Ultra low-power asynchronous-logic design for high variation-space and wide operation-space applications

Lin, Tong

2014

Lin, T. (2014). Ultra low-power asynchronous-logic design for high variation-space and wide operation-space applications. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/61794>

<https://doi.org/10.32657/10356/61794>

NANYANG
TECHNOLOGICAL
UNIVERSITY

ULTRA LOW-POWER ASYNCHRONOUS-LOGIC DESIGN
FOR
HIGH VARIATION-SPACE
AND
WIDE OPERATION-SPACE APPLICATIONS

LIN TONG

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

2014

**ULTRA LOW-POWER ASYNCHRONOUS-LOGIC DESIGN
FOR
HIGH VARIATION-SPACE
AND
WIDE OPERATION-SPACE APPLICATIONS**

LIN TONG

School of Electrical and Electronic Engineering

**A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy**

2014

Acknowledgements

First and foremost, I wish to thank my PhD advisors, Profs. Joseph S. Chang and Gwee Bah Hwee for leading me into the field of research. This has forever shaped my perspective, aroused my curiosity, and allowed me to experience first-hand one of the greatest joys in life – seeking and discovery.

Prof. Chang has taught me many things, amongst them, how to think critically, and the value of rigour whilst having fun. Perhaps the most contagious is his passion for uncompromising rigour, nuances in research, and for scholarliness. I owe him my deepest gratitude and appreciation for all the time and effort he spent working with me on my thesis and papers. It was and will always be fun to learn.

Prof. Gwee has always been so helpful, encouraging, and supportive. I am very grateful to him for allowing me the opportunity and freedom to do my research and introducing me to the field of asynchronous-logic, a field of increasing infinity with my every exploration. I wish to thank Dr. Chong Kwen Siong for guiding me through my initial, most struggling days of research and for demonstrating the notion of standard in work. I will always cherish and respect that. I also wish to thank my fellow researchers and friends for the fun while struggling together. I also thank NTU for the coveted Nanyang President's Graduate Scholarship and to the School of EEE for the availability of facilities.

Last but not least, I wish to dedicate this thesis to my beloved wife for her love and patience, and to my family for always being there.

Contents

Acknowledgements.....	i
Contents	ii
Author's Publications.....	iv
Abstract	v
List of Figures	vii
List of Tables	x
Nomenclature.....	xi
Chapter 1 Introduction.....	1
1.1 Motivation.....	1
1.2 Objectives	17
1.3 Contributions	19
1.4 Organization.....	21
Chapter 2 Literature Review.....	23
2.1 Low-Power and Ultra Low-Power Sub- V_t	24
2.1.1 Design-time Techniques	27
2.1.2 Operation-time Techniques.....	28
2.1.3 Ultra Low-Power Sub- V_t	31
2.1.4 Power Gating	36
2.2 Logic Families for Sub- V_t	37
2.2.1 Static Logic	38
2.2.2 Pass Transistor/Transmission Gate Logic	39
2.2.3 Ratioed Pseudo-NMOS Logic	40
2.2.4 Dynamic Logic	41
2.3 Design Approaches/Signaling Protocols for Sub- V_t	43
2.3.1 Synchronous-Logic	44
2.3.2 Asynchronous-Logic.....	46
2.4 Asynchronous-Logic for Sub- V_t	48
2.4.1 Fundamentals of Asynchronous-Logic.....	49
2.4.2 Asynchronous-Logic QDI for Sub- V_t	53
2.5 Summary of Literature Review.....	57

Chapter 3	Power Gating for Async MD and Ultra Low-Power Sub- V_t Async QDI	59
3.1	Introduction.....	59
3.2	Fine-Grain Power Gating for Reducing Wasted Powers in Async Matched Delay	61
3.2.1	Async MD Pipeline.....	63
3.2.2	Proposed Fine-Grain Power Gating for Async MD Pipeline	64
3.2.3	Benchmarking the Proposed Fine-Grain Power Gating	69
3.3	First-Order Delay Variations Estimation for Sync and its Comparison with Async QDI in Sub- V_t	73
3.3.1	First-Order Delay Variation Estimation due to V_t , V_{DD} and Temperature Variations.....	75
3.3.2	Benchmarking Sync and Async QDI in Sub- V_t	84
3.4	Conclusions.....	95
Chapter 4	An Ultra Low-Power Asynchronous-Logic In-Situ Self-Adaptive V_{DD} System for Wireless Sensor Networks, and Proposed ‘Pseudo-QDI’ Signaling Protocol.....	96
4.1	Introduction.....	96
4.2	Sub- V_t Self-Adaptive V_{DD} Scaling (SSAVS) System for Wireless Sensor Networks (WSNs).....	97
4.2.1	Adaptive V_{DD} Scaling Systems	102
4.2.2	System Design	104
4.2.3	Results and Benchmarking	118
4.3	A Robust Asynchronous Approach for Realizing Ultra Low-Power Digital Self-Adaptive V_{DD} Scaling System.....	134
4.3.1	Proposed Async Pseudo-QDI Realization Approach	136
4.3.2	Timing Analysis on the Proposed Pseudo-QDI Realization Approach.....	140
4.3.3	Benchmarking Results	142
4.4	Conclusions.....	144
Chapter 5	Conclusions and Recommendations for Future Work	146
5.1	Conclusions.....	146
5.2	Recommendations for Future Work	149
Bibliography		152

Author's Publications

Journal Papers

- [1] T. Lin, K.-S. Chong, J. S. Chang, and B.-H. Gwee, "An Ultra-Low Power Asynchronous-Logic In-Situ Self-Adaptive V_{DD} System for Wireless Sensor Networks," *IEEE Journal of Solid-State Circuits*, vol. 48, pp. 573–586, Feb. 2013.

Conference Papers and (Invited) Talks

- [2] Invited Talk
J. S. Chang, T. Ge, and T. Lin, "Fully-Additive Printed RFID on a Plastic Film," *IEEE MTT-S Int. Microwave Workshop Series on RF and Wireless Technologies for Biomedical and Healthcare Applications*, Dec 9-11, 2013, Singapore.
- [3] Invited Talk
J. S. Chang, T. Lin, and K.-S. Chong, "Asynchronous-logic: Low-Power/Ultra Low-Power Design, and High Variation-space Wide Operation-space Applications," *IEEE S3S Conference*, Oct 7-10 2013, Monterey, California, USA.
- [4] K.-L. Chang, T. Lin, W.-G. Ho, K.-S. Chong, B.-H. Gwee and J. S. Chang, "A Dual-Core 8051 Microcontroller System based on Synchronous-logic and Asynchronous-logic," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2013, pp. 3022-3025.
- [5] 'Best Student Paper' Award
T. Lin, K.-S. Chong, J. S. Chang, B.-H. Gwee, and W. Shu, "A Robust Asynchronous Approach for Realizing Ultra-Low Power Digital Self-Adaptive V_{DD} Scaling System," in *Proc. IEEE Sub-threshold Microelectronics Conf.*, 2012, pp. 1-3.
- [6] K.-L. Chang, T. Lin, W.-G. Ho, K.-S. Chong, B.-H. Gwee and J. S. Chang, "A Comparative Study on Asynchronous Quasi-Delay-Insensitive Templates," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2012, pp. 1819-1822.
- [7] W.-G. Ho, K.-S. Chong, T. Lin, B.-H. Gwee, and J. S. Chang, "Energy-Delay Efficient Asynchronous-Logic 16×16-Bit Pipelined Multiplier Based on Sense Amplifier-Based Pass Transistor Logic," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2012, pp. 492-495.
- [8] T. Lin, K.-S. Chong, B.-H. Gwee, J. S. Chang, and Z.-X. Qiu, "Analytical delay variation modelling for evaluating sub-threshold synchronous/asynchronous designs," in *Proc. IEEE Int. NEWCAS Conf.*, 2010, pp. 69–72.
- [9] T. Lin, K.-S. Chong, B.-H. Gwee and J. S. Chang, "Fine-grained power gating for leakage and short-circuit power reduction by using asynchronous-logic," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2009, pp. 3162-3165.

Abstract

This thesis pertains to the design of low-power/ultra low-power *high variation-space* and *wide operation-space* digital electronics for portable/mobile applications. High variation-space and wide operation-space respectively refer to error-free operation despite high variations in the prevailing conditions (including Process, Voltage and Temperature (PVT) variations) and under a wide range of activity levels or workload. In view of said spaces, we adopt the somewhat esoteric asynchronous-logic (async) vis-à-vis the conventional synchronous-logic (sync); more specifically, the Matched Delay (MD) and the Quasi-Delay-Insensitive (QDI).

For an MD pipeline operating under a wide operation-space (alternating between active and idle), we propose a fine-grain power gating methodology (applicable to three different gating configurations) to reduce short-circuit and leakage wasted powers. By exploiting the 4-phase handshake protocol, the ensuing overhead of the proposed power gating is low, specifically one inverter (per pipeline stage) and <15% delay.

For sake of robustness in view of the extreme/virtually intractable PVT in ultra low-power sub-threshold (sub- V_t) operation, where the circuit delay varies exponentially with PVT, we propose to adopt the QDI protocol. To quickly estimate to the first-order the delay variations (due to V_t , supply voltage (V_{DD}) and temperature; thus the required delay safety margin) of digital circuits in sub- V_t , we propose and derive a set of simple yet insightful analytical equations. The derived equations are verified by simulations, and we show that they are accurate for first-order estimations (with an inconsequential worst-case error of <12%). We thereafter benchmark, by means of adder circuits, the sync (with delay safety margins estimated from the derived equations) against the async QDI (with self-completion detection), and ascertain that neither the sync nor the async QDI is particularly advantageous in all conditions. This exercise depicts the usefulness of the derived equations, particularly

the insights provided thereto, and that delay variations are easily estimated from the nominal case.

We propose a Sub- V_t Self-Adaptive V_{DD} Scaling (SSAVS) system for a high variation-space and wide operation-space Wireless Sensor Network (WSN) with the objective of lowest possible power dissipation (in sub- V_t operation), yet high robustness and with minimal overheads. The effort to achieve the lowest possible power operation is by means of Dynamic-Voltage-Scaling (DVS) – self-adjusting V_{DD} to the minimum voltage (within 50mV) for the prevailing conditions. High robustness is achieved by adopting the QDI protocol, and by the embodiment of our proposed ‘Pre-Charged-Static-Logic’ (PCSL) logic style; when compared against competing async logic styles appropriate for sub- V_t , the PCSL is most competitive in terms of energy/operation, delay and IC area. By exploiting the already existing request and acknowledge signals of the QDI protocol, the ensuing overhead of the SSAVS is very modest – a simple counter and a FIFO buffer. The filter bank embodied in the SSAVS is shown to be ultra low-power and highly robust. The proposed async SSAVS is benchmarked against its conventional sync Dynamic-Voltage-Frequency-Scaling (DVFS) counterpart for two scenarios. We show that no one system is particularly advantageous when the operating conditions are known. Further, when the sync DVFS system is designed for the worst-case condition, the proposed async DVS SSAVS is somewhat more competitive. To reduce the overheads of async QDI to improve its competitiveness, we propose a hardware-simplified version of QDI (herein coined ‘pseudo-QDI’) with an implicit timing for said SSAVS, and show analytically that said implicit timing is easily satisfied whilst ensuring robust operation. This robustness is verified by measurements on prototype ICs over high variation-space and wide operation-space. By means of the pseudo-QDI, the ensuing energy and area are significantly reduced by $\sim 40\%$ and $\sim 1.34\times$ respectively compared to the standardized QDI, with virtually no compromise to robustness.

List of Figures

Fig. 1.1: Delay and power characteristics of inverters (@50kHz) 130nm CMOS, for different process options (LVT, RVT and LP); normalized with respect to RVT @1.2V	5
Fig. 1.2: Generic block diagram of a pipeline stage realized in: (a) sync, (b) async MD, and (c) async QDI	9
Fig. 2.1: E_{per} characteristics (normalized to the RVT design @ nominal $V_{\text{DD}}=1.2\text{V}$) of a 30-inverter chain (activity factor = 0.1) in 130nm CMOS process with different V_t options: LVT, RVT, and LP	26
Fig. 2.2: The degradation of on/off current ratio ($I_{\text{on}}/I_{\text{off}}$) of a MOS transistor in 180nm process (normalized to nominal $V_{\text{DD}}=1.8\text{V}$) [10].....	33
Fig. 2.3: 1000 Monte Carlo simulations on the delay of 80-inverter chain at sub- V_t V_{DD} (from 200mV to 400mV), and at various temperatures (extreme heat 125°C, nominal 25°C, and extreme cold -55°C) 35	
Fig. 2.4: Power gating configurations: (a) PMOS Gating, (b) NMOS Gating, and (c) Dual Gating [42].....	37
Fig. 2.5: Generic structure of a static logic gate	38
Fig. 2.6: A pass transistor/TG logic-based multiplexer in sub- V_t operation	40
Fig. 2.7: Generic structure of a pseudo-NMOS logic gate.....	41
Fig. 2.8: Dynamic logic in sub- V_t operation: (a) without keeper and (b) with keeper.	42
Fig. 2.9: (a) Generic block diagram of a sync pipeline stage working in sub- V_t ($V_{\text{DD}}=400\text{mV}$), and (b) signal waveforms (V_{DD} , $D1$, $D2$, $D3$, and CLK) for the sync circuit. The data is correctly synchronized for the first operation when V_{DD} is stable. The data is incorrectly synchronized for the second operation when V_{DD} is coupled with noise (V_{DD} variation). [53]	45
Fig. 2.10: (a) Generic block diagram of an async QDI pipeline stage, and (b) signal waveforms (V_{DD} , $D1.T$, $D2.T$, $D3.T$, and HS) for the async circuit. The data is correctly synchronized both for the first operation when V_{DD} is stable and for the second operation when V_{DD} is coupled with noise (albeit with a longer delay). [53]	48
Fig. 2.11: Block diagram of a generic async pipeline.....	50
Fig. 2.12: Async handshake protocols: (a) 2-phase NRZ and (b) 4-phase RZ	51
Fig. 2.13: Reported QDI designs	54
Fig. 2.14: Reported static QDI logic design styles for an AND/NAND gate: (a) static NULL-Convention-Logic (NCL), (b) static Delay-Insensitive-Minterm-Synthesis (DIMS), and (c) static Direct-Static-Logic-Implementation (DSLII).....	55
Fig. 2.15: Summary and classification of digital design approaches/signaling protocols. The approaches/protocols in bold are appropriate for sub- V_t operation.....	58
Fig. 3.1: Block diagram of an async MD pipeline	64
Fig. 3.2: Block diagram of the async MD pipeline with the proposed fine-grain power gating	66
Fig. 3.3: Schematic of the one-stage async MD pipeline with the proposed fine-grain power gating technique	67
Fig. 3.4: Signal Transition Graph (STG) of the Latch Controller employed in the async MD pipeline	68
Fig. 3.5: Signal timing diagram of the async MD pipeline with the proposed power gating.....	69
Fig. 3.6: Power Dissipations of the Combinational Block (including the power associated with the insertion of the gating transistor(s) where applicable) in the async MD pipeline at various input data rates.....	71

Fig. 3.7: Estimated inverter delay variations ($\Delta t_{d(V_t)\text{est}}$) at different V_{DD} due to $ V_t $ variations, and comparisons against simulations ($\Delta t_{d(V_t)\text{sim}}$).....	78
Fig. 3.8: Estimated inverter delay variations ($\Delta t_{d(V_{DD})\text{est}}$) at different V_{DD} due to V_{DD} variations, and comparisons against simulations ($\Delta t_{d(V_{DD})\text{sim}}$)	80
Fig. 3.9: Estimated inverter delay variations ($\Delta t_{d(T)\text{est}}$) at different V_{DD} due to T variations, and comparisons against simulations ($\Delta t_{d(T)\text{sim}}$)	83
Fig. 3.10: Pipeline stage: (a) Sync, and (b) Async QDI.....	85
Fig. 3.11: Full-adder design: (a) Single-rail sync and (b) Dual-rail async NCL.....	86
Fig. 3.12: Block diagram of the 8-bit async NCL CRA	87
Fig. 4.1: Block diagram of the WSN node.....	100
Fig. 4.2: Overall structure of the proposed SSAVS system with an async QDI FRM Filter Bank (FB); $V_{DD_NOM} = 1.2V$, V_{DD_ADJ} ranges from 150mV – 400mV	106
Fig. 4.3: An example of the variation of V_{DD_ADJ} with time. The logical numbers on the ordinate are V_{DD_Code} and their corresponding DC voltages (V_{DD_ADJ}).....	108
Fig. 4.4: (a) Proposed Pre-Charged Static-Logic (PCSL) architecture, and six basic cells embodying the proposed PCSL dual-rail QDI logic style: (b) 2-input AND/NAND gate, (c) 2-input OR/NOR gate, (d) 3-input AO/AOI gate, (e) 3-input OA/OAI gate, (f) 2-input XOR/XNOR gate, and (g) 2-input MUX.....	111
Fig. 4.5: Reported dual-rail AND/NAND circuit designs: (a) Delay-Insensitive-Minterm-Synthesis (DIMS), (b) NULL-Convention-Logic (NCL) with complex gates (NCL1), and (c) NCL with fast-reset complex gates (NCL2)	113
Fig. 4.6: Block diagram of one channel of the 8×8-Bit Quad-Channel Async QDI FRM FB	117
Fig. 4.7: Die microphotograph (left) and layout (right) of the fabricated test-chips: (a) proposed SSAVS system with async QDI FRM filter bank, and (b) sync benchmark filter	119
Fig. 4.8: (a) High V_{DD} variations @ 1kHz, 150mV-300mV, and (b) error-free response (<i>Ack</i> signal) from the proposed async QDI FRM filter bank	121
Fig. 4.9: Example of the captured waveforms depicting (a) self-adjustment of V_{DD_ADJ} and <i>Ack</i> from the async QDI FRM filter bank, and (b) self-adjustment of V_{DD_ADJ} and <i>Ack</i> under sudden temperature drop.	122
Fig. 4.10: Variation of the sync filter critical path delay under various PVT conditions: Monte Carlo simulations.....	124
Fig. 4.11: Scenario 1: Benchmarking delay and E_{per} of a sync DVFS filter bank and the async SSAVS filter bank for three temperature corners: (a) -40°C, (b) 25°C, and (c) 125°C. Note: Bold lines are measured while dotted lines are from simulations.....	130
Fig. 4.12: Scenario 1: Power consumption of the sync and async filter banks (a) @-40°C, (b) @25°C, and (c) @125°C.....	131
Fig. 4.13: Scenario 2: Benchmarking delay and E_{per} of a sync DVFS filter bank and the async SSAVS filter bank for three temperature corners: (a) -40°C, (b) 25°C, and (c) 125°C. Note: Bold lines are measured while dotted lines are from simulations.....	132
Fig. 4.14: Scenario 2: Power consumption of the sync and async filter banks (a) @-40°C, (b) @25°C, and (c) @125°C.....	133
Fig. 4.15: (a) The conventional async true-QDI pipeline, and (b) our proposed async pseudo-QDI pipeline embodying the PCSL cells	139

Fig. 4.16: (a) Die microphotograph and layout of the fabricated true-QDI and pseudo-QDI filter banks (@130nm CMOS), and (b) Robust sub- V_t operation of the fabricated pseudo-QDI filter bank under large V_{DD} variations	143
Fig. 4.17: Measured energy/operation (E_{per}) of the async filter banks.....	144

List of Tables

Table 1.1: International Technology Roadmap for Semiconductors (ITRS) 2011 [5].....	2
Table 1.2: The Dual-Rail Data Encoding	12
Table 2.1: Classification of the async design approaches.....	52
Table 2.2: Reported logic design styles (within specific logic families) for QDI realization.....	54
Table 3.1: Delays of the async QDI CRAs and the sync CRAs (without/with delay safety margins) @ $V_{DD}=0.15V$; the delays are normalized to the async QDI CRAs of respective wordlengths	89
Table 3.2: E_{per} of the async QDI CRAs and the sync CRAs (without/with delay safety margins) @ $V_{DD}=0.15V$; the E_{per} are normalized to the async QDI CRAs of respective wordlengths	92
Table 3.3: Transistor count of the async QDI CRA and the sync CRA.....	94
Table 4.1: Operation of the SSAVS controller	109
Table 4.2: Energy-per-operation (E_{per}), Delay and IC Area of Dual-rail Library Cells Embodying Various Logic Styles @ $V_{DD}=150mV$ and 130nm CMOS Process	114

Nomenclature

E_{per}	– Energy per Operation
I_{on}	– Transistor on Current
I_{off}	– Transistor off Current
V_{DD}	– Power Supply Voltage
V_t	– Transistor Threshold Voltage
V_{th}	– Thermal Voltage
μW	– Microwatt
ACK	– Acknowledge
ASIC	– Application Specific Integrated Circuit
Async	– Asynchronous-Logic
BD	– Bundled-Data
CAD	– Computer Aided Design
CD	– Completion Detection
CMOS	– Complementary Metal-Oxide-Semiconductor
DCD	– Datapath Completion Detection
DI	– Delay-Insensitive
DIMS	– Delay-Insensitive-Minterm-Synthesis
DSLI	– Direct-Static-Logic-Implementation
DSP	– Digital Signal Processor
DVFS	– Dynamic-Voltage-Frequency-Scaling
DVS	– Dynamic Voltage Scaling
EDA	– Electronic Design Automation
EMI	– Electromagnetic Interference
FB	– Filter Bank
FF	– Flip-Flop
FFT	– Fast Fourier Transform
FIFO	– First-In-First-Out
FIR	– Finite Impulse Response
FPGA	– Field Programmable Gate Array
FRM	– Frequency Response Masking
GALS	– Globally Asynchronous Locally Synchronous

HDL – Hardware Description Language
HS – Handshake
HVT – High Threshold Voltage Process Option
IC – Integrated Circuit
IFIR – Interpolated Finite Impulse Response
IIR – Infinite Impulse Response
ITRS – International Technology Roadmap for Semiconductors
LCD – Latch Completion Detection
LDO – Low-Dropout
Li/CF_x – Lithium/Carbon Fluoride
LVT – Low Threshold Voltage Process Option
MAC – Multiply-Accumulate
MC Simulation – Monte Carlo Simulation
MCU – Microcontroller Unit
MD – Matched Delay
MIPS – Million Instructions per Second
MUX – Multiplexer
NCL – NULL-Convention-Logic
NCL1 – NCL with complex gates
NCL2 – NCL with fast-reset complex gates
NRZ – None-Return-to-Zero
PCHB – Pre-charged Half Buffer
PCSL – Pre-Charged-Static-Logic
Pseudo-QDI – QDI with implicit timing
PVT – Process, Voltage and Temperature
QDI – Quasi-Delay-Insensitive
RCA – Ripple Carry Adder
REQ – Request
RF – Radio Frequency
RISC – Reduced Instruction Set Computer
RTL – Register Transfer Level
RTZ – Return-to-Zero
RVT – Regular Threshold Voltage Process Option
SAPTL – Sense Amplifier-based Pass Transistor Logic

SI – Speed-Independent

SRAM – Static Random Access Memory

SSAVS – Sub- V_t Self-Adaptive V_{DD} Scaling

SSTA – Statistical Static Timing Analysis

STAPL – Single-Track-handshake Asynchronous-Pulse-Logic

STFB – Single-Track Full-Buffer

STG – Signal Transition Graph

Sub- V_t – Sub-Threshold

Sync – Synchronous-Logic

VLSI – Very-Large-Scale-Integration

WSN – Wireless Sensor Network

Chapter 1 Introduction

This chapter describes the motivation, objectives, contributions and organization of the thesis.

1.1 Motivation

High Variation-Space and Wide Operation-Space Ubiquitous Computing

At the present juncture, it is generally well accepted that the future of computing will increasingly involve portable/mobile devices, including the “Internet of Things” (IoTs) objects [1], where intelligence/information processing capability is embedded therein. These devices typically acquire/process information and may communicate/coordinate directly with each other (for crowd-sourcing, etc) and/or via the internet, and with or without human intervention. Their realization requires a host of enabling technologies, and depending on their specific functionalities, these may include a Wireless Sensor Network (WSN) [2] for distributed information acquisition/processing; see Chapter 4 later for novel designs thereto.

For these devices to be ubiquitous or ‘everywhere’, they need to be operationally functional in a myriad of environments. The environmental conditions may be highly variable and the power supply unreliable, for example where the required energy is harvested from the environment [3]. Further, they have to accommodate a wide range of activity levels from inactivity or idle when no computation (and related activities including data acquisition, communication, etc) is required to ‘bursts’ of high activity when computation and related activities are required [4]. Put simply, it is desirable that the electronics of portable/mobile devices simultaneously feature high variation-space and wide operation-space. Specifically, high variation-space refers to functionally error-free operation despite the high variations in

the prevailing conditions including Process (P), Voltage (V) and Temperature (T) variations, otherwise commonly and collectively abbreviated as PVT variations. Wide operation-space, on the other hand, refers to functionally error-free operation under a wide range of activity levels or workload requirements.

The need for electronic devices to accommodate high variation-space is well recognized within the electronics design community. For example, the International Technology Roadmap for Semiconductors (ITRS) [5] has projected the variations of pertinent electrical parameters with respect to the minimum feature size of CMOS fabrication processes. The specific parameters of interest extracted therefrom are tabulated in Table 1.1 below. For completeness, note that these parameters from ITRS are strictly for nominal V_{DD} voltage operation, and the parameters for lower voltages are unavailable; see later.

Table 1.1: International Technology Roadmap for Semiconductors (ITRS) 2011 [5]

Parameter		2011	2012	2013	2014	2015	...	2026
1	CMOS Fabrication Process	40nm	32nm	28nm	24nm	21nm	...	6.3nm
2	% Process Parameter Uncertainty	11%	12%	14%	15%	18%	...	38%
3	% V_t Variability; all sources	42%	42%	42%	47%	47%	...	79%
4	% V_{DD} Variability; on-chip	10%	10%	10%	10%	10%	...	10%
5	% Circuit Performance Variability	42%	42%	42%	45%	45%	...	60%
6	% Asynchronous-logic in chips*	19%	20%	22%	23%	25%	...	54%

* For asynchronous interfaces, e.g., globally asynchronous locally synchronous (GALS) etc.

From Table 1.1, it is evident that the finer the minimum feature size (equivalently, the more advanced the CMOS fabrication process node; row 1), the higher are the process variations (rows 2 and 3). For example, the threshold voltage (V_t) variations (at nominal V_{DD}) is projected to increase from 42% for the current-art 28nm process to 79% for the impending 6.3nm process in 2026 (row 3). An on-chip 10% voltage rail (V_{DD}) variation (as a result of noise and imperfect voltage regulation; row 4) is expected and this variation needs to be tolerated by the associated digital circuit/system. For said variations, the ensuing circuit

performance variability is not unexpectedly projected to increase from 42% today to 60% in 2026 (row 5). It is instructive to note that the variations of the evolving CMOS process projected by ITRS are largely for process and voltage variations and at room temperature operation. This temperature dependency (as embodied in the general PVT variations) is also well established and appreciated within the electronics design community, particularly devices which are high-power and/or high-speed/performance (e.g. microprocessor (μ Ps)). Devices that operate in environments outside the home/laboratory (for example the WSN placed in open spaces; see Chapter 4 later for a WSN designed for large temperature range (-55°C to 125°C) operation) and which are industrial and military grade, will need to operate with a large temperature variation, hence under higher variation-space conditions.

Put simply, the PVT variations tabulated in Table 1.1 are largely Process variations ('P' of PVT wherein V_t is the major parameter thereof), limited Voltage variations ('V' of PVT, with limited 10% V_{DD} variations) and without temperature variations ('T' of PVT is not considered). As delineated earlier, if temperature variations are considered, the overall circuit performance variability will be significantly increased [6]; see Chapter 4 later.

The aforesaid overall circuit performance variability will yet further increase if V_{DD} is reduced as a means to reduce power dissipation. To depict the effect of V_{DD} on power dissipation, consider the well established power dissipation expression [7] for a CMOS circuit:

$$\begin{aligned} P_{\text{Total}} &= P_{\text{Dynamic}} + P_{\text{Short-Circuit}} + P_{\text{Leakage}} \\ &= \alpha C_L V_{DD}^2 f + I_{\text{Short-Circuit}} V_{DD} + I_{\text{Leakage}} V_{DD} \end{aligned} \quad (1.1)$$

where P_{Total} is the total power,

P_{Dynamic} is the dynamic power,

$P_{\text{Short-Circuit}}$ is the short-circuit power,

P_{Leakage} is the leakage power,

α is the switching activity,

C_L is the effective load capacitance,

V_{DD} is the supply voltage,

f is the switching frequency,

$I_{\text{Short-Circuit}}$ is the average short-circuit current, and

I_{Leakage} is the average leakage current.

Amongst the constituent powers, P_{Dynamic} is considered the useful power for computation while $P_{\text{Short-Circuit}}$ and P_{Leakage} are the wasted powers. From eqn. (1.1), it is apparent that the power dissipation of a CMOS circuit is greatly reduced if the supply voltage V_{DD} is reduced. Specifically, for P_{Dynamic} , V_{DD} being a quadratic function thereof, has the greatest impact amongst all controllable design parameters. $P_{\text{Short-Circuit}}$ and P_{Leakage} , on the other hand, can simply be reduced by decreasing V_{DD} , although the relationship thereto is linear.

