figure 5.28: Verilog code description to generate the Q_{matrix} coefficients.

As shown in Figure 5.28, all the coefficients are defined within a bi-dimensional array of paths. A path corresponds to the association of an action and two states⁶. For each possible path from each state, the code executes one of the following assertions:

- The path does not exist (i.e. `PATHS == 'ST_NOP`): no register is created. The corresponding output wires of the Q_{matrix} `q_matrix_output` are set to 1. It results to set the maximum value for this coefficient. Indeed, a value is still mandatory to execute the minimum function required by the QL. By using the worst value by default on this empty output of the matrix, every non-existing path is not selected. Moreover, it highly limits the number of registers instantiated.
- When the path exists: a register is instantiated and connected to `q_matrix_output`. The values stored in these registers are updated using a reset of when the QL computations are over. A clock gating technique is also used on these registers to avoid extra dynamic power consumption between data updates.

This solution is highly efficient in terms of implementation and configuration of the APM HW, however it increases the power consumption. Thus, optimizations on the implementation would enable the utilization of a small memory macro to store the Q_{matrix} coefficients. In fact, since these data are accessed and stored only when a new state goal has been reached, they are not timing critical.

A dual-port memory also facilitates the access to the Q_{matrix} coefficients. On the one side, it helps to analyze the algorithm execution by accessing to the stored Q-value during run

⁶This information are given in an external configuration file.

time. This information can be sent to the Python model live to check the correct operation of APM state selection. Moreover, assuming an embedded sensor that sends an energy information to the APM, the analysis of the Q_{matrix} coefficients retraces the energy consumed in each mode. Then, these data can be reused in the Python description to improve the QL model.

On the other side, instead of using 0 values at the initialization of the system, pre-determined values could be stored in the Q_{matrix} to improve convergence of the algorithm depending on the system activity. The Python model could be either used to get these values or a direct evaluation can be performed on running APM implementation. This concept is similar to the training phase of NNs. In the end, for a given SoC architecture embedding the APM, the user can roughly determine the expected activity of its system, set the proper Q_{matrix} coefficients and improve the convergence and energy gain. This information could even be given by the SoC data sheet next to the mode power and transition time. From this improved starting point, the QL algorithm would adapt to the system environment variation and power consumption.

5.4.2 Integration into an ARM Cortex-M0+ Based Architecture

In order to integrate the APM into the core, the whole SoC organization has to be reworked. The system integration of PMU, WUC and APM modules is presented in Figure 5.29. The APM performs the optimal mode learning. The PMU manages the correct state selection in accordance to the core status resulting from the application. Lastly, the WUC handles the peripherals and other integrated IPs. As a whole, these 3 modules constitute the HPM of the SoC.

PMU

The ARM Cortex-M0+ only accepts two low power modes as stated in Appendix A. Therefore, extra modes are added to the PMU to handle all the power states. Now, the sleep and deep sleep signal from the core must both be asserted to determine a suspend sequence. The PMU then handles the state selection. The power mode can be determined by the software like the previous architecture. The software-oriented timing sequence is given in Section 4.3.1. When APM-oriented power management is activated, the PMU follows the APM decisions. Contrary to the previous implementation of Chapter 4, the PMU is modified to handle transitions between suspend states before going back to active (e.g., Active → Sleep → Deep Sleep → Active). To wake the system up, a WAKEUP signal is asserted by the WIC, transferred by the A.ON part of the core. Therefore, no modification is necessary to handle several types of interruptions and masking.

APM

The APM is connected to the PMU to determine the states to be selected for a given suspend sequence. It receives its energy information from the top of the design, using an external measurement or with an embedded power meter (currently not available). Extra signals ERROR and IS_RECORDING are added for debug purposes. The APM can also be disabled and turned off. This ensures extra power reduction when the APM is not necessary. Moreover, it helps force sleep or deep sleep heuristic for evaluation purposes.

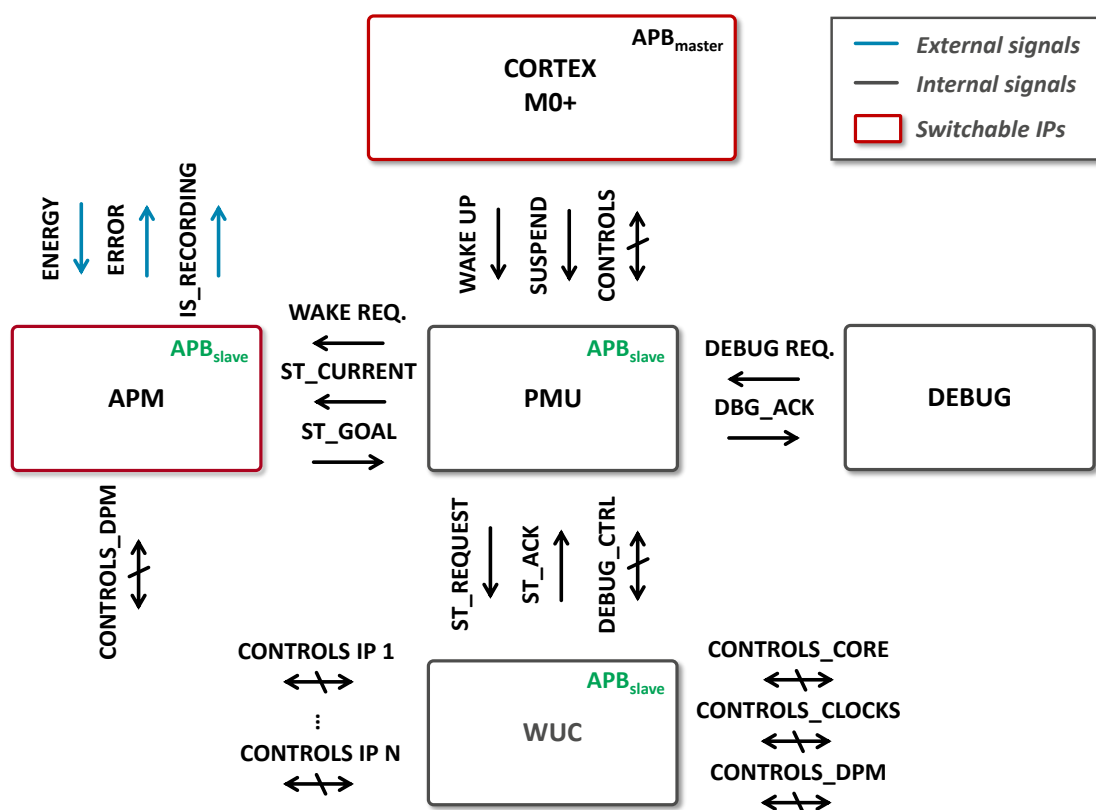


Figure 5.29: System integration of PMU, WUC and APM modules.

WUC

The WUC handles the IPs power management and control signals generation. Using a programmable FSM, an arbitrary number of peripherals can be added, controlled and associated to power states. The WUC also generates the control for the clock gating, power gating and clock switching that occurs on the ARM core. The WUC receives a state request from the PMU and acknowledge the state when the associated IPs have switched to the corresponding mode. Extra configuration registers are available to bypass the frequency switching acknowledgment currently operated off-chip. In such manner, only the transition time resulting from the embedded IPs is considered.

APB integration

All these IPs are treated as APB slave peripherals. The register mapping to configure the PMU, WUC and APM modules is given in Appendix F along with a C-code header file `hpm.h`. For instance, the last goal state defined by the learning algorithm is stored in a PMU register and available by the core when the APM is turned off. It allows to perform learning for a given amount of time then disables the APM. Moreover, the software control of the PMU or the enabling of the ultra deep mode are configurable via C-code.

Debug

The debug module has the priority on other modules. If activated, the whole system is woken-up by sending wake-up requests to all sub-systems and IPs, through the PMU and WUC.

Summary

Suspend sequences are initialized with a SUSPEND request from the core, and end with a WAKEUP signal. During a suspend sequence, the PMU communicates with the APM in order to execute the target state and update the QL functions. Each state change is acknowledged by the PMU when the WUC has finished the IPs configuration. This last validation depends on the SoC IPs transition time from one state to a new one.

5.4.3 SoC physical implementation

As previously stated in Section 4.5.1, the previous modules are integrated in a complete LVT 22 nm FD-SOI SoC. The block diagram of the system is recalled in Figure 5.30.

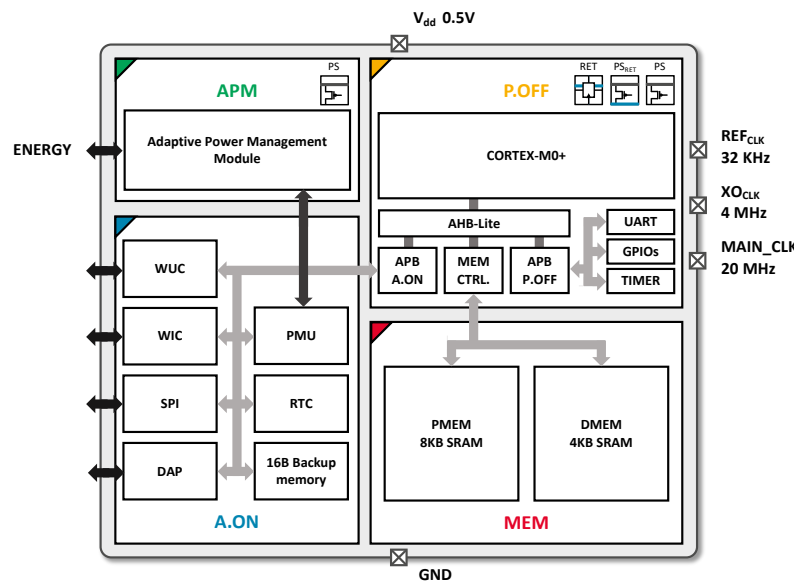


Figure 5.30: Block diagram of the NZP22 – V2.0 SoC integrating the PMU, WUC and APM modules.

The PMU and the WUC are integrated in the A.ON whereas the APM is placed in a separated switchable power domain. Dedicated power IOs are used for power measurements on P.OFF, MEM and A.ON. Hence, the APM power grid is shared with the A.ON. The impact of the module on the whole power consumption is mitigated thanks to power gating on the APM. However, no retention is available. Turning off the APM results in loosing of the Q_{matrix} information. The control of APM power mode and top power switches is determined by programming internal register to the WUC.

5.5 Measured Performances

This section gives the performances obtained from measurements on the APM module implemented in NZP22 – V2.0. The functional validation methodology and the tests performed are first described in Section 5.5.1.

5.5.1 Functional Validation

Figure 5.32 provides a simplified flow chart describing the functional validation of the APM module. Starting from a complete reset of the SoC, the core program memory is loaded with a C-code test program using the DAP. After initialization, depending on the GPIO inputs, two execution paths are available.

Software-Oriented Power Management

In this mode, the selection of the ULP mode is done using a determined software configuration. This type of management is used as a reference. It allows easy selection of a target mode and power measurements along with validation of the overall behavior of the system.

To be activated, the PMU and WUC internal registers are configured according to a targeted power mode (see Appendix F). Even though it is not mandatory, the APM module can be disabled for power reduction. Then, from the Active state of the core, a suspend request is asserted (see Section 4.3.1). In a standard operation, an acknowledgement signal is expected by the SoC to confirm the mode transition of external peripherals (e.g., clock generator). If only internal peripherals are used, this signal can be bypassed. When a wake-up signal is relayed by the WIC to the PMU, the system resumes its operation⁷. Again, acknowledgement signals are expected for external peripheral synchronizations or emulations.

Adaptive Power Management

This second mode validates the execution of the adaptive power management to determine the optimal low power mode. The APM is activated using the PMU and WUC internal registers while the software-oriented configuration is disabled.

Starting from the Active mode, a suspend sequence is initialized by the core. The PMU request a goal state to the APM module. Optional acknowledgement signals are requested by the WUC to reach the target suspend mode. If no wake-up signal is raised, the SoC can switch between modes according to the number of cycles spent in suspend. When a wake-up signal arrives, the energy consumed is read and stored by the APM module, then the SoC resumes to the active mode.

To validate the learning capability of the system, a software test is used. The test program loops between Active and Suspend modes until a counter value is reached, thus triggering a watchdog. Hence, depending on the system activity, the APM will converge towards a given suspend mode. At the end of each sequence, the last APM goal state is stored into the memory. Therefore, the whole behavior of the SoC can be replayed into the Python model. Combined

⁷The ultra deep mode requires context restorations steps (see Section 4.3.1) which are not represented on this graph.

with the power measurements obtained using the software-oriented management, the benefit of the APM is estimated over multiple trials.

Simulated Behavior

The previous behavior of the APM has been evaluated using power aware simulations. In Figure 5.31 is presented a subset of the top-level timing diagram of NZP22 – V2.0. After initialization, the software-oriented mode is tested. While power modes are requested to the PMU by the core, the APM is frozen. In the second part, adaptive power management is operated. The APM defines a goal state which is followed by the PMU. Suspend sequences are initiated using the sleeping signal asserted by the core whereas the wake-up is done using the NMI signal.

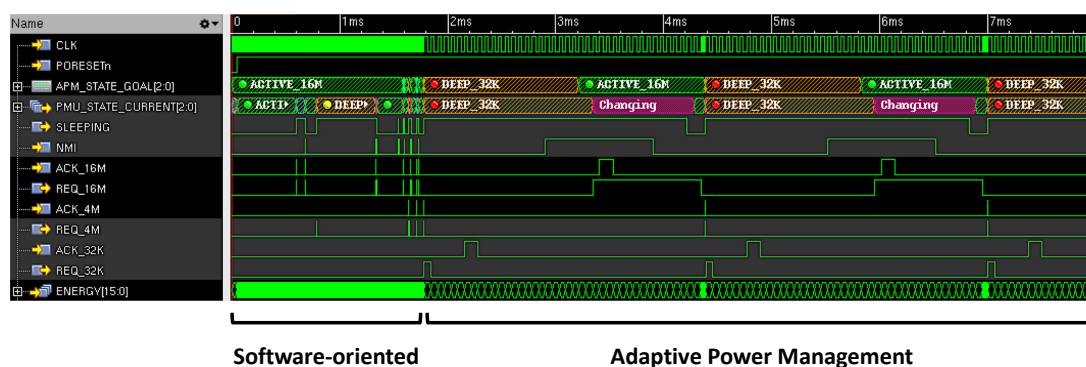


Figure 5.31: Simulated behavior of the APM module at the NZP22 – V2.0 SoC level.

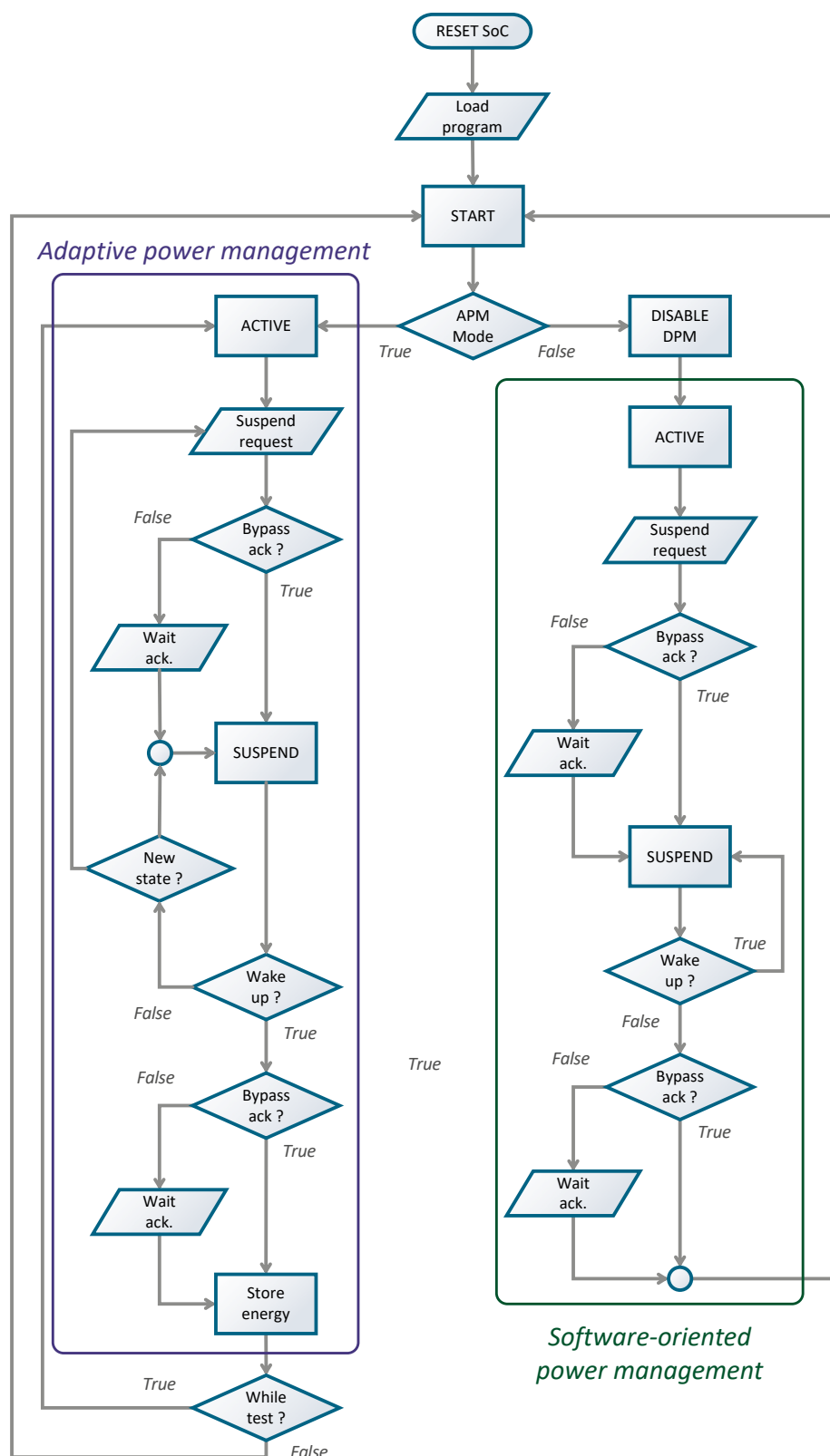


Figure 5.32: Simplified flow chart describing the functional validation of the APM module from the SoC perspective.

5.6 Summary and Perspectives

This chapter introduces a new model of Adaptive Power Management (APM) based on a Q-Learning (QL) algorithm to optimize suspend sequences during inactivity. The APM module offers the SoC the capability to select the most efficient ULP mode depending on its activity, its energy consumption and the external environment.

First, the general theory of the RL algorithm is defined on a generic SoC architecture. This preliminary analysis enables the proper definition of the algorithm parameters. Moreover, it resulted in the selection of the QL algorithm as a trial-and-error implementation to determine the optimal power mode. Simulations made with those parameters show a 17% energy saving for our system (compared to a heuristic solution). However, a learning time of 10 trials is required, which corresponds to a 87 nJ learning cost. This preliminary work led to an ISCAS 2018 publication. The complete reference of the paper can be found in Appendix G.

The APM hardware design was synthesized in 28 nm FD-SOI technology. Considering a NZP implementation, the APM impacts the SoC by adding 250 μ W extra power consumption at 0.7 V. Therefore, specific hardware optimizations were identified to improve the module energy consumption, such as removing coefficients in the number of available paths in the system MDP and reducing the size of variables while ensuring the precision required by the implementation.

Consequently, the APM HW is adapted to the architecture of Chapter 2. Thanks to the reduction of the states number, better performances are obtained. Less than 5 trials are required for the learning, resulting to 75 nJ of learning cost. Moreover, compared to a deep sleep heuristic, a 150 % gain is obtained. The APM impact is estimated to 42% of the SoC architecture and 13% of the area. Thus the good results shown by the technique are mitigated for tiny ULP SoCs. However, it paves the way towards efficient power management using machine learning techniques.

The whole system is simulated and evaluated using an in-house co-simulation environment based on Python and Verilog. Moreover, using scalable code techniques, the APM can be adapted to several SoC architectures. The physical implementation of the HW also involves the ULP techniques presented in the previous chapter to ensure a minimum power-overhead due to the module integration in the NZP SoC. Moreover, by using a memory macro as storing element of the Q_{matrix} , improvements are expected in terms of power consumption and algorithm pre-configuration.

A complete SoC demonstrator embedding the APM is currently under fabrication. Future work will focus on measuring the module performance in-situ, including the impact of APM on the overall power consumption. Moreover, an interesting benefit of the APM will be to dynamically adapt to PVT changes that will change the optimal point. Extra measurements are then expected to characterize such behavior.

This work is the first implementing QL into a complete SoC demonstrator for optimal power mode selection during suspend sequences.

Conclusion

THIS thesis addresses a problem that goes beyond solutions dedicated to present IoT connected objects. Hence, to enable sustainable fully autonomous connected devices with lifetime scale operations, the actual power consumption paradigms are moving towards Near-Zero-Power SoCs.

For applications relying on energy harvesting or small batteries, utmost energy efficiency is indeed a key metric. Unlike standard IoT devices which offer low and high-performance services, by taking advantage of a macroscopic energy reserve offered by their battery, NZP systems perform very specific operations at low energy cost while consuming little or no energy when they are in suspend modes. Consequently, this work has placed a particular emphasis on pushing the mass market microcontrollers down to the micro watt power consumption by using the FD-SOI technology combined with the latest industrial workflows.

The Near-Zero-Power (NZP) footprint will not only enable low activity applications, such as wireless sensor nodes processing physical and chemical environmental variations or monitoring people health and medical conditions, but also swarm robotics and remote applications. For such systems, the small on-board energy source, often associated to energy harvesters shows minimum dimensions and a reduced average power supply. This energy limitation requires overlapping innovation in the design of conventional ICs at all levels – device, circuits, gates/modules, system and architecture – in order to provide a relevant service while guaranteeing the energy autonomy of the circuit.

From the CMOS industry catalog of standard cells and technology features, to the utilization of Machine Learning techniques, through optimized low power IPs and system partition, the feasibility of ULP SoCs using exclusively the latest industrial components is demonstrated. This chapter reconsiders the previous chapters conclusions to highlight the challenges overcome throughout this thesis.

