

**On the Road towards Robust and Ultra Low
Energy CMOS Digital Circuits Using
Sub/Near Threshold Power Supply**

Pu Yu

National University of Singapore

2009

**On the Road towards Robust and Ultra Low
Energy CMOS Digital Circuits Using
Sub/Near Threshold Power Supply**

Pu Yu

*(Bachelor of Engineering,
Zhejiang University, China)*

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF ELECTRICAL AND COMPUTER
ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE

2009

Acknowledgements

By this opportunity, I would like to express my gratitude and appreciation to everyone who has helped to make this thesis possible.

Sincere appreciation first goes to my supervisor prof.dr. José Pineda de Gyvez, for his guidance, support, encouragement during my PhD study. José is the most well-informed and hardworking IC specialist I have met. He has made an incredible huge effort in coaching me. His constructive criticisms have surely led to a much higher quality of this research work. I also thank him very much for giving me the valuable opportunity to work in NXP Research Eindhoven for over two years. I will never forget the attitude he taught me: working hard for glory.

I would like to thank prof.dr. Henk Corporaal, for many inspiring and in-depth discussions over these years. Henk opened my mind for problem formulation at the initial uncertain phase of my PhD time. His expertise in processor architecture is a key to the successful outcome of this research.

I would like to thank prof.dr. Ha Yajun, for bringing me into the joint PhD program and offering me the freedom to follow my ideas wherever they led. I also highly appreciate his careful reviewing my scientific papers and

providing valuable feedback.

The other members in my doctorate committee, prof.dr. Ralph Otten, prof.dr. Lian Yong and prof.dr. Patrick Girard, are specially appreciated for reading the thesis, giving in-depth comments and participating in my PhD defense.

My PhD time in TU/e and NUS would not have been so amazing without the presence of many colleagues: Marja, Rian, Sander Stuijk, Akash Kumar, Hu Hao, He Yifan, Tang Yongjian, Yu Yikun, Deng Wei, Yu Jianghong, Yu Rui, He Lin, Cen Ling, Hu Yingping, Tian Xiaohua, Wei Ying, Zou Xiaodan, Kine Lynn, Chen Xiaolei and Lee Cheesing. I wish them all the best.

I will never forget my friends in the “office of glory” in the Mixed-Signal Circuit and System Group of NXP Research Eindhoven: Maurice Meijer, Leo Sevat, Cas Groot, Agnese Bargagli-stoffi. I could not progress my project without their wise help and encouragement. I also thank Jan Stuyt and Jos Huiskens for their kind and helpful support during my short staying in IMEC.

I am deeply indebted to my parents Pu Yicheng and Liu Guilan, my wife Sophie Lin Lei, for their constant love, support, and patience. I am really lucky to be a member of such a wonderful family.

Finally, I owe gratitude to all of the friends who are always there for me. The friendship will last forever in my heart. Particularly, I thank Andy Chen Hao for his generous help and encouragement when I was at the painful phase of designing the *SubJPEG* prototype chip.

Contents

Summary	vi
Glossary	xv
1 Introduction	1
1.1 Voltage Scaling for Low-Power Digital Circuits	1
1.2 Practical Limitation of Voltage Scaling	5
1.3 Related Sub-threshold Work	8
1.4 Contributions of This Work	11
1.5 Thesis Organization	13
2 System Level Analysis	14
2.1 Sub-threshold Modeling	14
2.1.1 Sub-threshold Current Model	14
2.1.2 Sub-threshold Propagation Delay Model	21
2.1.3 Sub-threshold Energy Model	23
2.2 Optimum Energy-per-Operation (EPO)	24
2.3 Parallelism for Fixed Throughput	26

2.4	Noise Margin Estimation for Sub-threshold Combinational Circuits	28
2.4.1	Estimating gate noise margin with rectified equivalent resistance model	31
2.4.2	Estimating statistical output noise margin with affine arithmetic model	38
2.4.3	Experimental results	42
3	Physical Level Effort	46
3.1	Adaptive V_T for Process Spread Control in Sub/Near Threshold	46
3.2	Gate Sizing Considering V_T Mismatch in Deep Sub-threshold	56
3.3	Improving Drivability by Exploiting V_T Mismatch between Parallelized Transistors	63
3.4	Sub-threshold Library Cell Selection	68
3.5	Turning Ratioed Logic into Non-ratioed Logic	71
3.6	Capacitive-based Level Shifter (CBLC)	71
4	Design of the <i>SubJPEG</i> Co-processor	76
4.1	Design Flow Overview	76
4.2	JPEG Encoding Standard	78
4.3	<i>SubJPEG</i> Architecture	84
4.3.1	Design challenge	84
4.3.2	<i>SubJPEG</i> Macro-Architecture	87
4.3.3	Control Path Design	87
4.3.4	Data Path Design	96
4.4	Implementation Issues	100

4.4.1	Logic Design	100
4.4.2	Physical Design	103
4.5	Fabrication and Packaging	106
4.6	Performance Evaluation	108
5	Conclusions, Future Work and Discussions	119
5.1	Conclusions	119
5.2	Future Work	121
5.3	Discussions: Are we ready for sub-threshold?	122
	Bibliography	126
	Curriculum Vitae	i
	List of Publications	ii

Summary

Voltage scaling is one of the most effective and straightforward means for CMOS digital circuit's energy reduction. Aggressive voltage scaling to the near or sub-threshold region helps achieving ultra-low energy consumption. However, it brings along big challenges to reach the required throughput and to have good tolerance of process variations. This thesis presents our research work in designing robust near/sub-threshold CMOS digital circuits. Our work has two features. First, unlike the other research work that uses sub-threshold operation only for low-frequency low-throughput applications, we use architectural-level parallelism to compensate throughput degradation, so a medium throughput of up to 100MB/s suitable for digital consumer electronic applications can be achieved. Second, several new techniques are proposed to mitigate the yield degradation due to process variations. These techniques include: (a) Configurable V_T balancer to control the V_T spread. When facing process corners in the sub-threshold, our balancer will balance the V_T of p/nMOS transistors through bulk-biasing. (b) Transistor sizing to combat V_T mismatch between transistors. This is necessary if the circuit needs to be operated with very deep sub-threshold supply voltage, i.e., below 250mV for 65nm CMOS standard V_T process. (c) Improving sub-threshold

drivability by exploiting the V_T mismatch between parallel transistors. While the V_T mismatch between parallel transistors is notorious, we proposed to utilize it to boost the driving current in the sub-threshold. This interesting approach also suggests using multiple-finger layout style, which helps reducing silicon area considerably. (d) Selection procedure of the standard cells and how they were modified for higher reliability in the sub-threshold regime. Standard library cells that are sensitive to process variations must be eliminated in the synthesis flow. We provided the basic guideline to select “safe” cells. (e) The method that turns risky ratioed logic such as latch and register into non-ratioed logic.

SubJPEG, an ultra low-energy multi-standard JPEG encoder co-processor with a sub/near threshold power supply has been designed and implemented to demonstrate all these ideas. This 8-bit resolution DMA based co-processor has multiple power domains and multiple clock domains. It uses 4 parallel DCT-Quantization engines in the data path. Instruction-level parallelism is also used. All the parallelism is implemented in an efficient manner to minimize the associated area overhead. Details about this co-processor architecture and implementation issues are covered in this thesis. The prototype chip is fabricated in TSMC 65nm 7-layer Low-Power Standard V_T CMOS process. The core area is $1.4 \times 1.4 \text{mm}^2$. Each engine has its own V_T balancer. Each V_T balancer is $25 \times 30 \mu\text{m}^2$. The measurement results show that our V_T balancer has very good balancing effect. In the sub-threshold mode the engines can operate with 2.5MHz clock frequency at 0.4V supply, with 0.75pJ energy per cycle per single engine for DCT and Quantization processing, i.e. 0.75pJ/(engine-cycle). This leads to $8.3 \times$ energy/(engine-cycle) reduction

when compared to using a 1.2V nominal supply. In the near-threshold regime the energy dissipation is about $1.1\text{pJ}/(\text{engine}\cdot\text{cycle})$ with a 0.45V supply voltage at 4.5MHz. The system throughput can meet 15fps 640×480 pixel VGA compression standard. By further increasing the supply, the test chip can satisfy multi-standard image encoding. Our methodology is largely applicable to designing sound/graphic and other streaming processors.

List of Tables

1.1	Summary of low-power digital techniques	3
1.2	Biomedical and sensor applications	7
1.3	Summary of existing sub-threshold work	10
2.1	Parameters for 65nm CMOS SV_T process	17
2.2	Estimated statistical noise margin from Cadence Spectre Monte-Carlo DC simulation and the new approach	42
2.3	Estimated statistical noise margins as % of V_{DD}	43
3.1	Minimum supply voltage for an inverter in 65nm CMOS	55
3.2	$\lg(I_{eff}/I_{idle})$ for a 2-input NAND	58
3.3	Gate size normalized to minimum gate size vs. V_{DD} (functional yield = 99.9% and 99.7%, 65nm CMOS process)	60
3.4	Mean frequency, mean energy/cycle of ringo ($L_d = 31$, with and without V_T balancing scheme)	62
3.5	Mean and standard deviation of driving current	66
4.1	Some DP-CP interactive signals in RDC	92
4.2	Some DP-CP interactive signals in WRC	95

4.3	Memory design choices	98
4.4	Register files used in <i>SubJPEG</i> data path	98
4.5	System throughput and possible image applications	116

List of Figures

1.1	Applicable throughput range of this work and other work . . .	11
2.1	Sources of leakage current	15
2.2	Calibrated transistor current model and SPICE simulation for 65nm SV_T nMOS transistor	20
2.3	Illustration of the simulated transistor	21
2.4	Normalized driving current variability arising from different variation sources	22
2.5	Dynamic/Leakage/Total energy per operation and the optimal V_{DD} in SV_T process	25
2.6	Total EPO and the optimal V_{DD} points for SV_T and HV_T process	26
2.7	Normalized EPO at different V_{DD} for the same throughput .	28
2.8	(a) Cell schematic (b) Inverter (c) Equivalent model	33
2.9	Noise margin generated from Spectre Simulator vs from Equa- tion 2.23	35
2.10	Noise margin by definition and by this work	36

2.11	3σ range of noise margin generated from Spectre Simulator vs from Equation 2.23	37
2.12	Noise margin uncertainty propagation with AA model	39
2.13	Noise margin estimation flowchart	40
2.14	Probability density function (pdf) plots for benchmark C880 at $V_{DD} = 180\text{mV}$	45
3.1	(a) n and p sections (b) CMOS inverter	47
3.2	k versus V_{DD}	50
3.3	Transistor threshold tuning of an inverter through bulk-biasing	51
3.4	The proposed V_T balancing scheme with only one bulk-control line	52
3.5	Proposed configurable V_T balancer	53
3.6	Simulated 3σ range of ζ (with and without our V_T balancing scheme)	55
3.7	Propagation delay for an inverter in 65nm CMOS from Monte- Carlo simulation (with and without our V_T balancing scheme)	56
3.8	(a) two-input NAND gate (b) two-input NOR gate	59
3.9	(a) nMOS transistor with aspect ratio (W, L) (b) N-parallelized nMOS transistors with aspect ratio (W/N, L)	63
3.10	Layout of configurable V_T balancer with multiple finger struc- tured power switch in a 65nm CMOS	67
3.11	Prohibited cell structures in near/sub threshold (only parallel and stacked pMOS transistors are drawn for clarity)	69

3.12 Monte-Carlo transient simulation for cross-coupling feedback inverters at $V_{DD}=400\text{mV}$	70
3.13 Turning ratioed logic into non-ratioed logic	72
3.14 Monte-Carlo simulation results at node X at $V_{DD} = 400\text{mV}$: (a) before turning ratioed logic into non-ratioed logic (b) after turning ratioed logic into non-ratioed logic	73
3.15 Capacitive-based level converter (CBLC)	75
3.16 Waveforms of the CBLC ($V_{DDL}=400\text{mV}$ and $V_{DDH}=800\text{mV}$)	75
4.1 Sub-threshold design flow	77
4.2 JPEG encoder processing steps	79
4.3 AC zig-zag sequence	82
4.4 Design challenge	85
4.5 (a) Area (b) energy breakdown for conventional JPEG encoder	86
4.6 The functionality of <i>SubJPEG</i> in the system	86
4.7 <i>SubJPEG</i> processor diagram	88
4.8 Configuration space overview	89
4.9 Read controller diagram	90
4.10 Pseudo code algorithm for RDC	92
4.11 Write controller diagram	93
4.12 Pseudo code algorithm for WRC	94
4.13 Data path diagram	98
4.14 Normalized energy per cycle for each engine [energy/(engine·cycle)]	

4.15 Area vs. throughput for the engines and possible real-time image applications	100
4.16 2-stage level-shifting scheme in <i>SubJPEG</i>	101
4.17 Simulation of the 2-stage level-shifting scheme (0.4V to 0.6V to 1.2V)	102
4.18 <i>SubJPEG</i> floorplan	103
4.19 Gradient process variations	105
4.20 <i>SubJPEG</i> area and simulated energy breakdown in the digital still image mode	105
4.21 The layout of <i>SubJPEG</i> IP core integrated with the V_T balancers in Cadence Encounter view	106
4.22 The final chip layout with I/O pads in Mentor Graphic Calibre view	107
4.23 Prototype chip micrograph	108
4.24 Pin-out bonding diagram	109
4.25 Testing boards	109
4.26 Measurement results of switching on the V_T balancer	111
4.27 Measurement results from logic analyzer: (a)(c) are zoomed in results of (b)	112
4.28 Pulse trains from engines at $V_{DDL} = 400\text{mV}$ and $V_{DDL} = 800\text{mV}$	113
4.29 Transient current measurement scheme	113
4.30 Transient and average current at (0.4V, 2.5MHz), (0.8V, 5MHz) and (1.2V, 10MHz)	114
4.31 Energy per cycle for each engine [pJ/(engine-cycle)]	115
4.32 System energy and throughput	117

Glossary

K	constant intrinsic to the process
α	average switching activity factor
β	velocity saturation effect factor
n	sub-threshold swing factor
η	DIBL coefficient
U	thermal voltage kT/q (around 26mV at room temperature)
I_0	zero-threshold leakage current for a unit width transistor
I_{0n}	zero-threshold leakage current for a nMOS transistor
I_{0p}	zero-threshold leakage current for a pMOS transistor
C_{load}	load capacitance of a FO4 inverter
I_d	average driving current of a FO4 inverter

L_d	logic depth
T_g	propagation delay of a FO4 inverter
T_{cp}	critical path delay
T_c	operating cycle time
f	operating frequency
f_{max}	maximum operating frequency
I_l	off-state leakage current of a digital block
$E_{leakage}$	leakage energy per operation
$E_{dynamic}$	dynamic energy per operation
M	degree of parallelism
$Area_{baseline}$	silicon area of baseline processor
$T_{baseline}$	operating cycle time of baseline processor
$T_{overhead}$	timing overhead due to parallelism
ρ	superlinear area growth factor
V_T	transistor threshold voltage
V_{T0}	process intrinsic parameter for zero substrate bias

γ	body effect coefficient
$2\varphi_B$	transistor surface potential
$\sigma\Delta$	intra-die V_T mismatch deviation
$A\Delta V_T$	technology conversion constant (in $\text{mV}\mu\text{m}$)
W	transistor's effective width
L	transistor's effective length

Chapter 1

Introduction

It is the time for the semiconductor industry to play a part in dealing with the global energy bottleneck and climate change that face our society. In this chapter, we will first overview the CMOS low-power digital design techniques. Then the practical limitation for aggressive voltage scaling is stated. Following that we will review the existing sub-threshold works. Finally, the contributions of this work and the organization of this thesis are presented.

1.1 Voltage Scaling for Low-Power Digital Circuits

As early as in the 1970s, Gordon Moore had observed that the number of transistors on a silicon die doubled every 18 months (Moore's law) [1] . It is reported that for the last two decades the CMOS technology has been conventionally scaled to provide 30% smaller gate delay with 30% smaller dimensions each year [2] [3] , and an ever-increasing amount of Intellectual Property (IP) cores are integrated on a single System-on-Chip (SoC). The

practice today is that, while the number of transistors integrated in a chip doubles approximately every two years, the capacity density of battery doubles only every ten years. As a result, the energy bottleneck becomes crucial to many consumer electronic applications. Taking an MP3 player as an example, consumers are strongly calling for new MP3 players with lower price but much longer playing time. In addition to the energy problem, the heat also becomes an issue. If the released heat from chips cannot be removed quickly, the whole system performance becomes very instable. It is then inevitable to use special IC packaging and more advanced cooling techniques that support quick heat removal, which will increase product cost remarkably. Therefore, exploring the design methodology for low energy, “green” sub-micron circuits is of very great importance.