The reduction in power by scaling V_{DD} is, however, not obtained without cost. Specifically, with reduced V_{DD} , the available current for switching the output of a transistor is also reduced, resulting in a rapid rise in circuit delay. To depict this, Fig. 1.1 plots our simulation results of the delay and total power dissipation of a CMOS inverter (@130nm; RVT process option (see below)) versus V_{DD} @50kHz switching rate. The delay herein is defined as the sum of high-to-low (t_{HL}) and low-to-high (t_{LH}) switching delays, where the low and high levels are defined as 10% and 90% V_{DD} respectively. Three process options, namely LVT (low- V_t ; $|V_t| \approx 0.25V$), RVT (regular- V_t ; $|V_t| \approx 0.4V$) and LP (low power, high- V_t ; $|V_t| \approx 0.55V$) are considered, and for sake of easy comparison, the plots are normalized to the RVT inverter @nominal $V_{DD} = 1.2V$.

Fig. 1.1: Delay and power characteristics of inverters (@50kHz) 130nm CMOS, for different process options (LVT, RVT and LP); normalized with respect to RVT @1.2V

In Fig. 1.1, the V_{DD} range is divided into two regimes: the super-threshold voltage regime (super- V_t , including nominal voltage and near- V_t voltage regimes) and the sub- V_t voltage regime. The attributes of these regimes are as follows:

- (a) Nominal voltage regime: $V_{DD} \gg V_t$

The transistor is in strong inversion, and the circuit dissipates high power and its delay is short (high speed);

- (b) Near- V_t voltage regime: $V_{DD} \sim V_t$

The transistor is in moderate inversion, and the circuit dissipates medium power and its delay is moderate (moderate speed); and

- (c) Sub- V_t voltage regime: $V_{DD} < V_t$

The transistor is in weak inversion, and the circuit dissipates very low power and its delay is extremely long (extremely low speed).

It can be observed from Fig. 1.1 that by reducing V_{DD} from nominal to near-/sub- V_t , the total power dissipation of an inverter is substantially reduced. For example, when V_{DD} is scaled from nominal $V_{DD}=1.2V$ to deep sub- V_t , $V_{DD}=0.15V$, the total power dissipation of the inverter based on the LVT and RVT processes is reduced by $\sim 43\times$ and $\sim 51\times$ respectively. Similarly, when V_{DD} is scaled from 1.2V to 0.2V (instead of 0.15V), the total power of the inverter based on the LP process is $\sim 37\times$ lower, and it fails to operate when $V_{DD} < 0.2V$. The effect of scaling V_{DD} is even more dramatic to delay, particularly in near-/sub- V_t . For example, for V_{DD} scaled from 1.2V to 0.15V, the delay of the LVT and RVT inverter is $\sim 689\times$ and $\sim 4262\times$ longer respectively, and similarly, for V_{DD} scaled from 1.2V to 0.2V, the delay of the LP inverter is $\sim 58819\times$ longer.

It is hence evident that for low-power/ultra low-power applications, operating digital circuits therein in the near-/sub- V_t regime is highly desirable from a power perspective, provided the ensuing long delay (low speed/low computation rate) can be tolerated. Conversely, when the delay of the digital circuit is required to be short (high speed/high computation rate), the voltage would need to be scaled upwards – this is Dynamic Voltage Scaling (DVS) [8]; see later. Put simply, operating in the near-/sub- V_t regime is particularly attractive to portable/mobile devices for ubiquitous computing, where the energy source (usually from a battery) is highly constrained and/or unreliable (in the sense of being highly variable), and the workload/computation requirement is modest and varying; see Chapter 4 later for such a device – a WSN.

Despite the attractiveness of operating in the near-/sub- V_t regime where applicable, the digital circuit/system design to accommodate the lower V_{DD} voltage operation is challenging, particularly in the sub- V_t regime. This is because the effects of PVT variations on circuit

performance variability as delineated earlier become increasingly variable – to the point of virtually intractable. This performance variability between nominal and sub- V_t V_{DD} operation is well established and evident from their drain current equations given respectively in eqns. (1.2) [9] and (1.3) [10] below; these are the simplified equations and a more comprehensive delineation will be provided in Chapter 2 later.

$$I_{\text{nom}} = v_{\text{sat}} C_{\text{ox}} W \left(V_{\text{GS}} - V_t - \frac{V_{\text{DSAT}}}{2} \right) \quad (1.2)$$

where v_{sat} is the saturation velocity for short-channel devices,

C_{ox} is the gate oxide capacitance per unit area,

W is the width of transistor,

V_{GS} is the gate source voltage,

V_t is the threshold voltage, and

$V_{\text{DSAT}} = \frac{L v_{\text{sat}}}{\mu}$ is the saturation drain voltage,

where L is the channel length of transistor, and

μ is the carrier mobility.

$$I_{\text{sub-}V_t} = \mu C_{\text{ox}} \frac{W}{L} (n - 1) V_{\text{th}}^2 \exp \left(\frac{V_{\text{GS}} - V_t}{n V_{\text{th}}} \right) \quad (1.3)$$

where n is the sub- V_t slope factor,

$V_{\text{th}} = \frac{kT}{q}$ is the thermal voltage,

where k is the Boltzmann constant,

T is the absolute temperature, and

q is the electron charge.

From (1.2) and (1.3), it can be seen that the parameters related to PVT for nominal and sub- V_t operation are respectively linear and exponential; note that process variations affect V_t , V_{DD} variations affect V_{GS} , and temperature variations affect both V_{th} and V_t . In other words, as the effects of PVT in sub- V_t are dominated by an exponential relationship as opposed to the

linear relationship in nominal V_{DD} , the former is significantly more severely affected than the latter. The degree is so severe that the variations in sub- V_t translate into intractable delay variations in a digital circuit; see our analytical derivations and measurements on prototype ICs in Chapter 3 and 4 later respectively.

In addition to the aforesaid high variation-space, it is also desirable that low power/ultra low-power portable/mobile devices (for ubiquitous computing) embody a wide operation-space attribute – a *dynamically* varying workload requirement. An example is a reported micro-controller unit (MCU) in a WSN [11] where the idle time of the MCU is >50% of the time and the computation speed/load varies for different functions. In such ‘ubiquitous computing’ devices, their design needs to simultaneously accommodate/adapt to high variation-space (under prevailing conditions possibly including intractable PVT in sub- V_t) and wide operation-space (comprising wide range of dynamically-varying workloads). Designing for said spaces is challenging, particularly where there is a need for low-power/ultra low-power operation. This is largely because to reduce power dissipation, the degree of delay (safety) margin would need to be compromised.

Digital Design Approaches for High Variation-Space and Wide Operation-Space

The concept of delay margin resides fundamentally with the operation modalities of the different digital circuit realization approaches, more specifically their data synchronization protocols. Fig. 1.2 below depicts the generic block diagrams of a digital pipeline stage realized based on three approaches, namely the prevalent (conventional) synchronous-logic (sync) approach, and the somewhat esoteric asynchronous-logic (async) Matched Delay (MD) [12] and Quasi-Delay-Insensitive (QDI) [13] approaches; see Chapter 2 later for a more in-depth review of the different digital circuit realization approaches.

Fig. 1.2: Generic block diagram of a pipeline stage realized in: (a) sync, (b) async MD, and (c) async QDI

The sync approach, as depicted in Fig. 1.2(a), embodies the single-rail logic circuit for computation, and flip-flops ('FF1' and 'FF2') for data registration where the FFs are controlled/timed by the global clock signal ('CLK'). Single-rail, as its denotation implies, refers to a specific logic representation of a binary data bit involving a single wire (and ground reference) with its associated low and high voltage levels are typically logic '0' (also data '0') and logic '1' (also data '1') respectively. As these logic levels represent valid data, the computation delay of a single-rail logic circuit (i.e. the delay to produce a valid data) cannot be derived from its output, thereby requiring its data synchronization to be performed independently with an assumption on the computation delay. This computation delay is in

general obtained by means of computer simulations of the circuit for the given operating conditions.

For error-free operation, the data synchronization period of a sync circuit (i.e. the period of the global ‘*CLK*’ signal, commonly known as the clock period) needs to be set longer than the (assumed) worst-case computation delay of the single-rail logic circuit therein. Further, this worst-case delay (hence the safety margin therein) has to be ascertained/assumed for the entire pipeline (encompassing all its constituent stages) and under all specified operating conditions – i.e. the *global* worst-case timing; global herein refers to the entire circuits/system under the same clock. In other words, with this general requirement for error-free operation and for the operation spaces for the portable/mobile devices delineated earlier, the sync circuit not unexpectedly requires a *large* delay safety margin to accommodate its *global worst-case* timing. For example, in [14], a very large delay safety margin of $\sim 200\times$ was reportedly allowed for in a sync device under sub- V_t operation to accommodate the PVT variations; also see our analytical derivation and Monte Carlo simulations in Chapters 3 and 4 later respectively. One reported method that attempts to reduce the size of the safety margin is Statistical Static Timing Analysis (SSTA) [15], where instead of worst-case delay, delay distributions (obtained by means of statistical simulations such as Monte Carlo simulations; see Chapter 4 later) are considered. However, SSTA greatly increases design and verification complexity, and the resulting circuits/system is still not guaranteed to be error-free; further, even by adopting SSTA, delay margins in sub- V_t is still likely to be large (see simulation results in Chapter 4 later) considering the intractable PVT.

Consider now the alternative to the sync approach, the somewhat esoteric MD and QDI async data synchronization protocols. The fundamental difference between the sync and async protocols is the replacement of the *global* clock signal of the former with a *local* handshake signal of the latter ('*HS*' in Fig. 1.2(b) and (c)). Particularly, for data registration, the FFs in the sync protocol timed by a global clock are replaced by latches 'timed' by a local handshake signal ('L1' and 'L2' in Fig. 1.2(b) and (c)).

In async MD, the data computation, as in the sync protocol, involves the single-rail logic circuit. However, instead of relying on the sync *global* clock signal for data synchronization, the async MD conversely employs a *local* delay element ('Matched Delay' in Fig. 1.2(b)) whose delay is designed to match the computation delay of the associated single-rail logic circuit, hence the denotation 'Matched Delay'. Because of its local handshake signal, an advantage of the async MD over its sync counterpart is its innateness to provide for fine-grain clock gating (from a sync perspective), where every logic/pipeline stage is controlled by the 'localness' of its own 'clock'. This contrasts, as delineated earlier, with the sync protocol whose clock is timed according to the worst-case *global* conditions. Put differently, the MD protocol innately provides unique 'opportunities' for realizing low power techniques (such as power gating to reduce the wasted powers when the circuit idles) in a fine-grain manner; see Chapter 3 later for a novel fine-grain power gating technique for the async MD protocol. The sync, on the other hand, has to implement this in a much more coarse-grain manner depending on the size of the circuits/system that share the same clock and for the entire circuits/system thereof.

From an operation robustness point of view, as *local variations* (in the form of PVT) exist between the delay element and its associated single-rail logic circuit, a certain amount

of delay safety margin is still nevertheless needed in an async MD circuit [12]. This delay margin, similar to its sync counterpart, needs to be derived and is likely to be large/extreme in view of the intractable PVT in sub- V_t . This is particularly the case as the delay element is typically a simple inverter chain, where its variations and that of its associated single-rail logic circuit are likely to be different under PVT variations. The margin nevertheless is likely to be smaller than the sync due to the ‘localness’ of the matched delay element. Overall, it can thus be argued that the async MD is advantageous for realizing low-power circuits/system at nominal V_{DD} by leveraging on its local fine-grain synchronization protocol and this advantage diminishes in ultra low-power sub- V_t operation due to the ensuing large delay margins required thereto.

Consider finally the async QDI approach, whose salient difference from its sync and async MD counterparts is the embodiment of a multi-rail logic circuit (typically dual-rail logic as shown in Fig. 1.2(c)) for data computation. Dual-rail, as its denotation implies, refers to a specific logic representation where a binary data bit involves two wires (Data True ($D.T$) and Data False ($D.F$); and ground reference) with their associated voltage levels. Table 1.2 tabulates the dual-rail encoding, where both $D.T$ and $D.F$ are initially at logic ‘0’ (i.e. No Data). After computation, only one of the wires will evaluate to logic ‘1’ to indicate either a valid data ‘0’ ($D.F = 1$) or a valid data ‘1’ ($D.T = 1$); both wires at ‘1’ is not allowed as this is an invalid state. Put simply, data validity (and conversely its absence) is innately encoded in a dual-rail logic circuit. In contrast, the single-rail logic (used in sync and async MD) does not possess this attribute.

Table 1.2: The Dual-Rail Data Encoding

	$D.T$	$D.F$
No Data	‘0’	‘0’
Valid Data ‘0’	‘0’	‘1’
Valid Data ‘1’	‘1’	‘0’
Invalid	‘1’	‘1’

By means of a completion detection circuit ('CD' in Fig. 1.2(c) – in its simplest form, a 2-input OR gate for the two wires for each dual-rail bit, where the assertion of the OR gate indicates the arrival of a valid data), the computation delay of a dual-rail logic circuit is *physically ascertained* under the prevailing conditions including under any PVT variations. As data synchronization is subsequently performed following the completion detection by the local handshake signal ('HS' in Fig. 1.2(c)), *no* delay safety margin is thus required by the async QDI. In other words, the accommodation of the computation delay is idiosyncratic of the QDI handshake protocol. Hence, the ensuing error-free operation is *unconditional* (save the isochronic fork timing [16]) regardless of the variations in its computation delay.

Viewed collectively, error-free operation in both sync and async MD is *conditional* as the computation delay of their single-rail logic cannot be ascertained, while in the async QDI, error-free operation is *unconditional* (save the isochronic timing) as its delay can be ascertained. Thus, from a robustness point of view, an async QDI circuit lends itself naturally to sub- V_t operation given the virtually intractable PVT thereof. In addition, as an async QDI circuit will innately adapt to the prevailing conditions, this potentially leads to shorter delay (and potentially lower power/energy) than the sync if the delay of the latter is limited by the global worst-case condition requiring very large delay safety margin; see Chapters 3 and 4 later. When compared to the async MD, it can be argued that an MD circuit can also, to a certain extent, adapt to the varying conditions if the delay element embodies the same variations. However, as delineated earlier, as the circuit of the delay element is typically different from the single-rail logic, there will be a mismatch between their variations. In view of this mismatch and the virtually intractable PVT in sub- V_t , a large delay safety margin would also be required, although the extent thereof is likely to be smaller than the sync case.

We will now henceforth limit our delineation (and comparison thereof) to that between the sync and the async QDI for sub- V_t operation (unless stated otherwise). This is also in part because the sync is presently the most prevalent (‘standard’) protocol adopted by the design community and QDI is the most robust async protocol (save the Delay-Insensitive (DI), which is not used in practical designs; see Chapter 2 later for a review of the different async protocols). For completeness, it is interesting that async signaling is projected by ITRS to be increasingly adopted (refer to row 6 in Table 1.1).

Despite the potential advantages of the async QDI in terms of its unconditional robustness, it is well established that it suffers from higher overheads than the sync, including IC area, and potentially generic circuit delay, and generic circuit power (i.e. delay and power without considering the delay safety margins). This is largely a consequence of the modality of dual-rail logic circuit (as opposed to the sync single-rail logic circuit), and, in part, to the overheads associated with completion detection; see Chapter 4 later for a novel QDI protocol with reduced completion detection overheads suitable for sub- V_t operation. Although somewhat contentious, it is generally accepted within the electronics design community that at nominal V_{DD} operation with small PVT variations, sync is advantageous over async QDI in terms of delay, power, and IC area. As delineated earlier, this sync advantage diminishes as the PVT variations increase (due to the ensuing larger delay margin). At sub- V_t where PVT becomes virtually intractable, the advantages of sync diminish further, possibly to the point where QDI becomes advantageous. The possible advantages are not just power and delay but unconditional robustness as well, whilst the IC area disadvantage of QDI will largely remain (due to $\sim 2\times$ hardware the dual-rail over the single-rail; see Chapter 4 later for a novel logic style coined ‘Pre-Charged-Static-Logic’ (PCSL) that mitigates the said generic overheads of

QDI). In short, to our knowledge, at this juncture, there is no general consensus as to which approach is advantageous *and* at what juncture in terms of variation and operation spaces.

A further advantage of QDI is that as its timing is implicit and innate, there is no intervention needed to adjust the timing or delay of the QDI circuit – it runs as fast as the prevailing conditions permit. Put differently, scaling V_{DD} as a means to save power dissipation (see eqn. (1.1) earlier) to accommodate the varying operation-space simply involves ‘dialing-up’ or ‘dialing-down’ the V_{DD} voltage for the given prevailing conditions, without need to consider ‘clocking’ rates; this innate accommodation of timing extends not only to PVT variations but also to workload/throughput – potentially full variation-space and full operation-space. This is the well-known ‘Dynamic-Voltage-Scaling’ (DVS); nevertheless, the issue of overheads remains (see Chapter 4 later for a novel DVS control scheme exploiting the QDI handshake with very low overheads). Conversely, in sync, the scaling of V_{DD} for the same includes a timing component. Specifically, when V_{DD} is ‘dialed-up’, the clock frequency can be increased, and the converse when V_{DD} is ‘dialed-down’. This is the well-known ‘Dynamic-Voltage-Frequency-Scaling’ (DVFS) [17] and from the practical perspective, there are several aspects to consider.

First, for sake of error-free operation, the clock signal in a sync DVFS system usually has to be stopped (thus its operation interrupted [18]) during a V_{DD} transition. This interruption in operation (with its ensuing performance penalty) will in part limit the frequency of V_{DD} transitions. Further, the computation can be resumed only when the clock (and clock infrastructure) is stable. This typically involves allowing several thousand clock cycles, hence some delay, upon adjustment to a new clock frequency. Second, the adjustment of the sync clock is a not necessarily trivial. This typically involves either

software adjustment of a clock divider or physical adjustment of a clock oscillator circuit. Third, given the worst-case timing based operation modality of sync, implementing DVFS requires the computation delay of the circuit to be pre-characterized at multiple V_{DD} levels under the worst-case of different conditions. This inevitably adds not only to the design complexity but the substantial pre-characterization effort. The degree of effort/complexity will escalate when sub- V_t V_{DD} s are involved given the virtually intractable PVT variations thereto [15].

Fourth, as the computation delay of a sync circuit cannot be ascertained under the prevailing conditions, a sync DVFS system thus cannot exploit ‘timing slacks’ created by a benign PVT variation. For example, in sub- V_t operation, an increase in temperature generally reduces the delay of the circuit, thereby the added ‘timing slack’. However, without the ability to ascertain its computation delay under the prevailing conditions, the sync circuit is unable to exploit the more benign conditions unless there is a means of physically measuring the new conditions, for example, by means of an environmental sensor; see Chapter 4 later for a sync DVFS system with a temperature sensor. This will however inevitably complicate the design, and the ensuing overheads may defeat any power/energy savings gained. Further, some ambiguity remains, for example, PVT variations that are difficult to ascertain such as aging, etc.

In summary, there are strong motivations to investigate digital design approaches that provide for the realization of robust portable/mobile low-power/ultra low-power devices for ubiquitous computing. The requirements of these devices include error-free operation under high variation-space (including PVT variations and the requirement of low-voltage sub- V_t operation) and wide operation-space (varying workload requirement including long period of

idle state), and yet with low hardware and power overheads. At this juncture, the most efficacious design method remains an open debate amongst the digital design community, particularly the adoption of various data synchronization protocols and novel design methodologies thereto to accentuate their attributes, particularly in variation and operation spaces and with low-power/ultra low-power operation.

1.2 Objectives

In view of the aforesaid motivations, the overall objectives of this thesis pertain to the design of low-power/ultra low-power *high variation-space* and *wide operation-space* digital electronics for portable/mobile applications. The specific signaling protocols adopted are the async MD and QDI, and the proposed designs herein are benchmarked against the prevalent conventional sync. The objectives can be divided into two parts.

The first part pertains to an investigation (and ensuing circuit design thereof) into the efficacy of the application of the async protocols for realizing *low-power/ultra low-power* digital circuits/systems. The specific objectives are:

- (i) To investigate (and propose) a novel fine-grain power gating technique, with low overheads, to reduce wasted powers (short-circuit and leakage powers) based on the 4-phase async MD protocol;
- (ii) To propose and derive a set of simple yet insightful analytical equations for estimating to the first-order the delay variations (due to V_t , V_{DD} and temperature variations; thus the required delay safety margin) of digital circuits in sub- V_t operation. Thereafter, to investigate and benchmark the efficacy of async QDI

against its sync counterpart for ultra low-power sub- V_t operation, with considerations for the extreme/virtually intractable PVT variations thereof.

The second part pertains to the design and realization of an adaptive DVS circuits/system for an ultra low-power WSN (operating in sub- V_t) based on the async QDI protocol and its benchmarking against its sync DVFS counterpart. The specific objectives are:

- (iii) To propose and realize in monolithic form (IC prototype) a novel Sub- V_t Self-adaptive V_{DD} Scaling (SSAVS) system based on the async QDI to realize the aforesaid ultra low-power WSN. Thereafter, to benchmark (on the basis of measurements on said IC prototypes) in terms of delay and power/energy, the proposed async QDI system against its sync DVFS counterpart under high variation-space and wide operation-space;
- (iv) Further to (iii), to investigate a means to reduce the overheads of the adopted QDI protocol embodied in the SSAVS for wide operation-space, particularly by exploiting the existing signaling of QDI; and
- (v) Further to (iii) and (iv), to propose a novel simplified QDI protocol (over the standardized QDI) to reduce the overheads associated with completion detection and with implicit timing.

1.3 Contributions

A number of contributions are made in this thesis, and they are now succinctly delineated in turn.

The contributions pertaining to objectives (i) and (ii) in the first part include:

- (a) The proposal of a fine-grain power gating methodology to reduce the short-circuit and leakage powers of an MD pipeline (applicable to three different gating configurations) over a wide operation-space. By exploiting the 4-phase handshake protocol, the ensuing overhead of the proposed power gating is low, specifically one inverter (per pipeline stage) and $<15\%$ delay;
- (b) To quickly estimate to the first-order the delay variations (due to V_t , V_{DD} and temperature variations; thus the required delay safety margin) of digital circuits in sub- V_t , the proposal and derivation of a set of simple yet insightful analytical equations. The derived equations are verified by simulations and shown to be accurate for first-order estimations (with an inconsequential worst-case error of $<12\%$);
- (c) Following (b), the benchmarking of the sync (with delay safety margins estimated by the derived equations) against the async QDI (with self-completion detection), on the basis of adder circuits, it is ascertained that neither the sync nor the async QDI is particularly advantageous in all conditions.

The contributions pertaining to objectives (iii), (iv) and (v) in the second part include:

- (d) The proposal of a Sub- V_t Self-Adaptive V_{DD} Scaling (SSAVS) system for a high variation-space and wide operation-space Wireless Sensor Network (WSN) with

the objective of lowest possible power dissipation (in sub- V_t operation), yet high robustness and with minimal overheads. The effort to achieve the lowest possible power operation is essentially DVS – by means of self-adjusting V_{DD} to the minimum voltage (within 50mV) for any given prevailing conditions. High robustness is achieved by adopting the QDI protocol;

- (e) The proposal of ‘Pre-Charged-Static-Logic’ (PCSL) logic style for the design of QDI logic cells that feature full-range DVS. Further to (d), the high robustness thereof is also in part achieved by the embodiment of our proposed PCSL. When our proposed PCSL is benchmarked against competing async logic styles suitable for sub- V_t , the PCSL is ascertained to be the most competitive in terms of energy/operation (E_{per}), delay and IC area;
- (f) The design of the filter bank (comprising PCSL cells) embodied in the SSAVS and shown to be ultra low-power and highly robust. The proposed async SSAVS is thereafter benchmarked against its conventional sync DVFS counterpart for two scenarios, and their merits and disadvantages delineated;
- (g) In conjunction with (f), to reduce the overheads of the QDI protocol in realizing SSAVS in wide operation-space and not requiring a priori information on the width of the operation-space or any other parameter, the proposal for the exploiting of the already existing request and acknowledge signals of the QDI protocols. The ensuing overhead of the SSAVS is very modest;

- (h) Further to (d) to (g), to yet further reduce the overheads (in terms of power/energy and area), the proposal of a hardware-simplified version of the standardized QDI, coined ‘pseudo-QDI’ herein, with an implicit timing for the aforesaid SSAVS. Analytical formulation to depict that said implicit timing is easily satisfied whilst ensuring robust operation, and verification of said robustness by measurement on prototype ICs. By means of the pseudo-QDI, the ensuing energy and area are significantly reduced by $\sim 40\%$ and $\sim 1.34\times$ respectively compared to the standardized QDI.

1.4 Organization

This thesis is organized as follows. Chapter 1 describes the motivation, objectives, contributions and organization of this thesis.

Chapter 2 presents a literature review of low-power/ultra low-power digital design and serves as a preamble to Chapters 3 and 4. The review emphasizes ultra low-power sub- V_t operation and the associated formidable challenges (over super- V_t operation); the review also includes power gating for low-power. For robust operation in sub- V_t , we review four logic families – the static logic, pass transistor/transmission gate logic, pseudo-NMOS logic and dynamic logic, and two digital design approaches/signalling protocols – the sync and the async. Amongst the reviewed async protocols, QDI async is the most practical and robust for sub- V_t operation due to its unconditional error-free operation under large PVT variations.

Chapter 3 describes a low-power fine-grain power gating technique for the async MD pipeline to reduce its wasted power. The proposed technique (with three different gating configurations) is benchmarked against the MD pipeline without power gating over a wide

operation-space. For ultra low-power sub- V_t , we propose and derive a set of simple yet insightful analytical equations to estimate to the first-order delay variations (due to V_t , V_{DD} and temperature variations) of digital circuits operating in sub- V_t . The derived equations are verified by simulations to show that they are accurate for first-order estimations. We thereafter benchmark, by means of adder circuits, the sync (with delay safety margins estimated by the derived equations) against the async QDI (with self-completion detection).

Chapter 4 describes a Sub- V_t Self-Adaptive V_{DD} Scaling (SSAVS) system for the Signal Processor module in a WSN based on a proposed methodology within the QDI async approach, and with a novel in-situ self-adjusting V_{DD} means. The proposed design methodology, coined ‘Pre-Charged-Static-Logic’ (PCSL) logic style is compared against competing logic styles in terms of E_{per} , delay and IC area. The proposed SSAVS system for the WSN is demonstrated by means of application to a filter bank. The filter bank embodied in the SSAVS is shown to be ultra low-power and highly robust. It is subsequently benchmarked against its conventional sync DVFS system counterpart for two scenarios, and their merits and disadvantages delineated. To address the usual power/energy overheads associated with standardized QDI, we further propose a hardware-simplified version of QDI (coined ‘pseudo-QDI’) with an easy-to-met implicit timing. We formulate and analyze this implicit timing, and by means of measurements on prototype ICs, we demonstrate the extreme robustness of pseudo-QDI in sub- V_t under very high variation-space. We further depict the E_{per} and IC area advantages of pseudo-QDI over its standardized QDI counterpart.

Chapter 5 concludes the thesis and recommends pertinent topics for further research.

Chapter 2 Literature Review

Very-Large-Scale-Integration (VLSI) digital circuits/systems are typically highly complex systems, in some cases embodying billions of transistors. To manage the design complexity, a digital circuit/system is often conceptualized at different hierarchical levels of abstractions. At the highest level, there is the architecture that describes the functionality of the circuit/system, e.g. a computer system from the programmer's point of view. Immediately below is the micro-architecture level (also known as the Register Transfer Level (RTL)) that implements the model of an architecture into a specific physical structure of the hardware. Below this is the logic level that implements the micro-architecture into a specific array of logic modules/gates such as a logic cell library. Electronic Design Automation (EDA) tools are usually employed to realize the transformation (logic synthesis) from the micro-architecture to the logic modules/gates. Below the logic is the circuit level that implements the logic into a specific arrangement of transistors such as the various logic families/styles. At the bottom of the hierarchy is the physical level that involves the specific sizing/drawing of each transistor in the circuit, the physical layout.

At each level of abstraction, a designer faces a plethora of design choices/implementations with different performance/power dissipation/operation robustness and other tradeoffs. In this thesis, we will explore/investigate some of these tradeoffs from the micro-architecture level downwards, in part, by means of proposing novel design approaches/realizations at different abstraction levels (and benchmark against their competing approaches/realizations where appropriate); see a novel WSN example in Chapter 4 later embodying said abstraction levels.

This chapter presents a literature review of low-power digital circuit design and serves as a preamble to Chapters 3 and 4. The review emphasizes ultra low-power sub- V_t operation and the associated formidable challenges (over super- V_t operation), particularly in view of operational robustness in high variation-space (including DVS where V_{DD} ranges from nominal to sub- V_t) and wide operation-space. To better depict these challenges, we augment herein our various simulations to illustrate the ensuing delay under PVT variations. This review includes a review of two general digital design approaches/signalling protocols, the sync and the async, including an overview of their idiosyncratic attributes when operating in the sub- V_t regime. The async protocols of interest here, as briefly discussed in Chapter 1, are the Matched Delay (MD) and Quasi-Delay-Insensitive (QDI). As the attributes of QDI lend itself readily to high variation-space and wide operation-space, including at sub- V_t operation, this review will emphasize the QDI async protocol and its idiosyncrasies thereto.