Achievements and Main Contributions

Advanced technology nodes are highly effective but show higher leakage currents. Hence, careful implementation trade-offs are compulsory to leverage the speed without impacting the power budget. Moreover, body biasing is a promising tuning knob to deal between the transistor's performances. Similarly, the application of Ultra-Low-Voltage (ULV) operations has proven to be highly effective to reach the power targets. However, it results in implementation constraints that must be taken into consideration.

Chapter 2 – System Definition This first chapter proposes an ULP SoC functional description resulting from the analysis of IoT devices energy flow, architectures and applications use cases. This general representation has been used along this work to demonstrate the technology, component and system optimizations while following the goal of a coherent system partition for mode switching depending on the activity and energy availability.

Chapter 3 – Clock Reference for Ultra-Low-Power Systems There, has been presented fully-integrated 28 nm FD-SOI IP designed for clock generation and combined with digital compensation unit for PVT compensations. The whole system is operating at 0.5 V and generates an accurate 32.768 kHz clock reference with 90 ppm/V for $V_{dd} \pm 8\%$ and 1.9 ppm/°C from 0 to 50 °C, while consuming 15 nW for the oscillator and 125 nW for the digital. By demonstrating an efficient low-cost clock, this solution offers a versatile time reference in suspend modes and clock source for standard digital systems.

Chapter 4 – ULP System Implementation The covered implementations of this chapter differ from the previous ULV and ULP publication by relying on simple, yet efficient, cost effective and robust techniques⁸, in order to facilitate adoption by the industry sector. ULP techniques and new innovative concepts were explored from technology to system level. First, extensive bench-marking, based on qFO4 and synthesis loops combined with tuning of the implementation flow, reduces the hardware and design margins energy imprint in 22 nm and 28 nm FD-SOI. This analysis, combined with silicon results, also provides a comparison of both technology nodes performances. Then, ULP techniques are explored and implemented for static and dynamic power reductions. Thanks to a careful selection of the design components (power gates, double-gated retention registers, low power IPs) and low-power techniques (clock gating, dynamic power management, frequency switching), a global optimization of the system architecture in FD-SOI is obtained. Consequently, relying on the FD-SOI intrinsic capabilities and adequate utilization of the technology features, SWBB enables the adoption of body-bias for energy/leakage/temperature trade-offs in low-power designs, without the power overhead of extra generators. Lastly, coupled with a learning of the different power modes offered by the Cortex-M0+ for power mode selection, versatile and efficient power modes implementations are implemented. It results in an ultra efficient SoCs with active power consumption fitting a ~100s μ W power budget.

⁸In that sense, it refers to the KISS design principle developed by Kelly Johnson, lead engineer at the Lockheed Advanced Development Programs (Skunk Works), in the 1960s [243].

Chapter 5 – Adaptive Power Management From the previous chapter considerations and targeted architectures, a functional APM module is designed. It uses ML techniques to add intelligence to generic MCUs and optimize suspend sequences during inactivity. Indeed, the APM allows the selection of the most efficient ULP mode depending on its activity, its energy consumption and the external environment. The APM HW is adapted to the architecture of Chapter 2. Less than 5 trials are required for the learning, resulting in 75 nJ of learning cost (relatively to a deep sleep heuristic). So, 150 % of energy saving is obtained. The whole system is simulated and evaluated using an in-house co-simulation environment based on Python and Verilog. The physical implementation of the HW also involves a large panel of power-mitigation techniques presented to ensure a minimum power-overhead due to the module integration. As the first implementing QL into a complete SoC demonstrator, for optimal power mode selection during suspend sequences, this works paves the way towards efficient power management using machine learning techniques.

Summary SoCs with a NZP footprint are demonstrated while enabling a power consumption in accordance with the surrounding energy that can be collected. The FD-SOI technology and ULV limitations have been overcome through innovative designs and low power techniques from the device up to the system level to preserve the operation efficiency in any situation. A corresponding summary of the main contributions of this work is proposed in Figure 5.33. Additionally, the insights presented in this thesis were presented at a selection of international conferences and published in full in conference proceedings and journals. The complete list of publications and patents is given in Appendix G. A testchip gallery is also available (see Table G.1) which summarizes the design and measurements resulting from this work. It covers 5 different test chips realized from July 2016 to July 2019 in 28 nm and 22 nm FD-SOI.

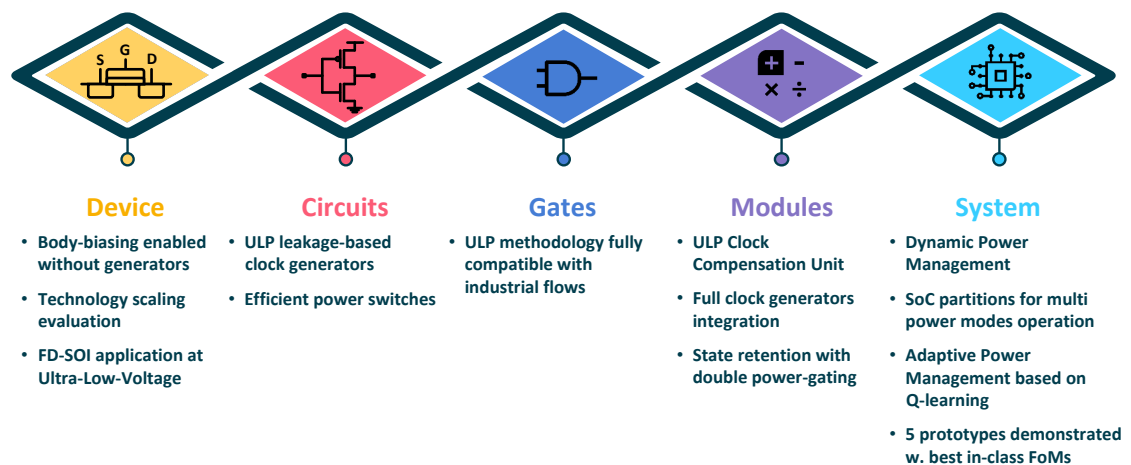


Figure 5.33: List of the main contributions from device up to system level.

Perspectives

Extension of low-power techniques to the memory IPs

While many researchers are focusing on the identified power-hungry components such as the radio or the sensor interfaces, this work focuses on the application of low-power techniques to reduce the power consumption of the digital core and processing elements.

The goal is to maintain a proper power-consumption balance between the main components of an IoT-oriented SoC. Then, the sub-components, such as the memory storage elements, are of particular interest to go further in power reduction and energy efficiency. Indeed, for large systems, the memory arrays consumed a great part of the energy available [244]. In advance technology nodes, it is worsened due to the leakage currents.

Most industrial companies tend to minimize the memory area to decrease the manufacturing cost, yet it comes at the price of extra power consumption. This power/area trade-off is also extended to the time required to write/read some data and the retention time offered by the macro for system switching between active and suspend modes.

Then, a thorough analysis and review of the memory solution available (SRAM, NVM, retention register,...) and their impact on the power budget is necessary to ensure proper power modes designing and the utilization of the optimal storage solution for correct spatial and temporal data handling.

Extension of the APM module utilization

The simulated behavior of the APM show benefit for the optimal suspend power mode. Using some refinements on the utilization and application of the QL algorithm, an extension to the active modes of the SoC is an interesting option to be explored.

Moreover, the application of the APM proposed in this work is easily justified for complex SoCs (like automotive applications), since the physical implementation of the module appear negligible relatively to the system size.

For smaller systems, the APM might be detrimental due to the power consumption of the implementation. While techniques to reduce the power impact of the block have been discussed in this work, other options should be explored.

First, in order to decrease the cost of the APM implementation, the QL application could be simplified by decreasing the number of available modes and the evaluation of their power consumption. Moreover, a periodic activity of the APM combined with power gating is a promising solution to cut-off the power consumption of the module when an optimal mode has been determined. Even though such utilization is currently available in NZP22 – V2.0, it requires extended qualification to determine the trade-off between the APM duty cycle and the resulting power savings.

Moreover, the APM module can be identified as a characterization module for power modes utilization and debug. Instead of requiring software benchmarking and accurate models of the SoC, the APM module could be activated during the application prototyping. Hence, assuming a given use case, it would give the user an idea of the optimal power mode. To validate such utilization and the benefit of the APM module, a comparison relatively to a standard power-aware code execution is required.

Lastly, besides incurring modification of the power consumption in each modes, the PVTs variations also result in modification of the transition time between mode (especially for closed loop module sub-systems, e.g., PLLs). Hence, the PVT sensitivity of the APM is a challenging assignment for accurate estimation of the APM power saving compared to software decision.

Towards a fully autonomous WSN

This thesis has advanced the field of ultra-low power digital systems by demonstrating simple, industrially compatible, and technology agnostic solutions and methods for efficient ULP SoCs and by developing prototypes pursuing the NZP paradigms. Nevertheless, some parts towards the development of a complete autonomous demonstrator remain untouched, although previously considered in the system analysis proposed. Improvements and research directions are left on several sub-systems.

1. Always-On Ultra-Low-Power Radios ULP radios are required to transmit data and information to the external world. Current implementations still present consumption in the mW range. Effort should be placed onto always-on radio able to sense valid information and wake-up only when necessary. Recent papers have demonstrated ultra-low-standby-power radio transmitters designed for applications with extreme energy storage and/or energy harvesting constraints [245]. Similarly to this thesis work, aggressive power gating techniques are combined within a low-complexity architecture. A 39.7 pW standby power consumption is achieved while a 38 pJ/bit is obtained to transmit data at a 5 Mbit/s rate. For maximum power reduction, a duty-cycling utilization with 5 bit/s is available leading to a consumed average power of 78 pW, thus fitting the NZP paradigm. Such radio would be complementary to the SoCs of this thesis. However, it would require technology compatibility or re-design.

2. Always-On Ultra-Low-Power Sensors Current sensors are always powered. Power efficiency can be obtained by “intelligent sensing” that detects variations in the environment and wake-up only when interesting information are observed. Such NZP sensor node solutions are presented in [246] based on piezoelectric sensors and tunable electrostatic switches. This sensor allows measurements of the acceleration, the rotation, and the magnetic field for the detection of a desired signal pattern and the generation of a wake-up trigger.

3. Integrated Energy Harvester To keep up with the ULP system architecture presented in Section 2.2, efforts should be pursued to integrate an energy harvester and the necessary supply sources [247]. Accordingly, the power mode of the system would be extended. It also requires the design of ULV and ULP power converters able to deliver the best performances depending on the harvester conditions and the system modes. Design efforts are also required on self-adapting PMU to automatically adapt to different battery voltages for down-conversion and different harvesting sources and conditions for up-conversion. A solution is proposed in [248], where the adaptation method enables harvesting with solar, microbial fuel cell, and thermal energy sources and increases harvesting efficiency by $\times 1.92$ while achieving a peak extraction efficiency of 99.8% for solar cell.

4. On-chip power and temperature sensors To provide tools for SoC-level power management, the design of individual low-power IPs such as the clock generators must be extended to power and temperature sensors. In fact, a temperature and voltage sensor can be combined to the digital compensation unit CLU of the low-power oscillator to provide a second feedback loop of compensation. Then, extra power saving can be observed by switching off the calibration when no PVT variation is detected. Moreover, an energy sensor is required by the APM to provide information on the power modes and their consumption during the suspend mode. Such sensor must be evaluated to accurately monitor the energy while ensuring a minimum impact on the system power consumption. Lastly, by introducing PVT sensors, a wide range of dynamic compensation techniques can be included into the SoC to target energy-efficient system with variation-resilience capabilities [103, 110].

5. Hardware/Software Co-optimizations Those co-optimizations pave the way towards Data Aware Power Management [249]. In fact, with a careful data control during the execution of the application, a fragmentation of the memory could be performed. During suspend sequences, power can be obtained by turning off the unused memory space. Whilst attractive, this concept opens a wide range of limitation, due to the non-predictability of the software execution.

Appendix A

ARM Cortex-M0+ Presentation

IN Chapter 1, a relatively general architecture for the IoT-oriented applications is given. Similarly, this appendix gives details on micro-processor architectures while focusing on the ARM Cortex-M0+ core selected for this work.

A.1 Reminders on Microprocessor Architectures

The last decade has seen an exponential growth of IoT-oriented application (see Chapter 1). These type of systems rely on data acquisition, control loops and, processing which are realized at the by the Microprocessor Unit. A typical one-core microprocessor architecture is presented in Figure A.1.

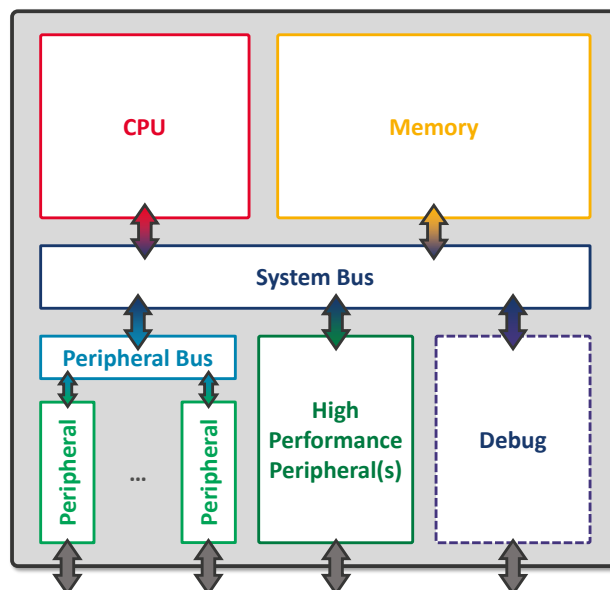


Figure A.1: Typical Microprocessor Unit architecture including its subsystems.

Central Processing Unit

The processor or Central Processing Unit executes the machine instructions of the computer programs (also known as code) that are defined by the software. As the name suggests, the CPU processes the information of the program or data to handle a specific task. Inside, an Arithmetic Logic Unit handles basic mathematics operations whereas a Control Logic Unit is responsible for the correct sequencing of operations. From software and machine language perspectives, the functional specifications of the processor are defined by the ISA. Possible implementations might differ, such as the Reduced Instruction Set Computer (RISC) or Reduced Instruction Set Computer (CISC), yet they all contain a set of available instructions, accessible registers as well as some architectural organisation information. The ISA must be distinguished from the microarchitecture itself, which is a map of the CPU at logic level [250].

Memory

The pieces of information which defines the calculation to be made on the data by the CPU is stored into a memory. In the Von Neumann architecture [251], the program instructions and the data are stored in a unique place. A solution, where data and instructions are split is called a Harvard architecture [250]. The memory is generally divided into a volatile element for instruction and data required during operation, and a permanent part for master instructions and data.

System Bus

The communication and connections between elements are done through a modular system bus. This major component is subdivided between a data bus which carries the information, an address bus to determine to which element the information must be sent and a control bus to manage the overall transfer flow. Nowadays, several widely used buses are regrouped under the open standard Advanced Microcontroller Bus Architecture (AMBA) [252]. Their utilization mostly depends on the selected CPU implementation.

Peripherals

Even though these elements could be considered as the minimum requirement for a Microprocessor Unit (MPU), without interfaces to the external world, the implementation would not be realistic. Hence, peripherals are added to communicate with external systems. The connectivity can be done using standard peripherals such as, I2C, SPI, UART, GPIOs, which are connected to the system bus by a peripheral bus, which adds some modularity and reduces system bus complexity at the price of a reduced bandwidth. Some high-performance peripherals might require a direct access to the core, the memory, or need a reduced latency. Hence, they are directly connected to the system bus.

Debug

Debugging system(s) are also added to check the correct execution of the code. They can be used during the application prototyping allowing on-chip instrumentation and then removed

in a final product to avoid security issues and attacks [253]. On the contrary, they can be maintained allowing on-run debug or self-checks for increased robustness. The most common standard for on-chip instrumentation is the Joint Test Action Group (JTAG) which defines a series of rules, protocol and physical port for unified debug features [254].

General Considerations

The binary representation of the system bus adopted for a specific implementation determines the bit-name architecture. For instance, 32-bits architecture use 32 bits size length instructions or data words to perform operations. The selected representation directly affects the CPU complexity – requiring more wires or systems to perform mathematics operations – but more importantly, the memory representation. Indeed, in a 32-bit architecture, a 32 bits binary word is used to address a memory element. Therefore, a 4 GB (2^{32}) memory can be addressed which limits for instance the available memory size directly accessible. Nowadays, several architectures are available, from the standardize 8-bit, 16-bit, 32-bit, 64-bit architectures to more exotic implementation such as the simple 1-bit architecture very useful for educational purpose [255] or up to the 512-bit architecture adopted in some Graphics Processing Units (GPUs) [256].

A.2 ARM Cortex-M0+ Presentation

Core Selection

For this work, the Microprocessor Unit (MPU) selection has been done by reviewing several low power cores¹ area-optimized intended for deeply embedded application. The selection was made on the ARM Cortex-M0+ and its older counterpart, the Cortex-M0. Both systems are designed – and advertised – with a very low gate count and high energy efficiency for ULP applications [161, 257].

Moreover, these MPUs differentiate themselves by their 32-bit architecture with an area and power similar to many popular 8 or 16-bit microprocessors. By allowing similar computational tasks to be carried out in shorter time – less instructions are needed for the same mathematical operations on data – this increases the energy-efficiency of the system. The whole system can stay in suspend modes for longer period of time. Alternately, to minimize power, the MPU can run at a slower clock frequency to perform the same required task [158].

The 32-bit architecture also appears to be very attractive over 8 or 16-bit thanks to the dynamic of the binary representation used². Lastly, the 32-bit combined with the ARM efficient instruction set results in high-code density [258]. The program size is decreased and the memory required to store is thus reduced. Consequently, the power consumption and costs of the implementation.

The Cortex-M0+ was selected over its older version, thanks to the internal STM availability. However, the Cortex-M0 has also shown great results in the literature (see Table 1.5 and

¹An open-source Reduced Instruction Set Computer five (RISC-V) solution has also been explored, yet the limited availability of ULP cores and support has discarded its selection.

²In 16-bit architectures, data vary between 0 and 65536 for an unsigned word and between -32768 to 32767 for a signed word. For sensor nodes, it narrows down the capture of some physical quantities and the addressable memory space.

Table 1.6). In fact, the *Lite* of the Cortex-M0 is also available for academia through the Design-Start University Program [259] to realize fast silicon prototyping. At the time of this work the new Cortex-M23 was not available [260]. This last core includes the aforementioned features yet it relies on the ARMv8-M ISA, facilitating, among other things, multi-core implementations.

Core Features

The Cortex-M0+ processor is a configurable, multistage, 32-bit RISC processor (see Figure A.2) with a Von Neumann bus architecture. It uses the Lite version of the AMBA AHB interface as system bus and includes a Nested Vectored Interrupt Controller (NVIC) component for interruption handling. It also includes optional features such as a proprietary optional hardware debug, a single-cycle IO interfacing, and a memory-protection functionality [161]. The processor can execute Thumb code and is compatible with other mobile-oriented Cortex-M processors [202].

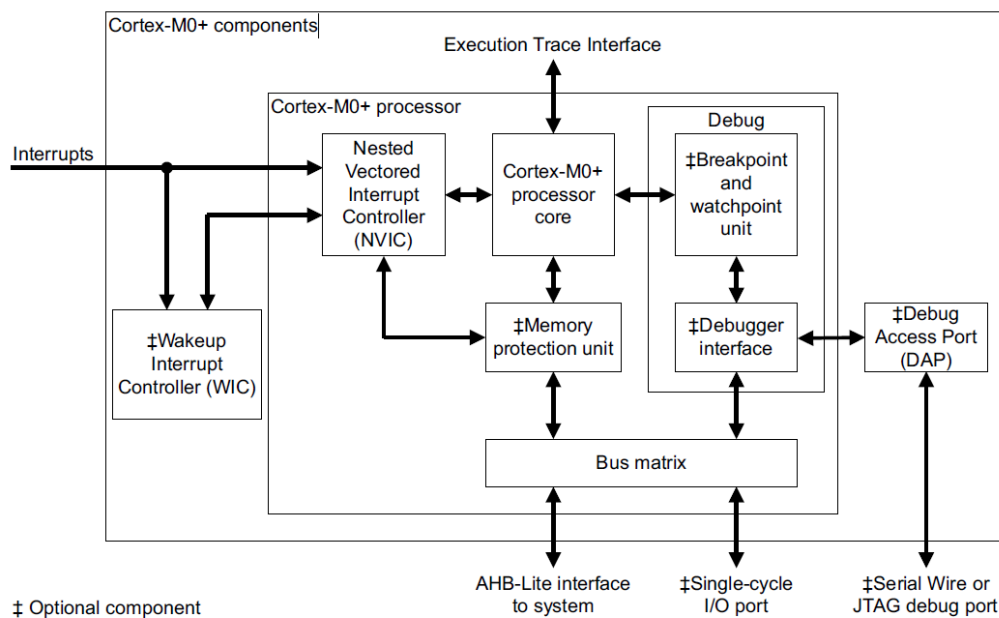


Figure A.2: Functional block diagram of the ARM Cortex-M0+. Extracted from the Cortex-M0+ Technical Reference Manual [202].

The Cortex-M0+ core includes the following features:

- The ARMv6-M Thumb instruction set with the Thumb-2 technology. It extends the architecture with 16 and 32-bits instructions for code size optimization and performances;
- A 32-bit hardware multiplier which can be a standard single-cycle multiplier, or a 32-cycle multiplier which results in a lower area during implementation and performance;
- Either little-endian or byte invariant big-endian data accesses support;
- Deterministic, fixed-latency interrupt handling with possible abandon and restart of instructions;
- Optional Unprivileged/Privileged support for improved system integrity;
- Full C-Application Binary Interface (C-ABI) compliant exception model.

Memory Model

The 4 GB memory space is architecturally split into regions giving a recommended usage for software porting between devices and access to the built-in component. It also ensures a unified programming model for interrupt control and debug between the Cortex-M family.