Targeting at broad and complex applications, SoCs normally integrate RF and analog modules such as transceivers, Phase (or Delay)-Locked-Loops (PLLs or DLLs), A/D-D/A converters, and digital modules such as multiple processors, memories, etc. The design trend has been to put more and more functionalities to digital modules for two reasons. First, modern Electronic Design Automation (EDA) tools support almost full automation of digital design flow. Integration of a large variety of processing functionalities into digital modules is much easier than into analog modules. Second, compared to analog signal processing, digital signal processing (DSP) is superior due to better noise immunity, smaller silicon area and less power consumption. Therefore, the digital modules are generally the dominant power consumer on a SoC.

The total power dissipation of a digital system is composed of the dy-

Table 1.1: Summary of low-power digital techniques

Design hierarchy	Reported low-power digital techniques
Algorithm level	<ol style="list-style-type: none"> 1. using more efficient DSP algorithms to eliminate unnecessary computations and reduce the number of computations
Mapping and architecture level	<ol style="list-style-type: none"> 1. ISA extension, e.g., ASIP 2. scenario based mapping, rescheduling, etc. 3. preserving data correlation and reference locality, reducing memory access 4. common expression elimination 5. pre-computation, etc. 6. using suitable pipelining and parallelism, enabling low supply voltage/frequency
System level	<ol style="list-style-type: none"> 1. multiple supply voltages (MSV) 2. dynamic voltage scaling (DVS) 3. dynamic voltage-frequency scaling (DVFS) 4. multiple clock domains 5. dynamic/variable V_T (adaptive body biasing) 6. sleep and power down modes
Circuit level	<ol style="list-style-type: none"> 1. power gating, clock gating 2. logic sizing and logic re-structuring 3. adiabatic logic circuits 4. low power SRAM, DRAM, etc. 5. power-efficient DC-DC converters
Device level	<ol style="list-style-type: none"> 1. multiple threshold CMOS (MTCMOS) 2. low temperature CMOS (LTCMOS) 3. Silicon-on-Insulator (SOI) 4. low power packaging

dynamic power, the leakage power and the short-circuit power. The dynamic power results from charging and discharging loading capacitances. It is often the dominant power consumer. The leakage power results from imperfect switch-off of nMOS/pMOS transistors. It is due to the current conducted even without any switching activity. Since millions of transistors are often integrated in a single SoC nowadays, the contribution of leakage power to the total power also becomes significant. The leakage current is sensitive to thermal conditions as its absolute value increases in an exponential fashion with the increasing temperature, so its significance can further increase if the released heat cannot be removed quickly. The short-circuit power dissipation is due to direct-path current when the nMOS and the pMOS transistors are conducting simultaneously during non-ideal rise/fall times. It only contributes a minor fraction ($<5\%$) of the total power dissipation.

Table 1.1 summarizes many low-power digital circuit techniques [52] [53]. These techniques are categorized by their level in the design hierarchy. To achieve low power, it needs a wide collaboration of designers from each level hierarchy. In general, these techniques trade-off flexibility, performance and silicon area for power. Among these techniques, the most straightforward and effective means are to scale the supply voltage V_{DD} along with the operating frequency. As V_{DD} scales, not only does the dynamic power reduce quadratically, the leakage current also reduces super-linearly due to the drain-induced barrier-lowering (DIBL) effect. In this way, the total power dissipation can be reduced considerably. In addition to power savings, V_{DD} scaling mitigates the transient current, hence lowering the notorious ground bounce noise (Ldi/dt). This also helps to improve the performance of sensi-

tive analog circuits on the chip, such as the delay-lock loop (DLL), which is crucial for the correct functioning of complex digital circuits.

In the techniques listed in Table 1.1, multiple supply voltages (MSV), dynamic voltage scaling (DVS), and dynamic voltage-frequency scaling (DVFS) are three means of voltage scaling. MSV is a static approach, which provides different supply voltages to different power domains. DVS and DVFS are two adaptive approaches. Both of them exploit the variation in processor utilization: lowering the frequency and voltage when the processor is lightly loaded, and running at maximum frequency and voltage when the processor is heavily executing. They have been widely deployed for commercial microprocessors, achieving significant power savings [4,5,6,7,8].

1.2 Practical Limitation of Voltage Scaling

For applications requiring ultra-low energy dissipation, such as wireless “motes”, sensor networks [10], in-vivo biomedicine (such as hearing aids, pace-makers, implantable device) [11] and wrist-watch computation [12], the techniques in Table 1.1 are not powerful enough. Table 1.2 lists some more biomedical and sensor applications that fall in this category. For each application, the associated sampling rates (in Hz) and the sample precision (in bits per sample) are also listed. Ideally, these applications should be self-powered, relying on scavenging energy from the environment, or at least be sustained by a small battery for tens of years. Such a stringent energy budget constrains the total system computation power to less than a hundred microwatts, which poses a great challenge to modern CMOS digital design.

Unlike analog circuit design where lowering the supply voltage to the sub-threshold region is generally avoided because of the low values of the driving currents and the exceedingly large noise, CMOS digital logic gates can work seamlessly from full V_{DD} to well below the threshold voltage V_T . Theoretically, operating digital circuits in the near/sub-threshold region ($V_{GS} < V_T$) can help obtaining huge energy savings. Therefore, sub-threshold techniques provide a potential solution for the ultra-low energy applications. They may also be applicable to applications with bursty characteristics, e.g., microprocessors which infrequently require high performance and most of the time it only makes sense to have a near-standby mode [13] [14] .

However, the design rules provided by foundries normally set $2/3$ of the full V_{DD} as the lower bound for V_{DD} scaling in deep sub-micron processes. Taking the Samsung's DVFS Design Technology [9] and the TSMC design rule as examples, the constraint of V_{DD} for digital circuits designed in CMOS 65nm Standard V_T Process is in the $0.8V \sim 1.2V$ range. The reasoning behind the lower constraint is twofold. First, as V_{DD} scales, the driving capability of transistors reduces accordingly. Because most electronic consumer applications need operating frequencies in the range of tens of MHz to reach certain throughput, which might not be fulfilled with aggressive V_{DD} scaling, $2/3 V_{DD}$ is tested to be a safe lower bound. Second, digital circuits become particularly sensitive to process variations when V_{DD} scales below $2/3 V_{DD}$. Process variations are likely to cause malfunctioning, and both the timing yield and functional yield tremendously decrease. As a result, $2/3 V_{DD}$ is generally chosen to maintain adequate margin to prevent high yield loss and to keep quality to the industry standard. Obviously, this limitation has pre-

Table 1.2: Biomedical and sensor applications

Application	Sample rate (in Hz)	Sample precision (in bits)
Body temperature	0.1 ~ 1	8
Heart rate	0.8 ~ 3.2	1
Blood pressure	50 ~ 100	8
EEG	100 ~ 200	16
EOG	100 ~ 200	16
ECG	100 ~ 250	8
Breathing sounds	100 ~ 5K	8
EMG	100 ~ 5K	8
Audio (hearing aids)	15 ~ 44K	16
Ambient light level	0.017 ~ 1	16
Atmospheric temperature	0.017 ~ 1	16
Ambient noise level	0.017 ~ 1	16
Barometric pressure	0.017 ~ 1	8
Wind direction	0.2 ~ 100	8
Seismic vibration	1 ~ 10	8
Engine temperature	100 ~ 150	16
Engine pressure	100 ~ 150	16

vented further power/energy reduction from voltage scaling. To safely evade this limitation and to enable wide range voltage scaling from the nominal supply to the near/sub threshold region is a goal to be achieved in this work.

1.3 Related Sub-threshold Work

In recent years, some design techniques for operating digital circuits in the sub-threshold region ($V_{GS} < V_T$) have been explored. Table 1.3 summarizes and categorizes the existing energy-efficient techniques that take advantage of sub-threshold operation. Most of these works are from the M.I.T sub-threshold circuit group headed by Professor Anantha Chandrakasan, in association with Texas Instruments. As can be seen from Table 1.3, the existing sub-threshold works span many different levels of abstraction. On the system level, some research has been done to model the characteristics of sub-threshold circuits, including current, delay, energy, variations, etc. Based on these models, the performance of a given sub-threshold system, the optimal energy point and the possible energy savings can be obtained. On the physical level, researchers have made effort to develop circuit styles for logic that can operate in the sub-threshold. The authors in [19] provide a closed-form solution for sizing transistors in a stack and introduce a new logical effort suitable to sub-threshold design. Traditional logic families like domino [60], pass transistor logic, pseudo nMOS [61] have also been considered for their usefulness in sub-threshold regime. In addition, sub-threshold on-chip SRAM architectures and circuits have been explored, as later it is found that SRAMs were the energy consumption bottleneck for micro-processors at ultra-low voltages.

Some very interesting prototype chips which function in the sub-threshold, have been presented. Among these chips, the most famous are the 180mV FFT processor in 180nm CMOS process designed by Alice Wang in 2004 [33] [34] . This is the first digital processor working in the sub-threshold. Ben Calhoun had designed the 256**kb** 10-T dual port SRAM in 65nm CMOS process [24] . It had been improved to 8-T dual port SRAM by Naveen Verma in 2007 [29] [30] . A sensor node processor having both sub-threshold logic and SRAMs is presented by University of Michigan [31][32]. It claims the highest energy savings. Recently, M.I.T group and Texas Instruments had jointly announced the newest sub-threshold MSP430 DSP processor with integrated DC-DC [38] [39] .

It is also worth mentioning some effort that has been made to create the “perfect” transistor for sub-threshold operation. Optimized MOSFET [62] [63] , SOI MOSFET [64] [65] , double gated MOSFET [66] may gain increasing popularity for their usage in sub-threshold design. SOI MOSFETs have much steeper subthreshold slope and more resistance to short-channel effects. [66] proposed to use double gated MOSFET in sub-threshold due to its steep subthreshold slope and a small gate capacitance. In addition, MTCMOS, VTCMOS, dual/multiple V_T partitioning are also claimed to benefit sub-threshold design.

However, the downsides of these existing works are still the considerable performance loss at ultra-low supply voltages and yield loss due to the effects of process variations.

Table 1.3: Summary of existing sub-threshold work

Category	Existing sub-threshold work
Sub-threshold modeling	[15] [16] [17] [18] : built up the analytical models for sub-threshold current, delay, energy and variations
Sub-threshold logic design	[19] [20] [21] [22] [60] [61] : explored sub-threshold logic cells
Sub-threshold memory	[23] [24] : 256 kb 10-T dual-port SRAM in 65nm CMOS [25] : 512×13 b dual-port SRAM in 180nm CMOS [26] : 480 kb 6-T dual-port SRAM in 130nm CMOS [27] [28] : 2 kb 6-T single-port SRAM in 130nm CMOS [29] [30] : 256 kb 8-T dual-port SRAM in 65nm CMOS
Sub-threshold processors	[31] [32] : 2.6pJ/inst 3-stage pipelined sensor node processor in 130nm CMOS [33] [34] : 180mV FFT processor in 180nm CMOS [35] [36] : 0.4V UWB baseband processor in 65nm CMOS [37] : 85mV 40nW 8×8 FIR filter in 130nm CMOS [38] [39] : 2-stage pipelined micro-controller with embedded SRAM and DC-DC converter in 65nm CMOS

1.4 Contributions of This Work

The major contributions of this work include:

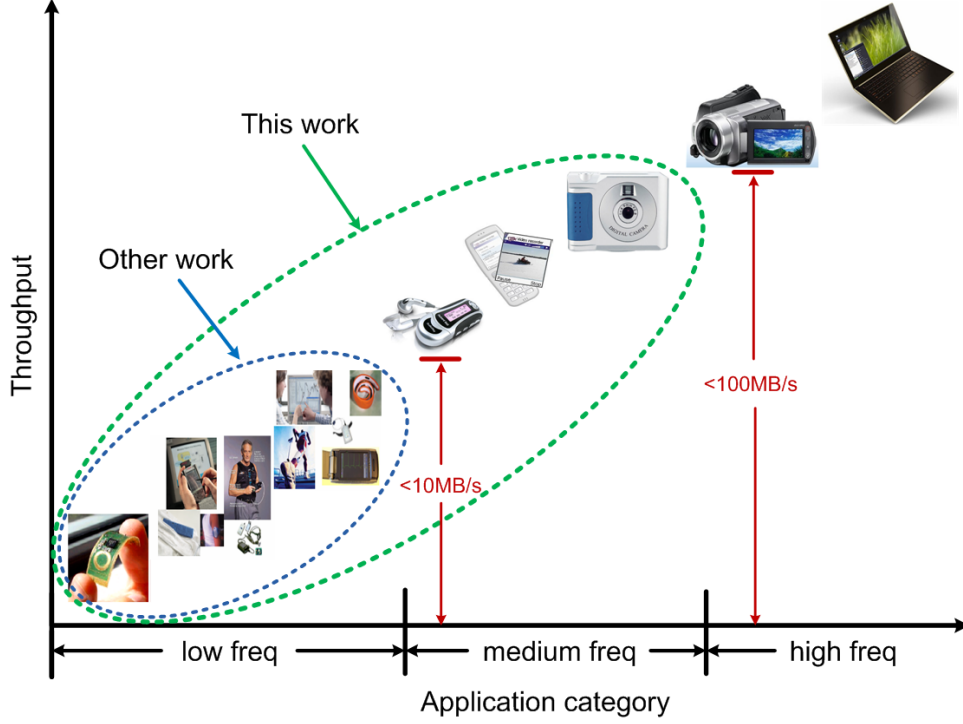


Figure 1.1: Applicable throughput range of this work and other work

- Although operating in the sub-threshold renders huge energy savings, it is believed only suitable for low-speed applications because the drivability is very small. This work explores the possibility to use architecture-level parallelism to compensate for throughput degradation. Through efficient parallelism, sub/near threshold techniques are extended to low-energy and medium throughput applications, such as mobile image processing. Figure 1.1 shows the applicable throughput range of this work

and the other work.

- Little attention has been given in previous art to the sub/near threshold circuit's yield. This work makes an effort to increase the reliability of sub/near threshold circuits. We propose a novel, configurable V_T balancer to balance the V_T between nMOS and pMOS transistors. Our V_T balancer helps increasing both the functional yield and timing yield.
- In addition to the V_T balancer, other sub-threshold physical level approaches including transistor sizing, utilizing parallel transistor V_T mismatch to improve drivability, selecting reliable library cells for logic synthesis, turning ratioed logic into non-ratioed logic, and level shifter design, are addressed in this thesis.
- To estimate noise margins, minimum functional supply voltage, as well as the functional yield in the sub-threshold, this work proposes a fast, accurate and statistical method based on *Affine Arithmetic* (AA). This method has an accuracy of 98.5% w.r.t. to transistor-level Monte Carlo simulations, but the running time is much shorter.
- *SubJPEG*, a state-of-the-art ultra-low energy multi-standard JPEG encoder co-processor is designed and implemented to demonstrate these ideas. This $1.4 \times 1.4 \text{mm}^2$ 8-bit resolution DMA based co-processor chip is fabricated with TSMC 65nm 7-layer standard V_T CMOS process. It contains 4 parallel DCT-Quantization engines, 2 voltage domains and 3 clock domains. For DCT and quantization operation, this co-processor dissipates only 0.75pJ energy per single engine in one clock

cycle, when using a 0.4V power supply at the maximum 2.5MHz in the sub-threshold mode, which leads to $8.3\times$ energy reduction compared to using the 1.2V nominal supply. In the near-threshold mode the engines can operate with 4.5MHz frequency at 0.45V, with 1.1pJ energy per engine in one cycle. The overall system throughput then still meets 640×480 15fps VGA compression requirement. By further increasing the supply voltage, the prototype chip can satisfy multi-standard image encoding. To our best knowledge, *SubJPEG* is the largest, sub/near threshold system so far.

1.5 Thesis Organization

This thesis is organized into five chapters. Chapter 1 presents the background of voltage scaling, reviews the related previous art about sub-threshold techniques and states the contributions that have been made by this thesis. In Chapter 2, many aspects of a sub-threshold system modeling, including current, delay, energy, variability and optimum V_{DD} are analyzed. The feasibility to compensate for throughput degradation by using architecture-level parallelism is also explored. An EDA approach for fast noise margin estimation for deep sub-threshold combinational circuits is introduced at the end of this chapter. Chapter 3 presents the physical level effort we have made to improve sub-threshold circuit's yield. In Chapter 4, the design of *SubJPEG* prototype chip is presented in detail. Finally, the conclusions, future work and discussions are given in Chapter 5.