2.1 Low-Power and Ultra Low-Power Sub- V_t

Design techniques to reduce power/energy dissipation of digital circuits, although an established art, continues to attract considerable interest within the digital design community. This is largely because of the increasing proliferation of portable/mobile electronic devices, where their energy source is limited, and the ever-increasing demand for extended battery life (between charges). The underlying principle for any low-power/power reduction technique is to avoid dissipating unnecessary power/energy, or equivalently, all power dissipation should be useful for computation in a digital circuit/system. At this outset (and as delineated in Chapter 1), one of the primary attraction of sub- V_t operation is the potential of operation at the theoretical minimum E_{per} . The E_{per} of a digital logic circuit in sub- V_t [10] involves both the dynamic energy ($E_{Dynamic}$) and the leakage energy ($E_{Leakage}$) (note that there is no short-circuit current/energy in sub- V_t as the transistors therein are never fully on (see later), and it is

further assumed that all the current/energy during transistor switching are captured by the term C_{eff} ; see (2.1) below):

$$E_{\text{Dynamic}} = C_{\text{eff}} V_{\text{DD}}^2 \quad (2.1)$$

where C_{eff} is the total effective switched capacitance.

$$\begin{aligned} E_{\text{Leakage}} &= I_{\text{Leakage}} V_{\text{DD}} t_d \\ &= W_{\text{eff}} I_{\text{off}} V_{\text{DD}} t_d \\ &= W_{\text{eff}} \mu C_{\text{ox}} \frac{W}{L} (n-1) V_{\text{th}}^2 \exp\left(\frac{-V_t}{nV_{\text{th}}}\right) V_{\text{DD}} t_{\text{d,inv}} L_D \\ &= W_{\text{eff}} K C_{\text{output,inv}} L_D V_{\text{DD}}^2 \exp\left(\frac{-V_{\text{DD}}}{nV_{\text{th}}}\right) \end{aligned} \quad (2.2)$$

where $I_{\text{Leakage}} = W_{\text{eff}} I_{\text{off}}$ is the total leakage current.

where W_{eff} is the total effective

leakage width, and

I_{off} is the transistor off current;

see eqn. (2.6) later, and

t_d is the critical path delay; see eqns. (3.1) and

(3.2) later.

E_{per} in sub- V_t is expressed in eqn. (2.3) [10] below:

$$\begin{aligned} E_{\text{per}} &= E_{\text{Dynamic}} + E_{\text{Leakage}} \\ &= V_{\text{DD}}^2 \left(C_{\text{eff}} + W_{\text{eff}} K C_{\text{output,inv}} L_D \exp\left(\frac{-V_{\text{DD}}}{nV_{\text{th}}}\right) \right) \end{aligned} \quad (2.3)$$

From eqn. (2.3), it can be seen that, as V_{DD} is reduced in sub- V_t , $E_{Dynamic}$ decreases while $E_{Leakage}$ increases (due to the rapid increase in sub- V_t circuit delay – the exponential term ‘ $\exp\left(\frac{-V_{DD}}{nV_{th}}\right)$ ’ dominates $E_{Leakage}$ in sub- V_t), and there exists a minimum energy point. To illustrate an example, we illustrate in Fig. 2.1 our simulations of E_{per} of a 30-inverter chain versus V_{DD} scaling (from nominal $V_{DD} = 1.2V$ to deep sub- V_t $V_{DD}=0.15V$ with an activity factor = 0.1; results normalized to the RVT inverters @nominal $V_{DD}=1.2V$, 0.001pJ, and the same 130nm CMOS LVT, RVT and LP process is used as that shown earlier in Fig. 1.1). The figure clearly depicts the minimum energy point of the LVT and RVT inverter chain occurring @ $V_{DD}=0.3V$. On the other hand, the minimum energy point of the LP 30-inverter chain occurs @ $V_{DD}<0.2V$, which is not depicted as the inverters fail to operate for $V_{DD}<0.2V$.

Fig. 2.1: E_{per} characteristics (normalized to the RVT design @ nominal $V_{DD}=1.2V$) of a 30-inverter chain (activity factor = 0.1) in 130nm CMOS process with different V_t options: LVT, RVT, and LP

In general, design techniques for low-power and ultra low-power sub- V_t can be classified [7] into design-time techniques and operation-time (standby-time and run-time) techniques. These will now be reviewed in turn.

2.1.1 Design-time Techniques

Design-time techniques, as the name implies, pertains to techniques at the juncture of circuits/system design, i.e. before/during the physical realization of the circuits/system. They include, at the architectural level, parallelism [19] and dedicated hardware/architecture [20]; and at the circuit/physical level, logical optimization and technology mapping [21].

Parallelism [19] refers to the replication of a single logic function into multiple copies (in hardware) and this offers an opportunity for lowering V_{DD} to achieve power reduction. The underlying principle is that power dissipation (assuming dynamic power dominates) scales quadratically with V_{DD} (see eqn. (1.1) earlier) while delay scales linearly/super-linearly with V_{DD} . Thus by employing multiple copies of the same logic function and enabling them to process data in parallel (with input steering and output rejoining), the same throughput can be achieved with reduced V_{DD} of each individual function, and the lower overall power dissipation is obtained. The potential power reduction gain can be costly in terms of overheads. The delay overhead associated with the input steering and output rejoining increases as the number of hardware copies increases. Further, the IC area cost of parallelism can be significant, and the associated leakage power increases rapidly (with each hardware copy).

Dedicated hardware/architecture refers to the use of dedicated hardware (e.g. an accelerator [20]) to meet a specific computation requirement. It is well established that a dedicated architecture (e.g. a dedicated Fast-Fourier Transform (FFT) machine as opposed to a general-purpose architecture such as that in a general-purpose microprocessor) can significantly improve the overall computational efficiency, thus reducing the ensuing

power/energy dissipation. However, the tradeoff is reduced flexibility, which needs to be carefully considered and ascertained at the juncture of the design-time.

Logical optimization and technology mapping [21] are procedures in the logic synthesis process where a logic function is being physically realized, i.e. synthesized into library cells. Specifically, logical optimization (in the context of low-power/power reduction techniques) refers to the *technology-independent* process of mapping a logic function to a network/topology of logic gates that minimizes its power/energy dissipation. This process may involve logic restructuring to reduce spurious transitions, algebraic transformation to simplify logic expressions, and/or buffer insertion to balance various logic paths, etc.

Technology mapping refers to the *technology-dependent* process of selecting a specific implementation (from a cell library) for each of the logic gate determined by logical optimization. In view of low-power, this may involve selecting transistor sizes, logic families/styles, process options (e.g. different V_t options), etc., that minimize the power/energy dissipation of the circuit.

2.1.2 Operation-time Techniques

Operation-time techniques embody both standby-time and run-time. Examples of operation-time techniques include adaptive body bias, DVS, power gating, etc.

Adaptive body bias [22], [23], as the name implies, refers to the adaptive adjustment of the threshold voltage (V_t) of transistors in a circuit (by controlling their body bias) to reduce its power/energy dissipation. For example, when the circuit requires high performance, the V_t of the transistors is reduced (by means of forward body bias) to provide higher switching

current, hence lower circuit delay. On the other hand, when the circuit idles (in standby) or the workload requirement is relaxed, the V_t of the transistors can conversely be increased (by means of reverse body bias), and the ensuing circuit leakage current/power is reduced and the delay increased. To realize adaptive body bias, the transistor body terminal would need to be accessible, hence requiring a triple-well fabrication process. In addition, to provide reverse body bias (for increasing V_t), supply voltages in addition to V_{DD} and ground are required – one higher than V_{DD} (for PMOS) and another lower than ground (for NMOS). In short, although adaptive body bias is well established and has some degree of acceptance within the digital design community, its implementation is applicable to specific fabrication processes and the overheads can be high. Further, it has been suggested [7] that the efficacy of body bias reduces with technology scaling – it may be effective only to relatively dated technology nodes of $\geq 90\text{nm}$ minimum feature size. For completeness, this technique is not applicable to SOI (no body terminal) and the emerging finFET [24] (ineffective due to the isolation of its channel above the substrate) transistors.

DVS [25]-[27] involves adjusting V_{DD} from nominal voltage downwards when the operating conditions permit, for example when the workload is reduced. From a practical perspective, DVS is probably the most effective technique to reduce power dissipation, and the influence on total power was expressed in eqn. (1.1) earlier. As described in Chapter 1, despite its potential, DVS at this juncture remains largely in the super- V_t voltage regime due to the need to ensure operational robustness. At the lowest V_{DD} of the DVS range, V_{DD} is reduced to sub- V_t [28]-[30] where the power dissipation is very low with its associated extremely long delay (see Fig. 1.1 in Chapter 1). In terms of energy dissipation, sub- V_t operation may be particularly attractive – it has been shown [31], [32] that a digital circuit

achieves theoretical minimum energy per operation (E_{per}) in sub- V_t ; see eqn. (2.3) and Fig. 2.1 earlier.

While DVS is effective in reducing the total (useful and wasted) power dissipation of a digital circuit, the associated cost in terms of delay of the circuit can be very severe, particularly the rapidly increasing delay below V_t – equivalently substantially reduced workload capacity; see Fig. 1.1. Consequently, the designer needs to carefully trade achievable power reduction with reduced throughput/workload. It is not surprising that some researchers advocate that an alternative to DVS is to operate at nominal speed (so that the computation is completed quickly including even faster operation by means of parallelism delineated earlier) and then cease operation otherwise when conditions allow. This is equivalent to the workload alternating between operation (typically full-load) and idling, and power gating [33]-[35] is applied to cease operation, thereby reducing power dissipation. At this juncture, there is no general consensus if DVS or power gating yields the best outcomes, and this remains a continuing debate within the electronics design community. It is however likely that due to the highly varied requirements of different digital circuits/systems, this debate will continue for some time. The work presented in this thesis, embodying investigations and proposals of new/different design approaches/signaling protocols for DVS and power gating, and providing some new/novel perspectives, will inevitably add to this continuing debate.

In view of the specific interest in this PhD thesis, we will now more comprehensively review ultra low-power digital sub- V_t operation and the power gating technique.

2.1.3 Ultra Low-Power Sub- V_t

As delineated earlier, sub- V_t operation is highly worthy where applicable because of its potential of theoretical minimum energy per operation (E_{per} , hence the highly desirable potential of maximum energy efficiency) despite the extremely long delay drawback. At this juncture, it is generally agreed within the digital electronics community that designing digital logic for sub- V_t operation presents formidable challenges/issues not normally considered in nominal V_{DD} /super- V_t operation. The two most important challenges/issues will be described in this section – first, relating to the choice of logic families, and second, relating to the choice of design approaches/signaling protocols.

Sub- V_t operation for digital circuits essentially involves operating the Metal-Oxide-Semiconductor (MOS) transistors therein in the *weak inversion* region, i.e. $V_{\text{DD}} < V_t$ where the ensuing drain current to switch the output is the sub- V_t current ($I_{\text{sub-}V_t}$) expressed in eqn. (1.3)¹ [10] earlier. Put simply, in sub- V_t , the transistors are never fully ‘on’.

The aforesaid first design challenge/issue, relating to the choice of logic families for sub- V_t operation, is to accommodate the degraded on/off current ratio ($I_{\text{on}}/I_{\text{off}}$, given by eqn. (2.5)/eqn. (2.6) or eqn. (2.7) below). This is due to the extremely low I_{on} current in sub- V_t . In addition, choosing appropriate logic families in sub- V_t further involves consideration for the effect of global and local process variations (e.g. V_t variations) that may

¹ A more comprehensive equation is given in eqn. (2.4) [36]. In this equation, η , the Drain-Induced-Barrier-Lowering (DIBL) coefficient, and the term, $\left(1 - \exp\left(\frac{-V_{\text{DS}}}{V_{\text{th}}}\right)\right)$, the low V_{DS} current roll-off (i.e. when V_{DS} drops to within a few times of V_{th}), are not included in simplified eqn. (1.3).

$$I_{\text{sub-}V_t} = \mu C_{\text{ox}} \frac{W}{L} (n-1) V_{\text{th}}^2 \exp\left(\frac{V_{\text{GS}} - V_t + \eta V_{\text{DS}}}{n V_{\text{th}}}\right) \left(1 - \exp\left(\frac{-V_{\text{DS}}}{V_{\text{th}}}\right)\right) \quad (2.4)$$

where η is the DIBL coefficient.

alter the relative strengths of transistors in the same circuit in terms of their current drivability; Section 2.2 later provides a comprehensive review on logic families for sub- V_t operation.

$$I_{\text{on (Sub-}V_t \text{ regime)}} = \mu C_{\text{ox}} \frac{W}{L} (n - 1) V_{\text{th}}^2 \exp\left(\frac{V_{\text{DD}} - V_t}{n V_{\text{th}}}\right) \quad (2.5)$$

$$I_{\text{off (All regimes)}} = \mu C_{\text{ox}} \frac{W}{L} (n - 1) V_{\text{th}}^2 \exp\left(\frac{-V_t}{n V_{\text{th}}}\right) \quad (2.6)$$

$$\frac{I_{\text{on}}}{I_{\text{off}}} (\text{Sub} - V_t \text{ regime}) = \exp\left(\frac{V_{\text{DD}}}{n V_{\text{th}}}\right) \quad (2.7)$$

Fig. 2.2 [10] below plots I_{on} versus V_{DD} (normalized to the $I_{\text{on}} (I_D) @ V_{\text{GS}} = 1.8\text{V}$ nominal). Noting that as I_{off} (eqn. (2.6)) is a constant, the I_{on} plot is hence representative of $I_{\text{on}}/I_{\text{off}}$. It can be seen in Fig. 2.2 that $I_{\text{on}}/I_{\text{off}}$ degrades exponentially with V_{DD} scaling in sub- V_t , and this is congruous with eqn. (2.7). The degradation of $I_{\text{on}}/I_{\text{off}}$ is not unexpected as I_{on} to switch the transistor output is, from the conventional (i.e. nominal V_{DD}) design perspective, effectively the extremely small sub- V_t leakage current.

Fig. 2.2: The degradation of on/off current ratio (I_{on}/I_{off}) of a MOS transistor in 180nm process (normalized to nominal $V_{DD}=1.8V$) [10]

The degradation of I_{on}/I_{off} is well recognized by the digital electronics community and they are cognizant for the need to account for this. This is because it may cause a commensurable degradation in the output logic level in certain logic circuits (e.g. see the static logic and the pass transistor/transmission gate logic in Sections 2.2.1 and 2.2.2 respectively later). Specifically, this is because the output logic voltage level of these circuits is usually determined by I_{on}/I_{off} , equivalently a voltage divider leading to a degradation in the output logic level (in terms of reduced output voltage swing) and reduced noise margin [37]. As in designs for the super- V_t regime, there is also the ‘fan-out’ issue to consider due to the limited I_{on} current. Furthermore, the ‘fan-in’ to a logic gate in sub- V_t deserves special attention [38] (a lesser consideration in super- V_t) as a higher ‘fan-in’ may lead to even lower I_{on} (due to longer transistor paths) and higher I_{off} (due to more parallel transistor paths); see Section 2.2 later.

Yet further, the effect of local and global process variations of V_t of the transistors is more significant in sub- V_t than in super- V_t – as delineated earlier, this is evident from the I_{on} exponential relationship with V_t for the former (eqn. (1.3)) and linear relationship for the latter (eqn. (1.2)). Perhaps less evident is that V_t variations of different transistors in the same circuit may easily alter their relative I_{on} , thereby increasing reliability issues in certain logic circuits whose functionality depends on the different I_{on} of the different transistors therein; e.g. the pseudo-NMOS logic and the dynamic logic in Sections 2.2.3 and 2.2.4 respectively later. In super- V_t designs, the well-established practice to accommodate this is by transistor sizing (by adjusting the transistor aspect ratio $\frac{W}{L}$) as a means to adjust I_{on} appropriately. However, this method, which only has a linear impact on transistor current (eqn. (1.3)), becomes less effective/unreliable in sub- V_t due to the undesirable more significant (exponential) impact of V_t variations [39].

The second design challenge/issue relates to the choice of design approaches/signaling protocols in sub- V_t is to accommodate the large/extreme circuit delay variations due to PVT variations. The large/extreme circuit delay variations are well established. The characteristic delay ($t_{d,inv}$) expressed in eqn. (2.8) [10] is for an inverter operating in sub- V_t . This delay is the time for I_{on} to charge (or discharge) the output node of the inverter through the PMOS (or NMOS) transistor (assuming symmetrical devices) to V_{DD} (or ground). For a circuit, the total delay along the critical path is simply multiples of $t_{d,inv}$.

$$t_{d,inv} = \frac{Q_{output,inv}}{I_{on}} = \frac{KC_{output,inv}V_{DD}}{\mu C_{ox} \frac{W}{L} (n-1)V_{th}^2 \exp(\frac{V_{DD}-V_t}{nV_{th}})} \quad (2.8)$$

where $Q_{output,inv}$ is the charge at the output node of the inverter,

K is a fitting parameter, and

$C_{\text{output,inv}}$ is the output load capacitance of the inverter.

To augment the literature review on the large delay variations due to PVT variations in sub- V_t , we will now illustrate these by means of statistical circuit simulations; our work here serves as a preamble to our WSN design in Chapter 4. Fig. 2.3 below plots our results of 1000 Monte Carlo (MC) simulations on the delay of an 80-inverter chain² circuit (@130nm CMOS) for V_{DD} ranging from 200mV to 400mV and for three operating temperatures, extreme heat 125°C, nominal 25°C, and extreme cold -55°C. In the MC simulations, both global and local process variations are considered. The abscissa is the delay (in log scale) and the ordinate is the corresponding delay occurrence. Each bell-shaped (more precisely lognormal [40]) distribution (at a given V_{DD} and temperature) represents the distribution of the inverter chain delays repeated 1000 times each with a random process variation.

Fig. 2.3: 1000 Monte Carlo simulations on the delay of 80-inverter chain at sub- V_t V_{DD} (from 200mV to 400mV), and at various temperatures (extreme heat 125°C, nominal 25°C, and extreme cold -55°C)

² The long inverter chain is chosen to allow for the averaging effect, i.e. the mitigation of the overall circuit delay variation as a result of the addition of individual gate delays (whose variations may cancel each other).

From Fig. 2.3, we make the following comments. First, both the delay and delay spread (the spread of delay distribution (due to process variations) at a given V_{DD} and T) increase in sub- V_t with reduced V_{DD} . Second, at a given V_{DD} , delay increases with reduced temperature and for completeness, the converse applies for super- V_t [41]. Third, the delay spread at a given V_{DD} increases with reduced temperature. Overall, these observations depict the challenges of sub- V_t operation (over super- V_t), and imperativeness of the choice of the design approaches/signaling protocols; see Section 2.3 later.

2.1.4 Power Gating

Consider now power gating as a technique to reduce power dissipation applicable to circuits that alternate between operation (typically full-load) and idle (i.e. no load where V_{DD} is gated, hence reducing wasted powers).

Fig. 2.4 [42] below depicts three different power gating configurations where high threshold ('High- V_t ', thus low leakage) gating transistors are inserted into the supply path (between V_{DD} and ground) of a combinational logic block. Specifically, PMOS gating transistor is inserted between V_{DD} and the combinational block and/or NMOS gating transistor is inserted between the combinational block and ground. The combinational block is usually implemented with low threshold ('Low- V_t ') transistors to achieve high computation speed. When operational ('active' mode), the gating transistor(s) are switched on (' SL '='0' and ' $\overline{SL}$ '='1') and the combinational block computes (hence the ensuing dynamic power and wasted powers). On the other hand, when the circuit is idle (i.e. no load, or the 'sleep' mode where no dynamic power is dissipated), the gating transistor(s) are switched off (' SL '='1' and ' $\overline{SL}$ '='0') and the leakage wasted current/power (through the combinational block) is reduced by the low leakage gating transistor(s).

Fig. 2.4: Power gating configurations: (a) PMOS Gating, (b) NMOS Gating, and (c) Dual Gating [42]

As mentioned earlier in Chapter 1, power gating in a sync circuit is usually implemented together with clock gating where a circuit block is only gated when its associated clock signal stops (i.e. when it is idling). Consequently, sync power gating is usually more coarse-grain due to its *global* clocking infrastructure where many circuits/systems share the same clock [34]. On the other hand, an async circuit (e.g. an async MD pipeline), where the computation is ‘clocked’ by the *local* handshake signal at every pipeline stage, attains the necessary local ‘clock-gating’ infrastructure to implement power gating in a much more fine-grain manner. This unique property of the async circuit will be explored in Chapter 3 later through our proposed novel fine-grain power gating technique specifically for the async MD protocol.

2.2 Logic Families for Sub- V_t

In this section, we will review the various digital logic families with emphasis on their circuit reliability in sub- V_t . The digital logic families of interest herein are the static logic, the pass transistor/transmission gate logic, the ratioed pseudo-NMOS logic, and the dynamic logic.

2.2.1 Static Logic

Static logic (particularly static CMOS) is the most commonly adopted logic family [43]. Fig. 2.5 below depicts the generic structure of a static logic gate, which comprises PMOS Pull-Up-Network ('PUN') and NMOS Pull-Down-Network ('PDN'). The complementary nature of the PUN and PDN ensures that the two transistor networks are never simultaneously switched on or off (except briefly during output transition in super- V_t , hence the ensuing 'short-circuit' current/power dissipation). In other words, there is always a low-resistive ('on') transistor path(s) connecting the output to either of the supply rails (V_{DD} or ground) while the other path(s) is 'off'. Static logic retains very high resistive difference (between I_{on}/I_{off}) in the path(s) driving the output node, hence an overall high noise margin [44]. However, when designing static logic for reliable sub- V_t operation, one needs to accommodate for the ensuing I_{on}/I_{off} degradation as delineated earlier. This basically puts a limit on the allowable number of 'fan-in' in a static gate [38].

Fig. 2.5: Generic structure of a static logic gate

2.2.2 Pass Transistor/Transmission Gate Logic

A distinctive feature of the pass transistor/transmission gate logic is that the inputs drive both the gate and the source-drain terminals of the transistors as opposed to the static logic, where only the transistor gate terminals are driven by the inputs. This modality allows the pass transistor/transmission gate logic to implement XOR-based circuits, such as multiplexers and full adders, with less number of transistors [45], and low leakage power dissipation [46].

If pass transistors (usually NMOS) are used in the logic, a level-restorer (a static buffer/inverter) is needed at the output to restore its logic level back to full- V_{DD} . This is because an NMOS transistor can only pass a voltage of $V_{DD}-V_t$ [43]. On the other hand, if transmission gates (comprising a parallel PMOS and NMOS transistors, see Fig. 2.6) are used, there is no V_t drop, albeit the overhead being an additional transistor and the need for complementary inputs. As pass transistor/transmission gate logic usually involves multiple transistor paths joining at the output, its output logic level is also susceptible to the degradation of I_{on}/I_{off} in sub- V_t [47]. An example is the 4-input multiplexer depicted in Fig. 2.6 where one I_{on} path is joined by three I_{off} paths at the output. In view of this, designing pass transistor/transmission gate logic in sub- V_t also requires a careful control of the number of ‘fan-in’ similar to the static logic. However, its ensuing output degradation is likely to be more problematic in cases where its output drives the source-drain terminal of the subsequent stage (thus causing further degradations). This contrasts with static logic, where only V_{DD} and ground are connected to the source-drain terminals, hence lesser output degradation.

Fig. 2.6: A pass transistor/TG logic-based multiplexer in sub- V_t operation

2.2.3 Ratioed Pseudo-NMOS Logic

Fig. 2.7 below depicts the generic structure of a ratioed pseudo-NMOS logic gate. Here the PUN in a static logic is replaced by an always ‘on’ single PMOS load transistor by its gate tied to ground. By removing the PUN, the pseudo-NMOS logic has the advantage of reduced transistor count and reduced input capacitance as compared to its static logic counterpart [48]. However, this logic family suffers from a static current issue (from V_{DD} to ground) when the PDN is switched on [43]. In sub- V_t , the disadvantage of the static current dissipation is unlikely to be acceptable in many designs given the long circuit delays in sub- V_t . Further, in terms of circuit reliability, pseudo-NMOS logic suffers from a current contention problem (for output ‘0’) when the I_{on} of the PMOS load and the I_{on} of the PDN compete with each other. In super- V_t designs, to ensure a sufficiently-low output ‘0’, the PMOS load is usually sized small (thus weaker with smaller I_{on}) than the PDN. However, as delineated earlier, this transistor sizing becomes less effective/unreliable in sub- V_t , where global and local process variations may easily alter the relative strengths (by altering their V_t) of the transistors, thereby undesirably degrading the output logic level(s) [10]. Under extreme cases,

the process variations may even inadvertently increase the drivability of the PMOS load transistor to be stronger than the PDN to the point where the output is erroneously permanently stuck at logic ‘1’.

Fig. 2.7: Generic structure of a pseudo-NMOS logic gate

2.2.4 Dynamic Logic

Fig. 2.8 below depicts the generic structure of a dynamic logic gate. Dynamic logic avoids the static current problem in pseudo-NMOS logic by replacing the always ‘on’ PMOS load transistor with a clocked pair PMOS ‘header’ and NMOS ‘footer’ transistors. The operation of a dynamic logic is divided into the ‘pre-charge’ phase and the ‘evaluation’ phase controlled by the clock signal (‘CLK’). During the ‘pre-charge’ phase where ‘CLK’=‘0’, the output node ‘Out’ is pre-charged to logic ‘1’ by the PMOS ‘header’. In the following ‘evaluation’ phase where ‘CLK’=‘1’, ‘Out’ is *conditionally* discharged by the PDN (through the NMOS ‘footer’) [49]. Dynamic logic typically achieves higher operating speed than static logic by replacing the latter’s PUN with the single pull-up PMOS ‘header’. Unlike the pseudo-NMOS and like the static logic, there is no static current/energy in dynamic logic as its ‘header’ and ‘footer’ transistors are never simultaneously switched on.

Fig. 2.8: Dynamic logic in sub- V_t operation: (a) without keeper and (b) with keeper.

A dynamic logic can be implemented without or with a feedback keeper (a PMOS transistor and an inverter depicted in Fig. 2.8(a) and (b) respectively). Without the keeper, a logic '1' state at the output 'Out' is held by the internal capacitance (C_{int}) of the node during the 'evaluation' phase, hence essentially 'floating'. This 'floating' state presents a reliability issue if the evaluation time is extended because the charge at the output node ($Q_{int} = C_{int}V_{DD}$) may leak away through I_{off} of the PDN. The unreliability is likely to exacerbate in sub- V_t where the node charge is extremely small and the circuit delay is long [50]. To avoid this, dynamic logic can be made 'semi-static' by augmenting a keeper circuit as depicted in Fig. 2.8(b). With this keeper, the previous 'floating' node for an output of logic '1' is now 'statically' held by the PMOS transistor. However, by adding the keeper, a current contention problem similar to the pseudo-NMOS logic delineated earlier is inadvertently created when the PDN is switched on (to produce an output '0') [51]. As in pseudo-NMOS logic, the PMOS keeper is thus needed to be sized small as compared to the PDN. Unfortunately, as delineated earlier, this solution is largely unsatisfactory in sub- V_t because

the circuit operation is unreliable due to process variations that may alter the relative strengths of the transistors therein.

In summary, amongst the four reviewed digital logic families, the static logic and the pass transistor/transmission gate logic do not suffer the current contention problem of their pseudo-NMOS and dynamic (with keeper) logic counterparts. In this aspect, they are arguably more reliable for sub- V_t operation. Nonetheless, the design of the former two families in sub- V_t need to carefully account for the degradation in I_{on}/I_{off} that affects the maximum number of ‘fan-in’. Amongst the former two families, it has been argued in literature [45] that the pass transistor/transmission gate is more efficient (in terms of lower transistor count) for XOR-based logic while static logic is more efficient for general-purpose logic. In view of this, for sub- V_t operation, we will adopt in this thesis, static logic for general-purpose logic and transmission gate (with level-restorer) for multiplexers. In particular, in Chapter 4 later we propose a novel static logic style, coined ‘Pre-Charged-Static-Logic’ (PCSL), for implementing sub- V_t async QDI circuits, where a 3 ‘fan-in’ limit is enforced.

2.3 Design Approaches/Signaling Protocols for Sub- V_t

We will now review the second design challenge/issue – relating to the choice of design approaches/signaling protocols with emphasis towards sub- V_t operation. This is particularly imperative in view of PVT variations being virtually intractable in sub- V_t for high variation-space and wide operation-space applications, and hence the ensuing intractable delay variations. In this section, we will review the two digital design approaches/signaling protocols, the prevalent sync and the somewhat esoteric async, with the emphasis on their operational robustness in sub- V_t .

2.3.1 Synchronous-Logic

The prevalent sync is widely accepted and adopted by the digital design community for super- V_t operation primarily due to its ease of conceptualization and implementation, and the availability of mature and sophisticated commercial EDA tools [52]. As delineated earlier in Chapter 1, the sync relies on a global clock signal (or variants thereof) as the timing reference for its data synchronization. The generic structure of a sync pipeline stage was shown earlier in Fig. 1.2(a) and repeated below in Fig. 2.9(a) for ease of readability. As mentioned earlier, as the computation delay of the single-rail logic circuit in sync cannot be derived from its output, the clock period of a sync circuit has to accommodate the worst-case delay based on pre-characterization(s) of the sub-circuits/circuit therein. However, the delay variations (equivalent to % circuit performance variability in Row 5 of ITRS projections in Table 1.1) become increasingly larger with the downward scaling of the minimum feature size of the transistors as a result of the increasing PVT variations. The variations could possibly reach the point of being intractable for high variation-space and wide operation-space applications when operating in sub- V_t [29].

Consider the sync pipeline stage depicted [53], [122] in Fig. 2.9(a) operating in sub- V_t . Several pertinent signal waveforms of the pipeline stage are plotted in Fig. 2.9(b) for two operating cases: one stable V_{DD} on the left half of Fig. 2.9(b), and the other with V_{DD} variations (or equivalently with noise) on the right half. For the first case where V_{DD} is stable at sub- V_t voltage of 0.4V, the output is synchronized correctly after 1 clock cycle ('CLK') as required. However, for the second case where V_{DD} is subjected to noise oscillating between 0.3V and 0.4V, the circuit fails to synchronize the output after 1 clock cycle. Instead, it erroneously synchronizes a data '0' instead of the correct data '1'. For completeness, it only synchronizes the data '1' after 3 clock cycles (i.e. a longer delay

required). Similar erroneous synchronizations may occur when the circuit is subjected to other process and temperature variations. In short, timing assumptions (and necessary delay safety margins thereof) are essential for the error-free operation of a sync circuit. However, such timing becomes ambiguous in the sub- V_t voltage regime (and increasingly so for nano-scaled fabrication processes).