The regions are located within the memory map at fixed locations (see Figure 2.9) and accessible by the software running. The memory space is shared between volatile and non-volatile memories, external and built-in peripherals, system regions and other reserved spaces. Contrary to other Cortex-M processors, the M0+/M0's debug features are only visible by the debug components, and thus do not appear in the memory map.

The memory connected to the CPU generally uses 32-bits data words yet, the memory system of the Cortex-M also supports byte (8-bits), half-word (16-bits) and word (32-bits) memory transfers. with a suitable memory interface hardware. This allows connecting memory of different data widths. Both little and big endianness are supported at implementation level.

System Interface

The 32-bit system bus interface of the Cortex-M0+ is based on the AHB-Lite bus protocol, defined in the AMBA 3 standard. It is a generic configurable master-slave communication protocol between blocks which support different data size transfer. The CPU acts as a master and any other blocks are considered as a slave (with exception for the Debug and the Direct Memory Access (DMA)). The simplest bus transaction consists of a one cycle address transfer and a one or more cycle data transfer, yet extended modes such as burst/locked transfer, protection controls are also available [261].

In addition to the system bus, a single-cycle IO optional port is available for very high-speed access to tightly-coupled peripherals. This extra port is accessible from the processor and the debugger using standard load and store instructions.

Debug Interface

A debugger system is available for instrumentation and checking of processor correct execution. Program breakpoints and data watchpoints can be added to determine the processor state at a specific moment of the code execution. When added to the whole system, the debug interface is a master of the system bus. It can place the core in halted state and suspend exe-

cution. The debug system is connected to the ARM DAP and the system bus by the debugger interface. The DAP is in the end connected to the external environment – generally using a debug probe [262] – through a JTAG-interface or a 2-ports serial wire port also known as Serial Wire Debug (SWD) [263].

Interrupt Controller(s)

The Cortex-M0+ has been designed with a processor's execution control part closely tied to a built-in interrupt controller called the NVIC. Besides offering a powerful and easy-to-use interrupt's management it also guarantees quick and deterministic interrupt responses which provides a low latency and handling of late arriving interrupts [202].

In a minimal configuration, the external interruption signals are directly connected to the NVIC which prioritize them. Some interruptions are not maskable due to their high-priority (NMI, Receive Event (RXEV)). The priority of the remaining exception signals, which are limited to 32 interruptions, are defined through software. In fact, the configuration registers of the NVIC are accessible using word transfer only to guarantee predictability.

The core implementation can also include an optional WIC. This last component enables the processor and the NVIC to be placed into low power mode, while identification of interrupts is still maintained by the WIC. Actually, in a low power implementation where the core might be powered down, and so the NVIC, the WIC becomes a mandatory component to handle interruptions.

Low Power Features

With other Cortex-M processors, the M0+ natively includes some low-power features. First, two architectural suspend modes are available: sleep and deep sleep. Within the processor, both modes work in a similar fashion, yet device-specific extension and control features are expected during the implementation. Such specific implementation will be particularly useful for this work and discussed in Section 4.3.1. These modes can be entered by software using the WFE and WFI instructions. The selected mode is determined by a configuration bit in the State Control Register (SCR), which should be set before execution of the aforementioned instruction. Otherwise, the default mode is sleep.

Besides, a Sleep-On-Exit feature allows interrupt driven application to wake-up from a specific exception, thus helping to stay in a low-power mode as often as possible. The specific breakdown between the WIC and NVIC also allows to place in low-power mode of a part of the design (the NVIC) while still maintaining the WIC for wake up and resume of operations.

Other low-power design implementations and design techniques are used specifically in the Cortex M0+ to reduce the power consumption as much as possible. Instead of using a three-stage pipeline as the rest of the Cortex family, the Cortex-M0+ is designed around a two-stage pipeline as shown in Figure A.3. The first stage fetch and pre-Decode the instruction while, in a second step, the main decoding and execution is performed [202].

With this reduced pipeline the overall power consumption is initially minimized by reducing the number of access to the memory. Moreover, contrary to the older three-stage pipe of Cortex-M core with no pre-decode, when a conditional branch is executed, the next instruction is not added in the pipe. Hence, pipeline does not need to be flushed every time there is a

branch instruction, reducing the latency by one clock cycle.

Moreover, the number of required flip-flops is reduced yet stage logic paths tend to be longer. Indeed, the decode logic now embedded into the fetch and execute stage increase (and impacts timings). However, as register are significantly bigger and more power hungry, this solution still reduces the area and power. The balance between the pre-Decode and decode stages has also been optimized to minimize the impact on power frequency. In some cases, it results in a 30% power reduction between the M0+ and M0 [161].

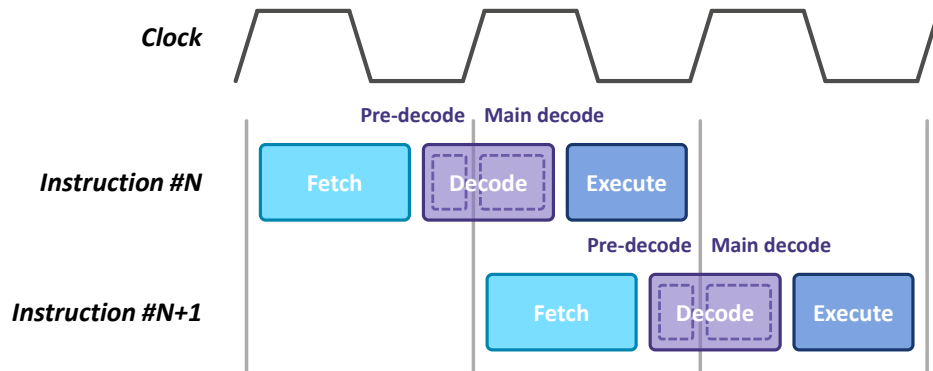


Figure A.3: Two-stage pipeline representation of the Cortex-M0+.

Appendix B

Introduction to the Allan deviation

THIS Appendix introduces a standard metric, known as the Allan Deviation, to compare the frequency stability of timers. This whole section is adapted from the author direct contribution in chapter 1 of the book entitled *Low Power Circuits for Emerging Applications in Communications, Computing, and Sensing* [158], published during this thesis. Extensive systems' frequency requirements and definitions are also given by the International Telecommunication Union (ITU) [264].

B.1 Stability and Accuracy

Overall, the accuracy expresses the measured timer deviation value from a reference of the quantity being measured. The time accuracy (i.e., period accuracy) and the frequency accuracy can be a single value or averaged on several samples. They are often normalized to a reference value. The dimensionless values $\frac{\Delta T}{T}$ and $\frac{\Delta f}{f}$ are generally used to describe time and frequency accuracy. In this case, it expresses the proper setting of a timer on a target period or frequency reference.

However, this metric does not indicate the variation around a reference set. Indeed, the frequency change caused by any environmentally and/or spontaneous action within a given time interval is expressed in term of frequency stability. Stability defines how well timers will keep the same accuracy over a given time interval. Stability will not guarantee that the output period or frequency is accurate but, only that it remains the same. These accuracy and stability considerations are given in Fig. B.1. It illustrates how an unstable device can be accurate while an inaccurate device can be temporarily stable.

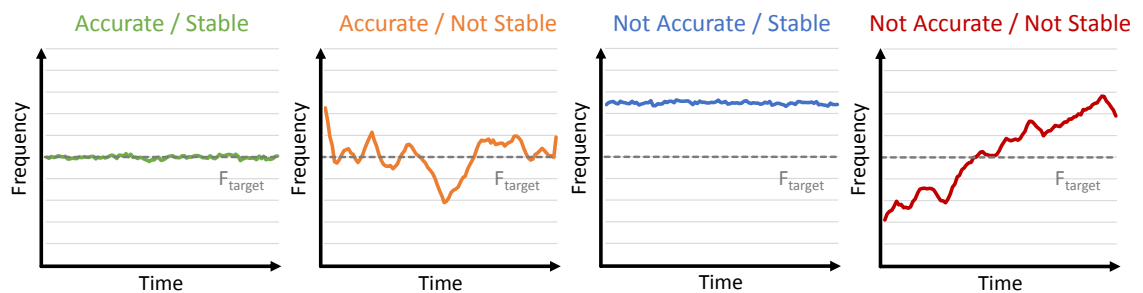


Figure B.1: Accuracy and stability evaluation.

B.2 Frequency Stability Characterization

Requirements

This work needs to define time references, thus a common set of frequency stability characterization parameters is necessary. If a source guarantees a certain stability but presents inaccuracy, the observed error can be seen as a constant offset and thus be removed. Therefore, in our application case, it is primordial to insist on the stability to ensure a proper time reference [265].

To ensure long term stability, requirements on different applications can be described with two kind of parameters depending of the problem's analysis domain selected. In the Fourier frequency domain, spectral parameters related to the spread of signal energy over the frequency spectrum are preferred. Spectral densities of phase and frequency fluctuation are particularly well adapted. In the time domain, the widely statistical time parameters used are variance or square-root of the variance also called standard-deviation. Due to the inherent association between these two reciprocal domains, integral relationships exist to switch from spectral densities to variances as shown in [266].

For low power time references, it is mandatory to characterize the stability of the timer over a time interval τ which can range from milliseconds to years. In that circumstance, we will prefer the time domain approach, since it easily expresses the stability of the timer over a time interval of a given length. However, because of the standard variance's divergence due to the flicker noise, a M-sample variance has been established by the early-1960's [267]. As shown in [268] the use of the standard deviation stands on the existence of an absolute mean frequency which is not a practical assumption.

The Allan Deviation

As the IEEE standard *Definitions of Physical Quantities for Fundamental Frequency and Time Metrology* [269] recommends, we will consider the timers frequency stability through the utilization of the 2-sample deviation $\sigma_y(\tau)$ also called Allan deviation. It is the square-root of the 2-sample variance $\sigma_y^2(\tau)$, also called Allan variance and its expression is given in Eq. B.1 for a given observation time τ .

$$\sigma_y(\tau) = \left[\frac{1}{2} \langle [\bar{y}_{k+1} - \bar{y}_k]^2 \rangle \right]^{1/2} \quad (\text{B.1})$$

The symbol $\langle \rangle$ denotes an infinite time average (i.e., $k \in [0; +\infty]$) whereas \bar{y}_k is the k^{th} instantaneous average of the fractional frequency deviation $y(t)$ over the time τ and defined by;

$$\bar{y}_k = \frac{1}{\tau} \int_{t_k}^{t_k+\tau} y(t) dt \quad (\text{B.2})$$

with;

- $t_k = t_0 + k\tau$ for some time origin t_0 .
- $y(t)$ normalized difference between the frequency $\nu(t)$ and the nominal frequency ν_n .

The Allan variance expresses the variance of the timing error accumulated after a time interval τ relatively to a reference clock even if the mean oscillator's frequency is changing.

This 2-sample variance is a function of the sample period as it depends on the time period used between samples, contrary to the distribution being measured. Hence, as the sampling time must be reported, the Allan variance is generally displayed as a whole graph rather than a single value. A timer with good stability will exhibit a low Allan variance [270].

In practical applications, the infinite time average requirement cannot be fulfilled. Hence, we will estimate the Allan deviation using the non-overlapped estimate of the Allan deviation where $y(t)$ is being averaged over non-overlapping intervals. For M number of frequency measurements Eq. B.1 becomes;

$$\sigma_y(\tau) \cong \left[\frac{1}{2(M-1)} \sum_{k=1}^{M-1} (\bar{y}_{k+1} - \bar{y}_k)^2 \right]^{1/2} \quad (\text{B.3})$$

The application of such estimators in various cases is left to the curiosity of the reader and further explained in [271]. Based on this key metrics and stability metrology tools, we are now able to perform an overview of the available timing reference currently available.

B.3 State of the Art Analysis

To offer an overview of the stability given by the solutions presented in Section 3.1, Figure B.2 reports published work on time references and their Allan deviation with regard to their power consumption. The corresponding references associated to each point are given in Table B.1.

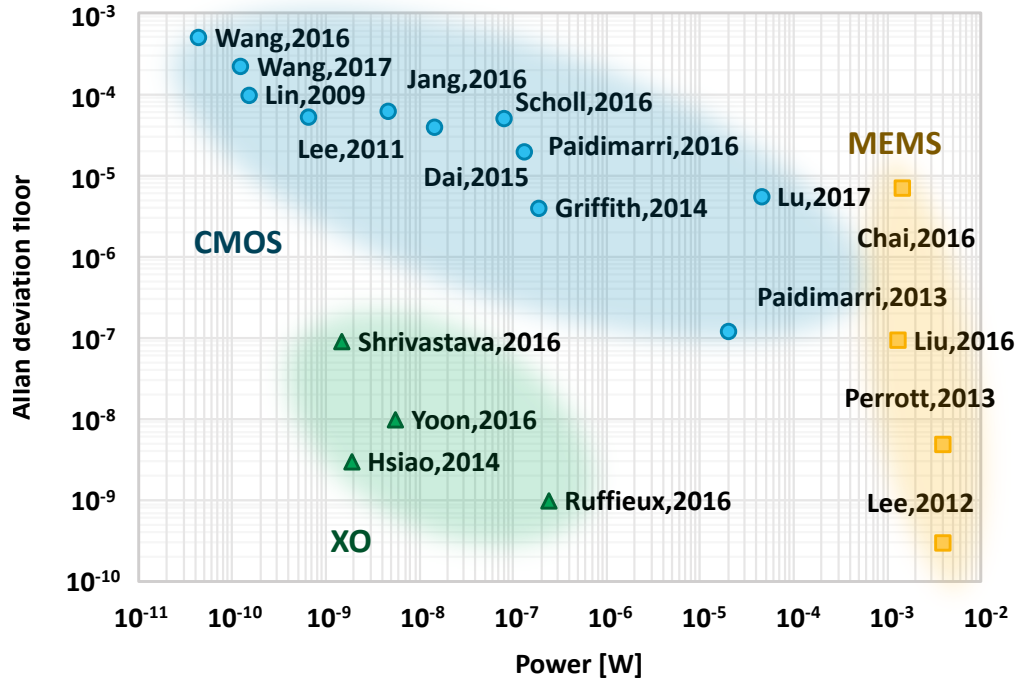


Figure B.2: Allan deviation of timers with regard to their power consumption.

Label	Reference	Power [W]	Allan deviation floor $\sigma_y(\tau)$	Observation window τ [s]	Type
Scholl2016	ESSCIRC'16 [191]	8.0×10^{-8}	5.0×10^{-5}	10	CMOS
Lin2009	ISSCC'09 [272]	1.5×10^{-10}	9.9×10^{-5}	1400	
Lee2011	ISSCC'11 [273]	6.6×10^{-10}	5.2×10^{-5}	1400	
Wang2017	ESSCIRC'17 [274]	1.2×10^{-10}	2.2×10^{-4}	N/A	
Wang2016	JSSCC'16 [275]	4.4×10^{-11}	5.0×10^{-4}	100	
Dai2015	CICC'15 [276]	1.4×10^{-8}	4.0×10^{-5}	1	
Griffith2014	ISSCC'14 [195]	1.9×10^{-7}	4.0×10^{-6}	10	
Lu2017	VLSI'17 [277]	4.5×10^{-5}	5.5×10^{-6}	2	
Paidimarri2013	ISSCC'13 [179]	2.0×10^{-5}	1.2×10^{-7}	100	
Jang2016	ISSCC'16 [178]	4.7×10^{-9}	6.3×10^{-5}	100	
Paidimarri2017	JSSCC'16 [180]	1.3×10^{-7}	2.0×10^{-5}	100	
Chai2016	TENCON'16 [278]	1.4×10^{-3}	7.0×10^{-6}	1	MEMS
Lee2012	IFCS'12 [170]	3.9×10^{-3}	3.0×10^{-10}	1	
Liu2016	IFCS'16 [153]	1.3×10^{-3}	9.5×10^{-8}	1	
Perrott2013	JSSCC'13 [279]	3.9×10^{-3}	5.0×10^{-9}	10	
Hsiao2014	ISSCC'14 [280]	1.9×10^{-9}	3.0×10^{-9}	1000	XO
Shrivastava2016	JSSCC'16 [166]	1.5×10^{-9}	9.0×10^{-8}	1000	
Yoon2016	JSSCC'16 [281]	5.6×10^{-9}	1.0×10^{-8}	1000	
Ruffieux2016	ISSCC'16 [282]	2.4×10^{-7}	1.0×10^{-9}	10	

Table B.1: Corresponding references and data associated to Figure B.2.

Appendix C

Electronic Design Automation Flow

THIS appendix introduces a conventional industrial EDA flow used to implement SoC from a logic abstraction level to a manufacturable description in a given CMOS technology process. A focus is then performed on the IEEE UPF standard which specifies power intent and power optimization for EDA tools. Lastly, a short presentation of low power simulations necessary for correct validation of a design is done.

C.1 Synthesis, Place-and-Route, Sign-Off

From the early ages of the Electronic Design Automation (EDA) the implementation of digital integrated circuit underwent some significant changes due to technology scaling [283]. A modern conception flow is proposed in Figure C.1. From specifications, it leads to a logic description of a system using an RTL language. After successive iterative steps, a final Graphic Database System (GDS) binary file is produced and sent to a foundry for manufacturing.

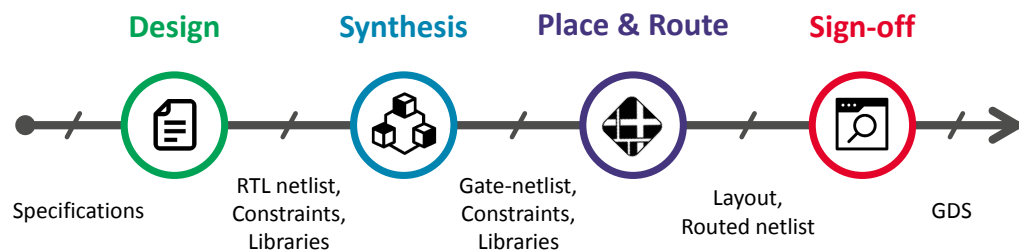


Figure C.1: Electronic Design Automation flow for digital circuit implementations.

All the information related to a given technology and required by the flow for the development of an IC are gathered into a Product Design Kit (PDK) which is provided by a silicon manufacturer. For a digital implementation flow, standard cells libraries – described using liberty files (.lib) – are mandatory. They contain HDL logic and/or functional descriptions, schematics and layouts and characterization data referring to the timing and power consumption performances of the cells, according to the Process Voltage Temperature Slope Capacitance (PVTSC) conditions. This data allows time and power verification of a circuit at each step.

Design

Following specifications and targeted results, an RTL netlist of the circuit is produced. It is a standard description of the circuit functionality which is done using a hardware description language. Nowadays, several languages are available with different level of abstraction. The industry uses the Very High Speed Integrated Circuit Hardware Description Language (VHDL), Verilog and its newer derivative version SystemVerilog, or even SystemC. Some high-level description languages are also developed and used in the academia, such as Chisel based on Scala [284], or MyHDL, an open-source package using directly Python [238]. This solution aims to propose a combined hardware description and verification language, accessible to as many people as possible. Unfortunately, they are not well supported by the industry tools.

Synthesis

Synthesis consists in translating the functionality of a circuit defined in the RTL netlist into a gate netlist composed of standard cells or IPs. The logical functions are converted using the logic description of the cells contained in libraries and some input constraints files.

Place & Route

This step carries out the placement of the cells contained in the gate-netlist elaborated during the synthesis. It relies on a floorplanning, positions of the IOs, and information on the metal layers. The cells are placed according to the space allocated and the filling ratio defined by the designer. During this step is also elaborated the clock tree that propagates the main clock to the input of the sequential cells. This step is mandatory to guarantee functionality of synchronous circuits.

Next, a route step ensures the metal connection between the placed cells and given timing/physical constraints. In fact, the added wires result in capacities and resistances that are extracted and integrated into the timing checks of the circuit. Using this analysis, the routing tool conducts a verification of compliance with the following constraints:

- Maximum capacitance – The maximum capacity value at the output of a gate does not exceed the value defined in the liberty files;
- Maximum fan-out – The number of cells driven by a logic gate does not exceed the number set by the designer;
- Maximum transition – The input slopes are lower than the maximum slope defined in the liberty files;
- Setup – The setup times of signals are not violated according to the input clock(s);
- Hold – The hold times of the signal are not violated according to the gate properties;
- Duty cycle – The clock signal is balanced between high and low levels.

When a constraint is not respected, the tool modifies the circuit for a better result. In the end, a layout file associated with a routed netlist including the power pins is produced. All the tools required during the flow for modeling, designing, testing, implementing, etc. are regrouped under the name CAD softwares.

Sign-Off

The last step, also known as Sign-Off, consists of formal verification to validate the layout and routed netlist obtained. Functional equivalence between the original RTL netlist and the final netlist is done. The conformity between the layout and routed netlist is also performed. Technology related design rules are checked as well as the timing constraints.

C.2 Unified Power Format

Historically, when the EDA tools and standards were developed by the industry in the 1980s, the main design constraint was the transistor area required to implement a particular functionality in a given technology. All power considerations were also often defined for a whole chip operated on a single power domain. Consequently, HDLs efficiently described the functional behavior of a system, but cannot capture the power architecture.

Hence, the Unified Power Format (UPF) has been developed to integrate power management techniques and description into an EDA flow, facilitating design automation and verification in implementation tools. The goal of this standard, also referred as the Institute of Electrical and Electronics Engineers (IEEE) Std. 1801 is to provide portable, low-power design specifications that can be integrated in open tools or products commercialized by CAD vendors. This results in mostly unified commands used for; power intent description, static and dynamic verification of power consumption, low-power attributes definitions for standard cells and liberty files, power-aware models and finally low power guidelines.

The latest version of the standard – as of December 13, 2019 – is the IEEE Std. 1801-2018 or UPF V4.0. For this whole work has been used the older V2.0 or IEEE Std. 1801-2009 version as it is better supported by tools.