Chapter 2

System Level Analysis

To quickly analyze the performance of a sub-threshold system, in this chapter we present the sub-threshold modeling, including current, delay, energy and variability. The optimum V_{DD} , at which the energy per operation is the lowest, is analyzed. The feasibility to compensate for throughput degradation by using architecture-level parallelism is also discussed. Finally, an EDA approach for fast sub-threshold noise margin estimation is introduced.

2.1 Sub-threshold Modeling

2.1.1 Sub-threshold Current Model

Sub-threshold design exploits leakage current as the driving current. We should first understand where the leakages come from. Figure 2.1 illustrates the leakage currents of a short channel device [54]. These leakage sources include:

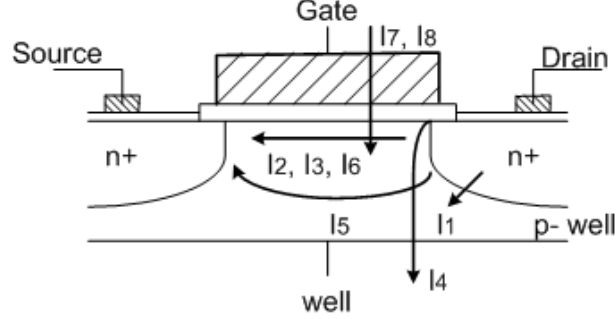


Figure 2.1: Sources of leakage current

a) pn Junction Reverse Bias Current (I_1)

A reverse bias pn junction leakage involves two key components. One is minority carrier diffusion/drift near the edge of depletion region and the other is due to electron-hole pair generation in the depletion region of the reverse bias junction. I_1 is a non-significant contributor to total leakage current.

b) Sub-threshold Leakage (I_2)

Sub-threshold conduction current between source and drain in a MOS transistor occurs when gate voltage is below V_T . Sub-threshold conduction is dominated by the diffusion current. The carriers move by diffusion along the surface. Weak inversion conduction dominates modern device off state leakage, especially when low V_T processes are used.

c) Drain -Induced Barrier Lowering - DIBL (I_3)

In a short-channel device, the source-drain potential has a strong effect on the band bending over a significant portion of the device. As a result, the threshold voltage and consequently the sub-threshold current of short-channel device vary with the drain bias. The barrier of a short-channel device reduces along with the increase of drain voltage, which causes a lower threshold voltage

and a higher sub-threshold current. This effect is referred as Drain-Induced Barrier Lowering (DIBL).

d) Gate -Induced Drain Leakage - GIDL (I_4)

Gate-Induced Drain Leakage (GIDL) is due to high field effect in the drain junction of MOS transistor. When the gate is biased to cause an accumulation layer at the silicon surface, the silicon surface under the gate has almost the same potential as the p-type substrate.

e) Punch Through (I_5)

Punch-through occurs when drain and source depletion regions approach each other and electrically “touch” in the channel. Punch-through is a space-charge condition that allows channel current to exit deep in the sub-gate region.

f) Narrow Width Effect (I_6)

Transistor V_T in non-trench isolated technologies increases for geometric gate widths on the order of $0.5\mu\text{m}$. No narrow width effect is observed when transistor sizes exceed significantly $0.5\mu\text{m}$.

g) Gate Oxide Tunneling (I_7)

Reduction of gate oxide thickness results in increase in field across the oxide. The high electric field coupled with low oxide thickness results in tunneling of electrons from substrate to gate and from gate to substrate through gate oxide, resulting in gate oxide tunneling current. Gate oxide tunneling current could surpass weak inversion and DIBL as a dominant leakage in the future as oxide get thin enough.

h) Hot Carrier Injection (I_8)

In a short channel transistor, because of high electric field near the Si/SiO₂ interface, electrons and holes can gain sufficient energy from the electric field

to cross the interface potential barrier, and enter into the oxide layer. This effect is known as hot carrier injection.

Among the leakage currents, sub-threshold leakage (I_2) and DIBL (I_3) are the source of leakage used as driving current in the sub-threshold design. Conventionally, this driving current of an nMOSFET is modeled by

$$I_D = \begin{cases} WI_0 e^{\frac{(V_{GS}-V_T-\gamma V_{SB}+\eta V_{DS})}{nU}} (1 - e^{-\frac{V_{DS}}{U}}) & (V_{GS} < V_T) \\ WKI_0(V_{GS} - V_T)^\beta & (V_{GS} \geq V_T) \end{cases} \quad (2.1)$$

where K is a constant intrinsic to the process, β is the velocity saturation effect factor, n is the sub-threshold swing factor, η is the DIBL coefficient, W is the transistor width. U is the so-called thermal voltage kT/q , which is around 26mV at room temperature. I_0 is the zero-threshold leakage current for a unit width transistor. Typical values for the parameters in a 65nm Standard V_T CMOS process are given in Table 2.1. Please note the slight discontinuity at $V_{GS}=V_T$ in the model. Equation (2.1) clearly indicates a super-linear decrease of sub-threshold driving current due to V_{DD} scaling, since V_{GS} is often considered approximately equal to V_{DD} in analysis.

Table 2.1: Parameters for 65nm CMOS $S V_T$ process

n	η	γ	V_T
1.37	0.03	0.33	0.41

Although the current model in equation (2.1) is well-known for its simplicity for back-of-the-envelope mathematic manipulations, we found it inadequate to capture device characteristics for very deep submicron CMOS technology. This model has two problems: 1) in the sub-threshold region, the current's absolute value is not very accurate. 2) It is unfavorable at the trans-regional part, from the sub-threshold to near-threshold. These drawbacks can be seen from Figure 2.2. Similar problems have also been observed by the MIT group [17] . To keep the simplicity but improve the accuracy, we have calibrated this trans-regional model, which is described by:

$$I_D = \begin{cases} WI_0 e^{\frac{(V_{GS}-V_T-k_1-\gamma V_{SB}+\eta V_{DS})}{nU}} (1 - e^{-\frac{V_{DS}}{U}}) & (V_{GS} < V_T + k_1) \\ WKI_0 (V_{GS} - V_T)^\beta & (V_{GS} \geq V_T + k_1) \end{cases} \quad (2.2)$$

where k_1 is a constant parameter obtained with a Levenberg–Marquardt algorithm (LMA) through curve fitting. If we define

$$V'_T = V_T + k_1 \quad (2.3)$$

Then equation (2.2) becomes equation (2.4) ,

$$I_D = \begin{cases} WI_0 e^{\frac{(V_{GS}-V'_T-\gamma V_{SB}+\eta V_{DS})}{nU}} (1 - e^{-\frac{V_{DS}}{U}}) & (V_{GS} < V'_T) \\ WKI_0(V_{GS} - V_T)^\beta & (V_{GS} \geq V'_T) \end{cases} \quad (2.4)$$

Figure 2.2 also compares this calibrated transistor current model with a SPICE simulation model for an nMOSFET in a CMOS 65nm Low Power Standard $V_T(\text{LP} - \text{SV}_T)$ technology. As shown, the model provides very good accuracy with respect to the SPICE simulation. The largest deviation occurs when V_{DD} is in the vicinity of V'_T , which is about 0.48V in our case.

The actual value of driving current is not to our interest. We are interested in the current scaling factor, which is needed to estimate circuit's performance at an ultra low V_{DD} based on our measurement results at nominal V_{DD} . Note that although changing the aspect-ratio of the nMOS transistor may result in different driving currents, it will not affect the scaling factor. Considering that the pMOS transistors in logic gates are normally carefully sized to have a symmetric characteristic with their nMOS counterparts, it is reasonable to assume pMOS transistors have the same scaling factor with the nMOS transistors. With the calibrated model, we alleviate the discontinuity at transregion hence making the estimation quicker and easier.

In super-threshold design, the supply voltage V_{DD} , the geometric L_{eff} and the threshold V_T , are the major variability sources. It is necessary to investigate how each of them contribute to the total current variation in the sub-

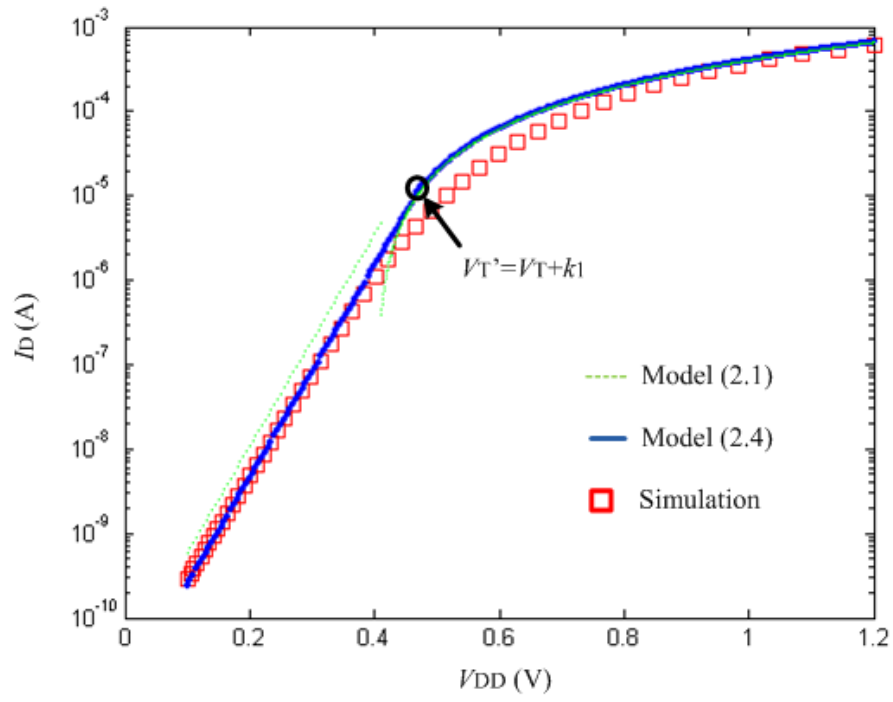


Figure 2.2: Calibrated transistor current model and SPICE simulation for 65nm $S V_T$ nMOS transistor

threshold. We take an nMOS transistor whose aspect ratio is $0.4\mu\text{m}/0.065\mu\text{m}$, and connect its gate to V_{DD1} and its drain to V_{DD2} , respectively. Its bulk and source are connected to G_{ND} , as shown in Figure 2.3. We assume that $V_{DD1}=0.9V_{DD2}$ and $V_{DD2}=V_{DD}$. The parameters that are varied to compute the envelope are L_{eff} ($\pm 5\%$ variation), V_T ($\pm 10\%$ variation) and V_{DD2} ($\pm 10\%$ variation). In Figure 2.4 the sensitivity $\Delta I_D/I_D$ arising from each different variability source is normalized to that arising from all variability sources at $V_{DD}=200\text{mV}$. It is clear that threshold voltage variation is the dominant criminal for sub-threshold current variation due to its exponential correlation, and therefore becomes our major enemy. In contrast, the other two variation sources have relatively small impact, which can be mitigated by designing with narrow margins. Although the absolute value and shares of the variability sources could change for different parameter settings, this conclusion still hold true.

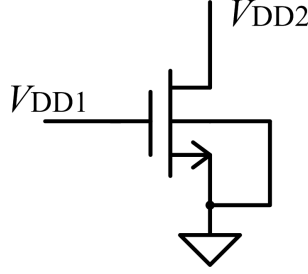


Figure 2.3: Illustration of the simulated transistor

2.1.2 Sub-threshold Propagation Delay Model

To model the sub-threshold propagation delay, we assume C_{load} the load capacitance of a FO4 inverter and I_d the average driving current of a FO4 in-

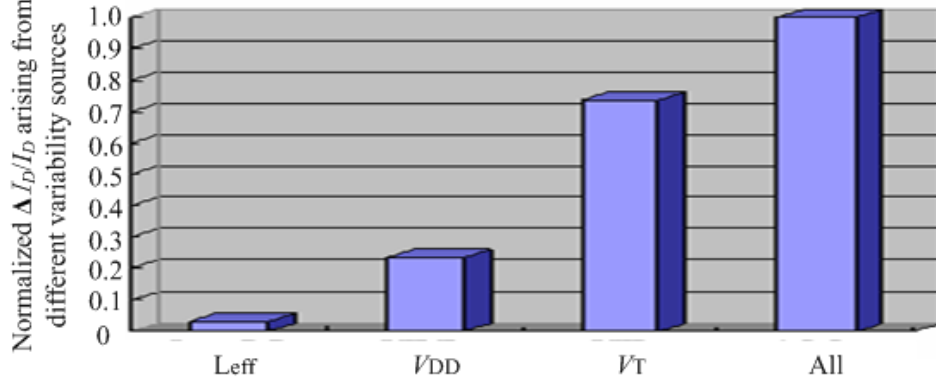


Figure 2.4: Normalized driving current variability arising from different variation sources

verter. L_d is the logic depth which represents how many inverters are chained to mimic the critical path delay. The propagation delay of a characteristic inverter T_g can be derived as

$$T_g = C_{load}V_{DD}/I_d \quad (2.5)$$

and the critical path delay is

$$T_{cp} = L_d T_g \quad (2.6)$$

The maximum operating frequency of the chip is then calculated,

$$f_{\max} = \frac{1}{T_{cp}} \quad (2.7)$$

2.1.3 Sub-threshold Energy Model

Instead of using power as the metric, we use energy-per-operation (EPO) in our study since it is the real metric to battery life. Dynamic energy and leakage energy are the two major sources of energy dissipation in CMOS digital circuits. The dynamic energy per operation is

$$E_{dynamic} = \alpha C V_{DD}^2 \quad (2.8)$$

where α is the average switching activity factor of all the output nodes, C is the total capacitance of all the output nodes, V_{DD} is the supply voltage.

The off-state leakage current I_l of a digital block is dominated by the zero sub-threshold leakage [40]. I_l can be modeled by letting $V_{GS}=0$ and $V_{DS}=V_{DD}$ in equation (2.2), i.e.

$$I_l = W I_0 e^{\frac{(-V'_T - \gamma V_{SB} + \eta V_{DD})}{nU}} (1 - e^{-\frac{V_{DD}}{U}}) \quad (2.9)$$

Thus, the leakage energy per operation can be obtained as

$$E_{leakage} = I_l V_{DD} T_c \quad (2.10)$$

where T_c is the operating cycle time. The total EPO of a digital circuit is

$$EPO = E_{dynamic} + E_{leakage} = \alpha C V_{DD}^2 + I_l V_{DD} T_c \quad (2.11)$$

2.2 Optimum Energy-per-Operation (EPO)

Above analysis shows that, as voltage scales, the dynamic energy reduces. However, because of the increased delay, the leakage energy increases. Therefore, whether the total EPO increases or decreases is uncertain. In fact, there is an optimum-energy supply voltage point, operating at which offers the best EPO. Theoretically, this point can be solved by

$$\partial \text{EPO} / \partial V_{DD} = 0 \quad (2.12)$$

This optimum voltage point can also be obtained experimentally. Let us introduce a baseline processor which is based on [55] from NXP. This real million-gate baseline processor is fabricated in a CMOS 65nm Low Power Standard V_T (LP-SV $_T$) technology. The average switching activity factor α is 0.12, the total switching capacitance for the entire block is 4.9nF, the nominal V_{DD} is 1.2V, average V_T of pMOS and nMOS is 0.41V, off-state leakage I_l is 648 μ A and $L_d = 24$. This baseline processor is supposed to run at its maximum speed, i.e., $T_c = T_{cp}$. Figure 2.5 shows how the dynamic, leakage and total energy of the baseline processor vary when V_{DD} scales. The simulated optimal V_{DD} point V_{opt} is indicated. Since nowadays high V_T (HV $_T$) processes are a popular option for low power digital design, a simulation has also been carried for the same block implemented through a HV $_T$ process. Figure 2.6 compares the total energy per operation for SV $_T$ and HV $_T$ processes. The behavior of these curves is similar.

As indicated by Figure 2.5 and Figure 2.6, the optimal energy operating

2.2. OPTIMUM ENERGY-PER-OPERATION (EPO)

supply voltage V_{opt} is in the sub-threshold region. Further lowering V_{DD} below V_{opt} does not yield any additional energy benefits. We also analyzed some other circuits, and found that their V_{opt} is normally greater than 0.3V. This suggests not scaling the V_{DD} to the very deep sub-threshold level but to stay at the weak sub-threshold or near threshold region. In fact, only for a digital block with extremely high switching density, there is a need to scale its V_{DD} into very deep sub-threshold region. In addition, we observe that using the HV_T process raises the EPO with 13% as compared to the SV_T process. Therefore, the SV_T process is selected for our research.