Fig. 2.9: (a) Generic block diagram of a sync pipeline stage working in sub- V_t ($V_{DD}=400mV$), and (b) signal waveforms (V_{DD} , D1, D2, D3, and CLK) for the sync circuit. The data is correctly synchronized for the first operation when V_{DD} is stable. The data is incorrectly synchronized for the second operation when V_{DD} is coupled with noise (V_{DD} variation). [53], [122]

To accommodate the said sub- V_t delay variation issues due to PVT variations for the sync, various design techniques/approaches have been reported, which include strict operating environments (e.g. expensive highly controlled fabrication processes and electrical conditions), transistor upsizing [32], [54] (to reduce the effects of random dopant fluctuations, etc), current-mode approaches [55], adaptive body bias [56], high-precision DC-DC converters and/or linear regulator [57] (to reduce V_{DD} variations), advanced cooling and packaging [7] (for controlling on-chip temperature gradients), self-calibration techniques [58], redundancy/duplication circuitry [59], and, to a large extent, ‘pessimistic’ designs with large delay safety margins (even with the aforesaid approaches fully or in part adopted). The large delay safety margins allowed for would typically include the worst-case delay, including clock skew, setup-time, and hold-time for registers, etc. Overall, the design of such systems for operation robustness based on the sync design approach (where a global clock is used) for sub- V_t operations would be challenging and/or such systems may be unnecessarily slower than warranted. Nonetheless, because a complete profile for PVT variations is ambiguous in the sub- V_t voltage regime, particularly for high variation-space and wide operation-space applications, the sync design approach is unable to guarantee robust error-free operation. Furthermore, the yield of sync designs for sub- V_t operation could be low, and their reliability issues cannot be assumed.

2.3.2 Asynchronous-Logic

To accommodate the large delay, possibly intractable, variations in sub- V_t operation, the alternative digital design approach/signaling protocol, the async may be adopted. As delineated earlier in Chapter 1, the data synchronization by means of the global timing (or variants thereof) in the sync is replaced with local sequencing of handshake protocols in the async [60]. Two types of async were briefly reviewed earlier, namely the async Matched

Delay (MD; also known as the async Bundled-Data protocol, see later) and the async Quasi-Delay-Insensitive (QDI). Of particular interest, as the async QDI protocol, with its dual-rail logic circuit and completion detection, achieves *unconditional* error-free operation (save the isochronic timing [16]) regardless of delay variations, it lends itself naturally to sub- V_t operation.

To depict the operation robustness of QDI circuits from a timing perspective, consider the same example in Fig. 2.9 but now the block diagram of a QDI pipeline [53], [122] in Fig. 2.10(a) (repeated from Fig. 1.2(c) earlier for sake of readability). Several pertinent signal waveforms of the circuit are plotted in Fig. 2.10(b) for two same operating conditions. Note that ‘ $D1$ ’, ‘ $D2$ ’, and ‘ $D3$ ’ are dual-rail encoded (see Table 1.2 earlier), where for simplicity, only the Data True waveforms (‘ $D1.T$ ’, ‘ $D2.T$ ’, and ‘ $D3.T$ ’) are depicted. In the first case where V_{DD} is stable at sub- V_t voltage of 0.4V, the handshake signal (‘ HS ’) which is generated by the Completion Detection (‘ CD ’) circuit, is correctly asserted to indicate that output ‘ $D3$ ’ has become valid. Similarly, in the second case where V_{DD} is subjected to noise oscillating between 0.3V to 0.4V, the signal sequence between output ‘ $D3$ ’ and ‘ HS ’ signals remains correct (albeit longer delay due to the reduced V_{DD}). In other words, a QDI circuit can innately adapt to its operating conditions and tolerate the delay variations therein, hence robust synchronization (error-free operation).

Fig. 2.10: (a) Generic block diagram of an async QDI pipeline stage, and (b) signal waveforms (V_{DD} , $D1.T$, $D2.T$, $D3.T$, and HS) for the async circuit. The data is correctly synchronized both for the first operation when V_{DD} is stable and for the second operation when V_{DD} is coupled with noise (albeit with a longer delay). [53], [122]

2.4 Asynchronous-Logic for Sub- V_t

Given that it is very worthwhile to have the option of DVS and that the conventional and prevalent sync approach is unable to provide designs with *unconditional* error-free operation in the sub- V_t regimes, we will now review the various async approaches/signaling protocols. The particular intention is to review the specific async approaches that can realize DVS

(including sub- V_t) with unconditional error-free operation, hence practical circuits/systems for high variation-space and wide operation-space applications.

2.4.1 Fundamentals of Asynchronous-Logic

In this section, we review the fundamentals of the async approach including their handshake protocols and delay models.

Handshake Protocols

As delineated earlier, the async adopts handshake protocols as a means for local operation sequencing/data synchronization. These protocols can be classified in terms of their data encodings and communication phases [61]. The data encodings include either single-rail or multi-rail (most commonly dual-rail or 1-of-4); see Chapter 1 earlier. The communication phases, on the other hand, include either 2-phase or 4-phase protocols.

Consider a generic async pipeline involving a ‘Sender’ and a ‘Receiver’ depicted in Fig. 2.11 where an N -bit data is being communicated from the sender to the receiver. This data communication (synchronization) is enforced by means of two handshake signals (‘HS’), the request signal (‘*Req*’) from the sender to the receiver and the acknowledge signal (‘*Ack*’) from the receiver back to the sender.

Fig. 2.11: Block diagram of a generic async pipeline

Fig. 2.12(a) and (b) depict the 2-phase non-return-to-zero (NRZ) and the 4-phase return-to-zero (RZ) protocols respectively. The 2-phase NRZ protocol, as the name implies, embodies two communication phases: (i) the sender issues valid data and produces a transition (either a low-to-high or a high-to-low transition) on '*Req*'; and (ii) the receiver receives the valid data and produces a transition on '*Ack*'. This completes the data communication/synchronization cycle and the sender is allowed to issue the next valid data. As the handshake signals, '*Req*' and '*Ack*', may not return to '0' after each data communication/synchronization cycle, hence the denotation NRZ. On the other hand, the 4-phase RZ protocol, as the denotation implies, embodies four communication phases (assuming an active-high protocol): (i) the sender issues valid data and asserts '*Req*' to '1'; (ii) the receiver receives the valid data and asserts '*Ack*' to '1'; (iii) the sender de-asserts '*Req*' to '0'; (iv) the receiver de-asserts '*Ack*' to '0'. This completes the data communication/synchronization cycle and the sender is allowed to issue the next valid data. As '*Req*' and '*Ack*' handshake signals always return to '0' after each data communication/synchronization cycle, hence the denotation RZ.

Fig. 2.12: Async handshake protocols: (a) 2-phase NRZ and (b) 4-phase RZ

From a cursory perspective, it may appear that the 2-phase protocol is more efficient than its 4-phase counterpart as it requires less transitions to complete a data communication/synchronization cycle. However, in practice, it is well recognized that the 2-phase protocol is more difficult to realize than its 4-phase counterpart because the former requires transition-based logic while the latter requires level-based logic [12]. Furthermore, for the same reason, the 2-phase protocol may incur higher overheads in terms of circuit area and power. Consequently, the 4-phase protocol is more widely adopted in practical async circuits/systems, and will be adopted herein.

Delay Models

The operation of async circuits may be viewed as signals flow through gate and wire delays, and the signaling therein is localized according to an async handshake protocol as delineated earlier. Depending on their delay properties, async circuits can generally be classified into four design approaches tabulated in Table 2.1.

Table 2.1: Classification of the async design approaches

No	Classification	Features
1	Quasi-Delay-Insensitive (QDI)	QDI circuits can operate correctly with arbitrary gate delays, and arbitrary wire delays except for certain wire branches (called isochronic forks [16] which assume the same wire delays). QDI is the most robust async used for practical applications.
2	Matched Delay (MD) (also known as Bundled-Data (BD))	MD/BD circuits can operate correctly with a bounded delay assumption on the ensuing gates and wires. A matched delay element is used to enforce proper data synchronization.
3	Delay-Insensitive (DI)	DI circuits can operate correctly with arbitrary gate and wire delays, However, such a strict delay property leads to circuit realizations comprising only inverters, buffers, and C-Muller circuits [16], hence not for practical applications.
4	Speed-Independent (SI)	SI circuits can operate correctly with arbitrary gate delays, and zero or negligible wire delays – a somewhat unrealistic assumption in state-of-the-art fabrication processes where wire delays cannot be ignored.

Of the four async approaches tabulated in Table 2.1, for DVS, especially in the sub- V_t regime, the QDI async approach is undoubtedly the most practical/realistic approach. Theoretically, if the design is appropriate, the circuit can be designed to operate error-free as long as the transistors therein can switch because it innately [13] detects the computation delays according to different workloads and operating conditions. In this sense, the QDI approach offers significant advantages for design simplicity for accommodating the PVT variations, particularly when the PVT variations are intractable, and operation robustness largely because QDI async circuits are virtually ‘delay insensitive’ (save the isochronic timing [16]).

As delineated in Chapter 1 earlier, the Matched Delay (MD) (also known as the Bundled-Data (BD)) async approach assumes bounded delay assumptions that may be unmatched/insufficient due to the PVT variations, somewhat akin to sync circuits. They are hence not necessarily robust for DVS in the sub- V_t regimes.

The DI async approach, although theoretically the most robust, is unfortunately not practical due to limited/impractical choices of implementation for many systems. Specifically, as this approach permits only inverters, buffers, and C-Muller circuits, the ensuing circuits are impractical. Finally, as the SI async approach can only operate correctly with zero or negligible wire delays, they are not only impractical but also insufficiently robust in the sub- V_t voltage regime.

In short, lower power or energy circuit/system realizations may be obtained by a combination of V_{DD} reduction (DVS) and appropriate async design approaches – in view of high variation-space and wide operation-space applications that include DVS (and in sub- V_t), the async QDI approach/signaling protocol is the most appropriate in terms of its unconditional error-free operation given the intractable PVT variations. For this reason, the QDI protocol is adopted for the WSN application in Chapter 4; Chapter 3 includes MD.

2.4.2 Asynchronous-Logic QDI for Sub- V_t

Async QDI design dates back to the 1950s (despite different denotations thereto until the late 1980s) and the first async microcontroller [62] was reported. The ‘milestone’ chronology of major reported QDI designs [62]–[80] is depicted in Fig. 2.13. The application and purpose of these reported QDI designs vary, including CAM [62] designed for proof-of-concept to delineate the properties of QDI circuits; MiniMIPS [69] was for high performance; NCL8051 [71] for low electromagnetic interference (EMI); the STFB prefix adder [77] for high throughput; and the TAM microprocessor [80] for low power dissipation. Interestingly, these reported designs were designed for super- V_t , largely nominal V_{DD} and, to some extent, near- V_t voltage regimes – except for recent reported work [14], the research of QDI circuits for the sub- V_t voltage regime is largely unexplored.

Fig. 2.13: Reported QDI designs

The logic families reviewed earlier in Section 2.2, namely the static logic, dynamic logic, and pass transistor/transmission gate logic, can be used to realize any digital-logic design approach, including async QDI designs. Based on the review of designs in Fig. 2.13, Table 2.2 below tabulates the specific logic family of the reported QDI logic design styles.

Table 2.2: Reported logic design styles (within specific logic families) for QDI realization

Logic Family	QDI Logic Design Styles	Design in Fig. 2.13
Static logic	1. Direct-Static-Logic-Implementation (DSL)	TITAC [66], TITAC II [68], TAM [80]
	2. Delay-Insensitive-Minterm-Synthesis (DIMS)	DIMS Multi-ring [65],
	3. Null-Convention-Logic (NCL)	NCL8051 [71]
Dynamic logic	1. Direct Logic Implementation	CAM [62]
	2. PS0	Ring Divider [63], FAM [64]
	3. Pre-charged Half Buffer (PCHB)	MiniMIPS [69], NEXUS [72], Lutonium [73], SNAP [74], BitSNAP [76], VORTEX [78]
	4. LP2/1	FIFO [79]
	5. Single-track Asynchronous Pulse Logic (STAPL)	STAPL Divider [75],
	6. Single-track Full Buffer (STFB)	Prefix Adder [77]
	7. Sunpulse	--
Pass transistor/transmission gate logic	1. Sense-amplifier Pass Transistor Logic (SAPTL)	--

In Table 2.2, there are three, seven and one logic design styles respectively within the static logic, the dynamic logic, and the pass transistor/transmission gate logic families. In general, QDI designs adopting the static logic family, where the associated sizing of transistors is not as critical, are robust for a wide range of V_{DD} (including sub- V_t). They are hence appropriate for DVS, but typically require a relatively large transistor count (larger IC area). Designs adopting dynamic logic, on the other hand, are usually for high performance while designs adopting pass transistor/transmission gate logic are primarily for low leakage power dissipation. At this juncture, our review has discovered that reported realizations adopting these various logic families are largely designed for and applied in the super- V_t regime (largely nominal and near- V_t), and hitherto their application in the sub- V_t voltage regime remains largely unreported.

Fig. 2.14 depicts the schematic of an AND/NAND gate adopting the three reported static QDI logic design styles tabulated in Table 2.2: (a) static NULL-Convention-Logic (NCL) [81], (b) static Delay-Insensitive-Minterm-Synthesis (DIMS) [65], and (c) static Direct-Static-Logic-Implementation (DSL) [82]. The operation modalities of these logic design styles will now be briefly reviewed.

Fig. 2.14: Reported static QDI logic design styles for an AND/NAND gate: (a) static NULL-Convention-Logic (NCL), (b) static Delay-Insensitive-Minterm-Synthesis (DIMS), and (c) static Direct-Static-Logic-Implementation (DSL)

Fig. 2.14(a) depicts the symbolic diagram (on the left) and schematic (on the right) of an NCL AND/NAND gate. NCL is realized based on an m -of- k threshold logic where k is the total number of inputs and m is the number of inputs necessary to assert its output. In other words, the output of the threshold gate will assert to logic ‘1’ when at least m inputs (among the k inputs) are asserted to logic ‘1’, and conversely the output will assert to logic ‘0’ only when all inputs are asserted to logic ‘0’. NCL can be implemented in either simple or complex logic gates; refer to Chapter 4 later for complex logic gate examples. Fig. 2.14(b) depicts a DIMS AND/NAND gate where the ‘minterms’ of the logic are realized in C-Muller gates and their outputs collected through an OR gate. It is generally recognized that DIMS is IC-area inefficient (due to larger transistor count), and usually larger than its NCL counterpart [83]. Fig. 2.14(c) depicts the DSLI AND/NAND gate where an ‘Input Validity’ and an ‘Output Validity’ block are employed for checking data validity. This logic style is inefficient in terms of transistor count [82].

As delineated earlier, the major shortcoming of async QDI is the need for dual-rail where the transistor count is approximately doubled compared to single-rail logic (used in sync and async MD) for the same functionality. By careful layout, the effective IC area³ is however typically $1.5\times$ [53]. Further, the dynamic power (due to its dual-rail logic and 4-phase protocol) and the leakage power (due to a larger IC area) of QDI may also be higher than its sync counterpart. In view of this disadvantageous overhead of QDI, we propose in Chapter 4 a novel static async QDI logic style coined ‘Pre-Charged-Static-Logic’ (PCSL) which simultaneously features lower power/energy dissipation, lower delay and smaller IC area than the competing reported static QDI logic styles.

³ Although this may imply an ensuing higher cost in manufacturing, the actual manufacturing cost may not be necessarily higher because the manufacturing yield is expected to be higher due to the inherent added operation robustness of QDI.

Interestingly, despite the potential advantages of async, its acceptance remains stymied and largely unaccepted by the digital electronics community and industry. Even at this juncture, async design remains esoteric and a confluence of major impediments to their general acceptance by the digital electronics community remains:

- (a) Unestablished design methodologies for high speed and for low power applications,
- (b) Lack of sophisticated computer aided design (CAD) or EDA tools,
- (c) Unestablished test methodologies for manufacturability,
- (d) Paucity of reported async designs, and their applications, and
- (e) A lack of critical mass of designers and users.

2.5 Summary of Literature Review

This chapter has described low-power digital circuit design techniques. The emphasis is on ultra low-power sub- V_t operation and the associated formidable challenges (over super- V_t operation), particularly in view of operational robustness in high variation-space (including DVS where V_{DD} ranges from nominal to sub- V_t) and wide operation-space. Power gating has also been reviewed as an effective method for reducing the wasted powers. In view of the need to accommodate the design challenges/issues in sub- V_t , four logic families (static, pass transistor/transmission gate, pseudo-NMOS, and dynamic) and two digital design approaches/signalling protocols (sync and async) have been reviewed with emphasis on their operation robustness in sub- V_t . In particular, the async QDI protocol has been reviewed in greater detail for its unconditional error-free operation given the intractable PVT variations, which, in our view, is the most appropriate for high variation-space sub- V_t operation.

In summary, Fig. 2.15 [53], [122] depicts a succinct generalized overview embodying the classification of digital logic circuit design for the realization of operationally robust

digital circuits in sub- V_t – from the highest-level digital design approaches/signaling protocols (sync or async) to async approaches/protocols (four possible approaches/protocols) to logic families (three logic families) to static QDI logic design styles (three reported logic design styles). The horizontal lines demarcate the various design levels and the nomenclature thereof (italicized text) is depicted on the right of the diagram. At the lowest level in the complete digital design space, there are three possible reported QDI logic design styles (particularly for robust operation in sub- V_t): DIMS, NCL and DSLI. The design approaches in bold are suitable for high variation-space (including DVS where V_{DD} ranges from nominal to ultra low-power sub- V_t) and wide operation-space applications.

Fig. 2.15: Summary and classification of digital design approaches/signaling protocols. The approaches/protocols in bold are appropriate for sub- V_t operation [53], [122]

Chapter 3 Power Gating for Async MD and Ultra Low-Power Sub- V_t Async QDI

3.1 Introduction

This chapter largely serves as a preamble to Chapter 4 where we propose, design and realize a novel ultra low-power async WSN for very high variation-space and very wide operation-space applications. For applications such as the WSN and in the perspective of sync and async operation modalities, there are several features that may be explored for low-power/ultra low-power, and we will herein describe two.

The first pertains to the application of async power gating. Interestingly, despite the ubiquity of power gating in sync circuits, power gating in async is rarely reported, perhaps in part due to the esotericism of async. This power gating serves to reduce the wasted power (mainly leakage power) during the idle period as part of the wide operation-space. As clock gating is innate in async, both power and clock gatings are hence simultaneously applied to reduce both dynamic and leakage power. Async power gating is particularly appropriate for the async-based WSN herein as it features relatively long idle periods and data/event-triggered active operation. Specifically, the active/passive (idle) operation is a 20/80 ratio and active operation is automatically (without added overheads with respect to hardware already present therein) triggered by the arrival of the input sample; see Chapter 4 later. In Section 3.2, we propose a fine-grain power gating for an async MD pipeline by exploiting its local handshake signals and thereafter investigate its efficacy in terms of wasted power reductions and in the context of the overall power. Despite the simplicity of our proposed gating method, this method [42] is among the first reported power gating methodologies for

an async pipeline/circuit [84]. Nevertheless, of late, there are other reported methods [85]-[87]. For completeness, it is instructive to note that this proposed technique (for MD async) is equally applicable to an async QDI pipeline (e.g. for the async WSN; see Chapter 5 later for our proposed future work).

The second pertains to the amount or degree of delay and delay margins of sync circuits operating in ultra low-power sub- V_t operation in view of high variation-space and wide operation-space, and its benchmarking against async QDI. For example, the sync and QDI async WSNs in Chapter 4 are designed to operate in a high variation-space environment for temperatures ranging from -55°C to 125°C , and in wide operation-space with a sampling rate range from 0.1 kSamples/s (kS/s) to 100 kS/s. In Chapters 1 and 2, it was explained that for sync, error-free operation in sub- V_t requires large/extreme delay safety margins while for async QDI, the varying delay is innately accommodated. In the design of the sync where the delay margins must be ascertained, the usual practice typically involves comprehensive time-consuming statistical static timing analysis (SSTA, such as by Monte Carlo simulations) [15]. To this end, we propose in this chapter, a simple analytical means to obtain a first-order estimation (the accuracy may be improved with additional heuristics; see Chapter 5 later) of the delay variations due to PVT variations in sub- V_t . This analytical means is particularly useful as it provides insights to the digital designer at the early juncture of his/her design to ascertain a first-order estimation of cost in terms of the delay-variations in view of variation-space and operation-space.

Further, as a cursory/preliminary study on the effect of delay safety margins to the sync and its benchmarking to the async QDI in sub- V_t , we benchmark a sync circuit example (with first-order delay safety margins estimated by the derived equations) against its async

QDI circuit counterpart (with self-completion detection) in terms of delay, energy and transistor count. In Chapter 4, we more comprehensively benchmark a sync (with $\pm 3\sigma$ delay safety margins ascertained by Monte Carlo simulations) and an async QDI filter bank, also with $\pm 3\sigma$ delay variations, for a WSN under very high variation-space and very wide operation-space in sub- V_t .

The work reported in this chapter is largely extracted from our two papers published in *Proc. IEEE ISCAS*, 2009 [42] and *Proc. IEEE Int. NEWCAS Conf.*, 2010 [88].

3.2 Fine-Grain Power Gating for Reducing Wasted Powers in Async Matched Delay

It was delineated in Chapter 1 that it is well established that the power dissipation of a typical CMOS digital circuit comprises dynamic power, wasted leakage and short-circuit powers. While dynamic power dissipation remains to be the dominant in many digital circuits (for super- V_t operation), wasted power dissipation (particularly leakage power) has become increasingly more significant especially when the minimum feature size of a transistor is deep sub-micrometer or nanometer scaled [89] and when operating in sub- V_t .

To reduce the wasted power dissipation (including both leakage and short-circuit power dissipations), many design techniques have been reported in literature. Common techniques [7] include power gating, body bias, transistor stacking, critical transistor-sizing, etc. Among these approaches, and where applicable (when the circuit/system is idle/sleep), power gating is one of the most effective for leakage power reduction. Power gating can be implemented by means of multi- V_t CMOS (MTCMOS) [90], self-controllable voltage level (SVL) [91], variable- V_t [92], etc. As reviewed earlier in Chapter 2, for efficacious power gating, low-

leakage (high- V_t , e.g. MTCMOS) gating transistors are employed to cutoff the power rails (V_{DD} and/or ground) to the combinational block during the sleep intervals (see Section 2.1.4 earlier).

As delineated earlier in chapters 1 and 2, there are several considerations when applying power gating in prevalent sync circuits [93]. First, in sync circuits, power gating is usually implemented in a coarse-grain manner, largely a consequence of the sync *global* clocking infrastructure. In many cases, attempting fine-grain power gating defeats any gains, even possibly increasing the power dissipation. Second, the transition between ‘active’ and ‘sleep’ modes may pose circuit reliability issues. These are well known and solutions thereto [34] include synchronization failure, noise margin degradation, timing violation, etc..

The async, specifically the async MD, adopts local handshake protocols for data synchronization, and provides the opportunity for a fine-grain ‘clocking’ infrastructure that is not easily achieved in its sync counterpart. Specifically, by means of local handshake controllers (see later), an async MD pipeline embodies local signaling that marks the beginning and ending of circuit operation (and conversely, the ending and beginning of its idle state) that may be exploited for implementing said fine-grain power gating.

Specifically, we propose herein a fine-grain power gating technique for an async 4-phase MD pipeline stage by means of handshake signaling-controlled gating transistors. An important consideration for a power reduction technique is its cost in terms of overheads which may otherwise defeat any advantages gained. In this perspective, the overhead of our proposed technique is small because it is simple, yet effective, augmentation to an existing async latch controller – one additional inverter (with necessary buffering) for driving the

PMOS gating. The proposed technique can be applied to the three different power gating configurations introduced earlier in Chapter 2, i.e PMOS gating, NMOS gating, and dual gating. The efficacy of the proposed technique will be investigated and ascertained for different workload levels (in terms of input data rates). By means of computer simulations, the amount of wasted power reduction (for each gating configuration) and the delay overhead (compared to a pipeline without power gating) will be evaluated.

The remaining of this section is organized as follows. Section 3.2.1 delineates the modus operandi of the async MD pipeline. Section 3.2.2 delineates our proposed fine-grain power gating technique. Section 3.2.3 presents the benchmarking on the proposed technique.

3.2.1 Async MD Pipeline

As a preamble to our proposed fine-grain power gating for an async MD pipeline, consider first the block diagram of an async MD pipeline stage (enclosed in the dashed box) based on the 4-phase handshake protocol (see Section 2.4.1 earlier) depicted in Fig.3.1. Its modus operandi is as follows. When the input data is ready, the request signal (R_{in1}) is asserted to '1'. This triggers Latch Controller 1, through En_1 , to enable Latch 1 to capture the input data and the Latch Controller subsequently asserts both the output request signal (R_{out1}) and the acknowledge signal (A_{in1}) to '1'. Latch Controller 1 will then wait for R_{in1} to be de-asserted (by the preceding stage, not shown), and respond by de-asserting A_{in1} (thus completing a 4-phase handshake with the preceding stage). While the data captured by Latch 1 is processed by the Combinational Block, the output request signal R_{out1} is simultaneously passed through the 'Matched Delay' whose delay is designed to be at least equal to (typically longer than for some delay safety margin) the worst-case delay of the

associated Combinational Block. This delay is to ensure that the Combinational Block has sufficient time to compute its computation, thereby ensuring the proper sequence of the handshake signals and error-free computed data arriving at the subsequent pipeline stage (Latch Controller 2 and Latch 2) for correct data synchronization. The same 4-phase operation will likewise repeat at Latch Controller 2 and Latch 2, and the data is passed down the pipeline accordingly.

Fig. 3.1: Block diagram of an async MD pipeline

3.2.2 Proposed Fine-Grain Power Gating for Async MD Pipeline

In the async MD pipeline, data computation in the Combinational Block initiates (hence the beginning of the ‘active’ mode) around the same time as the assertion of the output request signal R_{out1} . Similarly, when the computed data is captured by the subsequent stage (as acknowledged by the assertion of A_{out1}), R_{out1} is de-asserted marking the beginning of the

‘sleep’ (idle) mode. Clearly, this 4-phase handshake signaling and the computation in the Combinational Block may be exploited to implement a form of local power gating. Specifically, R_{out1} can be used to directly control (switch on and off) the gating transistor(s) to the Combinational Block transitioning between the ‘active’ and the ‘sleep’ mode and in a ‘just-in-time’ manner.

Fig. 3.2 depicts the block diagram of the async MD pipeline with the proposed fine-grain power gating technique, where as usual, high- V_t gating transistor(s) (PMOS Gating and/or NMOS Gating) are inserted between the Combinational Block and the power rails. The proposed fine-grain R_{out1} is used to control the gating transistors to enable/disable the active and idle operations of the associated Combinational Block which is, as usual, implemented in low- V_t transistors to facilitate high computation speed during the ‘active’ mode. Specifically, during the ‘active’ mode, R_{out1} is asserted ‘1’ and the gating transistor(s) are switched on to connect the power rails, thereby enabling the Combinational Block for computation. Conversely, during the ‘sleep’ mode, R_{out1} is de-asserted ‘0’ and the gating transistor(s) are switched off disconnecting the power rails, thereby reducing the leakage current/power of the Combinational Block. For completeness, as the computation in the Combinational Block initiates at about the same time as the gating transistor(s) are being switched on, the short-circuit wasted current/power (during this initial computation period) is somewhat reduced by the (not fully-on) gating transistors (see simulation results later).

Fig. 3.2: Block diagram of the async MD pipeline with the proposed fine-grain power gating

The overall schematic of the async MD pipeline (one stage) with the proposed fine-grain power gating technique (with dual gating configuration; see Section 2.1.4 earlier) is depicted in Fig. 3.3. The Latch Controller design [94] employed in the async MD pipeline features low-power for its normally-disabled control scheme. This scheme potentially eliminates/mitigates spurious transitions in the latch, because when its associated latch (to capture data), timed by the Matched Delay, is enabled, the input data bits should/are expected to be stable.

Fig. 3.3: Schematic of the one-stage async MD pipeline with the proposed fine-grain power gating technique

Fig. 3.4 depicts the Signal Transition Graph (STG) [95] of the Latch Controller, and its interpretation is as follows. A signal notation with a '+' and '-' symbol represents a '0'-to-'1' and '1'-to-'0' signal transition respectively. The arrows in the STG represent the causal relationships between the relevant signal transitions, where a solid arrow leads to a transition on the internal or output signal of the Latch Controller and a dashed arrow leads to a transition on the input signal of the Latch Controller.

Fig. 3.4: Signal Transition Graph (STG) of the Latch Controller employed in the async MD pipeline

With the STG specification, the Latch Controller can be synthesized using public-domain tools such as Petrify [96], for example, the Latch Controllers 1 and 2 in Fig. 3.3 earlier. In Fig. 3.3, the design of the Latch employed in the MD pipeline is a non-inverting latch, which is controlled by the normally-disabled Latch Controller. For sake of simplicity of the example herein, the Combinational Block is implemented as a 40-chain inverter. The Matched Delay is implemented as a 45-chain inverter whose delay is slightly longer than the Combinational Block for delay safety margin. In the proposed power gating, only the Combinational Block is power gated. The Latch Controllers and Latches remain power ungated as they are required to remain continuously powered for data retention and synchronization.