Power Intent Basics

The UPF standard defines a series of syntaxes and semantics to express power intent in energy-aware electronic system design. Figure C.2 proposes a dummy SoC where the various logic elements and their power relation can be described using the UPF. In this standard are described the following elements:

- Power Domains – Collection of instances that are treated as a group for power-management purposes. The instances of a power domain typically share a primary supply set. The assumption of a single default supply pair for each power domain is no more mandatory since UPF V2.0;
- Power Switches – Instance that conditionally connects one or more input supply nets to a single output supply net according to the logical state of one or more input controls;
- Level shifters – Instances that convert signal values from an input voltage swing to a different output voltage swing;
- Isolation strategies – Procedures which provide a defined behavior for a logic signal when its driving logic is in power-down state;
- Isolation cells – Instances that pass logic values during normal mode operation and clamp their outputs to some specified value when a control signal is asserted;
- Retention strategies – Procedures associated with selected sequential elements memories to preserve the circuit state during the shutdown of the primary supplies;

- Retention registers – Registers which extend the sequential functionality with memory retention state during power-down;
- Always-On cells – Instance of a library cell with more than one set of supply pins which remains functional when the supply connected to the switchable supply pins is powered off.

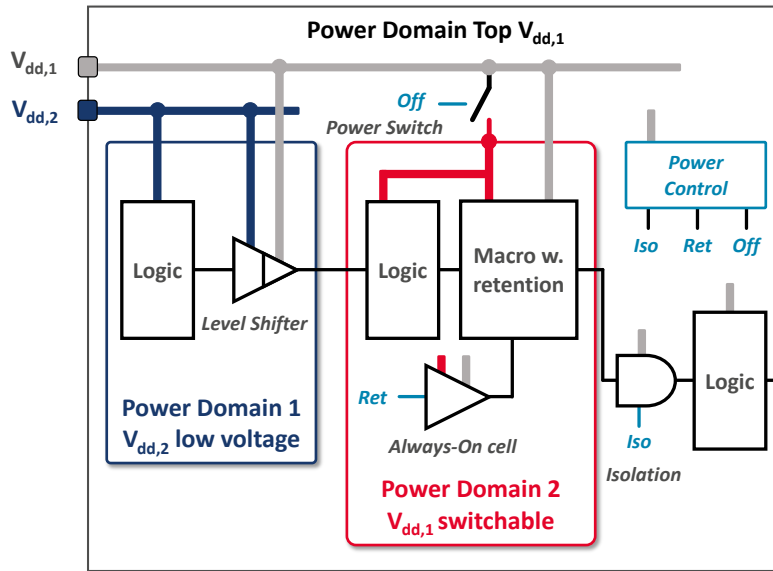


Figure C.2: Dummy system for power intent basic definitions.

UPF Implementation Flow

The UPF defines the relationship between the power intent described using commands from the standard and the design intents given by the standard HDL and cell libraries. However, the UPF is no more than a generalized abstraction of the HDL-based design hierarchy. Then, careful alignment between the logical and the physical hierarchies must be done to control the low-power implementation from the RTL level.

A consistent interpretation of the power behavior is thus provided, which can be integrated throughout the EDA flow for physical implementation and verification. A Golden UPF flow uses only one UPF file all along the steps and should be preferred for consistency between tools.

C.3 Simulations and Validation

All along the EDA flow, the correct operation of the circuit produced by a given step must be checked. Functional simulations combined with timing checks, obtained from a file containing the time of propagation of each gate and each wire composing the circuit, are done. The logic functions are thus checked when delays due to physical cells and wires are added.

Moreover, the low power functionalities brought by the UPF description should also be verified. Low-power techniques are known to bring new bugs. Hence, the following

things should be particularly taken into considerations; isolation/level shifting strategies, control sequencing bugs, retention scheme/control errors, electrical issue like memory corruption, power sequencing/voltage scheduling errors, hardware-software deadlock and power-on-reset/bring up problems. Therefore, a rigorous methodology is needed in the flow to check both static and dynamic potential issues.

Appendix D

UPF Descriptions for Power Gating

THIS appendix gives the UPF descriptions used in Chapter 4 to implement ULP techniques in a standard EDA flow. All the commands are based on UPF V2.0 (i.e., IEEE 18012009). The isolation strategies are left to the curiosity of the reader.

D.1 Power Gating Implementation

The two physical implementations of power gating presented in this section do not require custom tools or EDA flow modifications to be included into a specific design. Sizing and cell selection can be done at the early stages of the flow using SPICE simulator, then validated at any stage of the chip design. Power aware verification of the design can later be performed using an RTL description of the design together with the power intent.

The instantiation of a power gate (see Section 4.2.1) is described through an UPF command file as shown in Figure D.1. The power domain associated to the power gating strategy is defined as well as the IO supply port, control signal and ON/OFF state. An association of a physical cell to a strategy is also required.

```

1  # Create Power Switches
2  create_power_switch <PS_NAME>
3      -domain <POWER_DOMAIN_NAME>
4      -input_supply_port {VDD <VDD_NAME>}
5      -output_supply_port {VDD_OUT <VDD_OUT_NAME>}
6      -control_port {CMD <CMD_NAME>}
7      -on_state {on_state VDD {!CMD}}
8      -off_state {off_state { CMD}}
9
10 # Map Power Switches
11 map_power_switch <PS_NAME>
12     -domain <POWER_DOMAIN_NAME>
13     -lib_cells <PS_CELL_NAME>

```

Figure D.1: UPF standard decription to define a power gating strategy.

For a global custom power switch packaging, special care has to be taken during library creation to include cell and Power/Ground (PG) pins special attributes (see Figure D.2). The

control signal – necessary to instantiate the power switch in the UPF – is also added, defining the ON and OFF states of the device.

```

1  cell (POWER_SWITCH) {
2      switch_cell_type : "coarse_grain";
3      ...
4      # Input power pin
5      pg_pin (VDD) {
6          pg_type : primary_power;
7          ...
8      }
9      # Output power pin
10     pg_pin(VDD_OUT){
11         pg_type : internal_power;
12         # Switch OFF when "CMD" high
13         switch_function : "CMD";
14         ...
15     }
16 }

```

Figure D.2: Power switches liberty files attributes.

When power gating is activated, the sub-circuits associated are powered down, which results in boundary signal to reach non-logic levels. To avoid crowbarred conditions in the circuits connected downstream the power gated block, special isolations or clamp cells should be applied on the outputs. These cells are powered using global always-on power supplies to clamp the signal to a high or low value. Depending on the isolation strategy described in the UPF power intent, the equivalent AND-gate, OR-gate or Latch are added inside the power domain or the parent domain.

The explicit high or low isolation requires careful analysis of the signal state before and after power gating to avoid logic deadlock at start-up or during operation (e.g., retention signal in the wrong state at restart). If isolation latches are available in the standard cell offer, they could be used to avoid signal state enumeration on a block with a high number of outputs. However, as the complexity of the cell growth, it also increases timing delays and power consumption.

D.2 Double-Gated SRPG implementation

This section gives the power intent for the SRPG implementation proposed in 22 nm FD-SOI using a triple power grid description with two power domains *POFF* and *TOP POFF* (see Section 1.5.2).

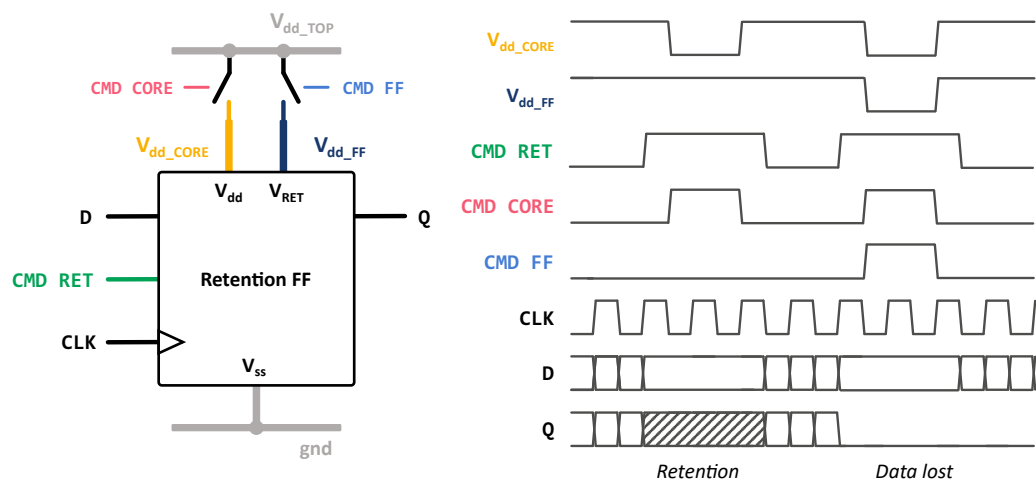


Figure D.3: Triple power grid description for SRPG implementation in 22 nm FD-SOI.

```

17 #####
18 # Create and logically connect power ports and nets
19 #####
20 # Always-ON Vdd TOP
21 create_supply_net Vdd_TOP
22 create_supply_port Vdd_TOP
23 connect_supply_net Vdd_TOP -ports Vdd_TOP
24 # Switched Vdd CORE
25 create_supply_net Vdd_CORE
26 # Switched Vdd FF
27 create_supply_net Vdd_FF
28 # gnds
29 create_supply_net gnds
30 create_supply_port gnds
31 connect_supply_net gnds -ports gnds
32 # vdds
33 create_supply_net vdds
34 create_supply_port vdds
35 connect_supply_net vdds -ports vdds
36 # ground
37 create_supply_net gnd
38 create_supply_port gnd
39 connect_supply_net gnd -ports gnd
40
41 #####
42 # Create Supply Set
43 #####
44 create_supply_set POFF_CORE_alim \
45     -function {power Vdd_CORE} \
46     -function {ground gnd} \
47     -function {nwell vdds} \
48     -function {pwell gnds} \
49
50 create_supply_set POFF_FF_alim \
51     -function {power Vdd_FF} \
52     -function {ground gnd} \
53     -function {nwell vdds} \
54     -function {pwell gnds} \
55
56 create_supply_set POFF_default \
57     -function {power Vdd_TOP} \
58     -function {ground gnd} \
59     -function {nwell vdds} \
60     -function {pwell gnds}

```

Figure D.4: Power intent description for SRPG implementation (1/3).

```

61 #####
62 # Create power domains
63 #####
64 # POFF power domain
65 create_power_domain POFF
66     -elements {<CORE LOGIC>}
67     -supply {primary POFF_CORE_alim}
68     -supply {extra_supplies_0 POFF_FF_alim}
69     -supply {extra_supplies_1 POFF_default}
70 # TOP POFF power domain
71 create_power_domain TOP_POFF
72     -elements {<NONE>}
73     -supply {primary POFF_FF_alim}
74     -supply {extra_supplies_0 POFF_default}
75
76 #####
77 # Create Power Switches
78 #####
79 # Core power switch strategy
80 create_power_switch SW_CORE_POFF -domain POFF
81     -input_supply_port {VDDC Vdd_TOP}
82     -output_supply_port {VDD Vdd_CORE}
83     -control_port {EN CMD_CORE}
84     -on_state {on_state VDDC {!EN}}
85     -off_state {off_state { EN}}
86
87 # Top power switch strategy
88 create_power_switch SW_FF_POFF -domain TOP_POFF
89     -input_supply_port {VDDC Vdd_TOP}
90     -output_supply_port {VDD Vdd_FF}
91     -control_port {EN CMD_FF}
92     -on_state {on_state VDDC {!EN}}
93     -off_state {off_state { EN}}
94 #####
95 # Map Power Switches
96 #####
97 # Core power switch
98 map_power_switch SW_CORE_POFF
99     -domain POFF
100     -lib_cells <PS_CELL_NAME>
101
102 # Top power switch
103 map_power_switch SW_FF_POFF
104     -domain TOP_POFF
105     -lib_cells <PS_CELL_NAME>

```

Figure D.5: Power intent description for SRPG implementation (2/3).

```

106 #####
107 # Set Retention
108 #####
109 # Retention strategy
110 set_retention          RETPOFF
111     -domain            POFF
112     -retention_supply_set POFF_FF_alim
113     -elements          {<CORE LOGIC>}
114
115 # Retention control
116 set_retention_control RETPOFF
117     -domain            POFF
118     -save_signal        {CMD_RET high}
119     -restore_signal     {CMD_RET low}
120
121 # Retention cell mapping
122 map_retention_cell     RETPOFF
123     -domain            POFF
124     -lib_cells          <RET_CELL_NAME>
125
126 ## EOF

```

Figure D.6: Power intent description for SRPG implementation (3/3).

Appendix E

Measurement Setup

ALL the SoCs presented in Section 4.5.1 are packaged using a standard 304 pins Super Ball-Grid Array (SBGA) package. For measurement purposes, a full test and instrumentation setup has been built around a Kintex-7 FPGA board from Xilinx.

As shown in Figure E.1, a custom development board (daughter board) encloses a Zero Insertion Force (ZIF) socket to insert the DUT. The required power supplies can be generated on-board using embedded generators. Using a series of jumpers external generators, equipment can also be connected to facilitate power measurements. In a similar fashion, body-bias power can be generated on-chip using four dedicated DACs for LVT and RVT transistors wells.

Most of the testchip signals are connected to the FPGA board through two FPGA Mezzanine Card (FMC) connectors. Some clock IOs signals can also be generated or monitored off-chip using SubMiniature version A (SMA) coaxial connectors and solder bridge connections. Lastly, for debug purposes and observability, standard logic analyzer connectors are placed to connect Keysight “Soft Touch Connectorless Probes”.

The measurement setup under utilization is shown in Figure E.2. The FPGA board is programmed with a bitstream generated from an HDL description. A compiled C-code program is stored on the FPGA and loaded on the SoC memories using the SWD protocol. Automated routines are programmed on the FPGA to communicate with the testchip. Using a complementary computer running a python program, additional commands are sent to the FPGA and the DUT. The same computer also controls external equipment using a Universal Serial Bus (USB) to General Purpose Interface Bus (GPIB) probe.

Contrary to Electrical Wafer Sorting (EWS) testing, this whole setup allows versatile and easy communications with the IPs to be tested which facilitates emulations of realistic SoC applications. Indeed, the testing procedures are described using FSMs embedded on the FPGA or written in Python with a high-level of abstraction. Unfortunately, even though the ZIF helps to change the DUT, the number of dies to be tested in a small amount of time with a wide range of voltage and temperature conditions is still limited. Thus, EWS tests are still required to facilitate experimentation on a wide distribution of samples, with simple testing procedure. Then, the FPGA setup is used for fine instrumentation and measurements.

Lately, recent FPGA boards incorporate CPU core able to run Linux-like distributions. In future developments, these kinds of board could be used to directly run the actual computer program on the FPGA and improve the integration of testing procedures.

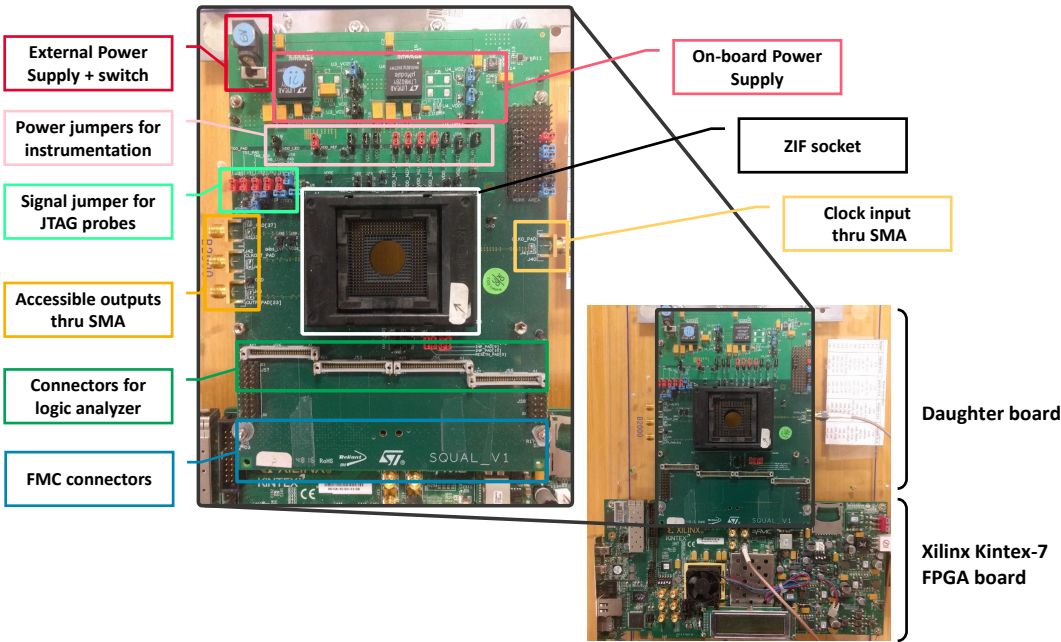


Figure E.1: Picture of the daughter board and Xilinx Kintex-7 FPGA board instrumentation setup.

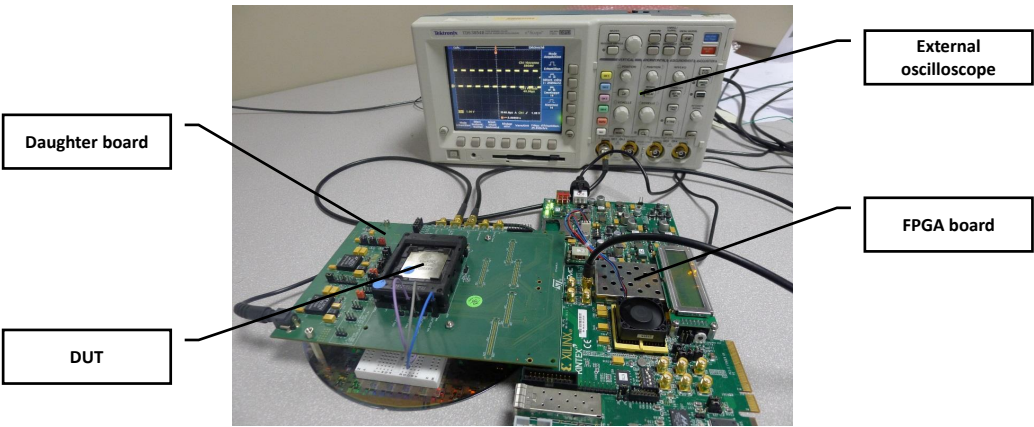


Figure E.2: FPGA board instrumentation setup under utilization.

Appendix F

Register Configurations of the Hardware Power Management Modules

THIS appendix gives the complete register configuration of the PMU, WUC and APM modules. These IPs are accessible through an APB bus connected to the ARM Cortex M0+ core and thus configurable using the application code. To access the registers, the final memory mapping is given in a C-code header file `hpm.h`.

F.1 Power Management Unit Register Mapping

PMU peripheral base address: `0xA0003000`

PMUCFG – `0x00000000`

Bits	Field Name	Reset	Type*	Description
[31]	SOFT_DRIVEN	0x01	RW	Enable the software control of the PMU 1: Soft driven – 0: APM driven
[30]	APM_ENABLE	0x00	RW	Enable the APM 1: Activated – 0: Disabled
[29]	SOFT_SLEEP_MODE	0x00	RW	Frequency mode during sleep mode 0: 16 MHz – 1: 4 MHz
[28]	SOFT_DEEPSLEEP_MODE	0x00	RW	Frequency mode during sleep mode 0: 4 MHz – 1: 32 kHz
[27:25]	APM_LAST_MODE	0x00	R	Last mode requested by the APM
[24:0]	UNUSED			

* R: Read only – W: Write only – RW: Read and Write

Table F.1: PMUCFG registers description.

F.2 Wake-Up Controller Register Mapping

WUC peripheral base address: 0xA0004000

WUCCFG – 0x00000000

Bits	Field Name	Reset	Type*	Description
[31]	ULTRA_DEEP	0x00	RW	Enable the retention FFs power rail 0: Activated – 1: Disabled
[30]	BYP_ACK_16M	0x00	RW	Bypass ACK_16M when switching request to 16 MHz done 1: ACK_16M pulse needed – 0: ACK_16M bypassed
[29]	BYP_ACK_4M	0x00	RW	Bypass ACK_4M when switching request to 4 MHz done 1: ACK_4M pulse needed – 0: ACK_4M bypassed
[28]	BYP_ACK_32K	0x00	RW	Bypass ACK_32K when switching request to 32 kHz done 1: ACK_32K pulse needed – 0: ACK_32K bypassed
[27]	APM_POWER	0x00	RW	APM Powering 0: Power ON – 1: Power OFF
[26:0]	UNUSED			

* R: Read only – W: Write only – RW: Read and Write

Table F.2: WUCCFG register description.

F.3 Adaptive Power Management Register Mapping

APM peripheral base address: 0xA0005000

ALPHA – 0x0000000C

Bits	Field Name	Reset	Type*	Description
[31:20]	UNUSED			
[19:0]	ALPHA	0xE6666	RW	Learning rate – C_1 constant

* R: Read only – W: Write only – RW: Read and Write

Table F.3: ALPHA register description.

ALPHAGAMMA – 0x00000010

Bits	Field Name	Reset	Type*	Description
[31:20]			UNUSED	
[19:0]	ALPHAGAMMA	0xDAE14	RW	Alpha Gamma product – C_2 constant

* R: Read only – W: Write only – RW: Read and Write

Table F.4: ALPHAGAMMA register description.

ALPHACOMPLEMENT – 0x00000014

Bits	Field Name	Reset	Type*	Description
[31:20]			UNUSED	
[19:0]	ALPHACOMPLEMENT	0x19999	RW	Alpha complement – C_3 constant

* R: Read only – W: Write only – RW: Read and Write

Table F.5: ALPHACOMPLEMENT register description.