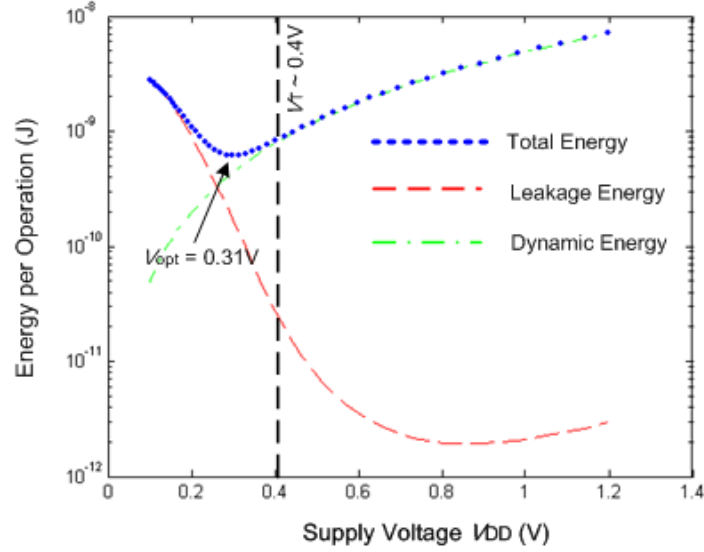


Figure 2.5: Dynamic/Leakage/Total energy per operation and the optimal V_{DD} in SV_T process

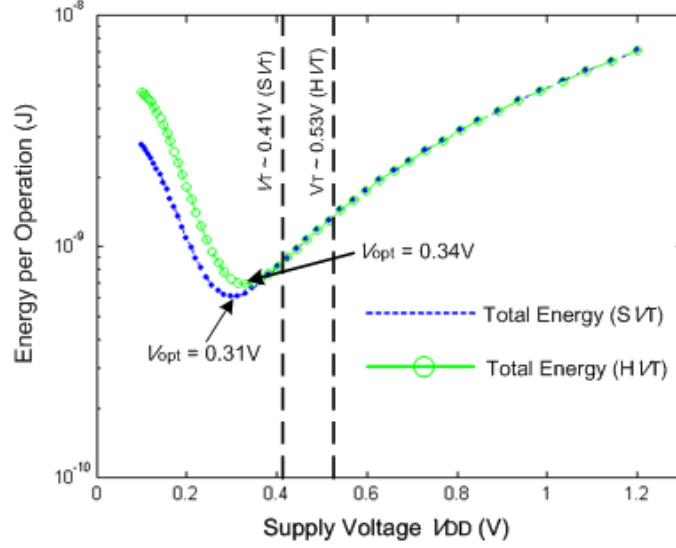


Figure 2.6: Total EPO and the optimal V_{DD} points for SV_T and HV_T process

2.3 Parallelism for Fixed Throughput

The circuit throughput degrades when V_{DD} scales. To maintain a fixed throughput, parallel processing units can be used. We assume that the computation tasks of individual units are independent, meaning that no performance penalty due to data or control dependencies is incurred from parallelism. This assumption is largely suitable for applications such as sound/graphic and other streaming processing, though there are still some sequential parts. Ideally, for a fixed V_{DD} , the degree of parallelism does not affect the EPO whereas a larger throughput can be obtained simply by using more parallelized units. However, in reality the multiplexer and demultiplexer circuits also contribute to increased overhead in the EPO. To take this overhead into account, the area and timing are approximated in

equations (2.13) ,(2.14) , and (2.15) ,

$$Area = Area_{baseline} \times M^{\rho} \quad (2.13)$$

$$T_{overhead} = \log_2 M \times FO4 \quad (2.14)$$

$$T_c = T_{baseline} + T_{overhead} \quad (2.15)$$

where M is the associated degree of parallelism, and ρ is the area growth factor which indicates that the circuit area grows super-linearly with M . In our simulation, we choose $\rho=1.1$ [56] . Referring to equation (2.11), the area overhead affects C and I_l , while the timing overhead affects T_c .

Figure 2.7 shows the normalized EPO for different values of V_{DD} , with the same throughput as that of the baseline processor operating at the nominal 1.2V supply voltage. The necessary degrees of parallelism for a few V_{DD} points are annotated in the plot. As shown, compared to operating at the 1.2V nominal V_{DD} , we could obtain $5\times$, $4\times$, $3\times$ EPO reduction when V_{DD} is at 0.4V, 0.5V, 0.6V, respectively. At first glance it is unwelcome to see the associated 245, 31 and 12 parallel widths, which implies an impossibly large silicon area. In addition, the larger the circuit's area, the more likely are defects, and so will fail to achieve commercially viable yields. However, it should be noted that in the analysis we assume the baseline processor is operating at its maximum speed, which is about 300MHz. For some consumer electronic applications which only need up to a few tens of MHz, the

2.4. NOISE MARGIN ESTIMATION FOR SUB-THRESHOLD COMBINATIONAL CIRCUITS

associated parallel width can become much smaller and thus more affordable. For applications that only run at KHz~MHz range frequencies, such as sensor networks, biomedical instrumentations and audio processors, operating at the V_{opt} is possible and there is not even a need to use parallel paths.

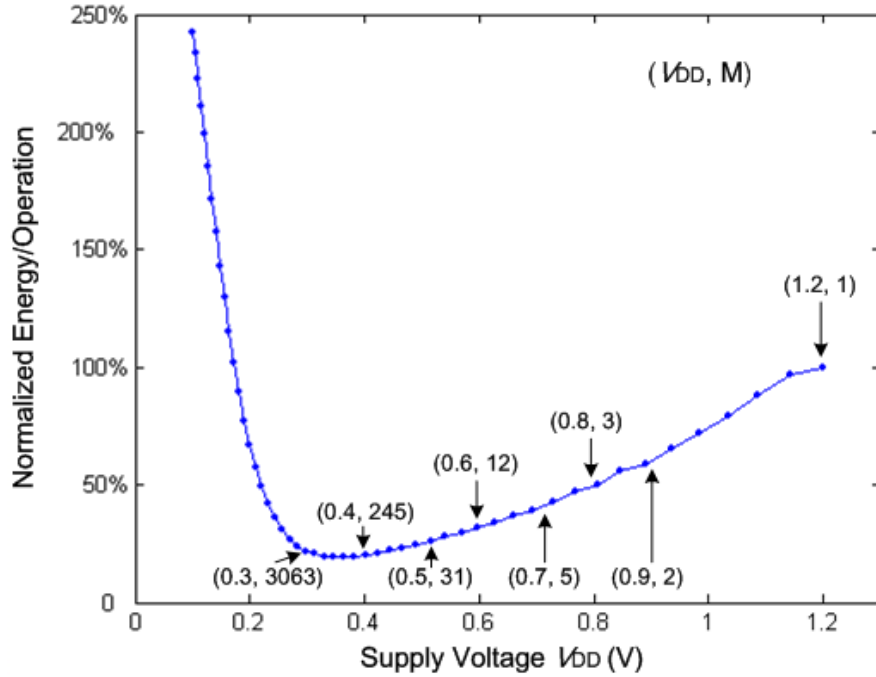


Figure 2.7: Normalized EPO at different V_{DD} for the same throughput

2.4 Noise Margin Estimation for Sub-threshold Combinational Circuits

In a digital circuit, the noise margin is the amount by which the signal exceeds the threshold for a proper “0” or “1”. The theoretical definition of noise

2.4. NOISE MARGIN ESTIMATION FOR SUB-THRESHOLD COMBINATIONAL CIRCUITS

margin is $V_{IL}-V_{OL}$ and $V_{OH}-V_{IH}$. The V_{OL} , V_{OH} are output of stage M , the V_{IL} , V_{IH} are input of stage $M + 1$. In industry, while this definition is widely used in memory design [23]-[30], a more strict definition is adopted for circuits based on standard cells. That is, a digital circuit's output might be designed to swing between 0.0 and V_{DD} , with anything below V_{OLmax} , e.g. $10\%V_{DD}$, considered a "0", and anything above V_{OHmin} , e.g. $90\%V_{DD}$ considered a "1". In this case, the noise margin for a "0" would be the amount that a signal is below $10\%V_{DD}$, and the noise margin for a "1" would be the amount by which a signal exceeds $90\%V_{DD}$. This definition guarantees the output's "0" or "1" intervals to be as large as possible, regardless of V_{IL} and V_{IH} of the next cascaded circuit. This definition helps achieve very high robustness and insensitivity to noise disturbances. It is used throughout this thesis.

When designs are moving from the super-threshold to the sub-threshold domain, the effective-to-idle current ratio (I_{eff}/I_{idle}) diminishes rapidly. Accordingly, the interval between V_{OH} and V_{OL} , is reduced. This may lead to a failure of the decoding logic values. Manufacturing variability further worsens circuit robustness. Therefore, guaranteeing sufficient output noise margins becomes a unique and important issue for sub-threshold designs. Targeting at a fixed V_{DD} , prior art [19] -[22] relies on device sizing as a means of ensuring enough noise margins for individual cells. This is because larger devices reduce the V_T mismatch [47]. This methodology neglects correlations between gates and results in a pessimistic estimation of the output's noise margin. For instance, a gate that outputs higher V_{OL} (lower V_{OH}) can tolerate higher V_{IL} (lower V_{IH}) from its preceding gate. Ignor-

2.4. NOISE MARGIN ESTIMATION FOR SUB-THRESHOLD COMBINATIONAL CIRCUITS

ing inter-cell correlations results in an overestimation of the minimum V_{DD} and device sizes, thus an increase of power/energy consumption. It would be also convenient for the designers to know the minimum functional V_{DD} in the design time. Unfortunately, nowadays this information is obtained only through post-silicon testing, such as the 180mV FFT processor [33] [34] and the 85mV FIR filter [37] , etc.

Theoretically, using Monte-Carlo DC simulations to extract the noise margin can solve these problems. Based on the extracted noise margin information, the designer can improve the robustness of the circuitry by means of gate resizing, buffer insertion, logic restructuring, etc. It also helps to estimate the minimum functional V_{DD} . In this way, the imposed additional area and power overhead are prevented. However, this is at the cost of a much longer design time. Usually, the design flow requires multiple iterations between noise margin extraction and circuit tuning. In our experience, spending tens of hours to extract the noise margins of a benchmark circuit composed of only thousands of logic gates is quite common. Therefore, exploring an approach that can promptly estimate the noise margin, minimum functional V_{DD} and the functional yield for a given circuit, taking into consideration the impact of process variations and inter-cell correlations, is of great importance.

This section introduces a novel noise margin extraction methodology for sub-threshold combinational circuits. Our methodology has the following features. First, instead of performing slow transistor-level DC simulations, we propose a fast gate-level noise margin modeling approach based on a new *equivalent resistance* model. We use curve-fitting to calibrate our model, so that the estimation results can perfectly match the results simulated from

2.4. NOISE MARGIN ESTIMATION FOR SUB-THRESHOLD COMBINATIONAL CIRCUITS

transistor level DC simulations. In analogy to the Elmore delay model for timing analysis, the gate-level model renders reasonably good accuracy, but is computationally much more efficient compared to its transistor-level counterpart. Second, we introduce the *Affine Arithmetic* (AA) approach to symbolically traverse the whole circuit from its inputs to outputs. Applying AA helps to model correlations of noise margins among cells. Besides, as the noise margins of the final outputs are expressed in the affine form, their statistical spread can be extracted. In this way, the minimum functional V_{DD} , as well as the functional yield of a circuit can be estimated. Our approach iterates only once per input vector, hence the running time can be reduced by several orders compared to the MC simulation. Experimental results show that our approach has 98.5% accuracy using MC simulations as a reference, but can reduce the running time by several orders of magnitude.

2.4.1 Estimating gate noise margin with rectified equivalent resistance model

As aforementioned, we first propose a gate-level noise margin model and show how to calibrate it to improve the estimation accuracy. Recall the models of the sub-threshold current for nMOS and pMOS transistors given in Section 2.1,

$$I_{nMOS} = I_{0n} e^{\frac{(V_{GS} - V'_{Tn} + \eta V_{DS} - \gamma V_{SB})}{nU}} (1 - e^{-\frac{V_{DS}}{U}}) \quad (2.16)$$

$$I_{pMOS} = I_{0p} e^{\frac{-(V_{GS} - V'_{Tp} + \eta V_{DS} - \gamma V_{SB})}{nU}} (1 - e^{\frac{V_{DS}}{U}}) \quad (2.17)$$

Conventionally, to model a logic gate, the transistors which are off as perfect switches, and transistors which are on as resistive loads. This model works

2.4. NOISE MARGIN ESTIMATION FOR SUB-THRESHOLD COMBINATIONAL CIRCUITS

perfectly in the super-threshold region where I_{eff}/I_{idle} is quite large. However, in the sub-threshold the I_{eff}/I_{idle} is small so that the off-transistors can no longer be modeled as perfect switches. In other words, the sub-threshold logic becomes a ratioed logic. Since in the super-threshold, a ratioed logic can be modeled by equivalent resistance model. To estimate the noise margin of a cell at gate-level, we extend the *equivalent resistance* model into the DC analysis in the sub-threshold region. The resistance is the derivative of the drain-to-source voltage V_{DS} , with respect to the drain-to-source current, at the DC point $V_{DS} = 0$. Ignoring for the moment body effects, we can approximate the equivalent resistances of nMOS and pMOS transistors as

$$R_{nMOS} = (I_{0n})^{-1} U e^{-(V_{in} - V'_{Tn})/nU} \quad (2.18)$$

$$R_{pMOS} = (I_{0p})^{-1} U e^{(V_{in} - V_{DD} - V'_{Tp})/nU} \quad (2.19)$$

A typical digital cell consists of a p-section with a common node tied to an n-section (see Figure 2.8(a)). Let us start the analysis with a CMOS inverter (Figure 2.8(b)). Its equivalent resistance model is shown in Figure 2.8(c).

Assuming $I_{0n} = I_{0p}$, we can obtain the output voltage of the inverter,

$$V_{out} = \left\{ 1 + e^{[2V_{in} - (V_{DD} + V'_{Tn} + V'_{Tp})]/nU} \right\}^{-1} V_{DD} \quad (2.20)$$

If we define

$$x = (V'_{Tn} + V'_{Tp})/2 \quad (2.21)$$

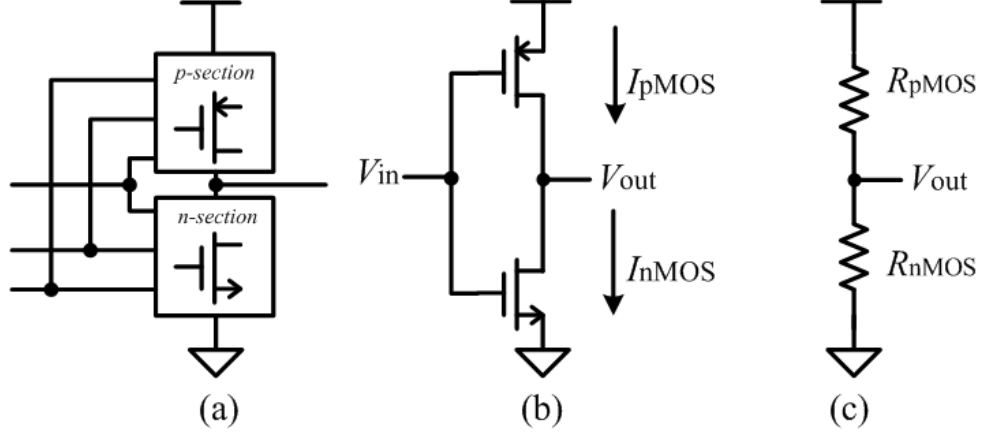


Figure 2.8: (a) Cell schematic (b) Inverter (c) Equivalent model

then Equation (2.20) can be expressed by

$$V_{out} = \left\{ 1 + \left[e^{(V_{in} - x - V_{DD}/2)/nU} \right]^2 \right\}^{-1} V_{DD} \quad (2.22)$$

The above analysis may have lost validity as we neglected the body effect and assumed $I_{0n} = I_{0p}$. To fix the accuracy, we intentionally insert a parameter λ into (2.22) for calibration,

$$V_{out} = \left\{ 1 + \left[e^{\lambda + (V_{in} - x - V_{DD}/2)/nU} \right]^2 \right\}^{-1} V_{DD} \quad (2.23)$$

where λ is a curve-fitting parameter, which can be extracted through *non-linear least square curve-fitting* from actual simulated results. λ varies when nMOS and pMOS transistor sizes change. It also varies with different operating conditions. Figure 2.9 gives the noise margin estimated by the Cadence Spectre Simulator and from Equation (2.23), for an inverter with $W_p/W_n=0.28\mu\text{m}/0.2\mu\text{m}$ in 65nm CMOS process under typical technology

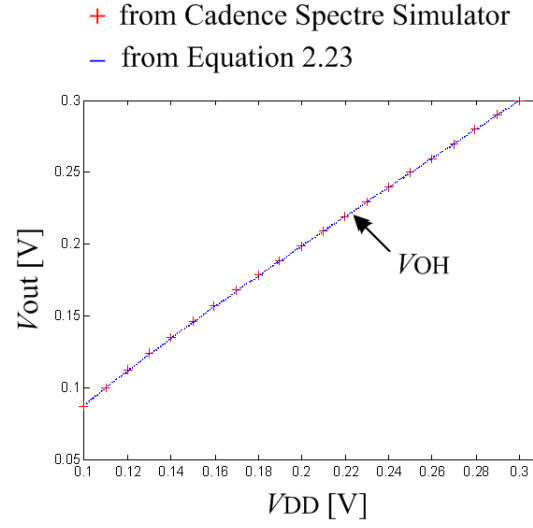
2.4. NOISE MARGIN ESTIMATION FOR SUB-THRESHOLD COMBINATIONAL CIRCUITS

(TT) when V_{DD} is swept in the sub-threshold region. By textbook definition, the V_{OL} and V_{OH} are the two operational points of the inverter where $d(V_{out})/d(V_{in}) = -1$. The V_{OL} and V_{OH} referred by this work are the steady high and low voltage output values, which are slightly different from the textbook definition values (Figure 2.10). Please note that the vertical axes in Figure 2.9 have different scales for each plot. As shown, both results perfectly match each other after curve-fitting.