Fig. 3.5 depicts the signal timing diagram of the async MD pipeline with the proposed power gating, where the 4-phase handshake protocol is adopted. It is clear from the timing diagram that the transitions between the ‘active’ and the ‘sleep’ modes are marked by the signal transitions on R_{out1} . In other words, the local 4-phase handshake protocol of the async MD innately provides the necessary timing and infrastructure for the proposed fine-grain power gating. It is also worthwhile to note that this timing and infrastructure is universal to all 4-phase async protocols (including the async QDI), hence the proposed power gating technique is likewise applicable to the other 4-phase async protocols; see Chapter 5 later for our proposed future work.

Fig. 3.5: Signal timing diagram of the async MD pipeline with the proposed power gating

3.2.3 Benchmarking the Proposed Fine-Grain Power Gating

To depict the efficacy of the proposed fine-grain power gating technique for the async MD pipeline, the design in Fig. 3.3 is simulated @130nm CMOS process, nominal $V_{DD}=1.2V$ for the three different power gating configurations (PMOS gating, NMOS gating, and dual gating) and for the case without power gating. Further, to depict the effect of (wide) operation-space (varying workloads), the pipelines are simulated with different input data rates.

It is well established that the insertion of gating transistor(s) carries both delay and power overhead when compared to a pipeline without power gating. The delay overhead is due to the charging/discharging of the power rails (V_{DD} and ground) to the Combinational Block during the transitions between the ‘active’ and the ‘sleep’ modes. As the voltage(s) of the power rails transition during these transition periods, the Combinational Block would need to wait until some voltage stability is reached, hence the overall computation of the Combinational Block takes longer time. In addition, this charging/discharging of the power rails and the associated charging/discharging of the (gate(s) of the) gating transistor(s) (typically large) also involve a (dynamic) power overhead. These delay and power overheads can be adjusted by sizing the gating transistor(s). For example, larger gating transistor(s) would allow faster charging and discharging of the power rails (thus shorter delay overhead) at the expense of increased (dynamic) power dissipation. In our simulations, we size the gating transistor(s) in the three gating configurations such that they have the same delay overhead (in this case, <15%) of that without power gating.

Fig. 3.6 depicts the simulated power dissipations of the Combinational Block (including the power associated with the insertion of the gating transistor(s) where applicable) in the async MD pipeline at various input data rates (from 10k bit-per-second (bps) to 100Mbps). The total power (including dynamic and wasted powers) and the wasted power (including short-circuit and leakage powers) are plotted.

Fig. 3.6: Power Dissipations of the Combinational Block (including the power associated with the insertion of the gating transistor(s) where applicable) in the async MD pipeline at various input data rates

From Fig. 3.6, we make the following observations, of which all are as expected:

- (i) The wasted power of the Combinational Block without power gating reduces with reduced input data rate. This reduction in wasted power is more evident at input data rate $\geq 1\text{Mbps}$, and is due to the reduction of the short-circuit power, which, similar to the dynamic power, is proportional to the input data rate. However, when the input data rate is $\leq 1\text{Mbps}$, the wasted power remains almost constant. This is because at these input data rates, the leakage power dominates, which remains constant (does not scale) with reduced input data rates;
- (ii) By applying the proposed power gating (with all three gating configurations), both the total power and the wasted power of the Combinational Block (including the power associated with the insertion of the gating transistor(s)) are reduced. Amongst the three gating configurations, dual gating achieves the largest total power reduction and wasted power reduction; and
- (iii) The efficacy of reducing total power by the application of the proposed power gating increases with reduced input data rate. For example, the dual gating configuration achieves $\sim 15\%$ reduction in total power at 100Mbps , $\sim 32\%$ reduction in total power at 1Mbps , and $\sim 97\%$ reduction in total power at 10kbps . In short, the proposed power gating is particularly efficacious at lower input data rate because power gating reduces wasted power (mainly leakage power) which is dominant therein. For completeness, this is different from DVS, which reduces both dynamic power and wasted power; see Chapter 4 later.

In summary, the proposed fine-grain power gating can reduce the wasted power hence the total power of the Combinational Block in the async MD pipeline across a wide

operation-space (with input data rate from 10kbps to 100Mbps). However, the delay overhead of the proposed power gating is within 15% of that without power gating. By leveraging on the existing async 4-phase handshake protocol, the proposed power gating can be similarly applied to other async pipelines adopting the 4-phase handshake protocol such as that of async QDI, and this constitutes part of our future work described in Chapter 5.

3.3 First-Order Delay Variations Estimation for Sync and its Comparison with Async QDI in Sub- V_t

The modus operandi of async QDI embodying dual-rail logic and completion detection, leading to its unconditional error-free operation under virtually all delay variations, was delineated earlier in Chapters 1 and 2. The penalty over its sync counterpart, nevertheless, is higher overheads in terms of transistor count, delay and energy per operation, E_{per} . These overheads are likely to be evident in super- V_t where, because the circuit delay variations due to PVT variations are moderate, the delay safety margins required by the sync are relatively small. However, these overheads are mitigated by the complexity of the clock infrastructure of the sync. At this juncture, we are unaware of any direct comparisons between sync and async, including how they compare for different design complexities; in Chapter 4, we attempt a direct comparison.

As delineated earlier, the delay variations increase with reduced V_{DD} and to the point of virtually intractability in sub- V_t operation, particularly in unknown and/or extreme/harsh conditions. To ensure error-free operation, the sync circuit in sub- V_t would require a commensurable large/extreme delay safety margin to accommodate said delay variations under the worst-case conditions. This will inevitably increase the delay and E_{per} (through the accumulation of leakage; for completeness, note that without self-completion detection as in

the async QDI, it is difficult to apply power gating to the sync to reduce this leakage current/energy as delineated earlier).

On the other hand, an async QDI circuit will operate as fast as the prevailing conditions allow and dissipate E_{per} accordingly with the completion of its operation indicated by its self-completion detection. Thus, to compare the efficacy (in terms of delay and E_{per}) of the sync against the async QDI in sub- V_t , the effect/amount of the delay safety margins for the sync has to be considered; also see Chapter 4 later. As delineated earlier, this delay safety margin is usually obtained through extensive statistical Monte Carlo simulations, typically a time-consuming exercise. However, as a priori at the outset of a design exercise, the (sync) circuit designer would need an estimation on the delay variations due to PVT variations (with respect to the nominal where there is no PVT variation, see eqn. (3.3) later). This would provide him/her a means to simply (re-)adjust the clock speed to (conditionally) accommodate said PVT variations.

To this end, in this section, we will propose and derive three simple analytical equations for estimating to the first-order delay variations due to P (in particular V_t), V_{DD} , and T respectively in sub- V_t operation. We will verify the derived equations by means of computer simulations to ascertain that they are sufficiently accurate for a first-order estimation of delay safety margins required by the sync. For completeness, more comprehensive analytical delay variation models were reported in literature [40], [97], [98], however, they are not suitable for first-order estimations due to their high complexity.

To study the effect of delay safety margins on the sync (with/without delay safety margins) against its async QDI counterpart, we will compare, by means of adder circuits (of

different wordlength), the delay and E_{per} of a sync and an async QDI pipeline in sub- V_t operation. For sake of circuit reliability in sub- V_t (see Chapter 2 earlier), we will adopt the static logic family for pipelines. Further, amongst the reported static async QDI logic styles, we will adopt the reported NULL-Convention-Logic (NCL) [81] (for its lower overheads, also see Chapter 2 earlier) as the representative async QDI for the comparison with the sync. We will thereafter ascertain the conditions under which either the sync or the async QDI is more competitive in terms of delay and E_{per} . For completeness, the transistor count overhead of the async QDI (independent of said delay safety margins and not taking into account the cost of the clock infrastructure of the sync) compared to its sync counterpart will also be delineated.

The remaining of this section is organized as follows. Section 3.3.1 derives the three analytical equations for estimating delay variations due to PVT variations in sub- V_t operation. The accuracy of the derived equations is thereafter verified by computer simulations. Section 3.3.2 benchmarks the sync (with delay safety margins) with the async QDI in sub- V_t operation.

3.3.1 First-Order Delay Variation Estimation due to V_t , V_{DD} and Temperature Variations

In this section, we derive three simple analytical equations for estimating first-order delay variations of digital circuits due to PVT variations operating in sub- V_t . For ease of readability, the well known characteristic delay of a CMOS inverter operating in sub- V_t earlier expressed [10] in eqn. (2.8) is repeated in eqn. (3.1) below.

$$t_{d,inv} = \frac{Q_{output,inv}}{I_{on}} = \frac{KC_{output,inv}V_{DD}}{\mu C_{ox} \frac{W}{L} (n-1)V_{th}^2 \exp(\frac{V_{DD}-V_t}{nV_{th}})} \quad (3.1)$$

From eqn. (3.1), the critical path delay t_d can be expressed as:

$$t_d = t_{d,inv} L_D \quad (3.2)$$

where L_D is the logic depth of the critical path in terms of characteristic inverter delays.

We define the delay variation due to PVT variations as:

$$\Delta t_d = \frac{t_{d2}}{t_{d1}} \quad (3.3)$$

where t_{d2} is the critical path delay with PVT variations,

and

t_{d1} is the nominal critical path delay without

PVT variations.

For sake of simplicity, we consider the worst-case scenario where the PVT variations affect the entire critical path equally and in the same manner; in reality, the variations may be different in different parts thereto and may average out. In other words, the delay variations of all logic gates due to PVT have the same magnitude and direction (either increasing or decreasing variation) along the critical path.

In eqn. (3.3), we define the delay variation as a ratio of the delay of the critical path with PVT variations over that without PVT variations. Although this definition is perhaps contentious, we deliberately define it in this fashion because we are interested in the delay of the former with respect to the latter, i.e. the number of times worse or better than the nominal (without PVT variations). Further, this definition yields simple yet insightful equations where only the PVT conditions are parameters thereto; see eqns. (3.4), (3.6), and (3.9) later. Put simply, to a sync circuit designer, this ratio directly indicates how many times the clock rate (arbitrary) needs to be slowed down to accommodate a given PVT (variation) condition.

Delay Variations due to V_t Variations

In our review in Chapter 2, it was delineated that of the parameters affected by process variations, the most important parameter is V_t ; also see Row 3 in Table 1.1 in Chapter 1 tabulating the ITRS roadmap. For this reason and for sake of simple first-order estimation, we will limit our scope of delay variations due to process variations to V_t variations alone. We denote $V_{t,nom}$ to be the nominal V_t without variation and $V_{t,var}$ to be the V_t with variations. From eqns. (3.2) and (3.3), we easily show that the delay variation due to V_t variations is:

$$\Delta t_{d(V_t)est} = \exp\left(\frac{V_{t,var} - V_{t,nom}}{nV_{th}}\right) \quad (3.4)$$

Not unexpectedly, eqn. (3.4) shows that the delay variation has an exponential dependence on the difference between $V_{t,var}$ and $V_{t,nom}$. To depict the accuracy/tolerance of eqn. (3.4) for estimating the first-order delay variations due to PVT variations for digital circuits operating in sub- V_t , HSPICE simulations are performed on an inverter implemented in a 130nm CMOS process ($|V_t| \approx 0.4V$, nominal $V_{DD} = 1.2V$) at sub- V_t from 0.15V to 0.3V and for different

$|V_t|$ variations, ranging from 10mV to 50mV. Fig. 3.7 depicts these simulated delay variations (bold lines).

Fig. 3.7: Estimated inverter delay variations ($\Delta t_{d(V_t)_{est}}$) at different V_{DD} due to $|V_t|$ variations, and comparisons against simulations ($\Delta t_{d(V_t)_{sim}}$)

To qualify the accuracy of the estimated delay variations from eqn. (3.4), we define the estimation error ($\% \Delta$) as the percentage difference between the estimated $\Delta t_{d(V_t)_{est}}$ and simulated delay variations $\Delta t_{d(V_t)_{sim}}$. These are also indicated in Fig. 3.7.

$$\% \Delta = \left| \frac{\Delta t_{d(V_t)_{est}} - \Delta t_{d(V_t)_{sim}}}{\Delta t_{d(V_t)_{est}}} \right| \times 100\% \quad (3.5)$$

From Fig. 3.7, we make the following observations:

- (i) As expected, the higher the $|V_t|$ variation, the higher are the delay variations. For example, the delay variations increase from $\sim 1.2\times$ to $\sim 4\times$ for $|V_t|$ variation of 10mV and 50mV respectively;
- (ii) The delay variations for a given $|V_t|$ variation is largely independent of the V_{DD} , i.e. the ratio of $|V_t|$ variation/ V_{DD} has little influence on the delay variation – even when the $|V_t|$ variation is a large $0.35V_{DD}$. This observation and hence ensuing insight is perhaps counter intuitive, and for completeness, this insight is difficult to observe from eqn. (3.1); and
- (iii) Despite the simplicity of derived eqn. (3.4), the estimation error ($\%\Delta$) is within 10% of the simulations – $V_{DD} = 0.3V$ with $|V_t|$ variation of 50mV. Further to (ii) above, the simulations show a slight droop in the delay variations with increasing V_{DD} . We conjecture that this may be attributed to higher-order V_{DD} dependency of parameters not considered in the derived equation.

In short, the derived equation to estimate the delay variation due to V_t variations given by eqn. (3.4) provides an insightful first-order estimation, and is accurate to the first-order.

Delay Variations due to V_{DD} Variations

For delay variations due to V_{DD} variations, we denote $V_{DD,nom}$ as the nominal V_{DD} without variation/noise and $V_{DD,var}$ as the V_{DD} with variations (e.g. the V_{DD} with noise depicted in Figs. 2.9 and 2.10 earlier). From eqns. (3.2) and (3.3), we easily show that the delay variation due to V_{DD} variations is:

$$\Delta t_{d(V_{DD})\text{est}} = \frac{V_{DD,\text{var}}}{V_{DD,\text{nom}}} \exp\left(\frac{V_{DD,\text{nom}} - V_{DD,\text{var}}}{nV_{th}}\right) \quad (3.6)$$

Not unexpectedly, eqn. (3.6) shows that the delay variation has an exponential dependence on the difference between $V_{DD,\text{var}}$ and $V_{DD,\text{nom}}$. Fig. 3.8 depicts the inverter delay variations estimated by eqn. (3.6) ($\Delta t_{d(V_{DD})\text{est}}$ plotted with dotted lines) at sub- V_t from 0.15V to 0.3V for different V_{DD} variations (ranging from -10mV to -50mV; a negative range is considered because a reduced V_{DD} is detrimental to the delay), and comparisons against simulated delay variations ($\Delta t_{d(V_{DD})\text{sim}}$ plotted with solid lines).

Fig. 3.8: Estimated inverter delay variations ($\Delta t_{d(V_{DD})\text{est}}$) at different V_{DD} due to V_{DD} variations, and comparisons against simulations ($\Delta t_{d(V_{DD})\text{sim}}$)

From Fig. 3.8, we make the following observations:

- (i) As expected, the higher the (negative) V_{DD} variation, the higher are the delay variations. For example, the delay variations increase from $\sim 1.2\times$ to $\sim 3\times$ for V_{DD} variations of -10mV and -50mV respectively;
- (ii) The delay variations for a given V_{DD} variation is largely independent of the V_{DD} , i.e. the ratio of V_{DD} variation/ V_{DD} has little influence on the delay variation; and
- (iii) The estimation error ($\%\Delta$) of eqn. (3.6) is within 12% of the simulations, and the largest estimation error is for $V_{DD} = 0.3V$ with V_{DD} variation of -50mV. Further to (ii) above, the estimations show a slight increase in the delay variations with increasing V_{DD} . This is, as expected, more evident for larger V_{DD} variations, e.g. for V_{DD} variation of -50mV, the estimated delay variation increases from $\sim 2.6\times$ @ $V_{DD}=0.15V$ to $\sim 3.2\times$ @ $V_{DD}=0.3V$. On the other hand, the simulations show a more constant (flat) delay variation relationship with increasing V_{DD} . Similar to the delay variations due to V_t variations depicted earlier in Fig. 3.7, we conjecture that this discrepancy between the simulations and the estimations may be attributed to higher-order V_{DD} dependency of parameters not considered in the derived equation.

In short, the derived equation to estimate the delay variation due to V_{DD} variations given by eqn. (3.6) provides an insightful first-order estimation, and is accurate to the first-order.

Delay Variations due to Temperature Variations

Temperature variation affects circuit delay through several parameters including thermal voltage V_{th} , carrier mobility μ and threshold voltage V_t . The temperature effect on V_{th} is well

established and given by $V_{th} = kT/q$. On the other hand, the temperature effect on μ and V_t is modeled (in BSIM4 level-54 transistor model [99]) respectively by the mobility temperature exponent (denoted UTE in BSIM4) and the temperature coefficient for V_t (denoted K_{T1} in BSIM4). Both μ and V_t decrease with increasing temperature. By our denotation of T_{nom} , μ_{nom} and $V_{t,nom}$ as the nominal parameters without variation while T_{var} , μ_{var} and $V_{t,var}$ as the parameters with variations, the BSIM 4 model states that

$$\mu_{var} = \mu_{nom} \left(\frac{T_{var}}{T_{nom}} \right)^{UTE} \quad (3.7)$$

$$V_{t,var} = V_{t,nom} + K_{T1} \left(\frac{T_{var}}{T_{nom}} - 1 \right) \quad (3.8)$$

where UTE and K_{T1} are -1.85 and -0.25 respectively in the chosen process (130nm CMOS).

We denote $V_{th,nom}$ as the nominal V_{th} and $V_{th,var}$ as the V_{th} with variations. From eqns. (3.2) and (3.3), we show that the estimated delay variation due to temperature variations ($\Delta t_d(T)_{est}$) is:

$$\begin{aligned} \Delta t_d(T)_{est} &= \frac{\mu_{nom} T_{nom}^2}{\mu_{var} T_{var}^2} \exp \left(\frac{V_{DD} - V_{t,nom}}{nV_{th,nom}} - \frac{V_{DD} - V_{t,var}}{nV_{th,var}} \right) \\ &= \left(\frac{T_{nom}}{T_{var}} \right)^{2+UTE} \exp \left((V_{t,nom} - K_{T1} - V_{DD}) \frac{(T_{nom} - T_{var})q}{nkT_{nom}T_{var}} \right) \end{aligned} \quad (3.9)$$

Not unexpectedly, eqn. (3.9) shows that the delay variation has an exponential dependence on the difference between T_{var} and T_{nom} . Fig. 3.9 depicts the inverter delay variations estimated

by eqn. (3.9) ($\Delta t_{d(T)\text{est}}$ plotted with dotted lines) at sub- V_t from 0.15V to 0.3V for different T variations (ranging from -5°C to -25°C with $T_{\text{nom}} = 25^\circ\text{C}$ (298K); a negative range is considered because a reduced temperature is detrimental to the delay in sub- V_t [6]), and comparisons against simulated delay variations ($\Delta t_{d(T)\text{sim}}$ plotted with solid lines).

Fig. 3.9: Estimated inverter delay variations ($\Delta t_{d(T)\text{est}}$) at different V_{DD} due to T variations, and comparisons against simulations ($\Delta t_{d(T)\text{sim}}$)

From Fig. 3.9, we make the following observations:

- (i) As expected, at a given V_{DD} , the higher the (negative) T variation, the higher is the delay variation. For example, the delay variation @ $V_{DD} = 0.15\text{V}$ increases from $\sim 1.2\times$ to $\sim 2.9\times$ for T variation of -5°C and -25°C respectively;
- (ii) The delay variation for a given (negative) T variation decreases with increasing V_{DD} , and particularly so for higher T variations. For example, for T variation of

-25°C, the delay variation decreases from $\sim 2.8\times$ @ $V_{DD}=0.15V$ to $\sim 2.0\times$ @ $V_{DD}=0.3V$; and

- (iii) Eqn. (3.9) is perhaps somewhat unexpectedly precise. The estimation error ($\%\Delta$) is within 0.9% of the simulations and the largest estimation error is for $V_{DD}=0.15V$ with T variation of -20°C.

In short, the derived equation to estimate the delay variation due to T variations given by eqn. (3.9) provides an insightful first-order estimation, and is accurate to the first-order.

In general, the delay variations estimated by the derived eqns. (3.4), (3.6) and (3.9) agree generally well with that obtained from circuit simulations (with an inconsequential worst-case error of $<12\%$), hence appropriate as a first-order estimation and insightful for its simplicity (for interpretation). The accuracy of the derived equations can easily be improved by adding heuristics and this constitutes part of our future work described in Chapter 5.

3.3.2 Benchmarking Sync and Async QDI in Sub- V_t

From the first-order delay variation estimations in the preceding section for V_t , V_{DD} and T variations for sync, we will now use said first-order delay estimations for ascertaining a sync pipeline delay variations under various PVT variations operating in sub- V_t . This sync pipeline will thereafter be benchmarked against its counterpart async QDI pipeline under the same conditions. The computational logic in both pipelines is a Carry Ripple Adder (CRA) of various wordlengths (8-, 16- and 32-bits) where single-rail and dual-rail logic are respectively used in the sync and async QDI.

Fig. 3.10(a) and (b) depict a sync pipeline stage and an async QDI pipeline stage respectively. At the outset, note that in the sync pipeline, as delineated earlier in Chapters 1 and 2, a global clock signal with a clock period longer than the worst-case delay of the critical path (to accommodate delay variations due to PVT variations) is required for error-free operation (correct data synchronization). Conversely, in the async QDI pipeline embodying dual-rail logic and completion detection (CD) circuits, the pipeline innately adapts to variations in circuit delay. Amongst the reported static QDI logic styles, the NCL is chosen for its lower area and power overheads than the competing designs (DIMS and DSLI, see Chapter 2 earlier); a novel QDI logic style is proposed in Chapter 4 that features even lower overheads than NCL, DIMS and DSLI.

Fig. 3.10: Pipeline stage: (a) Sync, and (b) Async QDI

Fig. 3.11(a) and (b) depict respectively the schematic of the sync and async QDI full-adders for the CRAs, and are simulated @130nm CMOS with the same input patterns @ $V_{DD} = 0.15V$.

Fig. 3.11: Full-adder design: (a) Single-rail sync and (b) Dual-rail async NCL

Fig. 3.12 depicts the block diagram of the 8-bit async NCL CRA, which includes a completion detection circuit (CD circuit) comprising 2-input OR gates and 2-input C-elements. The 16- and 32-bit async NCL CRA are similarly implemented.

Fig. 3.12: Block diagram of the 8-bit async NCL CRA

In this benchmarking exercise between the sync and async, the parameters of interest are delay, energy (E_{per}) and transistor count, and the benchmarking are tabulated in Tables 3.1, 3.2 and 3.3 respectively. The results are obtained from pre-layout simulations. For the sync pipeline, we consider the best-case (i.e. no delay safety margin required) and three cases of delay variations due to V_t , V_{DD} and T as predicted by eqns. (3.4), (3.6) and (3.9) respectively. For sake of more comprehensive benchmarking for a practical perspective, we further define a ‘turning-point’ delay safety margin, the additional clock delay margins needed by the sync pipeline so that it has the same delay or E_{per} as its async QDI counterpart.

Benchmarking Delay

Table 3.1 tabulates the delay benchmarking between the async QDI CRAs and the sync CRAs (without/with various first-order delay safety margins estimated by the derived equations) to accommodate different PVT variations). The delays of the sync CRAs (without delay safety margins) are taken as the computation delays associated with the longest carry propagations. On the other hand, the delays of the async QDI CRAs are taken from the time when the input data are ready to the time when the completion signal is asserted (see Fig. 3.12). Three specific PVT⁴ (variation) conditions are considered in the benchmarking, namely a 50mV $|V_t|$ variation ($|V_t|$ tolerance of the chosen process), a -15mV V_{DD} variation (10% of $V_{\text{DD}} = 0.15\text{V}$) and a -25°C T variation (equivalent to an operating temperature of 0°C, the lower range for commercial grade electronics (0°C to 70°C); see Chapter 4 later for our proposed WSN, where a wider temperature range of -55°C to 125°C for military grade electronics is considered). For ease of readability, in Table 3.1, the delays

⁴ These variations are congruous with that stipulated by the ITRS roadmap for nominal V_{DD} , see Table 1.1 earlier. When operating in sub- V_t , these variations are in fact optimistic because the variations therein may become extreme/virtually intractable.

of the sync CRAs are normalized to their respective async QDI counterparts of the same wordlength, and the actual values are shown within parentheses.

Table 3.1: Delays of the async QDI CRAs and the sync CRAs (without/with delay safety margins) @ $V_{DD}=0.15V$; the delays are normalized to the async QDI CRAs of respective wordlengths

	Delay (μs)		
	8-bit	16-bit	32-bit
Async QDI CRA	1.00 (20.6)	1.00 (36.1)	1.00 (67.2)
Sync CRA (without delay safety margin)	0.57 (11.8)	0.60 (21.7)	0.62 (41.7)
Sync CRA (to accommodate 50mV $ V_t $ variation)	2.26 (46.5)	2.38 (85.9)	2.46 (165.1)
Sync CRA (to accommodate -15mV V_{DD} variation)	0.85 (17.5)	0.89 (32.1)	0.92 (61.7)
Sync CRA (to accommodate -25°C T variation)	1.62 (33.3)	1.70 (61.4)	1.76 (118.0)
‘Turning-point’ delay safety margin	1.8×	1.7×	1.6×

From Table 3.1, we make the following observations:

- (i) As expected, when no delay safety margin (the best-case) is considered for the sync CRAs of all three wordlengths, they feature shorter delays (on average 40% shorter) than their async QDI counterparts. Nevertheless, this best-case timing for the sync is unrealistic considering the need to accommodate (for error-free operation) the extreme/virtually intractable delay variations due to PVT variations in sub- V_t operation (also see footnote 4);
- (ii) The delay advantage of the sync CRAs without delay safety margins (or conversely the delay overheads of the async QDI CRAs) decreases slightly with increasing wordlength. This is because the part of the delay overheads of the async QDI attributed to the completion detection circuit becomes less significant with increasing wordlength (i.e. the delay does not increase proportionally with the wordlength);

Note that in an actual circuit comprising multiple pipeline stages, part of the delay overhead associated with the completion detection circuit of the async QDI in one pipeline stage overlaps with the latching of the computed data by the subsequent stage. Consequently, this part of the delay overhead becomes even less significant, hence the diminishing delay advantage of the sync CRAs over the async QDI CRA. For example, see Chapter 4 later for the delay of the datapath completion detection circuit in the async QDI filter bank for a WSN; and

- (iii) As predicted, when varying amount of delay safety margins are considered for the sync CRAs to accommodate the different PVT (variation) conditions, it can be argued, perhaps somewhat contentiously that the delay advantages of the sync CRAs compared to their async QDI counterparts diminish and eventually defeated at the ‘turning-point’. The contention is that the delay of async QDI CRA may likewise increase under said conditions. Nevertheless, due to the adaptive nature of the async QDI, the delay of the async QDI may also decrease when the conditions are more benign (than the nominal condition), where, on the other hand, the delay of the sync will be fixed to the worst-case condition. Further, the delay of the async QDI is ascertained according to the prevailing condition and not deliberately designed to the absolute worst-case (as in sync).

For completeness, the benchmarking herein is, although useful, somewhat simplistic. In Chapter 4, the delay safety margin for the sync is ascertained by means of SSTA through Monte Carlo simulations, and 3σ delay variation is chosen to obtain 99.7% coverage. The delay of its async QDI counterpart is

likewise for $\pm 3\sigma$ delay variations. In view of this, the comparisons herein between the sync and the async QDI consider the (average-case) nominal condition. The same argument also holds for the E_{per} benchmarking; see Chapter 4 later (in particular, Figs. 4.11 and 4.13) for the delay and E_{per} benchmarking between a sync and an async QDI filter bank in a WSN.

For the aforesaid, when relatively modest delay safety margins of $1.8\times$, $1.7\times$ and $1.6\times$ are added to the sync CRAs of the three wordlengths, their delays are equal to their async QDI counterparts. Put simply, if the sync CRAs are designed to accommodate the three specific PVT conditions, their delays are longer than their async QDI counterparts when operating in the nominal (no PVT variations) conditions.

Overall, the delay overhead of the async QDI CRAs is expected to be small compared to their sync counterparts, possibly advantageous, because the latter in a practical application would need to be designed for the expected worst-case condition. This is particularly the case in sub- V_t because of the extreme/virtually intractable PVT variations therein. A more comprehensive benchmarking between the sync and async QDI is given in Chapter 4 later.

Benchmarking E_{per}

Table 3.2 tabulates the E_{per} benchmarking between the async QDI CRAs and the sync CRAs without/with various delay safety margins (first-order estimation from the derived eqns. (3.4), (3.6) and (3.9)) to accommodate different PVT variations. The E_{per} of the CRAs (both sync and async) are taken as the total energy dissipated during their delays defined respectively in Table 3.1. The same three PVT (variation) conditions considered in the

delay benchmarking are considered for the E_{per} benchmarking. As in the delay benchmarking, the E_{per} of the sync CRAs are normalized to their respective async QDI counterparts of the same wordlength, and the actual values are shown within parentheses.

Table 3.2: E_{per} of the async QDI CRAs and the sync CRAs (without/with delay safety margins) @ $V_{\text{DD}}=0.15\text{V}$; the E_{per} are normalized to the async QDI CRAs of respective wordlengths

	E_{per} (fJ)		
	8-bit	16-bit	32-bit
Async QDI CRA	1.00 (79.3)	1.00 (221.0)	1.00 (768.5)
Sync CRA (without delay safety margin)	0.53 (42.2)	0.45 (99.4)	0.41 (318.3)
Sync CRA (to accommodate 50mV $ V_t $ variation)	0.87 (69.2)	1.01 (223.2)	1.44 (1105.7)
Sync CRA (to accommodate -15mV V_{DD} variation)	0.58 (46.6)	0.54 (119.4)	0.58 (445.7)
Sync CRA (to accommodate -25°C T variation)	0.74 (58.9)	0.80 (175.9)	1.05 (805.0)
‘Turning-point’ delay safety margin	5.0×	3.9×	2.7×

From Table 3.2, we make the following observations:

- (i) As expected, when no delay safety margin (the best-case) is considered for the sync CRAs of all three wordlengths, they feature lower E_{per} (on average 54% lower) than their async QDI counterparts. However, as delineated earlier, this best-case timing is unrealistic for the sync in view of the extreme/virtually intractable PVT variations in sub- V_t operation;
- (ii) Using the same argument in comment (ii) for Table 3.1, as expected, when delay safety margins are considered for the sync CRAs to accommodate various PVT variations, the E_{per} advantages of the sync CRAs counterparts diminish and eventually defeated at the ‘turning-point’. This can be largely attributed to the increased accumulation of leakage energy of the sync CRAs over the longer

delays. However, the ‘turning-point’ delay margins for E_{per} are longer (on average $3.9\times$) than those for delay (on average $1.7\times$; see Table 3.1 earlier).