F.4 C-code memory mapping

A C-code header file is given in Figure F.1 to access the running application the previously described register.

```

1  #ifndef _HPM_H
2  #define _HPM_H
3
4  // Base addresses
5  #define APB_PMU_BASE 0xa0003000 // Base address of APB periph. PMU
6  #define APB_WUC_BASE 0xa0004000 // Base address of APB periph. WUC
7  #define APB_APM_BASE 0xa0005000 // Base address of APB periph. APM
8
9  // Register offsets
10 #define PMU_PMUCFG_OFFSET 0x00 // Only one register in the PMU
11 #define WUC_WUCCFG_OFFSET 0x00 // Only one register in the WUC
12 // APM
13 #define APM_ALPHA_OFFSET          0x0C
14 #define APM_ALPHAGAMMA_OFFSET    0x10
15 #define APM_ALPHACOMPL_OFFSET    0x14
16
17 // Register addresses
18 #define ADDR_PMU_PMUCFG (APB_PMU_BASE + PMU_PMUCFG_OFFSET) // PMU
19 #define ADDR_WUC_WUCCFG (APB_WUC_BASE + WUC_WUCCFG_OFFSET) // WUC
20 // APM
21 #define ADDR_APM_ALPHA          (APB_APM_BASE + APM_ALPHA_OFFSET          )
22 #define ADDR_APM_ALPHAGAMMA    (APB_APM_BASE + APM_ALPHAGAMMA_OFFSET    )
23 #define ADDR_APM_ALPHACOMPL    (APB_APM_BASE + APM_ALPHACOMPL_OFFSET    )
24
25 // Register access macros
26 #define PMUCFG (*(volatile int *) ADDR_PMU_PMUCFG) // PMU configuration register
27 #define WUCCFG (*(volatile int *) ADDR_WUC_WUCCFG) // WUC configuration register
28 // APM
29 #define ALPHA          (*(volatile int *) ADDR_APM_ALPHA          )
30 #define ALPHAGAMMA    (*(volatile int *) ADDR_APM_ALPHAGAMMA    )
31 #define ALPHACOMPL    (*(volatile int *) ADDR_APM_ALPHACOMPL    )
32 #endif /* _HPM_H */

```

Figure F.1: C-code header file giving the HPM modules memory mapping.

Appendix G

Summary of Contributions

Peer-reviewed Journals

A 1.1 pJ/cycle, 20 MHz, 0.42 V Temperature Compensated ARM Cortex-M0+ SoC with Adaptive Self Body-Biasing in FD-SOI

G. Lallement, F. Abouzeid, J. M. Daveau, P. Roche and J. L. Autran

IEEE Solid-State Circuits Letters (SSC-L), vol. 1, no. 7, pp. 174-177, July 2018

A 2.7 pJ/cycle 16 MHz, 0.7 μ W Deep Sleep Power ARM Cortex-M0+ Core SoC in 28 nm FD-SOI

G. Lallement, F. Abouzeid, M. Cochet, J. M. Daveau, P. Roche and J. L. Autran

IEEE Journal of Solid-State Circuits (JSSC), vol. 53, no. 7, pp. 2088-2100, July 2018

Peer-reviewed Conferences

A 140 nW, 32.768 KHz, 1.9 ppm/ $^{\circ}$ C Leakage-Based Digitally Relocked Clock Reference with 0.1 ppm Long-Term Stability in 28nm FD-SOI

G. Lallement, F. Abouzeid, T. Di Gilio, P. Roche and J. L. Autran

IEEE Asian Solid-State Circuits Conference (A-SSCC), Tainan, Taiwan, 2018

Q-Learning-based Adaptive Power Management for IoT System-on-Chips with Embedded Power States

Y. Debizet, G. Lallement, F. Abouzeid, P. Roche and J. L. Autran

IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 2018

A 0.40 pJ/cycle 981 μm^2 voltage scalable digital frequency generator for SoC clocking

M. Cochet, S. Clerc, G. Lallement, F. Abouzeid, P. Roche and J. L. Autran

IEEE Asian Solid-State Circuits Conference (A-SSCC), Seoul, Korea, 2017

A 2.7 pJ/cycle 16MHz SoC with 4.3 nW power-off ARM Cortex-M0+ core in 28 nm FD-SOI

G. Lallement, F. Abouzeid, M. Cochet, J. M. Daveau, P. Roche and J. L. Autran

IEEE European Solid State Circuits Conference (ESSCIRC), Leuven, Belgium, 2017

Book**Clock Generation and Distribution for Low-Power Digital Systems**

M. Cochet, G. Lallement, F. Abouzeid

Book section in Low Power Circuits for Emerging Applications in Communications, Computing, and Sensing. By Fei Yuan, CRC Press, 2018.

Patent and Invention Disclosures**Patent related to an Ultra Low Power Leakage based ring oscillator**

G. Lallement, F. Abouzeid

Submitted as U.S. Patent – Under prosecution.

Patent related to the body biasing for ultra low voltage digital circuits

G. Lallement, F. Abouzeid

Submitted as U.S. Patent – Under prosecution.

Live Demonstrations**Power Adaptive ARM-based SoC thru Frequency Self-Adjustments at 0.45V in 28nm FD-SOI**

M. Cochet, G. Lallement, F. Abouzeid, V. Lorquet, P. Roche

SEMICON Europa, Grenoble, France, 2016

Design Gallery

Table G.1 summarizes the design, measurements and published contributions resulting from this work. It covers 5 different test chips realized during the July 2016 to July 2019 research period in 28 nm and 22 nm FD-SOI.

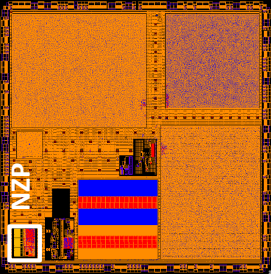
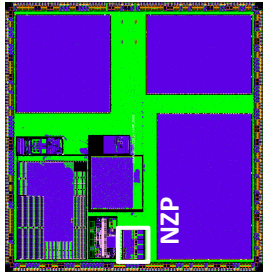
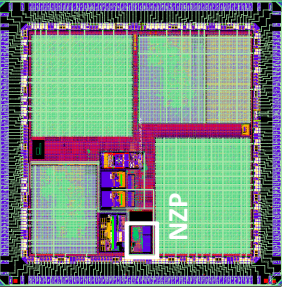
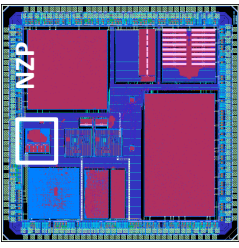
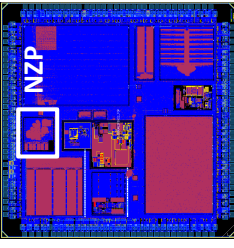
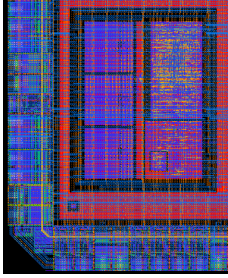
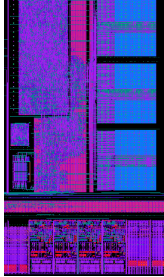
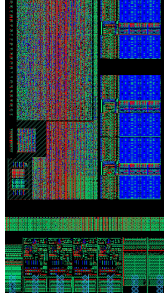
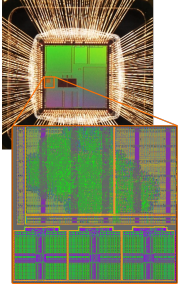
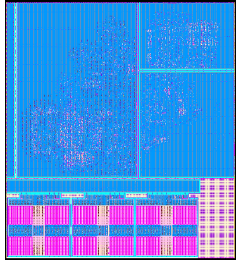
Testchip name	NZP28 V1.0	NZP28 V2.0	NZP28 V3.0	NZP22 V1.0	NZP22 V2.0
Tape-out date	July 2016	Jun. 2017	July 2017	Jun. 2018	Aug. 2019
Technology	28 nm FD-SOI	28 nm FD-SOI	28 nm FD-SOI	22 nm FD-SOI	22 nm FD-SOI
Contributions <i>Italic: meas. only</i>	NZP V1.0 DDSS PMU V1.0	NZP V2.0 LRO DDSS	NZP V3.0 w/ retention LRO DDSS PMU V2.0	NZP V4.0 Self-body bias PMU V3.0	NZP V4.0 APM PMU/WUC V1.0
Chip layout or micrograph	 9.3 mm ²	 9.3 mm ²	 9.3 mm ²	 7.1 mm ²	 7.1 mm ²
	 0.073 mm ²	 0.082 mm ²	 0.082 mm ²	 0.072 mm ²	 0.119 mm ²
Publications Bold: first author	ESSCIRC'16 A-SSCC'17 JSSC'17	A-SSCC'18 US Patent		SSCL'19 US Patent	ISCAS'18

Table G.1: Chips design gallery.

Biography

Guéno   Lallement was born in Nantes, France in 1991. He received with distinction the MSc. degree in Analog and Digital IC Design from Imperial College London, London, UK in 2016. The same year, he also received the Engineering Diploma in Electronic and Computer Science from T  l  com Paris, Paris, France. Under the supervision of Dr. Pantelis Georgiou he worked on creating bio-inspired CMOS DNA microarray. This project received the Hertha Ayrton Centenary Prize for the best MSc project with significant original contribution to the topic area. Since 2016 he is pursuing a PhD with the IM2NP institute from Aix-Marseille University and STMicroelectronics, Crolles under the co-supervision of Prof. Jean-Luc Autran and Dr. Fady Abouzeid. His current research targets the extension of SoCs mission capabilities by offering Near-Zero Power performances and enabling continuous functionality for IoT systems. With this regard, he has served as a reviewer for ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED) since 2017.

Bibliography

- [1] A. Bahai, "Ultra-low Energy systems: Analog to information," in *European Solid-State Device Research Conference*, vol. 2016-Octob, 2016, pp. 3–6. [pp. vii, 1, and 5.]
- [2] D. Bol, G. De Streel, and D. Flandre, "Can we connect trillions of IoT sensors in a sustainable way? A technology/circuit perspective," *2015 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference, S3S 2015*, pp. 5–7, 2015. [pp. vii and 3.]
- [3] C. Perera, C. H. Liu, S. Jayawardena, and M. Chen, "A Survey on Internet of Things from Industrial Market Perspective," *IEEE Access*, vol. 2, pp. 1660–1679, 2015. [pp. vii and 3.]
- [4] J. G. Koomey, S. Berard, M. Sanchez, and H. Wong, "Implications of Historical Trends in the Electrical Efficiency of Computing," *IEEE Annals of the History of Computing*, pp. 2–10, 2011. [pp. viii, 5, and 6.]
- [5] M. Alioto, "Ultra-low power VLSI circuit design demystified and explained: A tutorial," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 1, pp. 3–29, 2012. [p. viii.]
- [6] H. Reyserhove and W. Dehaene, "A 16 . 07pJ / cycle 31MHz Fully Differential Transmission Gate Logic ARM Cortex M0 core in 40nm CMOS," *2016 IEEE European Solid-State Circuits Conference (ESSCIRC)*, pp. 257–260, 2016. [p. viii.]
- [7] W. Brattain, "Brattain Lab Page." [Online]. Available: <https://www.pbs.org/transistor/science/labpages/labpg5.html> [p. xxi.]
- [8] E. Karlsson, "The Nobel Prize in Physics 1901-2000," pp. 1–24, 2012. [Online]. Available: <https://www.nobelprize.org/prizes/physics/1956/summary/> [p. xxi.]
- [9] G. E. Moore, "Cramming more components onto integrated circuits, Reprinted from *Electronics*, volume 38, number 8, April 19, 1965, pp.114 ff." *IEEE Solid-State Circuits Society Newsletter*, vol. 11, no. 3, pp. 33–35, sep 2006. [Online]. Available: <http://ieeexplore.ieee.org/document/4785860/> [pp. xxi and 14.]
- [10] R. Dennard, F. Gaensslen, L. Kuhn, and H. Yu, "Design of micron MOS switching devices," in *1972 International Electron Devices Meeting*. IRE, 1972, pp. 168–170. [Online]. Available: <http://ieeexplore.ieee.org/document/1477207/> [pp. xxi and 14.]
- [11] V. Roche, "Semiconductor innovation: Is the party over, or just getting started?" in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*. IEEE, feb 2018, pp. 8–11. [Online]. Available: <http://ieeexplore.ieee.org/document/8310164/> [p. 1.]

- [12] G. Bell, "Bell's Law for the birth and death of computer classes : A theory of the computer's evolution," *IEEE SSCS NEWS*, vol. 0, no. January, pp. 1–26, 2008. [p. 2.]
- [13] M. K. Tsai, "Cloud 2.0 clients and connectivity - Technology and challenges," *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, vol. 57, pp. 15–19, 2014. [Not cited.]
- [14] M. Alioto and Others, *Enabling the Internet of Things*, M. Alioto, Ed. Cham: Springer International Publishing, 2017. [Online]. Available: <http://link.springer.com/10.1007/978-3-319-51482-6> [pp. 2, 20, 25, 32, 34, and 51.]
- [15] E. Rogers, *Diffusion of innovations*. The Free Press, 1995. [p. 2.]
- [16] G. Bell, "The Mini and Micro Industries," *Computer*, vol. 17, no. 10, pp. 14–30, 1984. [p. 2.]
- [17] Oak Ridge Leadership Computing Facility, "SUMMIT - Oak Ridge National Laboratory's next High Performance Supercomputer." [Online]. Available: <https://www.olcf.ornl.gov/> [p. 1.]
- [18] "Top 500," 2018. [Online]. Available: <https://www.top500.org/lists/2018/11/> [p. 2.]
- [19] "Green500 List." [Online]. Available: <https://www.top500.org/green500/> [p. 2.]
- [20] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1, pp. 732–794, 2016. [p. 2.]
- [21] J. Scaramella and M. Eastwood, "Delivering Improved Energy Efficiency for the Datacenter," IDC, Tech. Rep. September, 2008. [Online]. Available: [https://sg.nec.com/en{}_SG/sites/default/files/idc{}_ecocenter.pdf](https://sg.nec.com/en/_SG/sites/default/files/idc{}_ecocenter.pdf) [p. 2.]
- [22] J. G. Koomey, "Worldwide electricity used in data centers," *Environmental Research Letters*, vol. 3, no. 3, 2008. [p. 2.]
- [23] Mueen Uddin, "Green Information Technology (IT) framework for energy efficient data centers using virtualization," *International Journal of the Physical Sciences*, vol. 7, no. 13, pp. 2052–2065, 2012. [Online]. Available: <http://www.academicjournals.org/IJPS/abstracts/abstracts/abstract2012/23Mar/Uddinetal.htm> [p. 2.]
- [24] M. Zuckerberg, "Mark Zuckerberg post on Facebook," 2016. [Online]. Available: <https://www.facebook.com/zuck/posts/10103136694875121> [p. 2.]
- [25] John Roach, "Under the sea, Microsoft tests a datacenter that's quick to deploy, could provide internet connectivity for years - Stories," 2018. [Online]. Available: <https://news.microsoft.com/features/under-the-sea-microsoft-tests-a-datacenter-thats-quick-to-deploy-could-provide-internet-connectivity-for-years/> [p. 2.]
- [26] S. Nomura, F. Tachibana, T. Fujita, C. K. Teh, H. Usui, F. Yamane, Y. Miyamoto, T. Yamashita, H. Hara, M. Hamada, and Y. Tsuboi, "A low-power multi-core media co-processor for mobile application processors," in *2009 IEEE International Conference on Integrated Circuit Design and Technology, ICICDT 2009*. IEEE, 2009, pp. 129–134. [p. 3.]
- [27] NVidia, "The Benefits of Quad Core CPUs in Mobile Devices," *nVidia White Paper*, pp. 1–19, 2011. [Online]. Available: http://www.nvidia.fr/content/PDF/tegra{}_white{}_papers/tegra-whitepaper-0911a.pdf [p. 3.]

- [28] C. Lichtenau, M. I. Ringler, T. Pflüger, S. Geissler, R. Hilgendorf, J. Heaslip, U. Weiss, P. Sandon, N. Rohrer, E. Cohen, and M. Canada, "PowerTune : Advanced Frequency and Power Scaling on 64b PowerPC Microprocessor," in *ISSCC*, no. July 2000, 2004, pp. 148–149. [p. 3.]
- [29] S. Nomura¹, F. Tachibana¹, T. Fujita¹, C. K. Teh¹, F. Usui¹, HiroyukiYAmane¹, Y. Miyamoto¹, H. Kumtornkittikul¹, ChaivasitHara¹, T. Yamashita¹, M. Tanabe¹, JunUCHiyama¹, Y. Tsuboi¹, T. Miyamori¹, H. Kitahara¹, TakeshiSAto¹, Y. Homma¹, S. Matsumoto², and Y. Seki², Keiko, W, "A 9.7mW AAC-Decoding, 620mW H.264 720p 60fps Decoding, 8-Core Media Processor with Embedded Forward-Body-Biasing and Power- Gating," in *2008 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, 2011, pp. 262–264. [p. 3.]
- [30] F. K. Shaikh, S. Zeadally, and E. Exposito, "Enabling technologies for green internet of things," *IEEE Systems Journal*, vol. 11, no. 2, pp. 983–994, 2017. [p. 3.]
- [31] S. Lucero, "IoT platforms : enabling the Internet of Things," 2016. [p. 3.]
- [32] K. Kaplan, "Robotic Spider Dress Powered by Intel Smart Wearable Technology," 2015. [Online]. Available: <https://iq.intel.com/smart-spider-dress-by-dutch-designer-anouk-wipprecht/> [p. 3.]
- [33] M. Tartagni, M. Belleville, E. Cantatore, and H. Fanet, *Energy autonomous systems: future trends in devices, technology, and systems*. Catrene, 2009. [Online]. Available: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Energy+Autonomous+Systems+:+Future+Trends+in+Devices+,+Technology+,+and+Systems{#}6> [pp. 4 and 126.]
- [34] J. F. M. Oudenhoven, R. J. M. Vullers, and R. Schaijk, "A review of the present situation and future developments of micro-batteries for wireless autonomous sensor systems," *International Journal of Energy Research*, vol. 36, no. 12, pp. 1139–1150, oct 2012. [Online]. Available: <http://doi.wiley.com/10.1002/er.2949> [pp. 4, 6, and 7.]
- [35] W. D. Raedt, L. Dussopt, M. Funk, P. Galvin, A. Ionescu, M. John, E. Jung, M. O. de Beeck, S. Pollin, R. Vullers, and F. Yazicioglu, "Smart Systems for Healthcare and Wellness," CATRENE Scientific Committee Working Group, Tech. Rep., 2014. [p. 4.]
- [36] W. Atkinson, "Interest in Smart Buildings Is Growing," 2018. [Online]. Available: <https://www.ecmag.com/section/systems/interest-smart-buildings-growing> [p. 4.]
- [37] S. Boisseau, G. Despesse, and B. Ahmed, "Electrostatic Conversion for Vibration Energy Harvesting," *Small-Scale Energy Harvesting*, pp. 1–39, 2012. [Online]. Available: <http://www.intechopen.com/books/small-scale-energy-harvesting/electrostatic-conversion-for-vibration-energy-harvesting> [p. 5.]
- [38] Intel, "Intel Core i9-9900K Processor." [Online]. Available: <https://www.intel.com/content/www/us/en/products/processors/core/i9-processors/i9-9900k.html> [p. 5.]
- [39] "GPU Performance - Returning To Lower Power - The Snapdragon 855 Performance Preview." [Online]. Available: <https://www.anandtech.com/show/13786/snapdragon-855-performance-preview/5> [p. 5.]

- [40] IBM, "IBM Archives – IBM 608 calculator." [Online]. Available: <http://www-03.ibm.com/ibm/history/exhibits/vintage/vintage{ }4506VV1003.html> [p. 5.]
- [41] Y. Nishi, "Lithium ion secondary batteries; past 10 years and the future," *Journal of Power Sources*, vol. 100, no. 1-2, pp. 101–106, nov 2001. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0378775301008874> [p. 6.]
- [42] K. Sit, P. Li, C. Ip, C. Li, L. Wan, Y. Lam, P. Lai, J. Fan, and D. Magnuson, "Studies of the energy and power of current commercial prismatic and cylindrical Li-ion cells," *Journal of Power Sources*, vol. 125, no. 1, pp. 124–134, jan 2004. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0378775303008334> [p. 6.]
- [43] L. Baggetto, R. A. H. Niessen, F. Roozeboom, and P. H. L. Notten, "High Energy Density All-Solid-State Batteries: A Challenging Concept Towards 3D Integration," *Advanced Functional Materials*, vol. 18, no. 7, pp. 1057–1066, apr 2008. [Online]. Available: <http://doi.wiley.com/10.1002/adfm.200701245> [p. 6.]
- [44] W. Brett, L. Matt, L. Brian, and S. J. P. Kristofer, "Smart Dust: Communicating with a Cubic-Millimeter Computer," *Computer*, vol. 34, no. 1, pp. 44–51, 2001. [p. 7.]
- [45] S. labs, "Sidewalk labs," 2018. [Online]. Available: <https://www.sidewalklabs.com/> [p. 7.]
- [46] R. H. Olsson, R. B. Bogoslovov, and C. Gordon, "Event driven persistent sensing: Overcoming the energy and lifetime limitations in unattended wireless sensors," in *Proceedings of IEEE Sensors*, 2017, pp. 3–5. [p. 8.]
- [47] DARPA, "Near Zero Power RF and Sensor Operations Microsystems Technology Office," DARPA, Tech. Rep., 2015. [Online]. Available: <https://govtribe.com/project/near-zero-power-rf-and-sensor-operations> [p. 8.]
- [48] Intel, "The Story of the Intel 4004," 2013. [Online]. Available: <http://www.intel.com/content/www/us/en/history/museum-story-of-intel-4004.html> [p. 14.]
- [49] K. Flamm, "Measuring Moore's Law: Evidence from Price, Cost and Quality Indexes," *National Bureau of Economic Research*, pp. 1–46, 2018. [p. 14.]
- [50] Ethan Mollick, "Establishing Moore ' s Law," *IEEE Computer Society*, pp. 62–75, 2006. [p. 14.]
- [51] B. Razavi, *Design of Analog CMOS Integrated Circuits*. McGraw-Hill Education, 2001, vol. 6, no. 7. [Online]. Available: <http://www.lavoisier.fr/notice/frLMO62SSCRRJLOO.html{% }5Cnhttp://doi.wiley.com/10.1111/j.1151-2916.1994.tb07040.x> [pp. 14, 17, and 70.]
- [52] R. J. Baker, *CMOS Circuit Design, Layout, and Simulation*. IEEE Press, 2010. [pp. 14, 25, 75, 79, and 100.]
- [53] "CPU DB - A complete database of processors for researchers and hobbyists alike." [Online]. Available: <http://cpudb.stanford.edu/manufacturers/17> [pp. 14 and 15.]
- [54] R. H. Dennard, F. H. Gaensslen, H. N. Yu, V. L. Rideout, E. Bassous, and A. R. LeBlanc, "Design for ion-implanted MOSFET's with very small physical dimensions," *IEEE Journal of Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, 1974. [p. 14.]