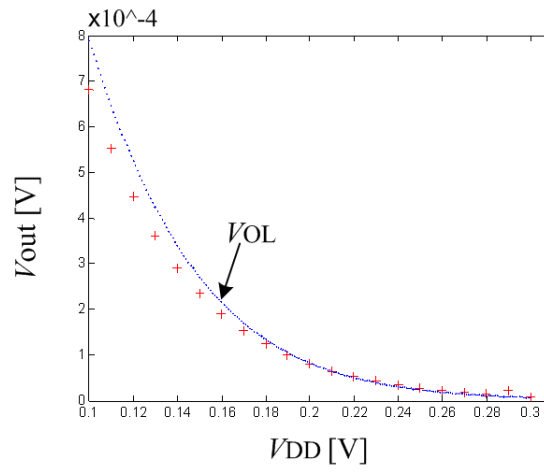
Next, we show how to incorporate process variations in our model. It is already shown in Section 2.1 that V_T variation is the dominant malefactor for the sub-threshold noise margin due to its exponential correlation with the sub-threshold current. The V_T mismatch of paired transistors also causes a wide range of sub-threshold current shifts [41]. In our model, the V_T variation is reflected on the variation of x . As V_{Tn} and V_{Tp} are normally distributed, x is also normally distributed, i.e., $x \sim N(\mu_x, \sigma_x^2)$. Parameters μ_x and σ_x are primarily dependent on the size of the transistors, and can also be characterized through transistor level simulations. Figure 2.11 shows the 3σ range of V_{OL} and V_{OH} obtained from Cadence Spectre Simulator and from our model. Once again, the results simulated from the transistor level model and our new model perfectly coincide. An observation from the two plots is that the variation of V_{OH} is much larger than that of V_{OL} due to the fact that the nMOS transistor can be much leakier than the pMOS transistor.

A similar analysis can be carried out for other static digital gates. For an N -input gate, we found that its output voltage can be approximately

2.4. NOISE MARGIN ESTIMATION FOR SUB-THRESHOLD COMBINATIONAL CIRCUITS



(a) V_{OH}



(b) V_{OL}

Figure 2.9: Noise margin generated from Spectre Simulator vs from Equation 2.23

2.4. NOISE MARGIN ESTIMATION FOR SUB-THRESHOLD COMBINATIONAL CIRCUITS

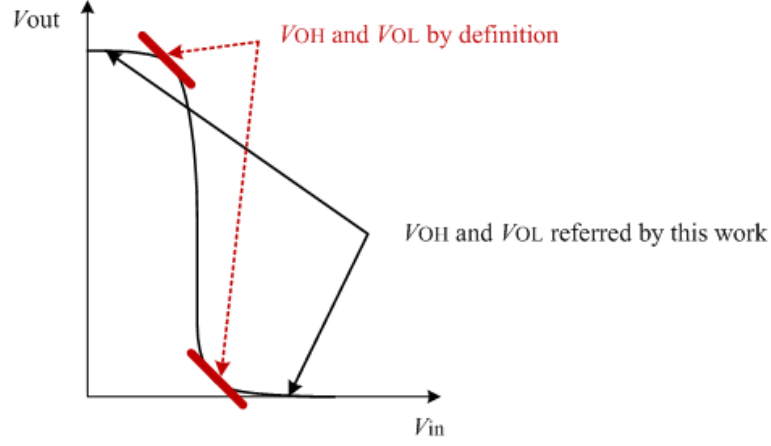


Figure 2.10: Noise margin by definition and by this work

expressed as a function,

$$V_{out} = f(V_{in}, X, V_{DD}) \quad (2.24)$$

where V_{in} denotes the set of N inputs' voltages, and X is the set that contains N normally distributed variables corresponding to the different inputs. For example, the output voltages of an N -input NAND and an N -input NOR gate can be expressed in Equations (2.25) and (2.26) , respectively,

$$V_{out,nand} = \left\{ 1 + \left[\sum_{i=1}^N \left[e^{\lambda i - (V_{in_i} - X_i - V_{DD}/2)/nU} \right] \right]^{-2} \right\}^{-1} V_{DD} \quad (2.25)$$

$$V_{out,nor} = \left\{ 1 + \left[\sum_{i=1}^N \left[e^{\lambda i + (V_{in_i} - X_i - V_{DD}/2)/nU} \right] \right]^2 \right\}^{-1} V_{DD} \quad (2.26)$$

where V_{in_i} is the voltage of the i^{th} input and $V_{in_i} \in V_{in}$, X_i relates to the V_T values of a pair of nMOS and pMOS transistors which have the same

2.4. NOISE MARGIN ESTIMATION FOR SUB-THRESHOLD COMBINATIONAL CIRCUITS

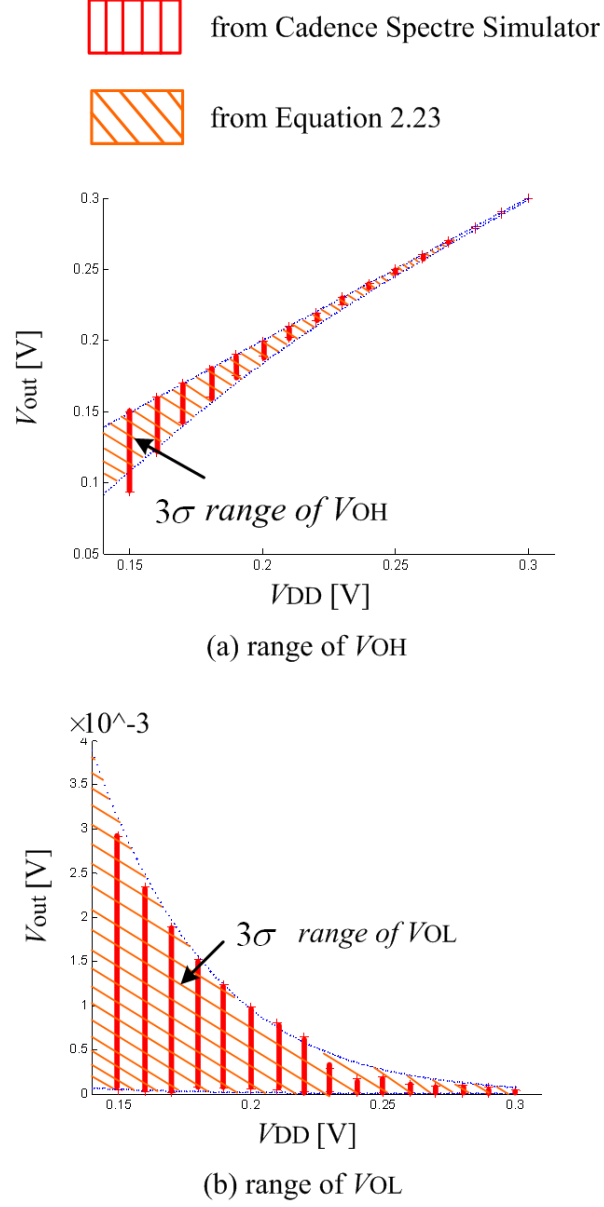


Figure 2.11: 3σ range of noise margin generated from Spectre Simulator vs from Equation 2.23

2.4. NOISE MARGIN ESTIMATION FOR SUB-THRESHOLD COMBINATIONAL CIRCUITS

input, and $X_i \in X$, and $X_i \sim N(\mu_{x_i}, \sigma_{x_i}^2)$. λ_i is the i^{th} fitted parameter. The noise margin model for each type of gate, including the pre-characterized constants $\mu_{x_i}, \sigma_{x_i}, \lambda_i(\forall i)$, can be embedded in a library file of the EDA tool.

Estimating the cell's noise margin with its equivalent resistance model renders reasonably good accuracy, and provides a much simpler expression when compared to the transistor-level model. The new noise margin model performs well at the gate-level, and avoids the need for solving a transistor-level matrix, hence tremendously reduces the computation intensity for the EDA software. However, if the statistical noise margins at the outputs are to be extracted, Monte-Carlo DC analysis is still needed. To totally eliminate using Monte-Carlo simulations, we introduce the *Affine Arithmetic* model for efficient computation and propagation of noise margins.

2.4.2 Estimating statistical output noise margin with affine arithmetic model

The Affine Arithmetic (AA) model is used for example in bit-width estimation and probabilistic error analysis ([42] -[45]). In the AA model, an uncertain variable x is expressed as

$$x = C_0 + \sum_i^N C_i \varepsilon_i \quad (2.27)$$

where C_0 is the *central value* of the affine form of x , ε_i is an independent *noise symbol* multiplied by its corresponding *coefficient* C_i . All *noise symbols* denote independent and identically-distributed variables. AA is very suitable for symbolic propagation. This is because if the operands are in AA form, the

2.4. NOISE MARGIN ESTIMATION FOR SUB-THRESHOLD COMBINATIONAL CIRCUITS

results of the arithmetic operations, such as addition, subtraction, multiplication, are also in AA form. Furthermore, AA is capable of carrying correlation information. Along a propagation path, one *noise symbol* ϵ_i may contribute to the uncertainties of two or more variables. When these variables are combined, the uncertainties may also be combined so that their correlations are taken into consideration. This property is especially useful for our case. As shown in Figure 2.12, the variation term ϵ_i in the noise margin expression at the output of INV1, will re-converge at the inputs of NAND1, and will proceed to the output of NAND1. In this way, the final results can be more accurately estimated.

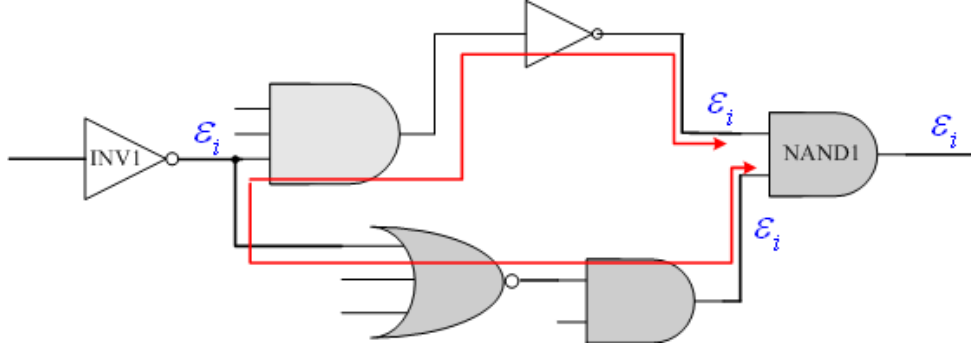


Figure 2.12: Noise margin uncertainty propagation with AA model

Figure 2.13 shows the statistical noise margin estimation flowchart of this work. The new approach takes 3 steps:

1. *Model Instantiation and AA form Initialization*

Given the synthesized gate-level netlist, we instantiate each gate with the noise margin model described in Section 2.4.1. Each parameter in X is

2.4. NOISE MARGIN ESTIMATION FOR SUB-THRESHOLD COMBINATIONAL CIRCUITS

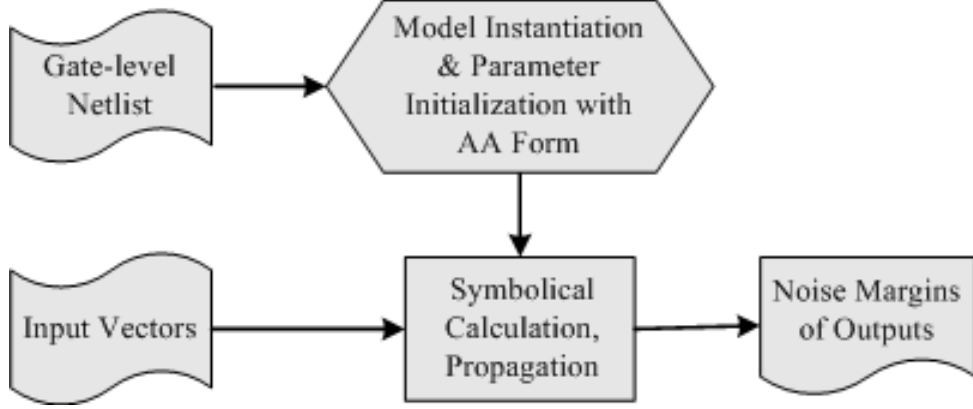


Figure 2.13: Noise margin estimation flowchart

initialized and stored in *AA* form, i.e.

$$X_{i,k} = X_{0i,k} + C_{i,k}\varepsilon_{i,k} \quad (2.28)$$

where $X_{i,k}$ denotes the i^{th} variable in the set X of the k^{th} gate. $\varepsilon_{i,k}$ is a unique and independent noise symbol associated with that variable and $\varepsilon_{i,k} \sim N(0, 1)$.

2. Symbolical Calculation and Propagation

For each input-vector, the program traverses the whole circuit from the inputs to the outputs in the *forward* direction, such that the voltage of each edge in the graph is annotated with a calculation result expressed in *AA* form. However, symbolic propagation would cause a range explosion when encountering special functions such as *exponential* and/or *power functions*, resulting in difficulty to maintain *AA* propagation. We solve this problem by approximating (2.24) linearly using a first order Taylor expansion, so that the

2.4. NOISE MARGIN ESTIMATION FOR SUB-THRESHOLD COMBINATIONAL CIRCUITS

output voltage of each gate is expressed as

$$V_{out} = V_{out,0} + \sum_{\forall i} [\partial f / \partial V_{in_i}]_0 \Delta V_{in_i} + \sum_{\forall i} [\partial f / \partial X_i]_0 \Delta X_i \quad (2.29)$$

where $V_{out,0}, [\partial f / \partial V_{in_i}]_0, [\partial f / \partial X_i]_0$ are the values calculated at the *central values* of the variables in the V_{in} and X sets of that gate.

It is reasonable to ask whether the approximation of function f by first-order Taylor expansion is valid. [67] had proven that, when process parameters (in our case the ΔV_{in_i} and ΔX_i) have relatively small variations, the first-order Taylor expansion is adequate and the approximation is acceptable with little loss of accuracy. This is generally true of intra-die variations, where the process parameter variations are relatively small in comparison with the nominal values. For this reason, first-order Taylor expansion is now widely used in timing, leakage and other analysis under process variations in modern EDA tools.