- (iii) The turning-point delay safety margin of E_{per} decreases with increasing wordlength (from $5.0\times$ for 8-bit CRA to $2.7\times$ for 32-bit CRA). This is expected as a longer wordlength (hence larger circuit) will dissipate higher leakage current, thereby accumulating E_{per} faster.

Overall, similar to the argument for the delay benchmarking, the E_{per} overhead of the async QDI CRAs is expected to be small compared to their sync counterparts, possibly advantageous, due to the latter’s need to accommodate for the worst-case condition in sub- V_t operation. A more comprehensive E_{per} benchmarking between the sync and async QDI for a WSN is given in Chapter 4 later.

Benchmarking Transistor Count

Table 3.3 tabulates the transistor count of the async QDI and the sync CRAs. It is not unexpectedly that the async QDI CRAs have, on average $\sim 3\times$ larger transistor count than their sync counterparts. As delineated in Chapters 1 and 2, this is attributed to the dual-rail encoded logic and completion detection circuit of the former. However, it is worthwhile to note that through careful design and layout techniques, the actual IC area overhead of the async QDI can be mitigated to within $\sim 1.5\times$ of the sync. In a practical larger circuit or system, the clocking infrastructure of the sync is typically a significant portion of the overall IC area. In this context, it is difficult to comment if the sync or the async QDI is

advantageous; see Chapter 4 later for our proposed low area overhead async QDI logic style – ‘Pre-Charged-Static-Logic’.

Table 3.3: Transistor count of the async QDI CRA and the sync CRA

	Transistor Count		
	8-Bit	16-Bit	32-Bit
Async QDI CRA	1854	3694	7392
Sync CRA	638	1234	2406
Overhead of Async QDI	2.9×	3.0×	3.1×

In summary, by means of the first-order estimations of delay variations given by our derived eqns. (3.4), (3.6) and (3.9), the delay variations of the sync pipeline can be easily estimated from the pipeline delay (from simulations) with no variations. From the above benchmarking, it is apparent that the delay and E_{per} advantages of the sync over its async QDI counterpart diminish in sub- V_t operation when varying amount of delay safety margins (to accommodate the high variation-space in terms of PVT variations) are considered for the sync. Under some circumstances, it can be argued that the async QDI is advantageous. In short, although the general view (within the digital design community) is that the sync is advantageous, this has not yet been conclusively or rigorously verified – as shown herein.

3.4 Conclusions

In this chapter, we have proposed a fine-grain power gating methodology (applicable to three different gating configurations) for an async MD pipeline in a very wide operation-space – the pipeline alternating between active and idle – to reduce the short-circuit and leakage wasted powers. The proposed methodology was shown to be efficacious in terms of reducing said wasted power, hence the total power of the conventional MD pipeline (for all said three configurations), yet the ensuing overhead is low, specifically one inverter (per pipeline stage) and $<15\%$ delay. In this chapter, we have proposed and derived a set of simple analytical equations to estimate to the first-order the delay variations (due to PVT) of digital circuits with respect to the same without delay variations (nominal), operating in sub- V_t . The derived equations have been verified by simulations to be shown to be useful, with a largely inconsequential (being first-order estimations) worst-case error of $<12\%$. On the basis of our simple derived equations, we have thereafter compared, by means of adder circuits, the sync (with delay safety margins) against the async QDI (with self-completion detection). It was ascertained that neither the sync nor the QDI async is particularly advantageous in all conditions, and this exercise depicted the usefulness and valuable insights provided by the simple derived equations.

Chapter 4 An Ultra Low-Power Asynchronous-Logic In-Situ Self-Adaptive V_{DD} System for Wireless Sensor Networks, and Proposed ‘Pseudo-QDI’ Signaling Protocol

4.1 Introduction

It was established in the preceding chapters that although async circuits, particularly QDI async, offer unprecedented operation robustness over their sync counterparts for high variation-space (in sub- V_t operation), this is often at the cost of added hardware and power overheads. Nevertheless, by exploiting its innate adaptation to operate at its inherent maximum speed for the given prevailing conditions (without delay safety margin), async QDI circuits may under some circumstances outperform their sync counterparts in terms of delay and E_{per} in sub- V_t . This demonstration in Chapter 3 was, however, only for a relatively simple circuit that may not necessarily be representative of real-life or complex systems. To this end, in this chapter, we will explore the merits and disadvantages of sync and async QDI for a complex practical application – a WSN whose operating conditions include very high variation-space (-55°C to $+125^{\circ}\text{C}$) and very wide operation-space (0.1kSamples/s (kS/s) to 100kS/s) in the sub- V_t regime. Of particular interest for the async QDI, by exploiting the signaling protocol thereof in a novel fashion, we design and realize monolithically a sub- V_t self-adaptive V_{DD} scaling system (SSAVS) for the WSN in an attempt to enable lowest power possible operation (by means of the lowest V_{DD} to within 50mV to meet the prevailing operating conditions). To fairly benchmark against the sync counterpart – the equivalent being a DVFS system requiring highly time-consuming comprehensive pre-characterizations – V_{DD} is manually tuned (as a priori information is unavailable) for a given operating

condition. To ensure that the benchmarking is both fair and useful, the benchmarking includes the benchmarking of delay, E_{per} and power dissipation under a myriad of real-life conditions and taking into consideration $\pm 3\sigma$ delay (due to process variations) and 10% V_{DD} variations.

To reduce the hardware, power and delay overheads of reported QDI async cells, we will describe our proposed ‘Pre-Charged-Static-Logic’ (PCSL) cells. To further reduce the hardware and power overheads of the standardized async QDI protocol, we propose a simplification to said protocol, and we coin this simplified protocol, ‘Pseudo-QDI’. For this proposed protocol, although requiring a timing assumption, we show that the timing assumption is easily satisfied in practical digital circuits and systems. To depict the efficacy of ‘Pseudo-QDI’, we benchmark, by means of measurements on prototype ICs, the pseudo-QDI against its standardized QDI counterpart under high variation-space and wide operation-space conditions.

The work reported in this chapter is largely extracted from our two papers published in the *IEEE Journal of Solid-State Circuits* [100] and *Proc. IEEE Sub-threshold Microelectronics Conf.*, 2012 [101]. The latter was awarded the ‘Best Student Paper’ at said conference.

4.2 Sub- V_t Self-Adaptive V_{DD} Scaling (SSAVS) System for Wireless Sensor Networks (WSNs)

Wireless Sensor Networks (WSNs) are increasingly ubiquitous, in part, due to their ultra low-power and high reliability operation. Fig. 4.1 depicts the WSN node of interest, comprising five main modules: Sensor Front-End, Signal Processor, Wireless Transceiver,

Energy Source, and Power Management. As the WSN is typically designed for multiple-year operational life-span [2], power is carefully budgeted and where pertinent, energized only when required, such that the overall average power is typically 10 – 100 μ W [20].

In our WSN depicted in Fig. 4.1, its overall active/passive operation ratio is approximately 20/80. In the passive mode, only the Sensor Front-End module is continuously energized. The Sensor and the Conditioning Circuits therein are powered directly by V_{DD_BAT} (~ 2.8 V), a Lithium/Carbon Fluoride (Li/CF_x) battery, via a Low-Dropout (LDO) Regulator. The Simple Processor is powered by V_{DD_NOM} (1.2V) via a power-efficient Buck DC-DC Converter. The Li/CF_x battery is appropriate largely because of its high energy density per weight and very wide operating temperature range (-60°C to 160°C), congruent with that required of our WSN [102]. The Simple Processor ascertains if the input is possibly useful, and if it is, the WSN goes into active mode where the Simple Processor signals the Power Management module to energize the Signal Processor module via V_{DD_ADJ} . The voltage of V_{DD_ADJ} , typically in the sub-threshold voltage (sub- V_t) range, is self-adjusted such that the lowest possible voltage is used – to enable ultra low-power operation. The Signal Processor module buffers (via a FIFO) the output of the Simple Processor, filters the output signal before final computation by the Microcontroller Unit (MCU). When the MCU ascertains that the filtered signal is useful, the Wireless Transceiver is energized and the processed signal is subsequently transmitted wirelessly. With the wireless transmission expected to be $<0.01\%$ active and with a 20/80 WSN active/passive operation, $\sim 50\%$ of the overall power is attributed to the Signal Processor module, which is of interest in terms of power dissipation.

The approaches taken to minimize power involve all levels of the design space including algorithmic design and at the hardware level. In the former, the filtering in the Signal

Processor module embodies the Frequency Response Masking (FRM) technique [103]. This involves the Interpolated Finite Impulse Response (IFIR) Filter and the FRM Filter Bank (FB), and is computationally more efficient than the usual FIR and IIR filter approaches. Ultra low-power design techniques in the latter are extensively reported in literature [104]-[106] and of these, operation in the sub- V_t region is one of the most effective. This is particularly applicable here because the speed of the digital circuits in the Signal Processor is modest – the clocking speed ranges from 1.4kHz to 1.4MHz for a sampling rate range from 0.1kSamples/s (kS/s) to 100kS/s.

Fig. 4.1: Block diagram of the WSN node

As delineated earlier in Chapters 1 and 2, despite the potential advantages of sub- V_t operation, this region of operation is challenging here for several reasons. First, the WSN is designed to work in a wide range of conditions, including extreme environments (-55°C to $+125^{\circ}\text{C}$) somewhat similar to [14]. Second, PVT variations for fine-dimensioned CMOS processes increase dramatically in sub- V_t operation, and the ensuing delay variations are very severe, possibly intractable. Typically, to accommodate for such high variation-space, a very large delay safety margin (for sync circuits) would need to be allowed for, for example $>200\times$ [14]. Third, the input signal to the Signal Processor module is variable (i.e. a wide operation-space). From a robust operation perspective, the circuits would need to be designed to meet the worst-case conditions – the fastest input rate and extreme temperatures.

To design the WSN for ultra low-power operation, we adopt a self-adjusting V_{DD} approach whilst operating in the sub- V_t region, termed ‘Sub-threshold Self-Adaptive V_{DD} Scaling’ (SSAVS) where the V_{DD} is in-situ dynamically self-adjusted. The modus operandi involves ‘dialing up’ V_{DD} when the need for computation increases or when the operating conditions are less favorable, and V_{DD} is ‘dialed-down’ when the conditions are the converse. Put simply, the lowest V_{DD} is used where possible because in general the lower the V_{DD} , the lower is the power dissipation due to dynamic and leakage currents (see eqn. (1.1) in Chapter 1 earlier). In this section, we describe an SSAVS system for the Signal Processor module in a WSN based on a proposed methodology within the Quasi-Delay-Insensitive (QDI) async approach, and with a novel in-situ self-adjusting V_{DD} means. The proposed design methodology, coined ‘Pre-Charged-Static-Logic’ (PCSL), is essentially a static-logic library cell architecture that exploits the fast reset feature and is appropriate for full-range Dynamic Voltage Scaling (DVS) [53] – for V_{DD} ranging from nominal voltage to deep sub- V_t . The proposed SSAVS system for the WSN is demonstrated by means of application to the FRM

FB. The novel self-adjustment is obtained very simply – by exploiting (and comparing) the existing Request (*Req*) and Acknowledge (*Ack*) signals of the QDI protocol signaling, and thereafter adjusting the V_{DD_ADJ} accordingly (see Section 4.2.2 later). The ensuing overhead is hence very low.

The remaining of this section is organized as follows. Section 4.2.1 reviews adaptive V_{DD} scaling systems. Section 4.2.2 presents the design of the proposed system. Section 4.2.3 presents the measurement results of prototype ICs and benchmarking thereof.

4.2.1 Adaptive V_{DD} Scaling Systems

The general modality of adaptive V_{DD} scaling systems to reduce power is to adaptively adjust V_{DD} as low as possible (with appropriate timing margin) to meet the throughput requirement for the prevailing operating conditions (including PVT variations). This largely requires the pertinent circuit delay variations to be tracked, observed, or inferred.

A reported delay tracking technique is based on a Look-Up Table [15], [18] comprising tabulated pre-characterized throughput versus V_{DD} data according to critical path circuit delay(s) under worst-case PVT conditions for the given throughput. As delineated earlier in Chapter 1, to avoid excessive timing margins, Statistical Static Timing Analysis [15] may be employed mostly to account for local (within-die) variations. Another reported technique [107] attempts to track real-time variations by adding PVT sensors. However, in sub- V_t operation, because of the exponential relationship of sub- V_t delay with PVT, even small errors in these sensor readings could lead to large circuit delay uncertainties, and the overheads associated with the sensors may defeat any advantage. The reported critical path

delay matching [108]-[111] involves a ring oscillator matched to the critical path delay to set the clock frequency, and V_{DD} is subsequently adjusted. For improved matching, the entire logic of the critical path may be replicated at high hardware cost [110]. Although this may be able to mitigate the delay uncertainties issues associated with global PVT variations, it may not comprehensively account for local variations, particularly in sub- V_t operation. Another reported technique employs timing error detection/correction [112]-[115], where V_{DD} is reduced until the ensuing computation is erroneous. V_{DD} is thereafter increased and the computation repeated. The applicability of this technique is arguably limited due to the severe/intractable PVT variations in sub- V_t operation, to possibly severe meta-stability issues due to the lack of timing margin, and to the need for re-computations. Another reported technique [116], [117] attempts to ascertain the circuit delay indirectly by measuring the variations in the supply current drawn to infer the ‘duration’ of the computation, and V_{DD} subsequently adjusted. This technique is likely to be ambiguous in sub- V_t operation where the ratio of the current during computation to idle is small.

On the basis of the aforesaid review, it can be argued that these reported tracked, observed and inferred techniques are inadequate in terms of robustness, particularly in sub- V_t operation. Further, the hardware/computation overheads are considerable, including the need to scale V_{DD} with the scaling of the clock frequency, i.e. DVFS; see Chapter 1 earlier.

We instead propose a *definitive* means by directly measuring the delay and comparing it against the throughput for the prevailing conditions, and V_{DD} is thereafter adjusted accordingly. To enable this, we adopt the self-timed async QDI (vis-à-vis the conventional sync) where its dual-rail encoding includes the Request (*Req*) signal which indicates that the input sample is ready and the Acknowledge (*Ack*) signal that indicates the completion of the

computation. By counting the number of *Req* against *Ack* within a given period, we ascertain if the delay of the circuit is excessive, or otherwise, with respect to the throughput for the prevailing conditions. V_{DD} is thereafter adjusted accordingly such that the delay is just slightly less than the delay between input samples, thereby satisfying the throughput. Further, as *Ack* is inherent in QDI async protocols, the computation is uninterrupted while V_{DD} is transitioning during its self-adjustment; in reported adaptive V_{DD} scaling systems, circuit operation typically ceases when V_{DD} is transitioning [18]. Of specific interest, note that the delay is definitive because the delay is that ascertained for the prevailing operating conditions, and we will show later that the associated hardware to adjust V_{DD} is very modest.

At this juncture, to the best of our knowledge, ultra low-power QDI circuits with self-adaptive V_{DD} , operating in the sub- V_t region and in extreme environments (hence requiring extremely high reliability), have yet to be reported or demonstrated. Further it would be interesting to compare their attributes, including IC area, delay, energy/operation (E_{per}) and power dissipation, against their conventional sync DVFS counterpart and under various conditions (see Section 4.2.3 later).

4.2.2 System Design

Fig. 4.2 depicts the proposed SSAVS system within the Power Management module embodying the SSAVS Controller and its associated adjustable V_{DD} means (a Buck DC-DC Converter), and the PCSL-based 8×8-Bit Quad-Channel Async QDI FRM FB within the FRM FB. There are two V_{DD} voltage rails in the overall proposed SSAVS system: a fixed $V_{DD_NOM}=1.2V$ and a variable V_{DD_ADJ} whose sub- V_t voltage typically ranges from 150mV to 400mV. For ease of illustration, the specific V_{DD} rail is shown in parenthesis for the supply

rails and for signals of the various modules. In Fig. 4.2, the voltage of *Input* and of *Req* signals is first adjusted from $V_{DD_NOM}=1.2V$ to V_{DD_ADJ} by the Step-Down Level Converter, and are thereafter buffered by the Async FIFO Buffer (depth of 50) before input (*Input_FB* and *Req_FB*) to the async FRM FB. The FB outputs (*Output1-4*) and their associated *Ack* (combined from *Ack1-4* via the Completion Detection Circuit) are output to the MCU for further processing. *Ack* is also fed back to the Async FIFO Buffer. The *Req* and *Ack* signals are input to the Power Management module, and *Ack* is stepped up from V_{DD_ADJ} to V_{DD_NOM} . The SSAVS Controller within the Power Management module monitors the number of *Req* and *Ack* signals in each *Req_vs_Ack_Clk* period (a 10 Hz clock generated by the Update V_{DD} Clock Generator for a target throughput of $<1kS/s$). The V_{DD_Code} is a 5-bit code that sets one of 24 voltage levels (in the Buck DC-DC Converter) ranging from ‘00000’=50mV to ‘10111’=1.2V (in 50mV steps) for V_{DD_ADJ} .

Fig. 4.2: Overall structure of the proposed SSAVS system with an async QDI FRM Filter Bank (FB); $V_{DD_NOM} = 1.2V$, V_{DD_ADJ} ranges from 150mV – 400mV

Fig. 4.3 graphically depicts an example of the self-adjustment of V_{DD_ADJ} . When the WSN is first initiated, the SSAVS Controller outputs $V_{DD_Code} = '10111'$, equivalently $V_{DD_ADJ} = 1.2V$, and the speed of the FB would far exceed the required computation. In this scenario, the number of FB *Ack* clocks will be equal to the number of *Req* clocks in each *Req_vs_Ack_Clk* period. In the next *Req_vs_Ack_Clk* period, the SSAVS Controller will subsequently decrement V_{DD_Code} by 1 bit to '10110' and V_{DD_ADJ} correspondingly reduces by 50mV to 1.15V. The process continues where V_{DD_Code} is continuously decremented as with the voltage of V_{DD_ADJ} commensurably reduced. Eventually, at period t in Fig. 4.3, V_{DD_Code} is decremented to '00010', equivalently $V_{DD_ADJ} = 150mV$. This is the juncture where the speed of the FRM FB is just slightly slower than the *Input* data rate for the prevailing conditions – the number of *Req* clocks hence exceeds the number of *Ack* clocks in one *Req_vs_Ack_Clk* period.

Fig. 4.3: An example of the variation of V_{DD_ADJ} with time. The logical numbers on the ordinate are V_{DD_Code} and their corresponding DC voltages (V_{DD_ADJ})

Although the speed of the FRM FB is slightly too slow, no error occurs because the unconsumed inputs are stored in the Async FIFO Buffer (Fig. 4.2). In the next period, $t+1$, the SSAVS Controller reacts accordingly by incrementing V_{DD_Code} by 1 bit to '00011' and the corresponding V_{DD_ADJ} increased by 50mV to 200mV. With V_{DD_ADJ} increased, the speed of the FRM FB now slightly exceeds the required computation and the unconsumed inputs stored in the FIFO buffer (*Input_FB*) are in turn computed at a slightly faster rate than the *Input* data rate. Consequently, the number of *Req* clocks is now less than the number of *Ack* clocks and at the end of this $t+1$ period, all unconsumed inputs in the FIFO may have been cleared; if not, the voltage of V_{DD_ADJ} remains (or increased further) in the next time period(s). If cleared, in the next period $t+2$, the number of *Req* clocks again equals to the number of *Ack* clocks (as in time periods preceding t). This is the same scenario where the FB, as a consequence of the slightly raised V_{DD_ADJ} , is capable of computing faster than the *Input* data rate. In the next period $t+3$, the scenario is that as in period t , and the operation repeats accordingly. Table 4.1 summarizes the three operational conditions.

Table 4.1: Operation of the SSAVS controller

	Operational Conditions	SSAVS Controller Action
(i)	Operation is too fast (No. <i>Req</i> clocks = No. <i>Ack</i> clocks)	Decrement V_{DD_Code} and decrease V_{DD_ADJ} (e.g. period $< t$ in Fig. 3)
(ii)	Operation is too slow (No. <i>Req</i> clocks > No. <i>Ack</i> clocks)	Increment V_{DD_Code} and increase V_{DD_ADJ} (e.g. period t in Fig. 3)
(iii)	Operation is slightly faster than required & the FIFO buffer is being cleared (No. <i>Req</i> clocks < No. <i>Ack</i> clocks)	Maintain V_{DD_Code} and maintain V_{DD_ADJ} (e.g. period $t+1$ in Fig. 3)

In short, the voltage of V_{DD_ADJ} of the FB is in-situ adaptively self-adjusted to be as low as possible (within 50mV) to meet the throughput for the prevailing operating conditions, and on average, the voltage of V_{DD_ADJ} is slightly higher than the actual required minimum. Hence, the FB is ultra low-power and highly power-efficient. Note that the overheads for this self-adjusting V_{DD} are very modest (a counter) and the circuit operation is uninterrupted whilst V_{DD} transitions.

As delineated earlier in Chapter 2, in view of the need for sub- V_t operation, it is imperative to adopt circuits based on the static logic family to mitigate the effects of critical transistor sizing; dynamic and pass transistor logic families are inappropriate. Fig. 4.4(a) depicts the basic architecture of our proposed async cells, coined ‘Pre-Charged-Static-Logic’ (PCSL) [53]. This basic architecture comprises an Inverting Static-Logic Cell, three transistors (for output pre-charging during the reset phase/evaluation during the computation phase), and two inverters (for output buffering). The outputs are $Q.T$ (Output True) and $Q.F$ (Output False). In PCSL cells, when Req is ‘0’, both outputs are ‘0’. On the other hand, when Req is ‘1’ (indicating that an operation is ready) and when the input signals are valid, the operation commences and an ensuing output is obtained. The architecture of the PCSL cell involves an integration of the subcircuit associated with the Req signal and a buffer (to each output) into the standard static-logic library cell (redesigned for dual-rail async), thereby sharing of (common) transistors. This reduces the number of transistors, resulting in simultaneous lower power/energy dissipation, faster speed and smaller IC area (see Table 4.2 later). On the basis of this architecture, Figs. 4.4(b)-(g) depict the schematic of six basic PCSL cells (all with 3-transistor limit in any stack, this is to mitigate the effect of I_{on}/I_{off} degradation, see Chapter 2 earlier).

Fig. 4.4: (a) Proposed Pre-Charged Static-Logic (PCSL) architecture, and six basic cells embodying the proposed PCSL dual-rail QDI logic style: (b) 2-input AND/NAND gate, (c) 2-input OR/NOR gate, (d) 3-input AO/AOI gate, (e) 3-input OA/OAI gate, (f) 2-input XOR/XNOR gate, and (g) 2-input MUX

To depict the hardware advantage of the proposed PCSL logic style, the 2-input AND/NAND gate in Fig. 4.4(b) can be compared to the same gate realized by three reported static logic QDI styles (see Chapter 2 earlier) in Figs. 4.5(a)-(c): (a) DIMS style [65], (b) NCL with complex gates [118] (denoted NCL1), and (c) NCL with fast-reset complex gates [119] (denoted NCL2). On the basis of simulations (130nm CMOS), Table 4.2 benchmarks E_{per} , delay and IC area of the aforesaid six basic cells of the various styles. The competing cells are normalized to the PCSL cells whose actual values are shown within parentheses. The average attributes are tabulated in the last row.

Fig. 4.5: Reported dual-rail AND/NAND circuit designs: (a) Delay-Insensitive-Minterm-Synthesis (DIMS), (b) NULL-Convention-Logic (NCL) with complex gates (NCL1), and (c) NCL with fast-reset complex gates (NCL2)

Table 4.2: Energy-per-operation (E_{per}), Delay and IC Area of Dual-rail Library Cells Embodying Various Logic Styles@ V_{DD} =150mV and 130nm CMOS Process

	E_{per} (normalized to PCSL (10^{-18} J))				Delay (normalized to PCSL (ns))				IC Area (normalized to PCSL (μm^2))			
	PCSL	DIMS	NCL1	NCL2	PCSL	DIMS	NCL1	NCL2	PCSL	DIMS	NCL1	NCL2
AND/NAND	1 (185)	3.2	0.9	1.2	1 (663)	3.2	1.0	1.2	1 (21)	4.5	1.6	1.6
OR/NOR	1 (185)	3.2	0.9	1.3	1 (661)	3.2	1.0	1.2	1 (21)	4.2	1.5	1.5
AO/AOI	1 (229)	6.8	2.0	2.2	1 (696)	7.0	2.2	2.3	1 (26)	6.8	3.1	3.3
OA/OAI	1 (229)	6.9	2.0	2.2	1 (696)	7.0	2.2	2.3	1 (26)	6.8	3.1	3.3
XOR/XNOR	1 (330)	1.8	2.0	2.1	1 (997)	2.1	2.3	2.3	1 (34)	2.9	3.1	3.3
MUX	1 (326)	1.8	2.0	2.1	1 (1021)	2.1	2.3	2.3	1 (32)	3.1	3.2	3.4
Average	1 (247)	4.0	1.6	1.9	1 (789)	4.1	1.8	1.9	1 (27)	4.7	2.6	2.7

It is apparent from Table 4.2 that the cells embodying the proposed PCSL logic style feature the lowest E_{per} , save the simple AND/NAND and OR/NOR gates of NCL1. On average, E_{per} of cells embodying the reported DIMS, NCL1, and NCL2 logic styles is significantly higher: $4.0\times$, $1.6\times$, and $1.9\times$ respectively. It is also apparent that the cells embodying the proposed PCSL logic style feature the shortest delay (the sum of two components, t_{LH} (computation phase) and t_{HL} (reset phase), averaged over all input combinations), save the simple AND/NAND and OR/NOR gates of NCL1. On average, the reported DIMS, NCL1, and NCL2 cells are significantly slower: $4.1\times$, $1.8\times$, and $1.9\times$ respectively. It is also apparent that the cells embodying the proposed PCSL logic style require the smallest IC area; the layouts are based on the standard-cell approach where the cell height is fixed at $4\mu\text{m}$ and the cell width is in multiples of $0.4\mu\text{m}$. On average, the IC area required for cells embodying the reported DIMS, NCL1, and NCL2 logic styles is significantly larger: $4.7\times$, $2.6\times$, and $2.7\times$ respectively; from a perspective of dual-rail async and (single-rail) sync circuits, the smaller IC area is worthwhile because the IC area overhead of the former is somewhat mitigated. In short, cells embodying the proposed PCSL logic style simultaneously exhibit the lowest E_{per} , shortest delay and smallest IC area.

With the proposed PCSL QDI logic style, an 8×8 -Bit Quad-Channel Async QDI FRM FB is designed. A semi-custom design flow is adopted, where the front-end is designed using an assortment of in-house design tools and commercial synthesis tools based on a flow similar to NCL-X [118]. The back-end implementation, on the other hand, is based on commercial EDA tools with our customized library cells (including the proposed PCSL). Each FB channel is independent and Fig. 4.6 depicts the block diagram of one FB channel embodying an FIR filter realizing the FRM algorithm. As the throughput requirement of the intended WSN is somewhat modest, a serial implementation is adopted, where each FB

channel comprises an Async Read/Write Controller, an 8×8-Bit Coefficient Memory, an 8×8-Bit Data Memory, an 8-Bit PCSL Multiplier, and a 20-Bit PCSL Adder. To preserve the QDI protocol and proper async handshaking, Datapath Completion Detection (DCD) and Latch Completion Detection (LCD) circuits are included with Muller C-elements (denoted by a gate symbol with ‘C’) [118]. All async dual-rail latches in the datapath are initialized to an ‘empty’ value except for Latch 3 which is used to hold the accumulated product and is initialized to a valid ‘0’.

The *Input_FB* data and *Req_FB* clock from the Async FIFO Buffer (Fig. 4.2) are input to each FB channel. The Async Read/Write Controller in Fig. 4.6 first initiates a write operation by providing a valid memory address on *Data_Addr* and asserting *Write_Req* to write the *Input_FB* data into the 8×8-Bit Data Memory. Upon write completion, the Async Read/Write Controller subsequently initiates the first read operation for the Multiply-Accumulate (MAC) operation from both the 8×8-Bit Data Memory and the 8×8-Bit Coefficient Memory by providing them with valid memory addresses on *Data_Addr* and *Coeff_Addr*, and then asserting *Read_Req*. The input data and its corresponding coefficient are respectively read out to Latch 1 and Latch 2, and subsequently multiplied by the 8-Bit PCSL Multiplier. The multiplication product is captured by Latch 4 and sign-extended to 20 bits to accommodate potential overflow. The 20-Bit PCSL Adder is used to add this product to the accumulated product stored in Latch 3. The result of the adder is looped back to Latch 3, thereby updating its value and completing the first MAC operation. The MAC operation repeats until the last tap of the filter. When *Output* (one of *Output1-4* in Fig. 4.2) is finally computed, the Async Read/Write Controller of each channel will assert its *Ack* clock to indicate completion. The overall *Ack* clock is output to the Async FIFO Buffer which subsequently resets *Input_FB*

Fig. 4.6: Block diagram of one channel of the 8x8-Bit Quad-Channel Async QDI FRM FB

and de-asserts the *Req_FB* clock. This in turn resets all FB channels and the system is now ready to process the next input data from the FIFO.