- [55] "International technology roadmap for semiconductors," 2005. [Online]. Available: <http://www.itrs2.net/> [p. 15.]
- [56] G. Baccarani, M. Wordeman, and R. Dennard, "Generalized Scaling Theory and Its Application," *IEEE Transactions on Electron Devices*, vol. 41, no. 4, pp. 1283–1290, 1984. [p. 15.]
- [57] "International Roadmap for Devices and Systems." [Online]. Available: <https://irds.ieee.org/> [p. 15.]
- [58] M. Riordan, L. Hoddeson, and C. Herring, "The invention of the transistor," *Reviews of Modern Physics*, vol. 71, no. 2, pp. S336–S345, mar 1999. [Online]. Available: <https://link.aps.org/doi/10.1103/RevModPhys.71.S336> [p. 16.]
- [59] M. Riordan, L. Hoddeson, and P. Platzman, "Crystal Fire: The Birth of the Information Age," *American Journal of Physics*, vol. 67, no. 7, pp. 648–650, jul 1999. [Online]. Available: <http://aapt.scitation.org/doi/10.1119/1.19345> [p. 16.]
- [60] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proceedings of the IEEE*, vol. 91, no. 2, pp. 305–327, 2003. [pp. 17, 18, and 20.]
- [61] W. M. Sansen, *Analog Design Essentials*, ser. The International Series in Engineering and Computer Science. Boston, MA: Springer US, 2006. [Online]. Available: <http://link.springer.com/10.1007/b135984> [p. 18.]
- [62] T. Sakurai and A. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, apr 1990. [Online]. Available: <http://ieeexplore.ieee.org/document/52187/> [p. 18.]
- [63] S. Hanson, B. Zhai, D. Blaauw, D. Sylvester, A. Bryant, and X. Wang, "Energy optimality and variability in subthreshold design," *Proceedings of the 2006 international symposium on Low power electronics and design - ISLPED '06*, p. 363, 2006. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1165573.1165660> [pp. 18, 19, and 29.]
- [64] C. C. Enz, F. Krummenacher, and E. A. Vittoz, "An analytical MOS transistor model valid in all regions of operation and dedicated to low-voltage and low-current applications," *Analog Integrated Circuits and Signal Processing*, vol. 8, no. 1, pp. 83–114, 1995. [p. 18.]
- [65] B. Pelloux-prayer and B. Pelloux-prayer, "Optimisation de l'efficacité énergétique des applications numériques en technologie FD-SOI 28-14nm," Ph.D. dissertation, Grenoble Université, 2014. [p. 19.]
- [66] N. Reynders and W. Dehaene, *Ultra-Low-Voltage Design of Energy-Efficient Digital Circuits*. Springer International Publishing, 2015. [pp. 19, 24, 26, 29, and 104.]
- [67] Neil H. E. Weste; David Money Harris, *CMOS VLSI Design - A Circuits and Systems Perspective*. Addison-Wesley, 2011. [p. 19.]
- [68] V. G. Oklobdzija, *Digital Design and Fabrication*. CRC Press, nov 2007. [Online]. Available: <https://www.taylorfrancis.com/books/9781315222226> [pp. 20 and 24.]

- [69] M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi, *Low power methodology manual: For system-on-chip design*. Springer International Publishing, 2007. [pp. 20, 30, 64, and 115.]
- [70] W. Zhao, Y. Ha, and M. Alioto, "Novel Self-Body-Biasing and Statistical Design for Near-Threshold Circuits with Ultra Energy-Efficient AES as Case Study," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 8, pp. 1390–1401, 2015. [p. 20.]
- [71] M. Cochet, "Efficiency Optimization in 28 nm FD-SOI: Circuit Design for Adaptive Clocking and Power-Temperature Aware Digital SoCs," Ph.D. dissertation, Aix-Marseille Université, 2016. [pp. 20, 40, 52, and 65.]
- [72] Y. Tsividis, *Operation and Modeling of the MOS Transistor (The Oxford Series in Electrical and Computer Engineering)*. McGraw-Hill Education, 1999. [pp. 21 and 34.]
- [73] X. Huang, S. Member, W.-c. Lee, C. Kuo, D. Hisamoto, L. Chang, J. Kedzierski, E. Anderson, H. Takeuchi, Y.-k. Choi, K. Asano, V. Subramanian, T.-j. King, J. Bokor, and C. Hu, "Sub-50 nm P-channel FinFET," *IEEE Transactions on Electron Devices*, vol. 48, no. 5, pp. 880–886, 2001. [Online]. Available: <http://ieeexplore.ieee.org/xpls/abs{ }all.jsp?arnumber=918235> [p. 21.]
- [74] M. LaPedus, "Semiconductor Engineering." [Online]. Available: <https://semiengineering.com/7nm-fab-challenges/> [p. 21.]
- [75] N. Singh, A. Agarwal, L. Bera, T. Liow, R. Yang, S. Rustagi, C. Tung, R. Kumar, G. Lo, N. Balasubramanian, and D.-L. Kwong, "High-performance fully depleted silicon nanowire (diameter /spl les/ 5 nm) gate-all-around CMOS devices," *IEEE Electron Device Letters*, vol. 27, no. 5, pp. 383–386, may 2006. [Online]. Available: <http://ieeexplore.ieee.org/document/1626464/> [p. 21.]
- [76] E. Dastjerdy, R. Ghayour, and H. Sarvari, "Simulation and analysis of the frequency performance of a new silicon nanowire MOSFET structure," *Physica E: Low-dimensional Systems and Nanostructures*, vol. 45, pp. 66–71, aug 2012. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1386947712002718> [p. 21.]
- [77] J. Hartmann, "FD-SOI technology development and key devices characteristics for fast, power efficient, low voltage SoCs," in *Technical Digest - IEEE Compound Semiconductor Integrated Circuit Symposium, CSIC*, 2014. [p. 22.]
- [78] E. G. Ioannidis, S. Haendler, A. Bajolet, T. Pahrton, N. Planes, F. Arnaud, R. A. Bianchi, M. Haond, D. Golanski, J. Rosa, C. Fenouillet-Beranger, P. Perreau, C. A. Dimitriadis, and G. Ghibaudo, "Low frequency noise variability in high-k/metal gate stack 28nm bulk and FD-SOI CMOS transistors," *International Electron Devices Meeting, IEDM*, pp. 449–452, 2011. [pp. 22 and 89.]
- [79] "12FDX - GLOBALFOUNDRIES." [Online]. Available: <https://www.globalfoundries.com/technology-solutions/cmos/fox/12fox> [p. 22.]
- [80] A. Cathelin, "Fully Depleted Silicon on Insulator Devices CMOS," *Sscm*, no. November, pp. 18–26, 2017. [pp. 22 and 99.]

- [81] F. Abouzeid, S. Clerc, B. Pelloux-Prayer, F. Argoud, and P. Roche, "28nm CMOS, energy efficient and variability tolerant, 350mV-to-1.0V, 10MHz/700MHz, 252bits frame error-decoder," *European Solid-State Circuits Conference*, pp. 153–156, 2012. [pp. 22, 29, and 98.]
- [82] D. Jacquet, F. Hasbani, P. Flatresse, R. Wilson, F. Arnaud, G. Cesana, T. Di Gilio, C. Lecocq, T. Roy, A. Chhabra, C. Grover, O. Minez, J. Uginet, G. Durieu, C. Adobati, D. Casalotto, F. Nyer, P. Menut, A. Cathelin, I. Vongsavady, and P. Magarshack, "A 3 GHz Dual Core Processor ARM Cortex TM -A9 in 28 nm UTBB FD-SOI CMOS With Ultra-Wide Voltage Range and Energy Efficiency Optimization," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 4, pp. 812–826, apr 2014. [Online]. Available: <http://ieeexplore.ieee.org/document/6725645/> [p. 22.]
- [83] M. Khater, J. Cai, R. H. Dennard, J. Yau, C. Wang, L. Shi, M. Guillorn, J. Ott, Q. Ouyang, and W. Haensch, "FDSOI CMOS with dielectrically-isolated back gates and 30nm Lghigh- κ /metal gate," *Digest of Technical Papers - Symposium on VLSI Technology*, no. 914, pp. 43–44, 2010. [pp. 22 and 126.]
- [84] D. Harris, R. Ho, G.-y. Wei, and M. Horowitz, "The Fanout-of-4 Inverter Delay Metric," *Vlsi*, pp. 4–5, 1997. [p. 25.]
- [85] M. Alioto, E. Consoli, and G. Palumbo, "From energy-delay metrics to constraints on the design of digital circuits," *International Journal of Circuit Theory and Applications*, vol. 40, no. 8, pp. 815–834, aug 2012. [Online]. Available: <http://doi.wiley.com/10.1002/cta.757> [p. 25.]
- [86] Y. Pu, J. d. j. P. de Gyvez, H. Corporaal, and Y. Ha, "V t balancing and device sizing towards high yield of sub-threshold static logic gates," in *Proceedings of the 2007 international symposium on Low power electronics and design - ISLPED '07*. New York, New York, USA: ACM Press, 2007, pp. 355–358. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1283780.1283857> [p. 25.]
- [87] M. Alioto, "Impact of NMOS/PMOS imbalance in ultra-low voltage CMOS standard cells," *2011 20th European Conference on Circuit Theory and Design, ECCTD 2011*, pp. 536–539, 2011. [p. 25.]
- [88] D. Bol, "Robust and energy-efficient ultra-low-voltage circuit design under timing constraints in 65/45 nm CMOS," *Journal of Low Power Electronics and Applications*, vol. 1, no. 1, pp. 1–19, 2011. [p. 25.]
- [89] S. Narendra, S. Borkar, V. De, D. Antoniadis, and A. Chandrakasan, "Scaling of stack effect and its application for leakage reduction," in *ISLPED'01: Proceedings of the 2001 International Symposium on Low Power Electronics and Design (IEEE Cat. No.01TH8581)*. ACM, 2001, pp. 195–200. [Online]. Available: <http://ieeexplore.ieee.org/document/945400/> [p. 25.]
- [90] D. Bol, C. Hocquet, D. Flandre, and J. D. Legat, "Robustness-aware sleep transistor engineering for power-gated nanometer subthreshold circuits," *ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems*, pp. 1484–1487, 2010. [pp. 25 and 35.]

- [91] M. Mittal and A. P. S. Rathod, "Digital circuit optimization using Pass Transistor Logic architectures," in *2016 International Conference on Emerging Trends in Communication Technologies (ETCT)*. IEEE, nov 2016, pp. 1–5. [Online]. Available: <http://ieeexplore.ieee.org/document/7882922/> [p. 26.]
- [92] A. Tajalli, E. J. Brauer, Y. Leblebici, and E. Vittoz, "Subthreshold Source-Coupled Logic Circuits for Ultra-Low-Power Applications," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 7, pp. 1699–1710, jul 2008. [Online]. Available: <http://ieeexplore.ieee.org/document/4550646/> [p. 26.]
- [93] D. Maksimovic, V. Oklobdzija, B. Nikolic, and K. Current, "Clocked CMOS adiabatic logic with integrated single-phase power-clock supply," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 8, no. 4, pp. 460–463, aug 2000. [Online]. Available: <http://ieeexplore.ieee.org/document/863629/> [p. 26.]
- [94] H. Hassan, M. Anis, and M. Elmasry, "MOS current mode logic: design, optimization, and variability," in *IEEE International SOC Conference, 2004. Proceedings*. IEEE, 2004, pp. 247–250. [Online]. Available: <http://ieeexplore.ieee.org/document/1362424/> [p. 26.]
- [95] E. Graniello, B. Chavan, A., Rodriguez, B., MacDonald, "Optimized circuit styles for sub-threshold logic," in *IEEE International Conference on Micro Electro Mechanical Systems*, 2005. [p. 26.]
- [96] H. Soeleman and K. Roy, "Ultra-low power digital subthreshold logic circuits," *Proceedings of the International Symposium on Low Power Electronics and Design, Digest of Technical Papers*, pp. 94–96, 1999. [p. 26.]
- [97] H. Reyserhove and W. Dehaene, "A Differential Transmission Gate Design Flow for Minimum Energy Sub-10-pJ/Cycle ARM Cortex-M0 MCUs," *IEEE Journal of Solid-State Circuits*, pp. 1–11, 2017. [pp. 26, 43, and 46.]
- [98] M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi, *Low Power Methodology Manual*. Boston, MA: Springer US, 2007. [Online]. Available: <http://link.springer.com/10.1007/978-0-387-71819-4> [p. 26.]
- [99] J. Rabaey, *Low Power Design Essentials*, ser. Integrated Circuits and Systems. Boston, MA: Springer US, 2009. [Online]. Available: <http://link.springer.com/10.1007/978-0-387-71713-5> [p. 27.]
- [100] D. Bol, R. Ambrose, D. Flandre, and J. D. Legat, "Interests and limitations of technology scaling for subthreshold logic," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 10, pp. 1508–1519, 2009. [p. 29.]
- [101] S. Jain, S. Khare, S. Yada, V. Ambili, P. Salihundam, S. Ramani, S. Muthukumar, M. Srinivasan, A. Kumar, S. K. Gb, R. Ramanarayanan, V. Erraguntla, J. Howard, S. Vangal, S. Dighe, G. Ruhl, P. Aseron, H. Wilson, N. Borkar, V. De, and S. Borkar, "A 280mV-to-1.2V wide-operating-range IA-32 processor in 32nm CMOS," *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, vol. 55, pp. 66–67, 2012. [p. 29.]
- [102] S. Khoshbakht and N. Dimopoulos, "Relating application memory activity to processor power," *Proceedings of the International Conference on Parallel Processing*, pp. 849–857, 2013. [p. 29.]

- [103] F. Rahman, R. Pamula, A. Boora, X. Sun, and V. Sathe, "Computationally Enabled Total Energy Minimization Under Performance Requirements for a Voltage-," *2019 IEEE International Solid-State Circuits Conference - (ISSCC)*, vol. 45, no. 12, pp. 312–314, 2019. [pp. 30 and 206.]
- [104] F. Assaderaghi, D. Sinitsky, S. Parke, J. Bokor, P. Ko, and C. H. C. Hu, "A dynamic threshold voltage MOSFET (DTMOS) for ultra-low voltage operation," *Proceedings of 1994 IEEE International Electron Devices Meeting*, vol. 15, no. 12, pp. 809–812, 1994. [pp. 30 and 105.]
- [105] S. Narendra, J. Tschanz, J. Hofsheier, B. Bloechel, S. Vangal, Y. Hoskote, S. Tang, D. Somasekhar, A. Keshavarzi, V. Erraguntla, G. Dermer, N. Borkar, S. Borkar, and V. De, "Ultra-low voltage circuits and processor in 180nm to 90nm technologies with a swapped-body biasing technique," *2004 IEEE International Solid-State Circuits Conference (IEEE Cat. No.04CH37519)*, pp. 156–518, 2004. [Online]. Available: <http://ieeexplore.ieee.org/document/1332641/> [pp. 31 and 121.]
- [106] M. J. Pelgrom and A. C. Duinmaijer, "Matching properties of MOS transistors," *Solid-State Circuits Conference, 1988. ESSCIRC '88. Fourteenth European*, vol. 24, no. 5, pp. 1433–1440, 1988. [p. 32.]
- [107] J. H. Kim and Y. H. Kim, "Effect of gate-level design margin relaxation on overall circuit performance metrics in VLSI design," in *2nd Asia Symposium on Quality Electronic Design (ASQED)*. IEEE, aug 2010, pp. 314–317. [Online]. Available: <http://ieeexplore.ieee.org/document/5548316/> [p. 33.]
- [108] S. Das, C. Tokunaga, S. Pant, W.-H. Ma, S. Kalaiselvan, K. Lai, D. M. Bull, and D. T. Blaauw, "RazorII: In Situ Error Detection and Correction for PVT and SER Tolerance," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 32–48, jan 2009. [Online]. Available: <http://ieeexplore.ieee.org/document/4735568/> [p. 33.]
- [109] D. Bol, J. D. Vos, S. Member, C. Hocquet, F. Botman, S. Member, F. Durvaux, S. Boyd, D. Flandre, and S. Member, "Microcontroller in 65-nm LP / GP CMOS for Low-Carbon Wireless Sensor Nodes," *JSSC*, vol. 48, no. 1, pp. 1–13, 2013. [pp. 39, 43, and 45.]
- [110] H. Reyserhove and W. Dehaene, "Margin Elimination Through Timing Error Detection in a Near-Threshold Enabled 32-bit Microcontroller in 40-nm CMOS," pp. 1–13, 2018. [pp. 33 and 206.]
- [111] Qing Wu, M. Pedram, and Xunwei Wu, "Clock-gating and its application to low power design of sequential circuits," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 3, pp. 415–420, 2002. [p. 33.]
- [112] E. Friedman, "Clock distribution design in VLSI circuits-An overview," in *1993 IEEE International Symposium on Circuits and Systems*. IEEE, 1993, pp. 1475–1478. [Online]. Available: <http://ieeexplore.ieee.org/document/394013/> [p. 33.]
- [113] S. Inc., "Power Compiler, White Paper," 2012. [Online]. Available: <https://www.synopsys.com/content/dam/synopsys/implementation{&}signoff/datasheets/power-compiler-ds.pdf> [pp. 33 and 34.]
- [114] S. Mutoh, S. Shigematsu, Y. Gotoh, and S. Konaka, "Design method of MTCMOS power switch for low-voltage high-speed LSIs," in *Proceedings of the ASP-DAC '99 Asia and*

- South Pacific Design Automation Conference 1999 (Cat. No.99EX198)*. IEEE, 1999, pp. 113–116 vol.1. [Online]. Available: <http://ieeexplore.ieee.org/document/759726/> [p. 34.]
- [115] M. De Nil, L. Yseboodt, F. Bouwens, J. Hulzink, M. Berekovic, J. Huisken, and J. Van Meerbergen, “Ultra low power ASIP design for wireless sensor nodes,” *Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems*, pp. 1352–1355, 2007. [p. 34.]
- [116] S. Kaijian and D. Howard, “Sleep transistor design and implementation - Simple concepts yet challenges to be optimum,” *2006 International Symposium on VLSI Design, Automation and Test, VLSI-DAT 2006 - Proceedings of Technical Papers*, pp. 51–54, 2007. [p. 34.]
- [117] M. R. Stan, “Low threshold CMOS circuits with low standby current,” in *Proceedings of the 1998 international symposium on Low power electronics and design - ISLPED '98*, no. 100. New York, New York, USA: ACM Press, 1998, pp. 97–99. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=280756.280807> [p. 35.]
- [118] M. K. James and B. K. Mathew, “An area-efficient asynchronous FPGA based on fine-grain power gating and time multiplexed dual rail encoding,” in *2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)*. IEEE, mar 2013, pp. 791–795. [Online]. Available: <http://ieeexplore.ieee.org/document/6526514/> [p. 35.]
- [119] M. Arora, S. Manne, I. Paul, N. Jayasena, and D. M. Tullsen, “Understanding idle behavior and power gating mechanisms in the context of modern benchmarks on CPU-GPU Integrated systems,” *2015 IEEE 21st International Symposium on High Performance Computer Architecture, HPCA 2015*, pp. 366–377, 2015. [p. 35.]
- [120] S. M. Nowick and M. Singh, “Asynchronous DesignPart 1: Overview and Recent Advances,” *IEEE Design & Test*, vol. 32, no. 3, pp. 5–18, jun 2015. [Online]. Available: <http://ieeexplore.ieee.org/document/7061401/> [p. 35.]
- [121] Z. Tabassam, S. R. Naqvi, T. Akram, M. Alhussein, K. Aurangzeb, and S. A. Haider, “Towards Designing Asynchronous Microprocessors: From Specification to Tape-Out,” *IEEE Access*, vol. 7, pp. 33 978–34 003, 2019. [Online]. Available: <https://ieeexplore.ieee.org/document/8660629/> [p. 35.]
- [122] J. C. Myers, “Muller Circuits,” in *Asynchronous Circuit Design*. New York, USA: John Wiley & Sons, Inc., 2001, pp. 207–258. [Online]. Available: <http://doi.wiley.com/10.1002/0471224146.ch6> [p. 35.]
- [123] D. E. Bellasi and L. Benini, “Smart Energy-Efficient Clock Synthesizer for Duty-Cycled Sensor SoCs in 65 nm/28nm CMOS,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 9, pp. 2322–2333, sep 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/7934447/> [p. 36.]
- [124] V. Pop, H. J. Bergveld, P. H. Notten, and P. P. Regtien, “State-of-the-art of battery state-of-charge determination,” 2005. [p. 36.]
- [125] STMicroelectronics, “BlueNRG-2 - Bluetooth® low energy wireless system-on-chip - STMicroelectronics.” [Online]. Available: <https://wwhttps://www.st.com/en/wireless-transceivers-mcus-and-modules/bluenrg-2.htmlw.st.com/en/wireless-transceivers-mcus-and-modules/bluenrg-2.html> [p. 37.]