3. Output Noise Margin Estimation

After calculation and propagation, the voltage at the output (s) of a circuit can be expressed as (2.30) ,

$$V_{output} = V_{output,0} + \sum_{\forall (i,k)} \eta_{i,k} \varepsilon_{i,k} \quad (2.30)$$

Recall that each $\varepsilon_{i,k}$ in (2.30) is an independent *noise symbol* and $\varepsilon_{i,k} \sim N(0, 1)$. $\eta_{i,k}$ is the corresponding accumulated coefficient. According to probability theory, the sum of these independent normally distributed terms is also

2.4. NOISE MARGIN ESTIMATION FOR SUB-THRESHOLD COMBINATIONAL CIRCUITS

Table 2.2: Estimated statistical noise margin from Cadence Spectre Monte-Carlo DC simulation and the new approach

Bench -mark	Sim	150mV		180mV		210mV		RunningTime /InputVector
		V_{OL}'	V_{OH}'	V_{OL}'	V_{OH}'	V_{OL}'	V_{OH}'	
C880	MC	2.4%	84.6%	1.2%	92.2%	0.3%	96.2%	> 10 hours
	New	2.9%	85.4%	1.1%	93.7%	0.4%	97.4%	0.08sec

normally distributed, so we can have

$$V_{output} \sim N(V_{output,0}, \sum_{\forall(i,k)} \eta_{i,k}^2) \quad (2.31)$$

Therefore, the mean value and variance of the output voltage can be easily obtained such that the statistical output noise margin can be estimated.

2.4.3 Experimental results

To prove the strength of our methodology, experiments have been conducted using the ISCAS combinational benchmark circuits. All simulations were performed for a CMOS 65nm Standard V_T (SV_T) technology from NXP. The benchmark circuits are synthesized to netlists with minimum size logic gates. We do not use gates that have more than 4 stacked transistors or 4 paralleled transistors, as sub-threshold design seldom exploits these gates due to severe robustness degradation [please refer to Chapter 3]. Our new approach was implemented in C++, and ran on a PC with Intel Pentium 1.86GHz and 1G RAM. To validate the new model, we performed transistor-level DC Monte-Carlo simulations for benchmark C880, and compared the results with those

2.4. NOISE MARGIN ESTIMATION FOR SUB-THRESHOLD COMBINATIONAL CIRCUITS

from our approach. The MC simulation was carried out with Cadence Spectre Simulator running on a HP UNIX server. The simulations ran for 2000 trials. Table 2.2 gives the simulation results. Here, V_{OL}' (V_{OH}') is defined as the maximum (minimum) value among all the outputs' 3σ values of V_{OL} (V_{OH}), normalized w.r.t. V_{DD} . As shown, our approach can predict the output noise margin with less than 1.5% deviation. However, the transistor-level DC MC simulation for benchmark C880 required more than 10 hours running time for one input vector, while the new approach only needed about 0.1 seconds! Our methodology reduces the design time for the output noise margin of a circuit by several orders of magnitude.

Table 2.3: Estimated statistical noise margins as % of V_{DD}

Bench -mark		150mV		180mV		210mV		RunningTime (sec)
		V_{OL}'	V_{OH}'	V_{OL}'	V_{OH}'	V_{OL}'	V_{OH}'	
C1355	3σ	2.5%	85.0%	1.8%	93.7%	0.3%	97.4%	0.172
	6σ	4.1%	73.2%	2.6%	88.8%	0.68%	95.4%	
C1908	3σ	2.4%	78.3%	1.7%	92.6%	0.4%	97.2%	0.204
	6σ	4.3%	61.1%	2.3%	86.8%	0.7%	95.0%	
C2670	3σ	3.0%	83.3%	1.2%	91.3%	0.4%	97.4%	0.484
	6σ	8.0%	70.1%	2.0%	86.7%	0.73%	95.0%	
C3540	3σ	3.4%	85.1%	1.1%	91.8%	0.4%	97.4%	0.688
	6σ	6.2%	73.3%	1.95%	88.4%	0.68%	95.4%	
C5315	3σ	3.5%	77.2%	1.1%	92.6%	0.4%	97.2%	1.203
	6σ	6.4%	59.4%	1.95%	88.9%	0.73%	95.1%	
C6288	3σ	7.1%	78.9%	2.4%	92.7%	0.8%	97.2%	1.422
	6σ	13.0%	62.2%	4.38%	86.9%	1.63%	95.0%	
C7552	3σ	2.7%	78.4%	1.1%	92.7%	0.4%	97.4%	1.781
	6σ	4.8%	61.2%	2.1%	86.8%	0.74%	95.1%	

2.4. NOISE MARGIN ESTIMATION FOR SUB-THRESHOLD COMBINATIONAL CIRCUITS

Table 2.3 gives the 3σ and 6σ statistical noise margins simulated with our methodology for the remaining ISCAS benchmarks. If targeting at ensuring sufficient noise margin ($V_{OH} > 90\%V_{DD}$ and $V_{OL} < 10\%V_{DD}$) for each individual gate, the required minimum V_{DD} is 220mV. However, observe that at $V_{DD}=180\text{mV}$, there is also enough 3σ noise margin ($V_{OH} > 90\%V_{DD}$ and $V_{OL} < 10\%V_{DD}$) for every output. The overestimation of minimum functional voltage V_{DD} is thus avoided, as the new approach can precisely estimate the output noise margins.

Based on the spread of noise margin, we are now able to estimate the circuit's functional yield for a given V_{DD} . Let us take benchmark C880 at $V_{DD} = 180\text{mV}$ as an example. Its V_{OH} and V_{OL} probability density function (pdf) plots, which are generated with the μ and σ estimated by our program, are shown in Figure 2.14. By intersecting a 90% V_{DD} line for V_{OH} and a 10% V_{DD} line for V_{OL} , the desired V_{OH} and V_{OL} ranges (shadow region) are obtained. Suppose p_1 and p_2 are the cumulative density within the acceptable ranges and neglecting the dependency between V_{OL} and V_{OH} , $p=p_1p_2$ is an estimation for the functional yield. In this example, p is 99.8%. This represents a 2000 ppm loss arising from malfunctioning of combinational circuits, excluding the timing yield loss. Obviously, it is very high for industrial design standards.

An interesting observation is that, the noise margin problem only happens when V_{DD} scales into very deep sub-threshold region, i.e. $V_{DD} < 250\text{mV}$ for circuit in 65nm SV_T CMOS. As pointed out in section 2.2, normally operating a circuit with such a low V_{DD} is not necessary as this V_{DD} has already fallen below the energy optimal supply voltage V_{opt} . Only for a

2.4. NOISE MARGIN ESTIMATION FOR SUB-THRESHOLD COMBINATIONAL CIRCUITS

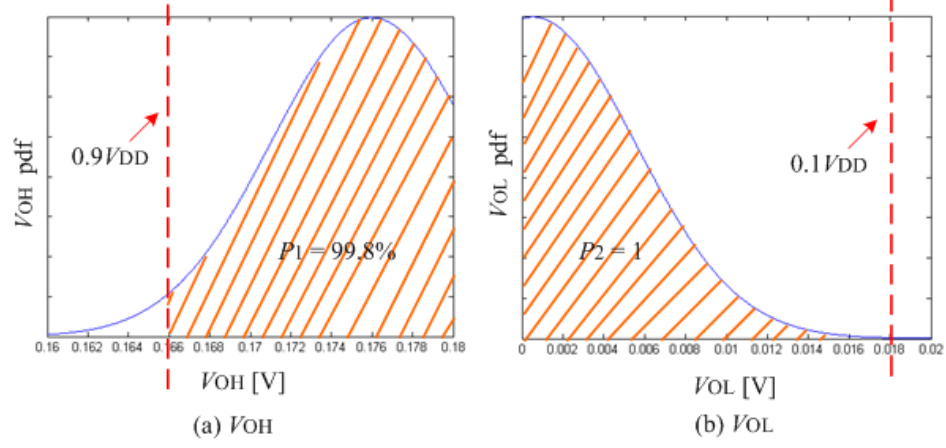


Figure 2.14: Probability density function (pdf) plots for benchmark C880 at $V_{DD} = 180\text{mV}$

circuit with extremely high switching density, there is a need to go to very deep sub-threshold. However, it is still very handy for designers to know quickly the lowest V_{DD} limitation imposed by the noise margin constraint for their sub-threshold design at the design time.

Chapter 3

Physical Level Effort

In this work we make effort on physical level to mitigate the impact of process variations on the near/sub-threshold design. First, we propose a novel configurable V_T balancer, which helps improving both the functional yield and timing yield by balancing the V_T of pMOS and nMOS transistors. In addition, some other approaches including transistor sizing, exploiting parallel transistor V_T mismatch to improve drivability, selecting reliable library cells for logic synthesis, turning risky ratioed logic into non-ratioed logic, and level shifter design, are also discussed in this chapter.

3.1 Adaptive V_T for Process Spread Control in Sub/Near Threshold

To perform standard cell based logic synthesis for sub-threshold design, traditional digital cells that are optimized in the super-threshold region need to be revised. Figure 3.1 (a) illustrates a typical digital cell consisting of a

3.1. ADAPTIVE V_T FOR PROCESS SPREAD CONTROL IN SUB/NEAR THRESHOLD

p-section with a common node tied to an n-section. We start analyzing the standard library from a CMOS inverter (see Figure 3.1 (b)).

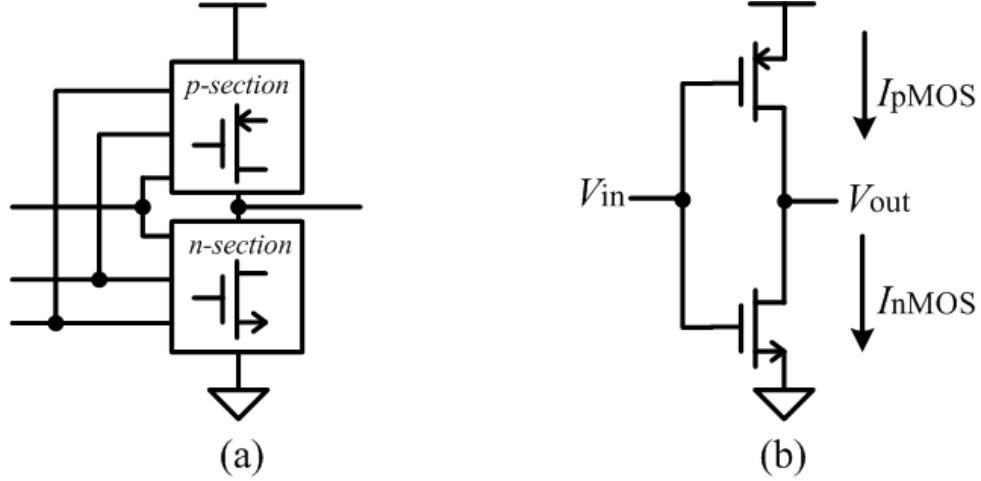


Figure 3.1: (a) n and p sections (b) CMOS inverter

To function correctly with enough noise margin, the gate must have sufficient high V_{OH} ($>\alpha V_{DD}$) for pull-up operation and sufficient low V_{OL} ($<\beta V_{DD}$) for pull-down operation. α and β are arbitrary limit parameters. Typical values for α and β are 0.9 and 0.1. Since in the sub-threshold region the effective drive current I_{eff} (the effective leakage, also known as active leakage) and idle current I_{idle} (the idle leakage) are comparable, the gate becomes ratioed logic, which demands careful device sizing. Process variations further magnify the design difficulty. For example, at the fast nMOS slow pMOS corner (FNSP) where the nMOS network is much leakier than the pMOS network, the pMOS network must be upsized to a large extent to guarantee a sufficiently high V_{OH} . However, doing so will result in insufficiently low V_{OL} when facing the fast pMOS slow nMOS corner (SNFP). A

3.1. ADAPTIVE V_T FOR PROCESS SPREAD CONTROL IN SUB/NEAR THRESHOLD

way to cope with unbalanced process corners is to increase the gate's supply voltage to increase I_{eff}/I_{idle} . A quantitative analysis on the minimum supply voltage of an inverter is as follows. For simplicity, we assume ΔV_{Tp} and ΔV_{Tn} to be the V_T variations for pMOS and nMOS transistors due to body-effect and process variation. During the pull-up operation, the effective current of the pMOS transistor gradually decreases and the idle current of the nMOS transistor gradually increases because of the DIBL effect. To have the output loading capacitor still get charged at V_{OH} , we need

$$I_{eff,pMOS} > I_{idle,nMOS} \quad \text{when } V_{out} = \alpha V_{DD} \quad (3.1)$$

$$I_{eff,pMOS} = I_{0p} e^{\frac{V_{DD} + (V'_{Tp} + \Delta V_{Tp}) - \eta(\alpha-1)V_{DD}}{nU}} (1 - e^{\frac{(\alpha-1)V_{DD}}{U}}) \quad (3.2)$$

$$I_{idle,nMOS} = I_{0n} e^{\frac{-(V'_{Tn} + \Delta V_{Tn}) + \eta\alpha V_{DD}}{nU}} (1 - e^{\frac{-\alpha V_{DD}}{U}}) \quad (3.3)$$

Similarly, for the pull-down operation, we need

$$I_{eff,nMOS} > I_{idle,pMOS} \quad \text{when } V_{out} = \beta V_{DD} \quad (3.4)$$

$$I_{eff,nMOS} = I_{0n} e^{\frac{V_{DD} - (V'_{Tn} + \Delta V_{Tn}) + \eta\beta V_{DD}}{nU}} (1 - e^{\frac{-\beta V_{DD}}{U}}) \quad (3.5)$$

$$I_{idle,pMOS} = I_{0p} e^{\frac{V'_{Tp} + \Delta V_{Tp} - \eta(\beta-1)V_{DD}}{nU}} (1 - e^{\frac{(\beta-1)V_{DD}}{U}}) \quad (3.6)$$

And usually, we set

3.1. ADAPTIVE V_T FOR PROCESS SPREAD CONTROL IN SUB/NEAR THRESHOLD

$$\alpha + \beta = 1 \quad (3.7)$$

$$I_{0n} = I_{0p} \quad (3.8)$$

The supply voltage V_{DD} is then solved as,

$$V_{DD} \geq \frac{knU + \left| (V'_{Tp} + \Delta V_{Tp}) + (V'_{Tn} + \Delta V_{Tn}) \right|}{1 + \eta(\beta - \alpha)} \quad (3.9)$$

where k is defined by,

$$k = \ln\left(\frac{1 - e^{-\alpha V_{DD}/U}}{1 - e^{-\beta V_{DD}/U}}\right) \quad (3.10)$$

Equation (3.9) is a non-linear equation; it is impossible to solve it analytically. However, for given α and β , k almost remains constant when V_{DD} is swept in the sub-threshold region (see Figure 3.2). Therefore, from Equation (3.9) it follows that the minimal supply voltage exists around the point where the threshold voltages of pMOS and nMOS transistors are balanced, i.e. $(V'_{Tn} + \Delta V_{Tn}) = -(V'_{Tp} + \Delta V_{Tp})$. As the imbalance between the pMOS and nMOS threshold voltages increases, it is inevitable to increase the supply voltage to guarantee correct functioning of the logic gates, hence more power/energy will be consumed.

The above analysis has clarified the importance of balancing V_T of pMOS and nMOS transistors. Since both V_T of pMOS and nMOS are controlled by separate doping process, their V_T can vary significantly with respect to each other. We proposed to use the transistor's body effect to tune transistor

3.1. ADAPTIVE V_T FOR PROCESS SPREAD CONTROL IN SUB/NEAR THRESHOLD

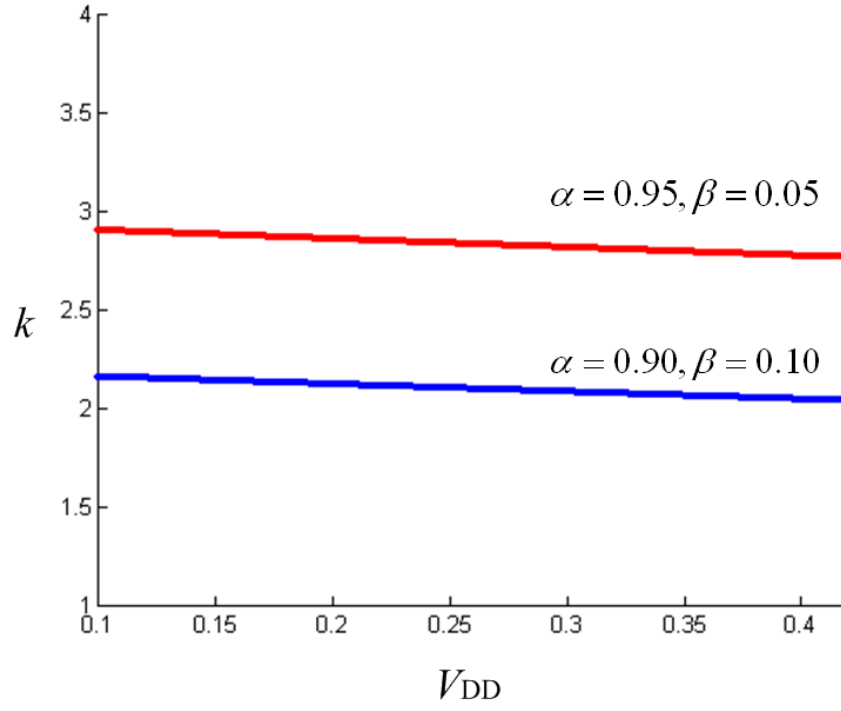


Figure 3.2: k versus V_{DD}

3.1. ADAPTIVE V_T FOR PROCESS SPREAD CONTROL IN SUB/NEAR THRESHOLD

threshold voltage through body biasing to tighten the distribution. Figure 3.3 shows the principle of threshold voltage tuning. Note that bias tuning requires triple-well process, which is optional for 90nm CMOS process but compulsory for 65nm and lower process. In Figure 3.3, V_{BBP} is the body bias voltage for the pMOS transistor, and V_{BBN} voltage is the body bias for the nMOS transistor.