4.2.3 Results and Benchmarking

We will first demonstrate the robustness of the proposed async FB to PVT variations, particularly large V_{DD} and temperature variations, on the basis of physical measurements on prototype ICs (@130nm CMOS) embodying the SSAVS system and the FB, and where pertinent, by simulations. Fig. 4.7(a) depicts the die microphotograph (left) and its layout (right). The async FB embodying 4 channels occupies an IC area of $\sim 0.18 \text{ mm}^2$. All 30 prototype ICs tested were fully functional for $V_{DD} \geq 130\text{mV}$ ($|V_t| \approx 400\text{mV}$), and this in some sense corroborates the robustness of the design. The functionality was verified by sampling the input data (generated from a pattern generator) and comparing the ensuing output data (by means of a logic analyzer) with that expected. We will thereafter delineate the efficacy of the SSAVS system embodying the async FB and benchmark it against the competing conventional DVFS system embodying a sync filter. The die microphotograph of DVFS system embodying one sync FB channel is depicted in the left of Fig. 4.7(b) and on the right, the layout; the 4-channel sync FB would occupy $\sim 0.10 \text{ mm}^2$, or $\sim 1.8\times$ smaller than the async FB. The lowest functional V_{DD} of the sync filter (probably attributed to the hold time violations of registers therein [120]) is $\geq 200\text{mV}$, a minimum voltage higher than that of the async FB (130mV).

Fig. 4.7: Die microphotograph (left) and layout (right) of the fabricated test-chips: (a) proposed SSAVS system with async QDI FRM filter bank, and (b) sync benchmark filter

Consider first the robustness of the proposed async FB against PVT variations, in this case V_{DD} varying at 1kHz between 150mV and 300mV as shown in the top trace of Fig. 4.8. Under this ‘harsh’ V_{DD} condition, the async FB, operates without error as verified by the *Ack* signal (and by means of a logic analyzer), depicted as the bottom trace in Fig. 4.8. It can be appreciated that as V_{DD} can be varied widely without error and since the FB operation is uninterrupted, the async FB readily lends itself to being self-adjusted using the SSAVS system to the lowest voltage possible that meets the throughput for the prevailing conditions.

Consider now two examples of the SSAVS system that demonstrate its in-situ self-adjusting V_{DD} . In the first example, the operation of the SSAVS system earlier delineated in Fig. 4.3 is now physically depicted in Fig. 4.9(a) with the top and bottom traces being V_{DD_ADJ} and *Ack* respectively. Fig. 4.9(b) depicts the second example where in addition to V_{DD_ADJ} self-adjusting to the throughput rate, it also self-adjusts to the prevailing conditions. In the top trace of Fig. 4.9(b), the prototype IC is subjected to a sudden temperature drop (by means of freezer spray onto the package thereof) at some juncture, and V_{DD_ADJ} self-adjusts by first increasing to between 200mV and 250mV, and thereafter to between 250mV and 300mV as the cold permeates the IC package. Although not shown here, the converse is obtained when the prototype IC is subjected to heat, e.g. from a hot air gun – V_{DD_ADJ} reduces and finally toggles between two lower voltage levels.

Fig. 4.8: (a) High V_{DD} variations @ 1kHz, 150mV-300mV, and (b) error-free response (*Ack* signal) from the proposed async QDI FRM filter bank

Fig. 4.9: Example of the captured waveforms depicting (a) self-adjustment of V_{DD_ADJ} and Ack from the async QDI FRM filter bank, and (b) self-adjustment of V_{DD_ADJ} and Ack under sudden temperature drop

We will now benchmark the proposed SSAVS system with the async FB against its sync DVFS FB counterpart. In the latter, to accommodate the extreme/intractable delay variations due to PVT (including temperature ranging from -55°C to 125°C [53], congruent with the WSN application) while operating in the sub- V_t region, a substantial amount of delay safety margin is needed to obtain operational robustness. To ascertain these margins, we employ statistical delay analysis on the critical path of the sync filter. In view of the intended WSN application and the availability of test equipment (particularly the environmental chamber), four temperature corners (extreme heat 125°C , nominal 25°C , and extreme cold -40°C (and -55°C)) are considered. To ascertain the spread of delay due to process variations, 1000 Monte Carlo simulations on the critical path delay of the sync filter are performed at each said temperature corner. The worst-case delay at 3σ of the given process parameters is chosen, in part, to obtain sufficient (99.7%) coverage. The same simulations are repeated across the intended V_{DD} in the sub- V_t voltage range. These ascertained delays are depicted in Fig. 4.10 for nominal process parameters (solid lines) and for that with 3σ process variations (dotted lines). Consistent with observations reported elsewhere [6], the 3σ delay variations are expectedly higher at lower temperatures, a consequence of steeper sub-threshold slope.

Fig. 4.10: Variation of the sync filter critical path delay under various PVT conditions: Monte Carlo simulations

Consider the benchmarking under two general scenarios. In Scenario 1, the sync DVFS system embodies a temperature sensor and on the basis of the measured temperature and pre-characterization of the sync filter, the clocking frequency is selected accordingly. In Scenario 2, the sync DVFS system is much simpler where the clocking frequency is fixed (to the worst-case) to accommodate all conditions. For Scenario 1, we will use a (delay) point along the 3σ plot of the pertinent temperature and adjust that point for 10% V_{DD} variation; the 10% V_{DD} variation is congruous with the International Technology Roadmap for Semiconductors. For example, for 25°C, the delay for $V_{DD}=300\text{mV}$ is that for $V_{DD}=270\text{mV}$ @25°C and 3σ , and equals to $3.9\times$ (of the nominal). For Scenario 2, the delay for $V_{DD}=300\text{mV}$ is that for the worst-case for $V_{DD}=270\text{mV}$ @-55°C and 3σ , and equals to $183\times$; in [14], the allowed delay safety margin was somewhat similar, $\sim 200\times$.

In both scenarios, the characteristics of prototype ICs (embodying both FBs) were measured at three temperature corners, i.e. 125°C for extreme heat, 25°C for nominal, and -40°C for extreme cold (limit of the environmental chamber), and plotted in Figs. 4.11-4.14. For completeness, the delays @Upper/Lower 3σ and 10% V_{DD} obtained by simulations for the async FB are also plotted.

Figs. 4.11(a)-(c) depict the delay (for computing one sample, equivalent 14 clock cycles) and E_{per} at the three aforesaid temperature corners; as we are only able to measure at -40°C (instead of -55°C), the remarks henceforth for the extreme cold temperature is for operation at -40°C. Note that E_{per} is ascertained at each V_{DD} over the delay of computing one sample. On the basis of the delay plots, we remark the following. First, in general and as expected, the delay increases with reducing V_{DD} for both FBs. Second, also in general and for both FBs, the delay increases for decreasing temperature. Third, with the temperature ascertained by the

sensor, the delay variations, hence the ensuing delay safety margins of the sync FB, are relatively small (vis-à-vis Scenario 2, see later). Consequently and not unexpectedly, the delay of the sync FB for 25°C and 125°C is largely comparable to its async counterpart at its nominal condition. Fourth, the delay of the sync FB is longer @-40°C – on average, 4.0× longer than the async FB. This can be attributed to the longer delay at 3 σ for -40°C compared to that at 125°C.

On the basis of the E_{per} plots, we remark the following. First, in general and as expected, the minimum E_{per} for both FBs decreases as the temperature decreases. Second, V_{DD} for minimum E_{per} reduces for reducing temperature for both FBs. Specifically, as the temperature drops from 125°C to -40°C, the minimum E_{per} for the async and sync FBs respectively shifts from V_{DD} equal to ~400mV to ~250mV and from ~450mV to ~300mV. Third, the sync FB, in general, is advantageous at the higher end of V_{DD} and this advantage diminishes at higher temperature. The async FB is conversely advantageous at the lower end of V_{DD} . This observation can, as before, be corroborated with Fig. 4.10.

As the interpretation of E_{per} to power dissipation is not *prima facie*, we plot in Figs. 4.12(a)-(c) the power dissipation of the FBs as a function of throughput for the three temperature corners. We make the following remarks. First, in general and as expected, the power dissipation of both FBs decreases with reducing throughput; in Fig. 4.12(c), the power dissipation continues to decrease for throughput <10kS/s albeit at a low rate. Second, the effect of throughput on power dissipation at the three corners are different. At -40°C, the power dissipation is roughly linearly related to the throughput, where as expected, it increases with higher throughput. At 25°C, the power dissipation remains roughly linearly related to the throughput (albeit at a slower rate than that at -40°C) for mid to high (>1kS/s) throughput,

and the relationship is only slight for low throughput, $<1\text{kS/s}$. At 125°C , the throughput has only a very slight effect on the power dissipation. Overall, the influence of throughput on power dissipation mitigates as the temperature rises. Third, at 125°C , the async FB dissipates lower power than the sync FB, while at -40°C , the converse is true. At 25°C , the async FB is advantageous at the low throughput range, while at the higher throughput range, the converse is true.

In the overall perspective of power dissipation in this Scenario 1, it would be prudent to be cognizant of the hardware and power dissipation costs associated with the temperature sensor. These costs apply only to the sync DVFS system, and practically, these costs would likely defeat any advantages offered by the sync DVFS system over the async SSAVS system.

Consider now Scenario 2 where the aforesaid temperature sensor is absent. Figs. 4.13(a)-(c) benchmark the delay and E_{per} for both FBs for the three temperature corners. The delay of the sync DVFS FB is preadjusted and fixed to satisfy the worst-case condition, i.e. 3σ delay with 10% V_{DD} variation at -55°C for the given operating V_{DD} voltage. It is hence not unexpected that the delay of the sync FB is substantially larger than its async counterpart (at nominal condition) for all three temperature corners. This disparity becomes most apparent when the conditions are most benign, at 125°C when the FBs can operate at a higher speed. In short, in Scenario 2, the async FB is advantageous in terms of delay to the sync FB for all conditions.

Consider now the E_{per} of the FBs. At -40°C , the E_{per} of the sync FB is lower than the async FB for $V_{\text{DD}} > 300\text{mV}$, and the converse is true for $V_{\text{DD}} < 300\text{mV}$. As the temperature increases, the E_{per} of the sync FB as expected increases significantly. Specifically, at 25°C ,

the sync FB dissipates higher E_{per} than its async counterpart for $V_{\text{DD}} < 400\text{mV}$. Further, at 125°C , the E_{per} of the sync FB is significantly higher than the async FB over the entire sub- V_t V_{DD} range. In short, in Scenario 2, the async FB is advantageous in terms of E_{per} to the sync FB at 125°C , advantageous for sub- V_t $V_{\text{DD}} < 400\text{mV}$ at 25°C , and at -40°C , only for sub- V_t $V_{\text{DD}} < 300\text{mV}$.

Figs. 4.14(a)-(c) depict the power dissipation of the FBs as a function of throughput for the same three temperature corners. At -40°C , the sync FB dissipates less power in most of the throughput range. At 25°C , the sync FB dissipates power comparable to its async FB counterpart in the high throughput range $> 10\text{kS/s}$, and higher power in the mid to low throughput range, $< \sim 10\text{kS/s}$. At 125°C , the sync FB dissipates substantially higher power than its async counterpart over the entire throughput range. In short, compared to the power dissipation of the sync FB, the async FB is disadvantageous at -40°C , comparable in the high throughput range at 25°C , and advantageous elsewhere.

The aforesaid remarks and observations pertaining to Scenarios 1 and 2 can largely be explained by noting that in sub- V_t , the delay of the circuits increases with decreasing temperature (vis-à-vis increasing temperature in supra- V_t), that the delay at 3σ increases the most at the extreme cold temperature (vis-à-vis at other temperatures), that at very low V_{DD} the leakage current is dominant (over dynamic), that the leakage current is exponentially related to temperature, and that because the FB is a relatively simple circuit, the delay of the critical path of the sync FB is only slightly longer than its non-critical paths (explaining the relatively low delay of the sync FB, particularly in Scenario 1).

Overall, this benchmarking depicts that in Scenario 1, no specific FB is particularly advantageous – the sync DVFS FB and async SSAVS FB are advantageous in different conditions. Nevertheless, the sync FB may be disadvantageous if the temperature sensor overheads associated with DVFS for Scenario 1 are considered. In Scenario 2, the async FB is advantageous in terms of reduced delay with respect to V_{DD} , usually lower E_{per} with respect to V_{DD} , and in terms of power dissipation, advantageous in some conditions (while the sync advantageous in other conditions). Further, in the context of continuous circuit operation and overheads associated with DVS, the proposed SSAVS is advantageous over the conventional DVFS in terms of uninterrupted circuit operation and not requiring external intervention (such as changing clock rate, pre-characterization, etc.).

Fig. 4.11: Scenario 1: Benchmarking delay and E_{per} of a sync DVFS filter bank and the async SSAVS filter bank for three temperature corners: (a) -40°C, (b) 25°C, and (c) 125°C. Note: Bold lines are measured while dotted lines are from simulations

Fig. 4.12: Scenario 1: Power consumption of the sync and async filter banks (a) @-40°C, (b) @25°C, and (c) @125°C

Fig. 4.13: Scenario 2: Benchmarking delay and E_{per} of a sync DVFS filter bank and the async SSAVS filter bank for three temperature corners: (a) -40°C, (b) 25°C, and (c) 125°C. Note: Bold lines are measured while dotted lines are from simulations.

Fig. 4.14: Scenario 2: Power consumption of the sync and async filter banks (a) @-40°C, (b) @25°C, and (c) @125°C

In summary, we have proposed an SSAVS system for a WSN with the objective of lowest possible power operation for the prevailing throughput and circuit conditions – V_{DD} adjusted to within 50mV of the minimum voltage, yet high operational robustness with minimal overheads. High robustness has been achieved by adopting the async QDI protocols, and the embodiment of our proposed PCSL logic style. Minimal overheads has been achieved by exploiting already existing signals in the QDI protocols. The proposed async SSAVS system has been benchmarked against its conventional sync DVFS system counterpart for two scenarios, and their merits and disadvantages delineated.

4.3 A Robust Asynchronous Approach for Realizing Ultra Low-Power Digital Self-Adaptive V_{DD} Scaling System

It was established in the previous section that self-adaptive V_{DD} scaling system attempts to achieve maximum power/energy efficiency by scaling V_{DD} to the lowest voltage possible for the prevailing conditions, including input data rate, temperature, etc. To realize reliable SSAVS, including accommodating the severe PVT variations thereof, we adopted the async QDI. Nevertheless, the costs – power/energy overheads associated with (conventional) async QDI are high, and this in part explains the conclusion that in terms of delay, E_{per} and power, neither sync nor (conventional) async QDI is particularly advantageous⁵.

In this section, we propose an alternative QDI approach, coined ‘Pseudo-QDI’, for SSAVS with the objective of reduced power/energy overheads compared to the standardized QDI, yet retaining robustness. The proposed approach comprises a simplified async 4-phase

⁵ However, on the basis of operational robustness, we argue that the async QDI would nevertheless be advantageous. Further, the FRM filter bank is a relatively small system where the sync version would only involve a commensurably small clock infrastructure. In other words, if the Signal Processor in Fig. 4.1 is a large system, e.g. a 32-bit processor, there is a good possibility that the async QDI version would be advantageous due to the absence of the complex clocking infrastructure required of the sync; see Chapter 5 later for our proposed future work.

pipeline structure (see Fig. 4.15(b) later) with our proposed PCSL dual-rail logic cell delineated earlier. The salient difference between the proposed pseudo-QDI pipeline and a standardized QDI pipeline (henceforth termed 'True-QDI') is the removal of the Datapath Completion Detection (DCD) while preserving the Latch Completion Detection (LCD) (see Section 4.3.1 later). This simplified technique places an additional timing requirement on the reset cycle of the 4-phase async operation – specifically that certain internal nodes must reset before the next cycle of evaluation commences, in part facilitated by the fast-reset nature of our proposed PCSL cells. We show that this timing requirement can be easily satisfied, thereby ensuring robust operation even under severe PVT variations in sub- V_t region (see Section 4.3.2 later).

On the basis of the true-QDI and our proposed pseudo-QDI approaches, we design and monolithically realize two async quad-channel FRM filter banks (@130nm CMOS). The true-QDI filter bank was the same embodied in the SSAVS system delineated in the previous section. On the basis of measurements on prototype ICs, our proposed async pseudo-QDI filter bank features ~40% lower energy and $\sim 1.34\times$ smaller IC area as compared to its true-QDI counterpart (see Section 4.3.3 later), yet it demonstrates extreme robustness against large sub- V_t PVT variations.

4.3.1 Proposed Async Pseudo-QDI Realization Approach

Consider first the design of a true-QDI pipeline embodying our proposed PCSL cells that provides for sub- V_t operation. To preserve its delay-insensitivity attribute (save the fundamental isochronic fork assumption [16]), the QDI pipeline needs to address the issues of ‘input completeness’ [121] (where all inputs need to be acknowledged before a new pipeline operation commences) and ‘gate orphan’ [121] (where an internal gate is enabled to switch its output but the switching is masked from the observable outputs of the entire circuit). To address these two issues, either the NCL-X pipeline structure [118] or the NCL-D pipeline structure [81] may be used. We adopt the former because it occupies a much smaller area due to its relatively simple realization [118] of datapaths where a functional circuit can first be synthesized (using a (single-rail) standard synthesis tool), followed by a single-rail to dual-rail conversion.

Fig. 4.15(a) depicts the adopted async true-QDI pipeline stage (i^{th} stage) comprising a QDI Handshake $_i$ (consisting of a Latch Controller $_i$, a Latches $_i$ and a Latch Completion Detection (LCD $_i$)) and an async QDI Datapath $_i$; a QDI Handshake $_{i+1}$ is also shown for ease of illustration. The QDI Handshake $_i$ controls the async QDI Datapath $_i$ according to a sequence of pre-defined handshake signals. Initially, $ACK_{i+1} = 0$ and $REQ_i = 1$, indicating that (dual-rail) Latches $_i$ are transparent and are waiting for valid Data $_i$. When Data $_i$ is all valid, LCD $_i$ will check the data and acknowledge Latches $_{i-1}$ (not shown) of the preceding pipeline. $ACK_i = 1$ also acknowledges Latch Controller $_i$ to ensure the input completeness of the pipeline. The valid Data $_i$ will trigger QDI Datapath $_i$ for computation. Once the output (Data $_{i+1}$) is valid and is stored in Latches $_{i+1}$ (if $REQ_{i+1} = 1$), LCD $_{i+1}$ will acknowledge Latch Controller $_i$. To address the gate orphan issues (if any), all outputs of the dual-rail PCSL circuits in the intermediate columns have to be checked by a Datapath Completion Detection (DCD $_i$ in Fig.

4.15(a)) before the intermediate detection signal, AVE_i , can be asserted. Latch Controller_{*i*} will thereafter de-assert REQ_i to reset the PCSL circuits, and both AVE_i and ACK_{i+1} will likewise reset to '0'. Once Data_{*i*} becomes empty, ACK_i is de-asserted to '0', and LCD_{*i*} will revert REQ_i back to its initial condition ($REQ_i = '1'$), awaiting Data_{*i*} to be valid again. This async pipeline (with DCD_{*i*}) fully satisfies the QDI protocol, hence 'true-QDI' as described earlier.

(a)

(b)

Fig. 4.15: (a) The conventional async true-QDI pipeline, and (b) our proposed async pseudo-QDI pipeline embodying the PCSL cells

It is well-established [118] that the area and energy overheads of DCD_i are large especially if the complexity of the functional circuits in $QDI\text{ Datapath}_i$ is high. Nevertheless, the delay overhead of DCD_i is largely insignificant as DCD_i executes in parallel with the functional circuits (and with $QDI\text{ Handshake}_{i+1}$).

To alleviate the area and energy overheads of DCD_i , DCD_i may be removed in the pipeline. We denote this async modality as ‘pseudo-QDI’ where an implicit timing condition is required to satisfy the QDI signal protocol. As the REQ signal is already integrated into our PCSL circuits, they immediately lead themselves to the pseudo-QDI pipeline depicted in Fig. 4.15(b). The pseudo-QDI pipeline operates exactly as its true-QDI counterpart except that the Latch Controller $_i$ no longer waits for the assertion and de-assertion of AVE_i as in the true-QDI pipeline. Note that as long as an implicit timing condition is abided by (see Section 4.3.2 below), the robustness of the pseudo-QDI pipeline is not compromised.

4.3.2 Timing Analysis on the Proposed Pseudo-QDI Realization Approach

Consider now the delay properties in the pseudo-QDI pipeline depicted in Fig. 4.15(b) by considering two scenarios:

- (a) $QDI\text{ Datapath}_i$ embodying only one level (column) of PCSL circuits – a fine-grain gate-level pipeline where every circuit is pipelined, and
- (b) $QDI\text{ Datapath}_i$ embodying multiple levels (columns) of PCSL circuits – a coarse-grain block-level pipeline where many circuits are collectively grouped to form a pipeline.

For brevity in the analysis, $t_{\uparrow\text{cycle}}$ is denoted as the forward cycle time ($REQ_{i+} \rightarrow REQ_{i-}$ for valid $Data_i$ sent to $Pipeline_i$ until $Latches_i$ is closed), and $t_{\downarrow\text{cycle}}$ the reset cycle time ($REQ_{i-} \rightarrow REQ_{i+}$ for empty $Data_i$ sent to $Pipeline_i$ until $Latches_i$ is re-opened for the next operation). The cycle delay $t_{\text{cycle}} = t_{\uparrow\text{cycle}} + t_{\downarrow\text{cycle}}$ is an indication of the speed of the async pipeline.

For scenario (a), the inputs of the PCSL circuits are checked and acknowledged by LCD_i , and their outputs are subsequently checked and acknowledged by LCD_{i+1} (of the next pipeline stage). In this scenario, the QDI property is preserved, and the pipeline operation is robust.

For scenario (b), an implicit delay assumption arises for the $t_{\downarrow\text{cycle}}$ path when $REQ_{i-} \rightarrow REQ_{i+}$; there is no delay assumption for the $t_{\uparrow\text{cycle}}$ path. This implicit delay assumption arises because LCD_{i+1} can only check the primary outputs of QDI Datapath_{*i*} at the last column, but not the intermediate output signals of the PCSL circuits (at the intermediate columns) where a ‘gate orphan’ may exist. We formulate the necessary implicit timing condition in eqn. (4.1) for error-free operation.

$$\begin{aligned} \max(t_{\downarrow(\text{PCSL}_{\text{col} \neq \text{last}})}) &< t_{\downarrow\text{cycle}} \\ &< \max[(t_{\downarrow\text{LCD}_i} + t_{\downarrow\text{Latches}_i}), (t_{\downarrow(\text{PCSL}_{\text{col}=\text{last}})} + t_{\downarrow\text{Latches}_{i+1}} + t_{\downarrow\text{LCD}_{i+1}})] + t_{\downarrow\text{LC}_i} \end{aligned} \quad (4.1)$$

where $t_{\downarrow(\text{PCSL}_{\text{col} \neq \text{last}})}$ is the reset delay for the PCSL circuits at the intermediate columns,
 $t_{\downarrow(\text{PCSL}_{\text{col}=\text{last}})}$ is the reset delay for the PCSL circuits at the last column,
 $t_{\downarrow\text{Latches}_i}$ is the reset delay for $Latches_i$,
 $t_{\downarrow\text{Latches}_{i+1}}$ is the reset delay for $Latches_{i+1}$,
 $t_{\downarrow\text{LCD}_i}$ is the reset delays for LCD_i ,
 $t_{\downarrow\text{LCD}_{i+1}}$ is the reset delays for LCD_{i+1} , and
 $t_{\downarrow\text{LC}_i}$ is the reset delay for Latch Controller_{*i*}.

From the viewpoint of the pipeline schematic, ideally $t_{\downarrow(\text{PCSL}_{col \neq last})} \cong t_{\downarrow(\text{PCSL}_{col=last})}$ when REQ_i switches from ‘1’ to ‘0’ for the reset phase, where all PCSL circuits are simultaneously reset. In general, this implicit timing assumption is easily satisfied – specifically, as long as the ratio of $t_{\downarrow cycle} / t_{\downarrow(\text{PCSL}_{col \neq last})} > 1$ under all possible PVT variations, the pseudo-QDI pipeline remains robust.

4.3.3 Benchmarking Results

We demonstrate the aforesaid by means of a true-QDI and a pseudo-QDI async quad-channel FRM filter banks. On the basis of measurements from the prototype ICs (@130nm CMOS as depicted in Fig. 4.16(a)), both filter banks were fully functional for $V_{DD} > 130\text{mV}$.

Further, as shown in Fig. 4.16(b) both filter banks were also fully functional for extreme V_{DD} variations, and fully functional for wide temperature variations (not shown) – thereby depicting their robustness under severe sub- V_t PVT variations.

Fig. 4.16: (a) Die microphotograph and layout of the fabricated true-QDI and pseudo-QDI filter banks (@130nm CMOS), and (b) Robust sub- V_t operation of the fabricated pseudo-QDI filter bank under large V_{DD} variations

Fig. 4.17: Measured energy/operation (E_{per}) of the async filter banks

Fig. 4.17 benchmarks the measured E_{per} of the two async filter banks in sub- V_t , depicting the $\sim 40\%$ lower E_{per} advantage of the proposed pseudo-QDI filter bank over its true-QDI counterpart. The proposed pseudo-QDI filter bank further features $\sim 1.34\times$ smaller IC area advantage over its true-QDI counterpart.

In summary, we have described our proposed alternative QDI – the pseudo-QDI approach – for simultaneous lower E_{per} and smaller IC area than the standardized true-QDI, yet robust in sub- V_t (appropriate for SSAVS) and under extreme PVT variations.

4.4 Conclusions

In this chapter, we have proposed a Sub- V_t Self-Adaptive V_{DD} Scaling (SSAVS) system for a varying-workload WSN with the objective of lowest possible power dissipation for the high variation-space and wide operation-space applications, yet high robustness and with minimal overheads. The effort to achieve the lowest possible power operation has been realized by means of an automatic DVS – self-adjusting V_{DD} to the minimum voltage (within

50mV) for the prevailing conditions. High robustness has been achieved by adopting the QDI protocol, and by the embodiment of our proposed PCSL design style; when compared against competing async logic styles that feature robustness in sub- V_t operation, the PCSL has been shown to be most competitive in terms of E_{per} , delay and IC area. By exploiting the already existing request and acknowledge signals of the QDI protocols, the ensuing overhead of the SSAVS is very modest – a simple counter and a FIFO buffer. The filter bank embodied in the SSAVS has been shown to be ultra low-power and highly robust. The proposed async DVS SSAVS has been benchmarked against its conventional sync DVFS counterpart. We have shown that no one system is particularly advantageous when the operating conditions are known. Further when the sync DVFS system is designed for the worst-case condition, the proposed async DVS SSAVS is somewhat more competitive. To improve the competitiveness of async QDI in terms of hardware and power, we have further proposed a hardware-simplified version of QDI (herein coined ‘pseudo-QDI’) with an implicit timing for the said SSAVS. We have shown analytically that said implicit timing is easily satisfied whilst ensuring robust operation, said robustness has also been verified by measurement on prototype ICs embodying the pseudo-QDI under very high variation-space and wide operation-space conditions. By means of the pseudo-QDI, the ensuing energy and area have been significantly reduced by $\sim 40\%$ and $\sim 1.34\times$ respectively compared to the standardized QDI.

Chapter 5 Conclusions and Recommendations for Future Work

5.1 Conclusions

We have delineated in this thesis research work pertaining to the design of low-power/ultra low-power *high variation-space* and *wide operation-space* digital electronics for portable/mobile applications. High variation-space and wide operation-space respectively refer to error-free operation despite high variations in the prevailing conditions (including PVT variations) and under a wide range of activity levels or workload. In view of said spaces, we have adopted the async MD and QDI protocols vis-à-vis the conventional sync protocol. The specific conclusions arising from investigations presented in this thesis can be divided into two parts, and will now be described in turn.

The first part pertained to the investigation (and design thereof) into the efficacy of the application of the async protocols for realizing *low-power/ultra low-power* digital circuits/system. The specific conclusions are:

- (a) We have proposed a fine-grain power gating methodology to reduce the short-circuit and leakage wasted powers of an async MD pipeline (applicable to three different gating configurations) over a wide operation-space. By exploiting the 4-phase handshake protocol, the ensuing overhead of the proposed power gating was shown to be low, specifically one inverter (per pipeline stage) and <15% delay;
- (b) To quickly estimate to the first-order the delay variations (due to V_t , V_{DD} and temperature variations; thus the required delay safety margin) of digital circuits in

sub- V_t , we have proposed and derived a set of simple yet insightful analytical equations. The derived equations have been verified by simulations and shown to be accurate for first-order estimations (with an inconsequential worst-case error of $<12\%$);

- (c) Following (b), the benchmarking of the sync (with delay safety margins estimated by the derived equations) against the async QDI (with self-completion detection), on the basis of adder circuits, has shown that neither the sync nor the async QDI is particularly advantageous in all conditions. This exercise depicted the usefulness of the derived equations, particularly the insights provided by the simple derived equations, and delay variations are easily estimated from the nominal case.