- [126] Nordic, “nRF51822 Technical Specifications,” 2014. [Online]. Available: <https://www.nordicsemi.com/Products/Low-power-short-range-wireless/nRF51822> [p. 37.]
- [127] T. Haine, D. Flandre, and D. Bol, “8-T ULV SRAM macro in 28nm FDSOI with 7.4 pW/bit retention power and back-biased-scalable speed/energy trade-off,” in *2018 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*. IEEE, oct 2018, pp. 1–3. [Online]. Available: <https://ieeexplore.ieee.org/document/8640170/> [p. 38.]
- [128] J.-L. A. Martin Cochet, Sylvain Clerc, Fady Abouzeid, Guénolé Lallement, Philippe Roche, “A 0.40 pJ / cycle 981 μ m² Voltage Scalable Digital Frequency Generator for SoC Clocking,” in *ASSCC*, 2018. [p. 38.]
- [129] P. P. Mercier and A. P. Chandrakasan, *Ultra-Low- Power Short- Range Radios*. Springer International Publishing, 2015. [p. 38.]
- [130] T. Jang, G. Kim, B. Kempke, M. B. Henry, N. Chiotellis, C. Pfeiffer, D. Kim, Y. Kim, Z. Foo, H. Kim, A. Grbic, D. Sylvester, H. S. Kim, D. D. Wentzloff, and D. Blaauw, “Circuit and System Designs of Ultra-Low Power Sensor Nodes With Illustration in a Miniaturized GNSS Logger for Position Tracking: Part I - Analog Circuit Techniques,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 9, pp. 2237–2249, 2017. [pp. 38, 64, and 89.]
- [131] Douseki and Kyuragi, “Ultralow-voltage MTCMOS/SOI circuits for batteryless wireless system,” in *2003 IEEE International Conference on Robotics and Automation (Cat No 03CH37422) SOI-03*. IEEE, 2003, pp. 5–8. [Online]. Available: <http://ieeexplore.ieee.org/document/1242877/> [p. 38.]
- [132] T. Burd, T. Pering, A. Stratakos, and R. Brodersen, “A dynamic voltage scaled microprocessor system,” *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1571–1580, nov 2000. [Online]. Available: <http://ieeexplore.ieee.org/document/881202/> [p. 38.]
- [133] I. Miro-Panades, E. Beigné, Y. Thonnart, L. Alacoque, P. Vivet, S. Lesecq, D. Puschini, A. Molnos, F. Thabet, B. Tain, K. Ben Chehida, S. Engels, R. Wilson, and D. Fuin, “A fine-grain variation-aware dynamic Vdd-Hopping AVFS architecture on a 32 nm GALS MP-SoC,” *IEEE Journal of Solid-State Circuits*, vol. 49, no. 7, pp. 1475–1486, 2014. [p. 39.]
- [134] C. T. Muller, M. Pons, D. Ruffieux, J.-L. Nagel, S. Emery, A. Burg, S. Tanahashi, Y. Tanaka, and A. Takeuchi, “Minimum Energy Point in Constant Frequency Designs under Adaptive Supply Voltage and Body Bias Adjustment in 55 nm DDC,” in *2019 15th Conference on Ph.D Research in Microelectronics and Electronics (PRIME)*. IEEE, jul 2019, pp. 285–288. [Online]. Available: <https://ieeexplore.ieee.org/document/8787736/> [p. 39.]
- [135] J. Tschanz, N. S. Kim, S. Dighe, J. Howard, G. Ruhl, S. Vangal, S. Narendra, Y. Hoskote, H. Wilson, C. Lam, M. Shuman, C. Tokunaga, D. Somasekhar, S. Tang, D. Finan, T. Karnik, N. Borkar, N. Kurd, and V. De, “Adaptive frequency and biasing techniques for tolerance to dynamic temperature-voltage variations and aging,” *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, pp. 292–294, 2007. [p. 39.]

- [136] S. Clerc, M. Saligane, F. Abouzeid, M. Cochet, J. M. Daveau, C. Bottoni, D. Bol, J. De-Vos, D. Zamora, B. Coeffic, D. Soussan, D. Croain, M. Naceur, P. Schamberger, P. Roche, and D. Sylvester, "A 0.33V/-40C process/temperature closed-loop compensation SoC embedding all-digital clock multiplier and DC-DC converter exploiting FDSOI 28nm back-gate biasing," *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, vol. 58, pp. 150–151, 2015. [pp. 39, 43, 45, and 121.]
- [137] J. S. Wang, J. S. Chen, Y. M. Wang, and C. Yeh, "A 230mV-to-500mV 375KHz-to-16MHz 32b RISC core in 0.18 μ m CMOS," in *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, 2007, pp. 192–193. [pp. 39, 45, 121, and 122.]
- [138] L. Benini, P. Glynn, G. D. Micheli, and T. Simunic, "Event-Driven Power Management," *IEEE TRANSACTIONS ON COMPUTER AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS*, vol. 20, no. 7, pp. 840–857, 2001. [pp. 39, 158, and 159.]
- [139] B. E. N. Keller, B. Nikoli, and K. Asanovi, "Raven : Fine-Grained Adaptive Voltage Scaling in 28nm Processor SoCs," BWRC, Tech. Rep., 2018. [pp. 39 and 114.]
- [140] G. Tagliavini, D. Rossi, A. Marongiu, and L. Benini, "Synergistic HW/SW Approximation Techniques for Ultralow-Power Parallel Computing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 5, pp. 982–995, 2018. [pp. 40 and 158.]
- [141] N. Ickes, Y. Sinangil, F. Pappalardo, E. Guidetti, and A. P. Chandrakasan, "A 10 pJ/cycle ultra-low-voltage 32-bit microprocessor system-on-chip," *European Solid-State Circuits Conference*, pp. 159–162, 2011. [pp. 43 and 45.]
- [142] J. Myers, A. Savanth, R. Gaddh, D. Howard, P. Prabhat, and D. Flynn, "A subthreshold ARM cortex-M0+ subsystem in 65 nm CMOS for WSN applications with 14 Power Domains, 10T SRAM, and integrated voltage regulator," *IEEE Journal of Solid-State Circuits*, pp. 31–44, 2016. [p. 46.]
- [143] S. Paul, V. Honkote, R. G. Kim, T. Majumder, P. A. Aseron, V. Grossnickle, R. Sankman, D. Mallik, T. Wang, S. Vangal, J. W. Tschanz, and V. De, "A Sub-cm³ Energy-Harvesting Stacked Wireless Sensor Node Featuring a Near-Threshold Voltage IA-32 Microcontroller in 14-nm Tri-Gate CMOS for Always-ON Always-Sensing Applications," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 4, pp. 961–971, 2017. [pp. 43 and 46.]
- [144] J. Myers, A. Savanth, P. Prabhat, S. Yang, R. Gaddh, S. O. Toh, and D. Flynn, "A 12.4pJ/cycle sub-threshold, 16pJ/cycle near-threshold ARM Cortex-M0+ MCU with autonomous SRPG/DVFS and temperature tracking clocks," *IEEE Symposium on VLSI Circuits, Digest of Technical Papers*, pp. C332–C333, 2017. [pp. 43 and 46.]
- [145] L. Lin, S. Jain, and M. Alioto, "A 595pW 14pJ/Cycle microcontroller with dual-mode standard cells and self-startup for battery-indifferent distributed sensing," *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, vol. 61, pp. 44–46, 2018. [pp. 43 and 46.]
- [146] W. Lim, I. Lee, D. Sylvester, and D. Blaauw, "Batteryless Sub-nW Cortex-M0+ Processor with Dynamic Leakage-Suppression Logic," *2015 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 146–148, 2015. [pp. 43 and 46.]

- [147] D. Bol, M. Schramme, L. Moreau, T. Haine, P. Xu, C. Frenkel, R. Dekimpe, F. Stas, and D. Flandre, "19.6 A 40-to-80MHz Sub-4 μ W/MHz ULV Cortex-M0 MCU SoC in 28nm FD-SOI with Dual-Loop Adaptive Back-Bias Generator for 20 μ s Wake-Up from Deep Fully Retentive Sleep Mode," *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, vol. 2019-Febru, pp. 322–324, 2019. [pp. 43 and 46.]
- [148] J. Myers, A. Savanth, D. Howard, R. Gaddh, P. Prabhat, and D. Flynn, "An 80nW retention 11.7pJ/cycle active subthreshold ARM Cortex-M0+ subsystem in 65nm CMOS for WSN applications," *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, vol. 58, pp. 144–145, 2015. [p. 43.]
- [149] H. Reyserhove and W. Dehaene, "A Differential Transmission Gate Design Flow for Minimum Energy Sub-10-pJ/Cycle ARM Cortex-M0 MCUs," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 7, pp. 0–30, 2017. [p. 43.]
- [150] S. Hanson, S. Member, M. Seok, Y.-s. Lin, Z. Foo, D. Kim, Y. Lee, N. Liu, D. Sylvester, S. Member, and D. Blaauw, "A Low-Voltage Processor for Sensing Applications With Picowatt Standby Mode," *IEEE Journal of Solid State Circuits*, vol. 44, no. 4, pp. 1145–1155, 2009. [p. 45.]
- [151] J.-s. Chen, C. Yeh, and J.-s. Wang, "Self-Super-Cutoff Power Gating with State Retention," *Isscc*, vol. 3, pp. 426–427, 2013. [p. 45.]
- [152] M. Fojtik, D. Kim, G. Chen, Y. S. Lin, D. Fick, J. Park, M. Seok, M. T. Chen, Z. Foo, D. Blaauw, and D. Sylvester, "A millimeter-scale energy-autonomous sensor system with stacked battery and solar cells," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 3, pp. 801–813, 2013. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6416956> [p. 45.]
- [153] X. Liu, M. Zhang, T. Xiong, A. G. Richardson, T. H. Lucas, P. S. Chin, R. Etienne-Cummings, T. D. Tran, and J. Van Der Spiegel, "A Fully Integrated Wireless Compressed Sensing Neural Signal Acquisition System for Chronic Recording and Brain Machine Interface," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 4, pp. 874–883, 2016. [pp. 50 and 220.]
- [154] J. A. Stankovic, "Research directions for the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, 2014. [Online]. Available: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84906842496&partnerID=40&md5=7312ed48ff4b81ea7b84c955d293c202> [p. 50.]
- [155] N. Shafiee, S. Tewari, B. Calhoun, and A. Shrivastava, "Infrastructure Circuits for Lifetime Improvement of Ultra-Low Power IoT Devices," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 9, pp. 2598–2610, 2017. [p. 50.]
- [156] M. Magno, S. Marinkovic, D. Brunelli, E. Popovici, B. O'Flynn, and L. Benini, "Smart power unit with ultra low power radio trigger capabilities for wireless sensor networks," *Proceedings -Design, Automation and Test in Europe, DATE*, pp. 75–80, 2012. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-0-84862113527&partnerID=40&md5=4389b385d5e457a58e003580555d4196> [p. 50.]
- [157] M. Pizzotti, L. Perilli, M. del Prete, D. Fabbri, R. Canegallo, M. Dini, D. Masotti, A. Costanzo, E. F. Scarselli, and A. Romani, "A long-distance RF-powered sensor node

- with adaptive power management for IoT applications,” *Sensors (Switzerland)*, vol. 17, no. 8, 2017. [p. 50.]
- [158] F. Yuan, *Low Power Circuits for emerging applications in communications, computing, and sensing*. CRC Press, 2018. [pp. 51, 211, and 217.]
- [159] Y. Lee, Y. Kim, D. Yoon, D. Blaauw, and D. Sylvester, “Circuit and system design guidelines for ultra-low power sensor nodes,” *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*, pp. 1037–1042, 2012. [pp. 52, 64, and 66.]
- [160] C. Salazar, A. Kaiser, A. Cathelin, and J. Rabaey, “A -97dBm-sensitivity interferer-resilient 2.4GHz wake-up receiver using dual-IF multi-N-Path architecture in 65nm CMOS,” in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*. IEEE, feb 2015, pp. 1–3. [Online]. Available: <http://ieeexplore.ieee.org/document/7063016/> [p. 52.]
- [161] J. Yiu, *The Definitive Guide to ARM® Cortex®-M0 and Cortex-M0+ Processors, 2nd Edition*. Academic Press, 2015. [pp. 56, 110, 116, 211, 212, and 215.]
- [162] H. Wang and P. P. Mercier, “A Reference-Free Capacitive-Discharging Oscillator Architecture Consuming 44.4 pW/75.6 nW at 2.8 Hz/6.4 kHz,” *IEEE Journal of Solid-State Circuits*, vol. 51, no. 6, pp. 1423–1435, 2016. [p. 64.]
- [163] D. W. Allan, N. Ashby, and C. C. Hodge, “The science of timekeeping,” *Hewlett Packard application note 1289*, pp. 1–88, 1997. [Online]. Available: <http://www.allanstime.com/Publications/DWA/Science{ }Timekeeping/TheScienceOfTimekeeping.pdf{%}5Cnhttp://literature.agilent.com/litweb/pdf/5965-7984E.pdf> [pp. 64 and 65.]
- [164] R. Duggirala, A. Lal, and S. Radhakrishnan, “Radioisotope thin-film powered microsystems,” *MEMS Reference shelf*, vol. 6, p. 198, 2010. [p. 64.]
- [165] V. Candelier, P. Canzian, J. Lamboley, M. Brunet, and G. Santarelli, “Space qualified 5MHz ultra stable oscillators,” *Ifcs*, pp. 575–582, 2003. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1275155> [p. 65.]
- [166] F. V. Supply, A. Shrivastava, D. A. Kamakshi, S. Member, B. H. Calhoun, and S. Member, “A 1.5 nW, 32.768 kHz XTAL Oscillator Operational from a 0.3V supply,” *IEEE Journal of Solid State Circuits*, vol. 51, no. 3, pp. 686–696, 2016. [pp. 65 and 220.]
- [167] ST Microelectronics, “Application note Oscillator design guide for STM8S , STM8A,” pp. 1–24, 2015. [p. 65.]
- [168] C. S. Lam, “A review of the recent development of mems and crystal oscillators and their impacts on the frequency control products industry,” *Proceedings - IEEE Ultrasonics Symposium*, pp. 694–704, 2008. [p. 65.]
- [169] C. Y. Liu, M. H. Li, C. Y. Chen, and S. S. Li, “An ovenized CMOS-MEMS oscillator with isothermal resonator and sub-mW heating power,” *2016 IEEE International Frequency Control Symposium, IFCS 2016 - Proceedings*, pp. 4–6, 2016. [p. 65.]
- [170] H. Lee, A. Partridge, and F. Assaderaghi, “Low jitter and temperature stable MEMS oscillators,” *2012 IEEE International Frequency Control Symposium, IFCS 2012, Proceedings*, pp. 266–270, 2012. [pp. 65 and 220.]

- [171] D. Kamakshi, A. Shrivastava, C. Duan, and B. Calhoun, "A 36 nW, 7 ppm/°C on-Chip Clock Source Platform for Near-Human-Body Temperature Applications," *Journal of Low Power Electronics and Applications*, vol. 6, no. 2, p. 7, 2016. [Online]. Available: <http://www.mdpi.com/2079-9268/6/2/7> [pp. 67, 69, 79, 93, and 94.]
- [172] A. Rusznyak, "Start-up time of CMOS oscillators," *IEEE Transactions on Circuits and Systems*, vol. 34, no. 3, pp. 259–268, mar 1987. [Online]. Available: <http://ieeexplore.ieee.org/document/1086137/> [p. 68.]
- [173] F. Yuan, *CMOS Time-Mode Circuits and Systems*. CRC Press, 2015. [Online]. Available: <https://www.taylorfrancis.com/books/9781315214375> [p. 69.]
- [174] A. I. Karsilayan and R. Schaumann, "A high-frequency high-Q CMOS active inductor with DC bias control," *Midwest Symposium on Circuits and Systems*, vol. 1, pp. 486–489, 2000. [p. 69.]
- [175] D. DiClemente and F. Yuan, "A passive transformer voltage-controlled oscillator with active inductor frequency tuning for ultra wideband applications," *Microsystems and Nanoelectronics ...*, no. 3, pp. 80–83, 2009. [Online]. Available: <http://ieeexplore.ieee.org/xpls/abs/all.jsp?arnumber=5338956> [p. 69.]
- [176] R. H. Najmeh Cheraghi Shirazi, Ebrahim Abiri, "A 5 . 5 GHz Voltage Control Oscillator (VCO) with a Differential Tunable Active and passive Inductor," *International Journal of Information and Electronics Engineering*, vol. 3, no. 1c, pp. 43–47, 2010. [p. 69.]
- [177] S. Jeong, I. Lee, D. Blaauw, and D. Sylvester, "A 5.8nW, 45ppm/°C on-chip CMOS wake-up timer using a constant charge subtraction scheme," *Proceedings of the IEEE 2014 Custom Integrated Circuits Conference, CICC 2014*, pp. 3–6, 2014. [p. 69.]
- [178] T. Jang, M. Choi, S. Jeong, S. Bang, D. Sylvester, and D. Blaauw, "A 4.7nW 13.8ppm/°C self-biased wakeup timer using a switched-resistor scheme," *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, vol. 59, pp. 102–103, 2016. [pp. 69 and 220.]
- [179] A. Paidimarri, D. Griffith, A. Wang, A. P. Chandrakasan, and G. Burra, "A 120nW 18.5kHz RC oscillator with comparator offset cancellation for ±0.25% temperature stability," *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, vol. 56, pp. 184–185, 2013. [pp. 69 and 220.]
- [180] A. Paidimarri, D. Griffith, A. Wang, S. Member, G. Burra, A. P. Chandrakasan, and A. A. R., "An RC Oscillator With Comparator Offset Cancellation," *IEEE Journal of Solid State Circuits*, vol. 51, no. 8, pp. 1–12, 2016. [pp. 69 and 220.]
- [181] F. Herzel and B. Razavi, "A study of oscillator jitter due to supply and substrate noise," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 1, pp. 56–62, 1999. [p. 69.]
- [182] V. Kratyuk, I. Vityaz, U. K. Moon, and K. Mayaram, "Analysis of supply and ground noise sensitivity in ring and LC oscillators," *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 5986–5989, 2005. [Not cited.]
- [183] A. A. Abidi, "Phase noise and jitter in CMOS ring oscillators," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 8, pp. 1803–1816, 2006. [p. 69.]

- [184] A. Shrivastava and B. H. Calhoun, "A 150nW, 5ppm/C, 100kHz on-chip clock source for ultra low power SoCs," *Proceedings of the Custom Integrated Circuits Conference*, pp. 12–15, 2012. [pp. 69, 70, and 79.]
- [185] D. W. Jee, D. Sylvester, D. Blaauw, and J. Y. Sim, "Digitally controlled leakage-based oscillator and fast relocking MDLL for ultra low power sensor platform," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 5, pp. 1263–1274, 2015. [pp. 69, 71, and 72.]
- [186] B. Consortium, "Bluetooth Core Specification v5.1," *Specification of the Bluetooth*, 2003. [Online]. Available: <http://www.bluetooth.com/> [p. 73.]
- [187] A. Cros, F. Monsieur, Y. Carminati, P. Normandon, D. Petit, F. Arnaud, and J. Rosa, "Silicon thickness monitoring strategy for FD-SOI 28nm technology," *IEEE International Conference on Microelectronic Test Structures*, vol. 2015-May, no. V, pp. 65–69, 2015. [p. 76.]
- [188] P. E. Acosta Alba, "Influence of Smart Cut technological steps on thickness uniformity of SOI wafers: Multi-Scale approach." Ph.D. dissertation, Toulouse University, 2014. [p. 77.]
- [189] Guénolé Lallement; Fady Abouzeid; Martin Cochet; Jean-Marc Daveau; Philippe Roche; Jean-Luc Autran, "A 2.7pJ/cycle 16MHz SoC with 4.3nW power-off ARM Cortex-M0+ core in 28nm FD-SOI," *2017 IEEE European Solid-State Circuits Conference (ESSCIRC)*, pp. 159–162, 2017. [p. 78.]
- [190] M. Scholl, R. Wunderlich, and S. Heinen, "A low complexity digital frequency calibration with high jitter immunity for ultra-low-power oscillators," *Advances in Radio Science*, vol. 17, no. 2016, pp. 145–150, 2019. [p. 80.]
- [191] M. Scholl, Y. Zhang, R. Wunderlich, and S. Heinen, "A 80 nW , 32 kHz Charge-Pump based Ultra Low Power Oscillator with Temperature Compensation," *2016 IEEE European Solid-State Circuits Conference (ESSCIRC)*, pp. 343–346, 2016. [pp. 80, 84, 93, 94, and 220.]
- [192] OpenCores, "Overview SPI Master_Slave Interface OpenCores," 2017. [Online]. Available: <https://opencores.org/projects/spi{ }master{ }slave> [p. 84.]
- [193] K. Tsubaki, T. Hirose, N. Kuroki, and M. Numa, "A 32.55-kHz, 472-nW, 120ppm/°C, fully on-chip, variation tolerant CMOS relaxation oscillator for a real-time clock application," *2013 IEEE European Solid-State Circuits Conference (ESSCIRC)*, pp. 315–318, 2013. [pp. 93 and 94.]
- [194] K.-J. Hsiao, "A 32.4 ppm/°C 3.2-1.6V Self-chopped Relaxation Oscillator with Adaptive Supply Generation," *Digest of Technical Papers - IEEE Symposium on VLSI Circuits*, pp. 14–15, 2012. [pp. 93 and 94.]
- [195] D. Griffith, P. T. Røine, J. Murdock, and R. Smith, "A 190nW 33kHz RC oscillator with ±0.21% temperature stability and 4ppm long-term stability," *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, vol. 57, pp. 300–301, 2014. [pp. 93, 94, and 220.]
- [196] F. Abouzeid, S. Clerc, C. Bottoni, B. Coeffic, J. M. Daveau, D. Croain, G. Gasiot, D. Sousan, and P. Roche, "28nm FD-SOI technology and design platform for sub-10pJ/cycle and SER-immune 32bits processors," *2016 IEEE European Solid-State Circuits Conference (ESSCIRC)*, pp. 108–111, 2015. [p. 98.]