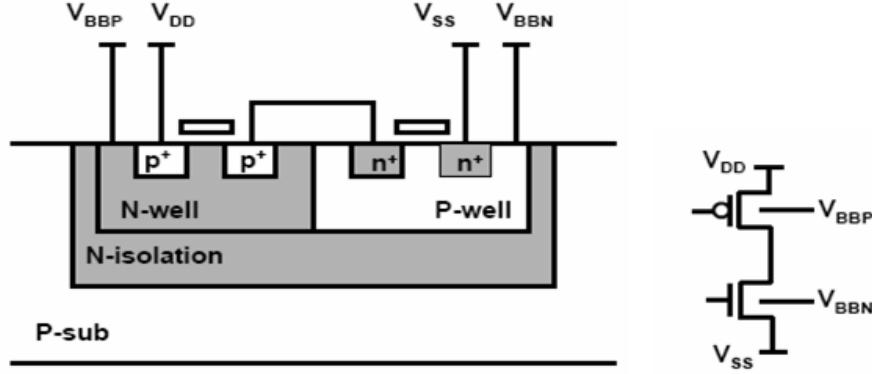


Figure 3.3: Transistor threshold tuning of an inverter through bulk-biasing

The formula for the transistor threshold voltage is given as

$$V_T = V_{T0} + \gamma(\sqrt{2\phi_B - V_{BB}} - \sqrt{2\phi_B}) \quad (3.11)$$

where V_{T0} is a process intrinsic parameter for zero substrate bias, γ denotes the body effect coefficient. The value of γ typically lies in the range from 0.3 to 0.4 $V^{1/2}$. $2\phi_B$ is the surface potential.

[46] has presented a pMOS/nMOS V_T balancing method, which is composed of a logical threshold detector, a reference supply, a comparator, a shift-register and a resistor-based nMOS bias generator. This approach, while

3.1. ADAPTIVE V_T FOR PROCESS SPREAD CONTROL IN SUB/NEAR THRESHOLD

accurate, introduces considerable area and power overheads. In addition, its large area overhead prevents it from being distributed on a die, hence it fails to take into account the ever increasing intra-die variations.

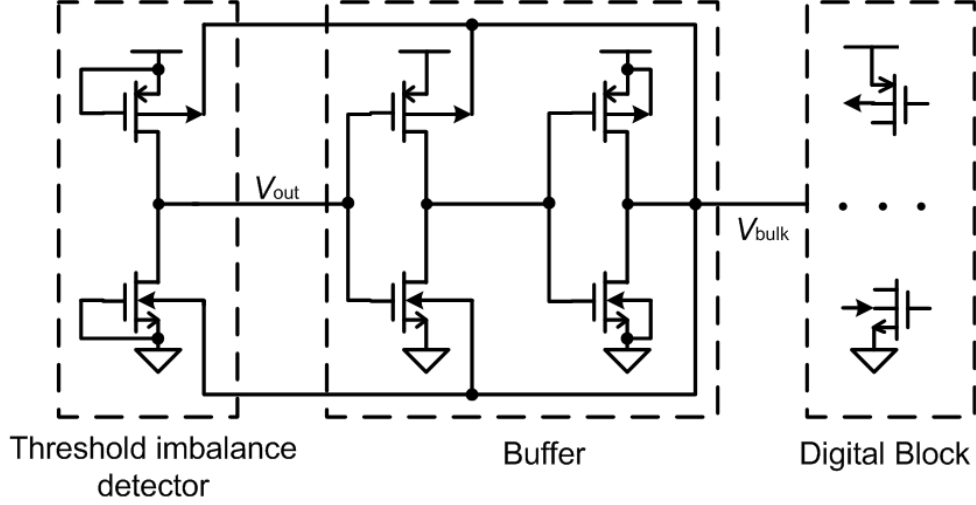


Figure 3.4: The proposed V_T balancing scheme with only one bulk-control line

In our work, we propose a simple V_T balancing scheme, which is shown in Figure 3.4. A CMOS inverter, whose pMOS and nMOS transistors are off, functions as a process-corner V_T imbalance detector. In contrast to previous works where separate bulk controlling lines for pMOS and nMOS transistors are used, our approach makes use of only one controlling line. This is possible because the bulk controlling line is never higher than $|V_T|$ preventing in this way the junction diodes from turning on. V_{out} and V_{bulk} are designed in advance as $V_{DD}/2$ in the typical process corner (TT). V_{out} fluctuates with the variations of process and temperature. A buffer detects and amplifies the swing of V_{out} . The buffer's output V_{bulk} , which supplies the bulk voltage for the logic gates, is fed back to the bulk of the threshold balancing detector to

3.1. ADAPTIVE V_T FOR PROCESS SPREAD CONTROL IN SUB/NEAR THRESHOLD

force pMOS/nMOS V_T balancing. For instance, if the nMOS is leakier than the pMOS, V_{out} will go down, triggering a much larger drop on V_{bulk} . This drop will make the nMOS increase its V_T and the pMOS decrease its V_T , so the process-corner V_T imbalance can be mitigated.

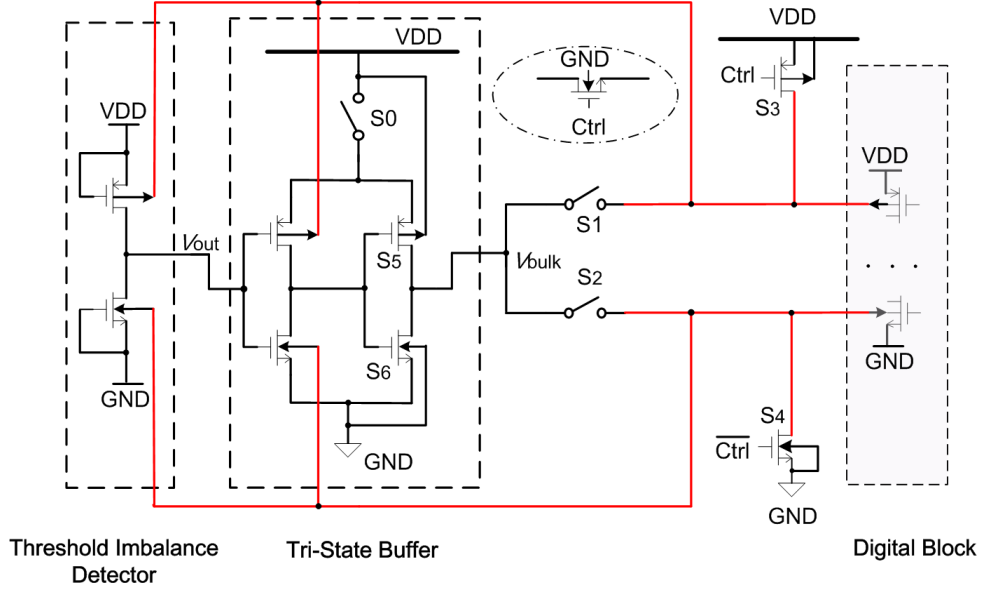


Figure 3.5: Proposed configurable V_T balancer

To support ultra-wide voltage scaling from the nominal supply to sub-threshold, a configurable V_T balancer has been designed, as shown in Figure 3.5. When the processor is set in super-threshold mode, S_0 is off so the tri-state buffer is configured to be in high impedance state. The power switch transistors S_3 and S_4 are on, and S_1 , S_2 are off, so the bulk of pMOS transistors is connected to V_{DD} , and the bulk of nMOS transistors is connected to G_{ND} . When the processor is configured to be in the sub/near threshold mode, S_0 is on so the tri-state buffer starts to function. In this mode, S_1 , S_2 are on, and

3.1. ADAPTIVE V_T FOR PROCESS SPREAD CONTROL IN SUB/NEAR THRESHOLD

S_3, S_4 are off. Therefore, the buffer's output voltage passes through S_1, S_2 to supply the bulk of the logic gates.

The design of the power switch transistor S_0, S_1 and S_2 should be careful as their equivalent on-resistance R_{on} must be small enough to avoid large voltage drop across the transistors. Small R_{on} also improves the configuration setup time. However, in the sub/near threshold mode, the R_{on} of a transistor becomes hundreds times bigger than in the super-threshold mode. As a result, if pMOS transistors are used as the switches, they must be upsized largely, which would introduce a huge amount of area as well as configuration energy overhead. Instead, we use smaller nMOS transistors with their gate voltage boosted. As nMOS transistor has better on-current characteristic than pMOS transistor and the boosted gate voltage over-drives the transistor, the R_{on} and transistor area can be greatly reduced. The potential drop across a transistor is also avoided. In our design, the boosted gate voltage is obtained from another high voltage domain.

We use a metric $\zeta = (V_{out} - V_{DD}/2)/V_{DD}$ to represent the V_T imbalance. V_{out} is the output voltage of the threshold imbalance detector, as indicated in Figure 3.5. In fact, ζ depicts how far V_{out} deviates from $V_{DD}/2$ due to unbalanced V_T devices. Obviously, the larger ζ is, the larger the V_T imbalance is. Figure 3.6 shows the simulated 3σ range of ζ , with and without our V_T balancing scheme. As can be seen, the imbalance between V_T of pMOS and nMOS transistors is confined to a much tighter range after V_T balancing. More importantly, since our scheme's overhead is negligible, we may copy this circuit to different blocks across the whole die to reduce intra-die variations.

3.1. ADAPTIVE V_T FOR PROCESS SPREAD CONTROL IN SUB/NEAR THRESHOLD

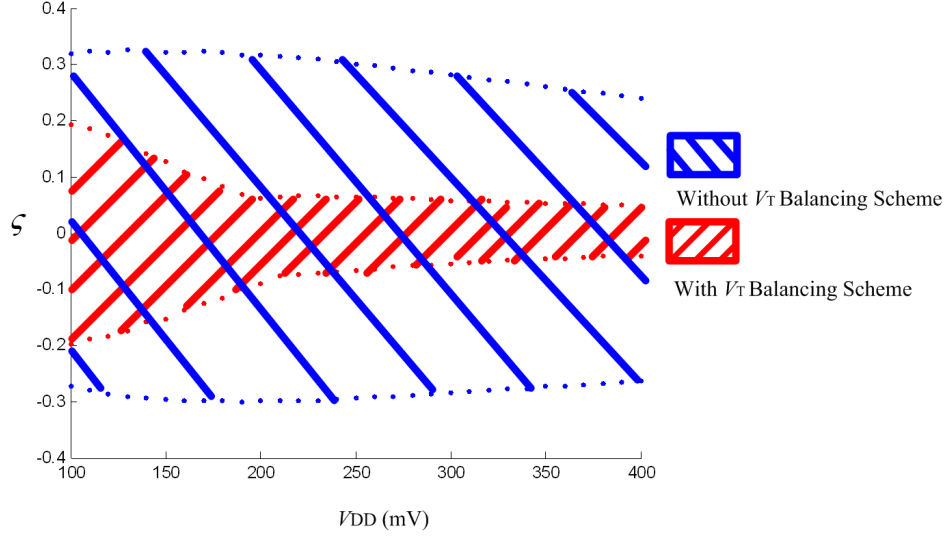


Figure 3.6: Simulated 3σ range of ζ (with and without our V_T balancing scheme)

Table 3.1: Minimum supply voltage for an inverter in 65nm CMOS

Process	with V_T balancing	w/o V_T balancing
Typical	96 mV	96 mV
slow nMOS, slow pMOS	98 mV	100 mV
fast nMOS, fast pMOS	105 mV	120 mV
slow nMOS, fast pMOS	98 mV	100 mV
fast nMOS, slow pMOS	120 mV	140 mV

Table 3.1 shows the simulated minimum functional V_{DD} for an inverter with $Wp/Wn=0.28\mu\text{m}/0.20\mu\text{m}$ in the CMOS 65nm LP-SV $_T$ process technology at different process corners. In the simulation, we assume room temperature, $\alpha=0.9$, $\beta=0.1$. It is clear that the minimal V_{DD} with the V_T balancing scheme is lower compared to that without the V_T balancing scheme.

Figure 3.7 shows the Monte-Carlo simulated propagation delay for an

3.2. GATE SIZING CONSIDERING V_T MISMATCH IN DEEP SUB-THRESHOLD

inverter with aspect ratio of $W_p/W_n=1.1\mu\text{m}/0.40\mu\text{m}$ to drive a capacitive load of 5fF at $V_{DD}=400\text{mV}$ in the CMOS 65nm LP-SV_T process technology. Compared to the conventional design, the standard deviation σ is reduced by $4.7\times$ and the σ/μ is reduced by $3.6\times$ when the proposed configurable V_T balancer is used.

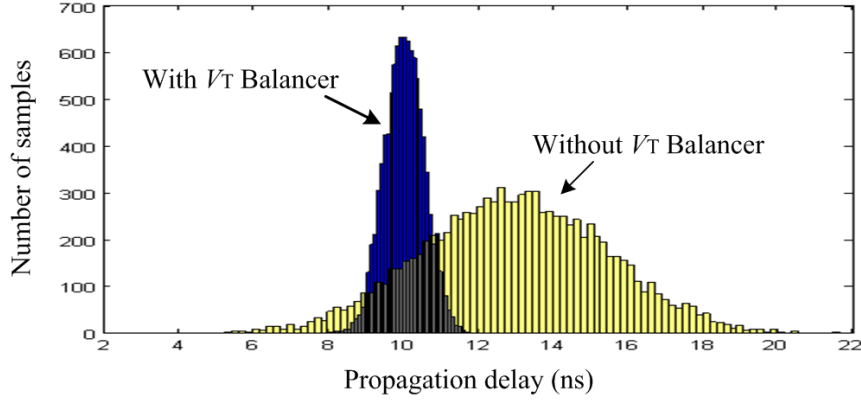


Figure 3.7: Propagation delay for an inverter in 65nm CMOS from Monte-Carlo simulation (with and without our V_T balancing scheme)

3.2 Gate Sizing Considering V_T Mismatch in Deep Sub-threshold

Once the V_T of the pMOS and nMOS transistors are balanced, sizing an inverter in the deep sub-threshold region is not difficult. Unfortunately, for gates with paralleled/stacked topologies, such as NAND, NOR, NXOR, sizing is still non-trivial. This is due to V_T mismatch of paired transistors, as will be discussed in this section.