The second part pertained to the design and realization of an adaptive DVS circuits/system for a WSN (operating in sub- V_t) based on the async QDI protocol and its benchmarking against the sync DVFS. The general intention herein is a WSN that operates at its minimal V_{DD} (within 50mV), yet robust operation. The specific conclusions are:

- (d) We have proposed a Sub- V_t Self-Adaptive V_{DD} Scaling (SSAVS) system for a high variation-space and wide operation-space Wireless Sensor Network (WSN) with the objective of lowest possible power dissipation in sub- V_t operation, yet high robustness and with minimal overheads. The effort to achieve the lowest possible power operation has been realized by means of adjusting V_{DD} to the minimum voltage (within 50mV) for any given prevailing conditions. High robustness has been achieved in part by adopting the QDI protocol;

- (e) Further to (d), the high robustness thereof has been also in part achieved by the embodiment of our proposed PCSL logic style. The proposed PCSL logic style is a worthy logic style because when compared against competing async logic styles appropriate (in terms of robust error-free operation) for sub- V_t , the PCSL has been shown to be most competitive in terms of E_{per} , delay and IC area;
- (f) The filter bank (comprising PCSL cells) embodied in the SSAVS has been shown to be ultra low-power and highly robust. When the proposed async SSAVS was benchmarked against its conventional sync DVFS counterpart for two scenarios, we have shown that no one system is particularly advantageous when the operating conditions are known. However, when the sync DVFS system is designed for the worst-case condition, the proposed async DVS SSAVS was shown to be somewhat more competitive;
- (g) In conjunction with (f), to reduce the overheads of the QDI protocol in realizing SSAVS in wide operation-space, we have proposed to exploit the already existing request and acknowledge signals of the QDI protocol, and the ensuing overhead of the SSAVS is very modest. This proposal is interesting not just because of said exploitation but also because it does not require a priori information on the width of the operation-space or any other parameter. Conversely, the DVFS sync requires both a priori information and the other prevailing conditions unless it is designed to already accommodate the worst-case conditions;
- (h) Further to (d) to (g), to yet further reduce the overheads (in terms of power/energy and area) of async QDI, we have proposed a hardware-simplified version of QDI,

coined ‘pseudo-QDI’ herein, with an implicit timing for the aforesaid SSAVS. We have analytically depicted that said implicit timing is easily satisfied whilst ensuring robust operation (hence applicable for the proposed SSAVS), and said robustness has also been verified by measurement on prototype ICs embodying the pseudo-QDI under very high variation-space conditions. By means of the pseudo-QDI, the ensuing energy and area have been shown to be significantly reduced by $\sim 40\%$ and $\sim 1.34\times$ respectively compared to the standardized QDI.

Overall, the conclusions are that the work in this thesis has been significant to the digital design community as it provides insights to the designers on the mechanisms for low-power/ultra low-power yet robust error-free operation in high variation-space and wide operation-space applications, and a means of selecting the most appropriate design approaches/techniques for said applications.

5.2 Recommendations for Future Work

Further to the research work presented in this thesis, we will now describe some recommendations for future work.

- (i) In Chapter 3, we described our proposed techniques to power gate async MD circuits. It is interesting and perhaps surprising that hitherto reported work on async (MD and QDI) remains somewhat paltry [84]-[87]. Our literature review has discovered that one reported power gating for async QDI is for the NCL [84], where the gating transistors are embedded in every logic gate (a gate-level approach as opposed to our proposed power gating where gating transistors are

inserted at every pipeline stage). This reported approach is likely to involve higher overhead in terms of delay, energy and IC area than our pipeline-level approach with the PCSL (note that the NCL logic style without power gating is already shown to be less competitive than our proposed PCSL, see benchmarking in Chapter 4). To this end, our first recommendation pertains to reducing the wasted power of the async QDI (embodying our proposed PCSL) by applying the fine-grain power gating technique, including benchmarking the two aforesaid techniques. In this recommended future work, the application of the proposed power gating to async QDI is expected to be similar to that described for the async MD as they both adopt the 4-phase handshake protocol. It would thereafter be interesting to benchmark the efficacy of power gating for the async QDI against our proposed SSAVS for high variation-space and wide operation-space applications;

- (ii) In Chapter 3, we derived a set of simple yet insightful equations for estimating to the first-order delay variations (due to V_t , V_{DD} and temperature variations) of digital circuits operating in sub- V_t . The derived equations were shown to be accurate for first-order estimations. Nevertheless, the accuracy of the said equations may be further improved by adding heuristics, which may thereafter be employed for calculating delay safety margins in real-time. To this end, our second recommendation pertains to improving the accuracy of said equations, in particular, by adding heuristics to the equations on V_t and V_{DD} variations (eqns. (3.4) and (3.6)) to account for the effects of different V_{DD} ; see Figs. 3.7 and 3.8 earlier.

- (iii) Further to (ii), with said improved accuracy, we further recommend employing these equations in a sync DVFS system to estimate the required delay safety margins in real-time based on readings from embedded PVT sensors (see scenario 1 of the sync in Chapter 4), and adjust the clock rate accordingly. This recommended approach may replace the current LUT (Look Up Table) approach for sub- V_t , where the sync needs to be pre-characterized under all variation-space of PVT. The likely positive outcome may be substantial – simplified pre-characterization, smaller overheads (than LUT), and possibly self-tuning/correction.
- (iv) In Chapter 4, the FRM filter bank embodied in the WSN is a relatively small system where the sync version would only involve a commensurably small clock infrastructure. In other words, if the Signal Processor in Fig. 4.1 is a larger system, e.g. a 32-bit processor, there is a good possibility that the async QDI version would be advantageous due to its absence of the complex clocking infrastructure required of the sync. To this end, our final recommendation pertains to the realization of the proposed async SSAVS embodying a larger circuit/system and benchmarking against its sync DVFS counterpart.

Bibliography

- [1] K.-L. Chang, J. S. Chang, B.-H. Gwee, K.-S. Chong, “Synchronous-Logic and Asynchronous-Logic 8051 Microcontroller Cores for Realizing the Internet of Things: A Comparative Study on Dynamic Voltage Scaling and Variation Effects,” *IEEE JESTCAS*, v3, n1, pp. 23–34, Mar. 2013.
- [2] G. Chen, S. Hanson, D. Blaauw, and D. Sylvester, “Circuit design advances for wireless sensing applications,” *Proc. IEEE*, v98, n11, pp. 1808–1827, Nov. 2010.
- [3] S. Roundy, P. K. Wright and J. M. Rabaey, *Energy Scavenging for Wireless Sensor Networks with Special Focus on Vibrations*. Kluwer Academic Press, 2003.
- [4] A. Sinha and A. Chandrakasan, “Dynamic power management in wireless sensor networks,” *IEEE Design Test Comput.*, vol. 18, pp. 62–74, Mar./Apr. 2001.
- [5] International Technology Roadmap for Semiconductors 2011 [Online]. Available: <http://www.itrs.net>.
- [6] D. Bol *et al.*, “The detrimental impact of negative Celsius temperature on ultra-low-voltage CMOS logic,” in *Proc. ESSCIRC*, Sep. 2010, pp. 522–525.
- [7] J. Rabaey, *Low Power Design Essentials*. Springer Publishing Company, 2009.
- [8] V. Gutnik and A. P. Chandrakasan, “Embedded power supply for low-power DSP,” *IEEE Trans Very Large Scale Integr.(VLSI) Syst.*, vol. 5, pp. 425–435, 1997.
- [9] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits, A Design Perspective*, 2nd Ed. Prentice Hall, 2001.
- [10] A. Wang, B. H. Calhoun, and A. P. Chandrakasan, *Sub-threshold Design for Ultra Low-Power Systems*. Springer, 2006.
- [11] J. Hill, “System architecture for wireless sensor networks,” Ph.D. dissertation, University of California at Berkeley, 2003.
- [12] J. Sparsø, and S. Furber, *Principle of Asynchronous Circuit Design: A System Perspective*. Norwell, MA: Kluwer Academic, 2001.
- [13] A. J. Martin and M. Nysyrom, “Asynchronous techniques for system-on-chip designs,” *Proc. IEEE*, v96, n6, pp. 1104–1115, Jun. 2006.
- [14] R. D. Jorgenson *et al.*, “Ultralow-power operation in subthreshold regimes applying clockless logic,” *Proc. IEEE*, v98, n2, pp.299–314, Feb. 2010.
- [15] J. Kwong *et al.*, “A 65nm sub- V_t microcontroller with integrated SRAM and switched-capacitor DC-DC converter,” *IEEE JSSC*, v44, n1, pp. 115–126, Jan. 2009.

- [16] A. J. Martin, "The limitations to delay-insensitivity in asynchronous circuits," In *Proc. Sixth MIT Conf. on Advanced Research in VLSI*, 1990, pages 263–278.
- [17] S. Gary, P. Ippolito, G. Gerosa, C. Dietz, J. Eno, and H. Sanchez, "Powerpc 603TM, A Microprocessor for Portable Computers," *IEEE Design & Test of Computers*, vol. 11, no. 4, pp. 14-23, 1994.
- [18] D. N. Truong *et al.*, "A 167-processor computational platform in 65 nm CMOS," *IEEE JSSC*, v44, n4, pp. 1130–1144, Apr. 2009.
- [19] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE JSSC*, vol. 27, pp. 473-484, 1992.
- [20] M. Hempstead, D. Brooks, and G.-Y. Wei, "An accelerator-based wireless sensor network processor in 130 nm CMOS," *IEEE JESTCAS*, v1, n2, pp. 193–202, Jun. 2011.
- [21] A. P. Chandrakasan and R. W. Brodersen, *Low Power CMOS Digital Design*. Norwell, MA: Kluwer, 1996.
- [22] J. W. Tschanz *et al.*, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *IEEE JSSC*, vol. 37, no. 11, pp. 1396–1402, Nov. 2002.
- [23] J. W. Tschanz *et al.*, "Dynamic sleep transistor and body bias for active leakage power control of microprocessors," *IEEE JSSC*, vol. 38, no. 11, pp. 1838 -1845, 2003.
- [24] D. Hisamoto *et al.*, "FinFET-a self-aligned double-gate MOSFET scalable to 20 nm," *IEEE Trans. Electron Devices*, vol. 47, no. 12, pp. 2320-2325, Dec. 2000.
- [25] L. S. Nielsen *et al.*, "Low-power operation using self-timed circuits and adaptive scaling of the supply voltage," *IEEE Trans. VLSI Syst.*, v2, n4, pp. 391–397, Dec. 1994.
- [26] M. Nakai *et al.*, "Dynamic voltage and frequency management for a low power embedded microprocessor," *IEEE JSSC*, v40, n1, pp. 28–35, Jan. 2005.
- [27] D. Ma and R. Bondade, "Enabling power-efficient DVFS operations on silicon," *IEEE Circuits Syst. Mag.*, vol. 10, no. 1, pp. 14–30, Mar. 2010.
- [28] A. Raychowdhury *et al.*, "Computing with subthreshold leakage: device/circuit/architecture co-design for ultralow-power subthreshold operation," *IEEE Trans. VLSI Syst.*, v13, pp. 1213–1224, Nov. 2005.
- [29] S. Hanson *et al.*, "Exploring variability and performance in a sub-200-mV processor," *IEEE JSSC*, v43, n4, pp. 881–891, Apr. 2008.

- [30] I. J. Chang, S. P. Park, and K. Roy, "Exploring asynchronous design techniques for process-tolerant and energy-efficient subthreshold operation," *IEEE JSSC*, v45, n2, pp. 401–410, Feb. 2010.
- [31] B. Zhai *et al.*, "Theoretical and Practical Limits of Dynamic Voltage Scaling," in *IEEE DAC Digest of Technical Papers*, 2004, pp. 868-873.
- [32] B. Calhoun, A. Wang, and A. Chandrakasan, "Modeling and Sizing for Minimum Energy Operation in Subthreshold Circuits," *IEEE JSSC*, vol. 40, no. 9, pp. 1778-1786, Sept. 2005.
- [33] D. Chinnery and K. Keutzer, *Closing the Power Gap between ASIC and Custom Tools and Techniques for Low Power Design*. New York: Springer, 2007, ch. 10.
- [34] M. Keating *et al.*, *Low Power Methodology Manual For System-on-Chip Design*. Springer, 2007.
- [35] V. Kursun and E. G. Friedman, *Multi-voltage CMOS Circuit Design*. John Wiley & Sons, 2006.
- [36] V. De *et al.*, "Techniques for Leakage Power Reduction," in *Design of High-Performance Microprocessor Circuits*, A. Chandrakasan, W. Bowhill, and F. Fox, Eds. IEEE Press, 2001, ch. 3, pp. 46-62.
- [37] B. H. Calhoun, A. Wang, and A. Chandrakasan, "Device Sizing for Minimum Energy Operation in Subthreshold Circuits," in *CICC Digest of Technical Papers*, Oct. 2004, pp. 95-98.
- [38] A. Wang, and A. P. Chandrakasan, "A 180-mV Subthreshold FFT Processor Using a Minimum Energy Design Methodology," *IEEE JSSC*, vol. 40, no. 1, pp. 310-319, Jan. 2005.
- [39] B. H. Calhoun and A. P. Chandrakasan, "A 256-kb 65-nm subthreshold SRAM design for ultra-low-voltage operation," *IEEE JSSC*, vol. 42, no. 3, pp. 680–688, 2007.
- [40] B. Zhai *et al.*, "Analysis and mitigation of variability in subthreshold design," in *Proc. Int. Symp. on Low-Power Electronics and Design (ISLPED)*, 2005, pp. 20-25.
- [41] D. Bol *et al.*, "Technology flavor selection and adaptive techniques for timing constrained 45nm subthreshold circuits", in *Proc. ISLPED*, 2009, pp. 21-26.
- [42] T. Lin, K.-S. Chong, B.-H. Gwee and J. S. Chang, "Fine-grained power gating for leakage and short-circuit power reduction by using asynchronous-logic," in *Proc. IEEE ISCAS*, 2009, pp. 3162-3165.
- [43] N. Weste and D. Harris, *CMOS VLSI Design: A Circuit and System Perspective*, 4th ed. Reading, MA: Addison Wesley, 2010.

- [44] B. H. Calhoun, S. Khanna, R. Mann, and J. Wang, "Sub-threshold circuit design with shrinking CMOS devices," in *Proc. ISCAS*, 2009, pp. 2541-2544.
- [45] A. Parameswar, H. Hara, and T. Sakurai, "A swing restored pass-transistor logic-based multiply and accumulate circuit for multimedia applications," *IEEE JSSC*, vol. 31, pp. 804-809, 1996.
- [46] L. Alarcón, T.-T. Liu, M. Pierson, and J. Rabaey, "Exploring very low energy logic: A case study," *J. Low Power Electron.*, vol. 3, no. 3, pp. 223-233, Dec. 2007.
- [47] B. Zhai *et al.*, "Energy-Efficient Subthreshold Processor Design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, pp. 1127-1137, 2009.
- [48] C. Y. Kim and L. S. Kim, "Low-power and high-performance equality comparator using pseudo-NMOS NAND gates," *Electronics Letters*, vol. 40, pp. 1100-1101, 2004.
- [49] S. M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits: Analysis and Design*. McGraw-Hill, New York, 2002.
- [50] N. Verma, J. Kwong, and A. P. Chandrakasan, "Nanometer MOSFET Variation in Minimum Energy Subthreshold Circuits," *IEEE Trans. on Electron Devices*, pp. 163-174, January 2008.
- [51] S. M. Sharroush *et al.*, "Impact of technology scaling on the performance of domino CMOS logic," in *Proc. ICED*, 2008, pp. 1-7.
- [52] R. J. Baker, *CMOS Circuit Design, Layout, and Simulation*, Revised 2nd ed. Wiley-IEEE Press, 2008.
- [53] J. S. Chang *et al.*, "Digital Asynchronous-Logic: Dynamic Voltage Control," Final Technical Report for DARPA Project, HR0011-09-2-0006, Aug. 2010.
- [54] B. Zhai, S. Hanson, D. Blaauw, and D. Sylvester, "A variation-tolerant sub-200mV 6-T subthreshold SRAM," *IEEE JSSC*, vol. 43, no. 10, pp. 2338-2348, Oct 2008.
- [55] A. Tajalli, M. Alioto, and Y. Leblebici, "Improving power-delay performance of ultra-low-power subthreshold SCL circuits," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 56, no. 2, pp. 127-131, Feb. 2009.
- [56] N. Jayakumar and S. P. Khatri, "A variation-tolerant sub-threshold design approach," in *Proc. Design Automation Conf.*, 2005, pp. 716-719.
- [57] Y. K. Ramadass, and A. P. Chandrakasan, "Minimum Energy Tracking Loop With Embedded DC-DC Converter Enabling Ultra-Low-Voltage Operation Down to 250 mV in 65 nm CMOS," *IEEE JSSC*, pp. 256-265, January 2008.
- [58] W. B. Wilson, M. Un-Ku, K. R. Lakshmikumar, and D. Liang, "A CMOS self-calibrating frequency synthesizer," *IEEE JSSC*, vol. 35, no. 10, pp. 1437-1444, 2000.

- [59] S.-C. Chang, C.-T. Hsieh, and K.-C. Wu, "Re-synthesis for delay variation tolerance," in *Proc. Design Automation Conf.*, 2004, pp. 814-819.
- [60] S. Hauck, "Asynchronous design methodologies: an overview," *IEEE Proc.*, vol. 83, no. 1, pp. 69-93, 1995.
- [61] P. A. Beerel, R. O. Ozdag, and M. Ferretti, *A designer's guide to asynchronous VLSI*. Cambridge University Press, Mar. 2010.
- [62] A. J. Martin, S. M. Burns, T. K. Lee, D. Borkoivc, and P. J. Hazewindus, "The first asynchronous microprocessor: the test results," *Computer Architecture News*, vol. 17, no. 4, pp. 95-110, Jun. 1989.
- [63] T. E. Williams and M. A. Horowitz, "A zero-overhead self-timed 160-ns 54-b CMOS divider," *IEEE JSSC*, vol. 26, no. 11, pp. 1651-1661, Nov. 1991.
- [64] K. R. Cho, K. Okura and K. Asada, "Design of a 32-bit Fully Asynchronous Microprocessor (FAM)", in *Proc. Midwest Symp. Circuits Syst.*, vol. 2, 1992, pp. 1500-1503.
- [65] J. Sparsø, J. Staunstrup, and M. Dantzer-Sorensen, "Design of delay insensitive circuits using multi-ring structures," in *Proc. European Design Automation Conf.*, 1992, pp. 7-10.
- [66] T. Nanya, Y. Ueno, H. Kagotani, M. Kuwako, and A. Takamura, "TITAC: design of a quasi-delay-insensitive microprocessor," *IEEE Design & Test of Computers*, vol. 11, no. 2, pp. 50-63, Feb. 1994.
- [67] U. V. Cummings, A. M. Lines, and A. J. Martin, "An asynchronous pipelined lattice structure filter," in *Proc. Int. Symp. Advanced Research in Asynchronous Circuits Syst.*, 1994, pp. 126-133.
- [68] A. Takamura *et al.*, "TITAC-2: a 32-bit asynchronous microprocessor based on scalable-delay-insensitive model," in *Proc. Int. Conf. Comput. Design*, 1997, pp. 288-294.
- [69] A. J. Martin *et al.*, "The design of an asynchronous MIPS R3000 microprocessor," in *Proc. Conf. Advance Research in VLSI*, 1997, pp. 164-181.
- [70] M. Renaudin, P. Vivet, and F. Robin, "ASPRO-216: a standard-cell QDI 16-bit RISC asynchronous microprocessor," in *Proc. Symp. Advanced Research on Asynchronous Circuits Syst.*, 1998, pp. 22-31.
- [71] Camgian. [Online]. Available: <http://www.camgian.com/integratedcircuits.html>
- [72] A. Lines, "Nexus: an asynchronous crossbar interconnect for synchronous system-on-chip designs," in *Proc. High Performance Interconnects*, 2003, pp. 2-9.

- [73] A. Martin et al, "The Lutonium: a sub-nanojoule asynchronous 8051 microcontroller," in *Proc. IEEE Int. Symp. Asynchronous Circuits Syst.*, 2003, pp. 14–23.
- [74] C. Kelly IV, V. Ekanayake, and R. Manohar, "SNAP: a sensor network asynchronous processor," in *Proc. IEEE Int. Symp. Asynchronous Circuits Syst.*, 2003, pp. 24–33.
- [75] M. Nystrom, E. Ou, and A. J. Martin, "An eight-bit divider implementation in asynchronous pulse logic," in *Proc. IEEE Int. Symp. Asynchronous Circuits Syst.*, 2004, pp. 19–23.
- [76] V. Ekanauake, C. Kelly IV, and R. Manohar, "BitSNAP: dynamic significance compression for low power sensor network," in *Proc. IEEE Int. Symp. Asynchronous Circuits Syst.*, 2005, pp. 144–154.
- [77] M. Ferrretti, and P. A. Beerel, "High performance asynchronous design using single-track full-buffer standard cells," *IEEE JSSC*, vol. 41, no. 6, pp. 1444–1454, Jun. 2006.
- [78] A. Lines, "The Vortex: a superscalar asynchronous processor," in *Proc. IEEE Int. Symp. Asynchronous Circuits Syst.*, 2007, pp. 39–48.
- [79] M. Singh and S. M. Nowick, "The design of high-throughput asynchronous dynamic pipelines: lookahead pipelines," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 11, pp. 1256–1269, Nov. 2007.
- [80] Tiempo. [Online]. Available: <http://www.tiempo-ic.com>
- [81] K. M. Fant, and S. A. Bandt, "Null conventional logic: a complete and consistent logic for asynchronous digital circuit synthesis," in *Proc. Intl. Conf. Appl.-Spec. Syst. Arch. Processors*, 1996, pp. 261–273.
- [82] T. E. Williams, *Self-timed Rings and Their Applications to Divisor*. Ph.D Thesis, Standard University, 1991.
- [83] M. Ligthart, K. Fant, R. Smith, A. Taubin, A. Kondratyev, "Asynchronous Design Using Commercial HDL Synthesis Tools", in *Proc. IEEE Int. Symp. Asynchronous Circuits Syst.*, 2000, pp. 114-125.
- [84] A. Bailey *et al.*, "Multi-Threshold Asynchronous Circuit Design for Ultra-Low Power," *J. Low Power Electron.*, v4, n3, pp. 1-12, 2008.
- [85] C. Ortega, J. Tse, and R. Manohar, "Static power reduction techniques for asynchronous circuits," in *Proc. IEEE Symp. Asynchronous Circuits Syst.*, May 2010, pp. 52–61.
- [86] T. Kawano *et al.*, "Adjacent-State monitoring based fine-grained power-gating scheme for a low-power asynchronous pipelined system," in *Proc. IEEE ISCAS*, 2011, pp. 2067 - 2070.

- [87] M.-C. Chang and W.-H. Chang, "Asynchronous Fine-Grain Power-Gated Logic," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 6, pp. 1143–1153, Jun. 2013.
- [88] T. Lin, K.-S. Chong, B.-H. Gwee, J. S. Chang, and Z.-X. Qiu, "Analytical delay variation modelling for evaluating sub-threshold synchronous/asynchronous designs," in *Proc. IEEE Int. NEWCAS Conf.*, 2010, pp. 69–72.
- [89] V. De and S. Borkar, "Technology and design challenges for low power and high performance," in *Proc. ISLPED*, 1999, pp. 163–168.
- [90] S. Mutoh *et al.*, "1-V power supply high-speed digital circuit technology with multi-threshold voltage CMOS," *IEEE JSSC*, vol. 30, pp. 847–854, Aug. 1995.
- [91] T. Enomoto, Y. Oka, and H. Shikano, "A self-controllable voltage level (SVL) circuit and its low-power high-speed CMOS circuit applications," *IEEE JSSC*, vol. 38, pp. 1220–1226, 2003.
- [92] T. Kuroda *et al.*, "A 0.9V 150MHz 10mW 4mm 2-D discrete cosine transform core processor with variable-threshold-voltage scheme," in *Proc. IEEE ISSCC*, pp. 166–167, 1996.
- [93] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proc. IEEE*, vol. 91, pp. 305–327, 2003.
- [94] K.-S. Chong, B.-H. Gwee, and J. S. Chang, "Energy-efficient synchronous-logic and asynchronous-logic FFT/IFFT processors," *IEEE JSSC*, v42, n9, pp. 2034–2045, Sep. 2007.
- [95] C. J. Myers, *Asynchronous Circuit Design*. John Wiley & Sons, 2001.
- [96] J. Cortadella *et al.*, "Petrify: a tool for manipulating concurrent specifications and synthesis of asynchronous controllers," *IEICE Trans. Information and Systems*, E80-D(3), pp. 315–325, Mar. 1997.
- [97] Y. Cao and T. Clark, "Mapping statistical process variations toward circuit performance variability: An analytic modelling approach," in *Proc. IEEE DAC*, Anaheim, CA, Jun. 13–17, 2005, pp. 658–663.
- [98] F. Frustaci, P. Corsonello, and S. Perri, "Analytical Delay Model Considering Variability Effects in Subthreshold Domain," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 59, no. 3, pp. 168–172, Mar. 2012.
- [99] C. Hu, "BSIM model for circuit design using advanced technologies," in *Digest of Technical Papers Symp. VLSI Circuits*, 2001, pp. 5–10.

- [100] T. Lin, K.-S. Chong, J. S. Chang, and B.-H. Gwee, "An Ultra-Low Power Asynchronous-Logic In-Situ Self-Adaptive V_{DD} System for Wireless Sensor Networks," *IEEE JSSC*, vol. 48, pp. 573–586, Feb. 2013.
- [101] T. Lin, K.-S. Chong, J. S. Chang, B.-H. Gwee, and W. Shu, "A Robust Asynchronous Approach for Realizing Ultra-Low Power Digital Self-Adaptive V_{DD} Scaling System," in *Proc. IEEE Sub-threshold Microelectronics Conf.*, 2012, pp. 1-3.
- [102] T. Reddy and D. Linden, *Linden's Handbook of Batteries*, 4th ed. McGraw-Hill Professional, 2010.
- [103] Y. C. Lim, "Frequency response masking approach for the synthesis of sharp linear phase digital filters." *IEEE Trans. Circuits and Systems*, v33, n4, pp. 357-364, Apr. 1986.
- [104] J. S. Chang and Y.-C. Tong, "A micropower-compatible time-multiplexed SC speech spectrum analyzer design" *IEEE JSSC*, v28, n1, pp. 40–48, Jan. 1993.
- [105] E. Beigne *et al.*, "An asynchronous power aware and adaptive NoC based circuit," *IEEE JSSC*, v44, n4, pp. 1167–1177, Apr. 2009.
- [106] K.-S. Chong *et al.*, "Synchronous-logic and globally-asynchronous-locally-synchronous (GALS) acoustic digital signal processors," *IEEE JSSC*, v47, n3, pp. 769–780, Mar. 2012.
- [107] J. Tschanz *et al.*, "Adaptive frequency and biasing techniques for tolerance to dynamic temperature-voltage variations and aging," in *Proc. IEEE ISSCC*, Feb. 2007, pp. 292–293.
- [108] J. Kao, M. Miyazaki, and A. Chandrakasan, "A 175-mV multiply-accumulate unit using an adaptive supply voltage and body bias architecture," *IEEE JSSC*, v37, n11, pp. 1545–1554, Nov. 2002.
- [109] B. H. Calhoun and A. P. Chandrakasan, "Ultra-dynamic voltage scaling (UDVS) using sub-threshold operation and local voltage dithering," *IEEE JSSC*, v41, pp. 238–245, Jan. 2006.
- [110] M. Elgebaly and M. Sachdev, "Variation-aware adaptive voltage scaling system," *IEEE Trans. VLSI Syst.*, v15, n5, pp. 560–571, May 2007.
- [111] D. Bol *et al.*, "A 25MHz 7 μ W/MHz ultra-low-voltage microcontroller SoC in 65nm LP/GP CMOS for low-carbon wireless sensor nodes," in *Proc. IEEE ISSCC*, Feb. 2012, pp. 490-492.
- [112] S. Das *et al.*, "A self-tuning DVS processor using delay-error detection and correction," *IEEE JSSC*, v41, n4, pp. 792–804, Apr. 2006.

- [113] S. Das *et al.*, “*Razor II*: in situ error detection and correction for PVT and SER tolerance,” *IEEE JSSC*, v44, n1, pp. 32–48, Jan. 2009.
- [114] K. A. Bowman *et al.*, “A 45nm resilient microprocessor core for dynamic variation tolerance,” *IEEE JSSC*, v46, n1, pp. 194–208, Jan. 2011.
- [115] J. Mäkipää *et al.*, “Timing-Error Detection Design Considerations in Subthreshold: An 8-bit Microprocessor in 65 nm CMOS,” *J. Low Power Electron. Appl.*, v2, n2, pp. 180-196, 2012.
- [116] O. C. Akgun, J. Rodrigues, and J. Sparsø, “Minimum-energy subthreshold self-timed circuits: design methodology and a case study,” in *Proc. 16th ASYNC*, 2010, pp. 41–51.
- [117] W.-C. Hsieh and W. Hwang, “Adaptive power control technique on power-gated circuitries,” *IEEE Trans. VLSI Syst.*, v19, n7, pp. 1167–1180, Jul. 2011.
- [118] A. Kondratyev and K. Lwin, “Design of asynchronous circuits using synchronous CAD tools,” *IEEE Design Test Comput.*, v19, n4, pp. 107–117, 2002.
- [119] J. Cortadella *et al.*, “Coping with the variability of combinational logic delays,” in *Proc. ICCD*, Oct. 2004, pp.505–508.
- [120] D. Bol, “Robust and Energy-Efficient Ultra-Low-Voltage Circuit Design under Timing Constraints in 65/45 nm CMOS,” *J. Low Power Electron. Appl.*, v1, n1, pp. 1-19, 2011.
- [121] S. C. Smith and J. Di, *Designing Asynchronous Circuits using NULL Convention Logic (NCL)*. Morgan & Claypool, 2009.
- [122] K. L. Chang, *Asynchronous-Logic 8051 Microcontroller and Circuits: Dynamic Voltage Control*. Ph.D Thesis, Nanyang Technological University, 2011.