- [197] D. Bol, "Pushing Ultra-Low-Power Digital Circuits into the Era Pushing Ultra-Low-Power Digital Circuits into the Era," *Power*, no. December 2008, 2008. [p. 98.]
- [198] F. Abouzeid, S. Clerc, F. Firmin, M. Renaudin, and G. Sicard, "A 45nm CMOS 0.35v-optimized standard cell library for ultra-low power applications," *Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design - ISLPED '09*, no. August, p. 225, 2009. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1594233.1594288> [p. 100.]
- [199] G. De Streel and D. Bol, "Impact of back gate biasing schemes on energy and robustness of ULV logic in 28nm UTBB FDSOI technology," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 255–260, 2013. [p. 103.]
- [200] P. Galy, J. Bourgeat, and D. Marin-Cudraz, "New modular bi-directional power-switch and self ESD protected in 28nm UTBB FDSOI advanced CMOS technology," *ICICDT 2014 - IEEE International Conference on Integrated Circuit Design and Technology*, no. 1, pp. 4–7, 2014. [p. 105.]
- [201] J. Le Coz, B. Pelloux-Prayer, B. Giraud, F. Giner, and P. Flatresse, "DTMOS power switch in 28 nm UTBB FD-SOI technology," *2013 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference, S3S 2013*, pp. 2–3, 2013. [p. 105.]
- [202] A. R. M. Limited, "Cortex -M0 + - Technical Reference Manual," ARM Limited, Tech. Rep., 2012. [pp. 113, 212, and 214.]
- [203] Y. Akgul, D. Puschini, P. Benoit, and L. Torres, "Power management through DVFS and dynamic body biasing in FD-SOI circuits," in *Design Automation Conference (DAC), 2014 51th ACM/EDAC/IEEE*, 2014. [p. 117.]
- [204] M. Cochet, S. Clerc, M. Naceur, P. Schamberger, D. Croain, J.-l. Autran, and P. Roche, "A 28nm FD-SOI Standard Cell 0.6-1.2V Open-Loop Frequency Multiplier for Low Power SoC Clocking," *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1206–1209, 2016. [p. 118.]
- [205] M. Cochet, A. Puggelli, B. Keller, B. Zimmer, M. Blagojevic, S. Clerc, P. Roche, J. L. Autran, and B. Nikolić, "On-chip supply power measurement and waveform reconstruction in a 28nm FD-SOI processor SoC," in *2016 IEEE Asian Solid-State Circuits Conference, A-SSCC 2016 - Proceedings*, 2017, pp. 125–128. [p. 118.]
- [206] K. J. Dhorì, H. Chawla, A. Kumar, P. Pandey, P. Kumar, L. Ciampolini, F. Cacho, and D. Croain, "High-yield design of high-density SRAM for low-voltage and low-leakage operations," in *2017 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*. IEEE, oct 2017, pp. 1–6. [Online]. Available: <http://ieeexplore.ieee.org/document/8244429/> [p. 120.]
- [207] A. Quelen, G. Pillonnet, P. Flatresse, and E. Beigne, "A 2.5 μ W 0.0067mm² automatic back-biasing compensation unit achieving 50% leakage reduction in FDSOI 28nm over 0.35-to-1V VDD range," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*. IEEE, feb 2018, pp. 304–306. [Online]. Available: <http://ieeexplore.ieee.org/document/8310305/> [pp. 121 and 130.]

- [208] G. Lallement, F. Abouzeid, M. Cochet, J. M. Daveau, P. Roche, and J. L. Autran, "A 2.7 pJ/cycle 16 MHz, 0.7 μW Deep Sleep Power ARM Cortex-M0+ Core SoC in 28 nm FD-SOI," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 7, pp. 2088–2100, 2018. [pp. 122 and 154.]
- [209] R. Uytterhoeven and W. Dehaene, "A sub 10 pJ/Cycle over a 2 to 200 MHz Performance Range RISC-V Microprocessor in 28 nm FDSOI," in *ESSCIRC 2018 - IEEE 44th European Solid State Circuits Conference*, 2018, pp. 326–329. [pp. 122 and 152.]
- [210] F. Abouzeid, C. Bernicot, S. Clerc, J.-M. Daveau, G. Gasiot, D. Noblet, D. Soussan, and P. Roche, "30% static power improvement on ARM Cortex ® -A53 using static Biasing-Anticipation," in *2016 46th European Solid-State Device Research Conference (ESSDERC)*. IEEE, sep 2016, pp. 29–32. [Online]. Available: <http://ieeexplore.ieee.org/document/7599581/> [p. 123.]
- [211] G. Lallement, F. Abouzeid, J.-M. Daveau, P. Roche, and J.-L. Autran, "A 1.1-pJ/cycle, 20-MHz, 0.42-V Temperature Compensated ARM Cortex-M0+ SoC With Adaptive Self Body-Biasing in FD-SOI," *IEEE Solid-State Circuits Letters*, vol. 1, no. 7, pp. 174–177, 2019. [p. 154.]
- [212] STMicroelectronics, "POWERSTUDIO - ST PowerStudio dynamic electrothermal simulation software for power devices - STMicroelectronics," 2019. [Online]. Available: <https://www.st.com/en/embedded-software/stsw-powerstudio.html> [p. 158.]
- [213] L. Benini, A. Bogliolo, and G. D. Micheli, "A survey of design techniques for system-level dynamic power management - Very Large Scale Integration (VLSI) Systems, IEEE Transactions on," *Integration The Vlsi Journal*, vol. 8, no. 3, pp. 299–316, 2000. [pp. 158 and 159.]
- [214] K. Huang, L. Santinelli, J. J. Chen, L. Thiele, and G. C. Buttazzo, "Adaptive dynamic power management for hard real-time systems," *Proceedings - Real-Time Systems Symposium*, pp. 23–32, 2009. [p. 158.]
- [215] Yung-Hsiang Lu and G. De Micheli, "Adaptive hard disk power management on personal computers," *Proceedings Ninth Great Lakes Symposium on VLSI*, pp. 50–53, 2003. [p. 159.]
- [216] M. K. Stojčev, M. R. Kosanovic, and L. R. Golubovic, "Power management and energy harvesting techniques for wireless sensor nodes," *Proceedings of the 9th International Conference on Telecommunication in Modern Satellite, Cable, and Broadcasting Services 2009 (TELSIKS '09)*, no. NOVEMBER 2009, pp. 65–72, 2009. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5339410> [p. 159.]
- [217] D. Ernst, M. Glavic, and L. Wehenkel, "Power Systems Stability Control: Reinforcement Learning Framework," *IEEE Transactions on Power Systems*, vol. 19, no. 1, pp. 427–435, 2004. [p. 159.]
- [218] R. C. Hsu, C. T. Liu, and H. L. Wang, "A reinforcement learning-based ToD provisioning dynamic power management for sustainable operation of energy harvesting wireless sensor node," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 2, pp. 181–191, 2014. [p. 159.]

- [219] X. Lin, Y. Wang, P. Bogdan, N. Chang, and M. Pedram, "Reinforcement learning based power management for hybrid electric vehicles," in *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, nov 2014, pp. 33–38. [Online]. Available: <http://ieeexplore.ieee.org/document/7001326/> [pp. 159 and 162.]
- [220] Y. Zhang, Z. Xiong, D. Niyato, P. Wang, and D. I. Kim, "Towards a Perpetual IoT System: Wireless Power Management Policy with Threshold Structure," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5254–5270, 2018. [p. 159.]
- [221] X. He, K. Wang, H. Huang, T. Miyazaki, Y. Wang, and S. Guo, "Green Resource Allocation based on Deep Reinforcement Learning in Content-Centric IoT," *IEEE Transactions on Emerging Topics in Computing*, vol. 6750, no. c, pp. 1–15, 2018. [p. 159.]
- [222] U. Gupta, S. K. Mandal, M. Mao, C. Chakrabarti, and U. Y. Ogras, "A Deep Q-Learning Approach for Dynamic Management of Heterogeneous Processors," *IEEE Computer Architecture Letters*, vol. PP, no. c, pp. 1–5, 2019. [p. 159.]
- [223] Y. Tan, W. Liu, and Q. Qiu, "Adaptive power management using reinforcement learning," *2009 IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers*, p. 461, 2010. [p. 159.]
- [224] S. Yue, D. Zhu, Y. Wang, and M. Pedram, "Reinforcement learning based dynamic power management with a hybrid power supply," *Proceedings - IEEE International Conference on Computer Design: VLSI in Computers and Processors*, pp. 81–86, 2012. [p. 171.]
- [225] Wei Liu, Ying Tan, and Qinru Qiu, "Enhanced Q-learning algorithm for dynamic power management with performance constraint," *2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010)*, pp. 602–605, 2013. [p. 159.]
- [226] Y. Wang and M. Pedram, "Model-Free Reinforcement Learning and Bayesian Classification in System-Level Power Management," *IEEE Transactions on Computers*, vol. 65, no. 12, pp. 3713–3726, 2016. [pp. 159 and 166.]
- [227] M. Triki, A. C. Ammari, Y. Wang, and M. Pedram, "Reinforcement learning algorithms for dynamic power management," *2014 World Symposium on Computer Applications and Research, WSCAR 2014*, pp. 1–6, 2014. [p. 159.]
- [228] STMicroelectronics, "BlueNRG-2 Bluetooth® low energy wireless system-on-chip." [Online]. Available: http://www.st.com/content/st_{ }com/en/products/wireless-connectivity/bluetooth-bluetooth-low-energy/bluenrg-2.html [p. 162.]
- [229] Intel, "Intel® Edison Compute Module (IoT)," 2016. [p. 162.]
- [230] Eui-Young Chung, L. Benini, and G. De Micheli, "Dynamic power management using adaptive learning tree," in *ICCAD '99 Proceedings of the 1999 IEEE/ACM international conference on Computer-aided design*, 2003, pp. 274–279. [p. 162.]
- [231] A. A. Markov, *Theory of Algorithms*. Academy of Sciences of the USSR, 1954. [p. 163.]
- [232] R. A. Howard, *Dynamic Programming and Markov Processes*, 1st ed. Press, The MIT, 1960. [p. 163.]

- [233] S. Russell and P. Norvig, *Artificial Intelligence A Modern Approach Third Edition*. Pearson Education, Inc., 2010. [pp. 164, 166, and 167.]
- [234] R. S. Sutton and A. G. Barto, *Reinforcement learning*. The MIT Press, 2012, vol. 3. [pp. 165, 166, 167, and 177.]
- [235] DeepMind, “AlphaStar: Mastering the Real-Time Strategy Game StarCraft II,” pp. 1–16, 2019. [Online]. Available: <https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/> [p. 166.]
- [236] R. Bellman, *Dynamic Programming*, 1st ed. Princeton, NJ, USA: Princeton University Press, 1957. [p. 167.]
- [237] L. Baird, “Residual Algorithms: Reinforcement Learning with Function Approximation,” *Machine Learning Proceedings 1995*, vol. 10, no. 8, pp. 30–37, 1995. [Online]. Available: <http://www.embase.com/search/results?subaction=viewrecord{&}from=export{&}id=L11019759https://linkinghub.elsevier.com/retrieve/pii/B978155860377650013X> [p. 167.]
- [238] MyHDL, “MyHDL,” 2019. [Online]. Available: <http://www.myhdl.org/> [pp. 172 and 222.]
- [239] Cocotb, “COroutine based COsimulation TestBench environment,” 2019. [Online]. Available: <https://cocotb.readthedocs.io/en/latest/introduction.html> [p. 173.]
- [240] A. D. Tijsma, M. M. Drugan, and M. A. Wiering, “Comparing exploration strategies for Q-learning in random stochastic mazes,” in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, dec 2016, pp. 1–8. [Online]. Available: <http://ieeexplore.ieee.org/document/7849366/> [p. 177.]
- [241] Q. Gao, B. Hong, Z. He, J. Liu, and G. Niu, “An improved Q-learning algorithm based on exploration region expansion strategy,” in *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, vol. 1, 2006, pp. 4167–4170. [p. 178.]
- [242] K. Saito, A. Notsu, and K. Honda, “Discounted UCB1-tuned for q-learning,” *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems, SCIS 2014 and 15th International Symposium on Advanced Intelligent Systems, ISIS 2014*, pp. 966–970, 2014. [p. 178.]
- [243] B. R. Rich, *Clarence Leonard (Kelly) Johnson 19101990: A Biographical Memoir*. National Academies Press, Washington, DC, 1995. [p. 202.]
- [244] F. Berthier, E. Beigne, P. Vivet, and O. Sentieys, “Power gain estimation of an event-driven wake-up controller dedicated to WSN’s microcontroller,” in *2015 IEEE 13th International New Circuits and Systems Conference (NEWCAS)*. IEEE, jun 2015, pp. 1–4. [p. 204.]
- [245] P. P. Mercier, S. Bandyopadhyay, A. C. Lysaght, K. M. Stankovic, and A. P. Chandrakasan, “A 78 pW 1 b/s 2.4 GHz radio transmitter for near-zero-power sensing applications,” *European Solid-State Circuits Conference*, pp. 133–136, 2013. [p. 205.]
- [246] V. Pinrod, L. Pancoast, B. Davaji, S. Lee, R. Ying, A. Molnar, and A. Lal, “Zero-power sensors with near-zero-power wakeup switches for reliable sensor platforms,” *Proceedings of the IEEE International Conference on Micro Electro Mechanical Systems (MEMS)*, pp. 1236–1239, 2017. [p. 205.]

- [247] A. Morel, A. Quelen, P. Gasnier, R. Grezard, S. Monfray, A. Badel, and G. Pillonnet, "A Shock-Optimized SECE Integrated Circuit," *IEEE Journal of Solid-State Circuits*, pp. 1–14, 2018. [p. 205.]
- [248] S. Bang, Y. Lee, I. Lee, Y. Kim, G. Kim, D. Blaauw, and D. Sylvester, "A fully integrated switched-capacitor based PMU with adaptive energy harvesting technique for ultra-low power sensing applications," in *Proceedings - IEEE International Symposium on Circuits and Systems*, 2013. [p. 205.]
- [249] T. Nakada, T. Shigematsu, T. Komoda, S. Miwa, H. Nakamura, Y. Sato, H. Ueki, M. Hayashikoshi, and T. Shimizu, "Data-aware power management for periodic real-time systems with non-volatile memory," *2014 IEEE Non-Volatile Memory Systems and Applications Symposium, NVMSA 2014*, 2014. [p. 206.]
- [250] D. A. Hennessy, John L and Patterson, *Computer architecture*. Elsevier, 2011. [p. 210.]
- [251] J. von Neumann, "First draft of a report on the EDVAC," *IEEE Annals of the History of Computing*, vol. 15, no. 4, pp. 27–75, 2002. [p. 210.]
- [252] A. R. M. Limited, "AMBA Generic Flash Bus Protocol Specification," ARM Limited, Tech. Rep., 2018. [p. 210.]
- [253] K. Rosenfeld and R. Karri, "Attacks and defenses for JTAG," *IEEE Design and Test of Computers*, vol. 27, no. 1, pp. 36–47, 2010. [p. 211.]
- [254] J. Consortium, "1149.7-2009 - IEEE Standard for Reduced-Pin and Enhanced-Functionality Test Access Port and Boundary-Scan Architecture." [Online]. Available: <https://ieeexplore.ieee.org/document/5412866> [p. 211.]
- [255] "Qibec - 1-bit, 1-instruction CPU made from transistors." [Online]. Available: <http://mircad.com/q/> [p. 211.]
- [256] NVidia, "GeForce GTX 285 Specifications." [Online]. Available: <https://www.geforce.com/hardware/desktop-gpus/geforce-gtx-285/specifications> [p. 211.]
- [257] A. Limited, "Cortex-M0 ARM Developer." [Online]. Available: <https://developer.arm.com/products/processors/cortex-m/cortex-m0> [p. 211.]
- [258] —, "Code Size a comprehensive comparison of microMIPS32 and Thumb code size using many Megabytes of customer code." [Online]. Available: <https://community.arm.com/developer/ip-products/processors/b/processors-ip-blog/posts/code-size-a-comprehensive-comparison-of-micromips32-and-thumb-code-size-using-many-megabytes-of-customer-code> [p. 211.]
- [259] ARM Limited, "DesignStart for University Arm." [Online]. Available: <https://www.arm.com/resources/designstart/designstart-university> [p. 212.]
- [260] —, "Cortex-M23 ARM Developer." [Online]. Available: <https://developer.arm.com/products/processors/cortex-m/cortex-m0> [p. 212.]
- [261] —, "ARM: AHB-Lite technical documentation," 2014. [Online]. Available: <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.set.amba/index.html> [p. 213.]

- [262] STMicroelectronics, "ST-LINK/V2 ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32 - STMicroelectronics." [Online]. Available: <http://www.st.com/web/catalog/tools/FM146/CL1984/SC724/SS1677/PF251168> [p. 214.]
- [263] E. Ashfield, I. Field, P. Harrod, S. Houlihane, and W. Orme, "Serial Wire Debug and the CoreSight," 2000. [p. 214.]
- [264] I. T. Union, "Definitions and terminology for synchronization networks," INTERNATIONAL TELECOMMUNICATION UNION, Tech. Rep., 1996. [p. 217.]
- [265] F. Stability, "Ieee-Nasa Symposium on Short-Term Frequency Stability," *Nasa*, 1964. [p. 218.]
- [266] L. S. Cutler and C. L. Searle, "Some Aspects of the Theory and Measurement of Frequency Fluctuations in Frequency Standards," *Proceedings of the IEEE*, vol. 54, no. 2, pp. 136–154, 1966. [p. 218.]
- [267] D. W. Allan, "Statistics of Atomic Frequency Standards," *Proceedings of the IEEE*, vol. 54, no. 2, pp. 221–230, 1966. [p. 218.]
- [268] J. A. Barnes, A. R. Chi, L. S. Cutler, D. J. Healey, D. B. Leeson, E. T. McGunigal, J. A. Mullen, W. L. Smith, R. L. Sydnor, R. F. C. Vessot, and G. M. R. Winkler, "Characterization of Frequency Stability," *IEEE Transactions on Instrumentation and Measurement*, vol. IM-20, no. 2, pp. 105–120, 1971. [p. 218.]
- [269] I. Standards and C. Committee, *IEEE Std 1139-2008 (Revision of IEEE Std 1139-1999) IEEE Standard Definitions of Physical Quantities for Fundamental Frequency and Time Metrology—Random Instabilities*. IEEE Press, 2009, vol. 2008, no. February. [p. 218.]
- [270] D. W. Allan, "Time and Frequency (Time-Domain) Characterization, Estimation, and Prediction of Precision Clocks and Oscillators," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 34, no. 6, pp. 647–654, 1987. [p. 219.]
- [271] W. J. Riley, *Handbook of Frequency Stability Analysis*. Hamilton Technical Services, 1994, vol. 31, no. 1. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/0148906294927065> [p. 219.]
- [272] Y. S. Lin, D. M. Sylvester, and D. T. Blaauw, "A 150pW program-and-hold timer for ultra-low-power sensor platforms," *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, pp. 326–328, 2009. [p. 220.]
- [273] Y. Lee, B. Giridhar, Z. Foo, D. Sylvester, and D. Blaauw, "A 660pW multi-stage temperature-compensated timer for ultra-low-power wireless sensor node synchronization," *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, pp. 46–47, 2011. [p. 220.]
- [274] H. Wang and P. P. Mercier, "A 1.6%/V 124.2 pW 9.3 Hz Relaxation Oscillator Featuring a 49.7 pW Voltage and Current Reference Generator," *ESSCIRC 2017-43rd IEEE European Solid State Circuits Conference*, vol. 1, pp. 6–9, 2017. [p. 220.]
- [275] —, "A 51 pW reference-free capacitive-discharging oscillator architecture operating at 2.8 Hz," *Proceedings of the Custom Integrated Circuits Conference*, vol. 2015-Novem, pp. 0–3, 2015. [p. 220.]

- [276] S. Dai and J. K. Rosenstein, "A 14.4nW 122KHz dual-phase current-mode relaxation oscillator for near-zero-power sensors," *Proceedings of the Custom Integrated Circuits Conference*, vol. 2015-Novem, pp. 4–7, 2015. [p. 220.]
- [277] S. Y. Lu and Y. T. Liao, "A 45 μ W, 9.5MHz current-reused RC oscillator using a swing-boosting technique," in *2017 International Symposium on VLSI Design, Automation and Test, VLSI-DAT 2017*, 2017, pp. 5–8. [p. 220.]
- [278] K. T. Chai, C. Wang, J. Tao, J. Xu, L. Zhong, and R. S. Tan, "High-performance differential capacitive mems sensor readout with relaxation oscillator front-end and phase locked loop time-to-digital converter back-end," *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, pp. 1528–1531, 2016. [p. 220.]
- [279] M. H. Perrott, J. C. Salvia, F. S. Lee, A. Partridge, S. Mukherjee, C. Arft, K. Jintae, N. Arumugam, P. Gupta, S. Tabatabaei, S. Pamarti, L. Haechang, and F. Assaderaghi, "A Temperature-to-Digital Converter for a MEMS-Based Programmable Oscillator," *Solid-State Circuits, IEEE Journal of*, vol. 48, no. 1, pp. 276–291, 2013. [Online]. Available: <http://ieeexplore.ieee.org/ielx5/4/6399535/06341095.pdf?tp=&arnumber=6341095&isnumber=6399535> [p. 220.]
- [280] K. J. Hsiao, "A 1.89nW/0.15V self-charged XO for real-time clock generation," *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, vol. 57, pp. 298–299, 2014. [p. 220.]
- [281] D. Yoon, T. Jang, D. Sylvester, and D. Blaauw, "A 5.58 nW Crystal Oscillator Using Pulsed Driver for Real-Time Clocks," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 2, pp. 509–522, 2016. [p. 220.]
- [282] H.-D. R. T. C. Module, O. Accuracy, and C.-r. Scheme, "DTCXO RTC Module with an Overall Accuracy of Compensation-Resolution Scheme at 1Hz," *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, pp. 208–210, 2016. [p. 220.]
- [283] A. Sangiovanni-Vincentelli, "The tides of EDA," *IEEE Design & Test of Computers*, vol. 20, no. 6, pp. 59–75, nov 2003. [Online]. Available: <http://ieeexplore.ieee.org/document/1246165/> [p. 221.]
- [284] J. Bachrach, H. Vo, B. Richards, Y. Lee, A. Waterman, R. Avižienis, J. Wawrzynek, and K. Asanović, "Chisel: Constructing Hardware in a Scala Embedded Language," pp. 1212–1221, 2012. [Online]. Available: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6241660%5Cnhttp://ieeexplore.ieee.org.eproxy1.lib.hku.hk/xpl/articleDetails.jsp?arnumber=6241660> [p. 222.]