The intra-die V_T variation of a single transistor has been modeled in [47]

3.2. GATE SIZING CONSIDERING V_T MISMATCH IN DEEP SUB-THRESHOLD

by

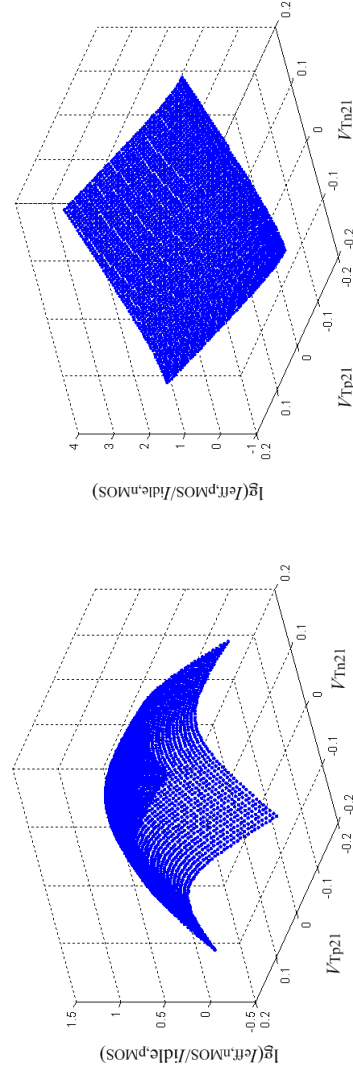
$$\sigma\Delta = \frac{A\Delta V_T}{\sqrt{WL}} \quad (3.12)$$

where $A\Delta V_T$ is a technology conversion constant (in $\text{mV}\mu\text{m}$), and WL is the transistor's active area. $A\Delta V_T$ is about $5\text{mV}\mu\text{m}$ for nMOS and PMOS transistors in the super-threshold region in a 65nm process. Our previous research [41] has already revealed that V_T mismatch of paired transistors working in sub-threshold can be worse by a factor of two as compared to transistors working in the super-threshold region, i.e., $A\Delta V_T$ is about $2\times$ larger in the sub-threshold than in super-threshold. Therefore, assuming transistors with minimum dimensions, e.g., $W/L = 0.12\mu\text{m}/0.065\mu\text{m}$, the V_T mismatch deviation $\sigma\Delta$ can be 113.2mV in sub-threshold! Analog circuit designers always size transistors towards a $\sigma\Delta$ which is less than 10mV. In the sub-threshold region, $\sigma\Delta = 10\text{mV}$ translates into a transistor size which is more than $100\times$ larger compared to minimum transistor size! Obviously, doing so for digital gates will result in unaffordable silicon area, and vanished energy savings, which we otherwise could have obtained from operating circuits in the sub-threshold.

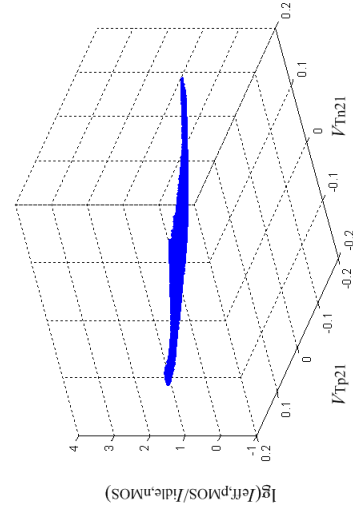
In our methodology, instead of using $\sigma\Delta$ as the metric, we use the metric ($I_{eff,nMOS}/I_{idle,pMOS} \geq 1$ at $V_{out}=V_{OL}$) for pull-down operation and the metric ($I_{eff,pMOS}/I_{idle,nMOS} \geq 1$ at $V_{out}=V_{OH}$) for pull-up operation. Recall that the I_{eff} is the effective drive current and I_{idle} is the idle leakage current. Table 3.2 shows the influence of V_T mismatch on I_{eff}/I_{idle} for a 2-input NAND (NAND2) gate (see Figure 3.8 (a)), where the dimen-

3.2. GATE SIZING CONSIDERING V_T MISMATCH IN DEEP SUB-THRESHOLD

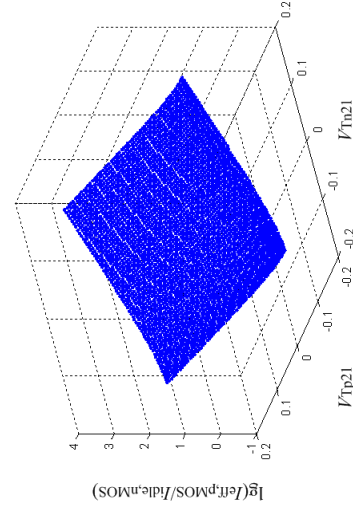
Table 3.2: $\lg(I_{eff}/I_{idle})$ for a 2-input NAND



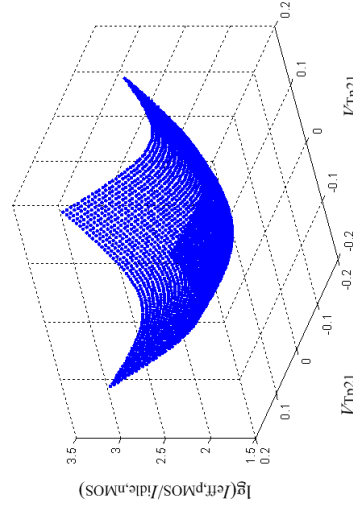
(a) $V_1 = V_{DD}$, $V_2 = V_{DD}$, $V_{out} = 0.1V_{DD}$



(c) $V_1 = G_{ND}$, $V_2 = V_{DD}$, $V_{out} = 0.9V_{DD}$



(b) $V_1 = V_{DD}$, $V_2 = G_{ND}$, $V_{out} = 0.9V_{DD}$



(d) $V_1 = G_{ND}$, $V_2 = G_{ND}$, $V_{out} = 0.9V_{DD}$

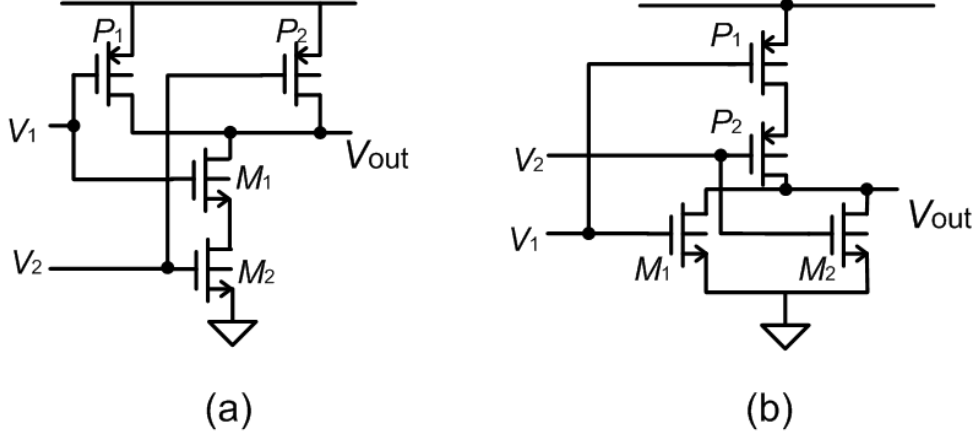


Figure 3.8: (a) two-input NAND gate (b) two-input NOR gate

sion of the pMOS transistor is $0.17\mu\text{m}/0.065\mu\text{m}$, and of the nMOS transistor $0.12\mu\text{m}/0.065\mu\text{m}$. The supply voltage V_{DD} is 160mV. V_{Tn21} is the V_T mismatch between two stacked nMOS transistors, and V_{Tp21} the V_T mismatch between two parallel pMOS transistors. The unit of V_{Tn21} and V_{Tp21} is Volt. In Table 3.2, (a),(b),(c),(d) represent the following cases: a) $V_1=V_{DD}$, $V_2=V_{DD}$, $V_{out}=0.1V_{DD}$, b) $V_1=V_{DD}$, $V_2=G_{ND}$, $V_{out}=0.9V_{DD}$, c) $V_1=G_{ND}$, $V_2=V_{DD}$, $V_{out}=0.9V_{DD}$, d) $V_1=G_{ND}$, $V_2=G_{ND}$, $V_{out}=0.9V_{DD}$. For a proper logic operation we must satisfy that $I_{eff} > I_{idle}$. As shown, case (d) is the only situation that benefits from paired transistor V_T mismatch, as I_{idle} is always lower than I_{eff} . For the other three cases, the I_{eff} and I_{idle} can either increase or decrease, depending on the direction of the V_T mismatch shift. Since I_{idle} spreads widely, and may exceed I_{eff} , the gate's functional yield cannot be guaranteed.

As the V_T mismatch of paired transistors in a digital gate occurs quite

3.2. GATE SIZING CONSIDERING V_T MISMATCH IN DEEP SUB-THRESHOLD

Table 3.3: Gate size normalized to minimum gate size vs. V_{DD} (functional yield = 99.9% and 99.7%, 65nm CMOS process)

V_{DD} (mV)	NAND2		NOR2	
	$\epsilon=99.9\%$	$\epsilon=99.7\%$	$\epsilon=99.9\%$	$\epsilon=99.7\%$
150	3.6	3.2	4.0	3.4
180	1.8	1.6	2.8	2.2
210	1.2	1.0	2.0	1.8
240	1.0	1.0	1.6	1.0
270	1.0	1.0	1.0	1.0

locally, device sizing is the only way to mitigate it. For each type of gate in the existing super-threshold library, we pick out the minimum sized gate (gate with $1\times$ driving capability). We do not vary its nMOS and pMOS transistors width ratios, as our simulation results showed that the gate that gives nearly equal rising and falling delays in the super-threshold also renders nearly equal rising and falling delays in the sub-threshold, provided that the V_T of the pMOS and nMOS transistors are balanced. We iteratively size up all the transistors in the paralleled and stacked topologies, until the gate meets a specified statistical functional yield in Monte-Carlo simulations. Table 3.3 lists the gate sizes of NAND2 and NOR2 for a functional yield ϵ of 99.9% and 99.7% respectively, while V_{DD} is varied with 30mV steps. For each gate, the gate size is normalized to the minimal size gate in the existing super-threshold library.

It can be seen that transistor upsizing is necessary if V_{DD} approaches an extremely low value. Once V_{DD} goes above certain critical value, the

3.2. GATE SIZING CONSIDERING V_T MISMATCH IN DEEP SUB-THRESHOLD

minimum sized gates in the existing super-threshold library suffice for the functional yield constraint in the sub-threshold domain, thus no re-sizing is needed. In our MC simulation, for 99.7% functional yield this critical value is typically around 250mV for all the minimum size gates except for the latch and flip-flop cells in the 65nm super-threshold library. The minimum size latch and flip-flop normally need a V_{DD} higher than 300mV because of the feed-back logic structure, as will be discussed in section 3.4.

We simulated a NAND3X1 (minimum 3-input NAND gate) ring oscillator (ringo) with 31 stages (LD=31). Targeting a functional yield of 99.7% and $V_{DD} = 180\text{mV}$, would require that the NAND3 gate is sized $2\times$ larger compared to the minimum size NAND3 gate in the super-threshold library. Table 3.4 shows the simulated mean frequency and mean energy per cycle before and after using our V_T balancing scheme. The energy overhead introduced from the V_T balancing scheme has been included in the simulation. Interestingly, although the minimum NAND3 gate satisfies our functional yield constraint only when $V_{DD} > 180\text{mV}$, the ringo can still function at 150mV with confidence, and after V_T balancing, it even functions without error at around 130mV! The reason why the ringo is able to operate at a lower V_{DD} than we expected is because the individual-gate functional yield does not directly translate into the yield of the whole ringo. As stated in chapter 2.4, considering only the functional yield of each individual gate and neglecting the spatial correlations between these gates results in pessimistic estimates of the minimum V_{DD} . For example, a gate that outputs higher V_{OL} (lower V_{OH}) can tolerate higher V_{IL} (lower V_{OH}) from its preceding gate. Therefore, the functional yield of the whole circuit is actually higher than the individual

3.2. GATE SIZING CONSIDERING V_T MISMATCH IN DEEP SUB-THRESHOLD

Table 3.4: Mean frequency, mean energy/cycle of ringo ($L_d = 31$, with and without V_T balancing scheme)

V_{DD} (mV)	Mean Frequency (KHz)			Mean Energy / Cycle (fJ)		
	before	after	% diff	before	after	% diff
130	N.A.	6.948	N.A.	N.A.	22.080	N.A.
150	7.675	9.055	17.985	27.667	21.370	22.761
200	19.187	26.594	38.608	26.422	20.975	20.614
250	55.399	83.108	50.016	25.360	20.954	17.371
300	167.914	267.448	59.277	25.546	19.863	22.243
350	493.808	847.867	71.700	25.620	20.482	20.053
400	1397.480	2430.090	73.891	27.387	23.284	14.981
Avg.			51.913			19.670

gate's functional yield. Since in our methodology a high functional yield of an individual gate is guaranteed, a high functional yield of the circuit is also guaranteed.

It is worth noting the significant improvement of the mean frequency and energy/cycle after implementing our V_T balancing scheme. On average, with the V_T balancing scheme we observed a frequency speedup of 51.91%, and energy/cycle savings of 19.67%. This is mainly due to the reduced delay variability. At process corners, either the rising or falling time is exceedingly long, such that the cycle time of the ringo becomes very long. As a result, more leakage energy is dissipated per cycle. With the V_T balancing scheme, the rising and falling times are balanced, and the cycle time is reduced, so lower energy per cycle is consumed. As V_{DD} increases, our V_T balancing scheme becomes more effective, so greater improvement on speed is obtained.

3.3. IMPROVING DRIVABILITY BY EXPLOITING V_T MISMATCH BETWEEN PARALLELIZED TRANSISTORS

Without V_T balancer, in this case the V_{opt} is 250mV. Compared with the V_{opt} in Fig. 2.5, which is about 300mV, the V_{opt} shift is because the switching density of ringo is much higher thereby yielding a higher dynamic/leakage energy ratio and consequently a lower V_{opt} value.

3.3 Improving Drivability by Exploiting V_T Mismatch between Parallelized Transistors

As discussed in section 3.2, the V_T mismatch is catastrophic to circuit design. In this section, we will propose an interesting approach to improve sub/near threshold current drivability by exploiting the V_T mismatch between parallel transistors. Our approach is based on theoretical proof and simulation results that the V_T mismatch between parallelized transistors always results in an increased driving current in the sub-threshold.

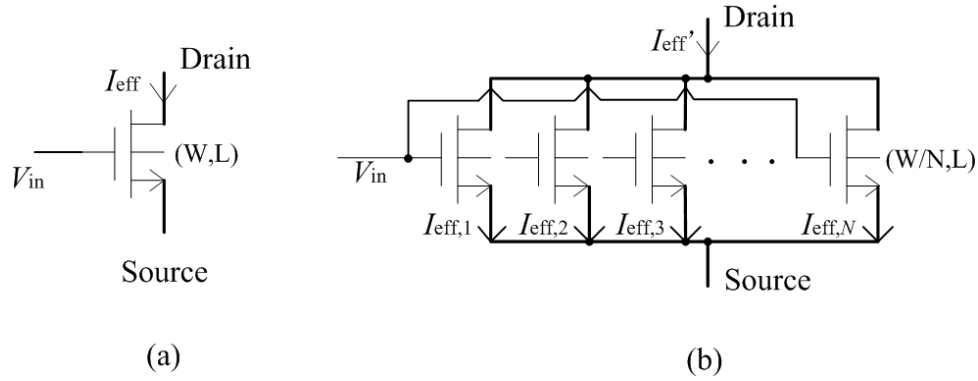


Figure 3.9: (a) nMOS transistor with aspect ratio (W, L) (b) N -parallelized nMOS transistors with aspect ratio $(W/N, L)$

Suppose $\mu(V_T), \sigma(V_T)$ are the mean and standard deviation of V_T for an nMOS transistor in Figure 3.9 (a). Considering V_T variation and according

3.3. IMPROVING DRIVABILITY BY EXPLOITING V_T MISMATCH BETWEEN PARALLELIZED TRANSISTORS

to the properties of log-normal distribution, the mean value of I_{eff} is:

$$\mu(I_{eff}) = I_{0n} e^{\frac{[V_{GS} - \mu(V'_T) + \eta V_{DS} - \gamma V_{SB}]}{nU} + \frac{[\frac{\sigma(V'_T)}{nU}]^2}{2}} (1 - e^{-\frac{V_{DS}}{U}}) \quad (3.13)$$

Suppose the transistor is equally divided as N -parallel nMOS transistors (see Figure 3.9(b)). For every individual small transistor we have

$$\mu(V_{T,1}) = \mu(V_{T,2}) = \dots = \mu(V_{T,N}) = \mu(V'_{Tx}) \quad (3.14)$$

$$\sigma(V_{T,1}) = \sigma(V_{T,2}) = \dots = \sigma(V_{T,N}) = \sigma(V'_{Tx}) \quad (3.15)$$

The mean value of the total sub-threshold current $\mu(I'_{eff})$ in Figure 3.9(b) is calculated by,

$$\begin{aligned} \mu(I'_{eff}) &= \sum_{i=1}^N \mu(I_{eff}, i) \\ &= I_{0n} e^{\frac{[V_{GS} - \mu(V'_{Tx}) + \eta V_{DS} - \gamma V_{SB}]}{nU} + \frac{[\frac{\sigma(V'_{Tx})}{nU}]^2}{2}} (1 - e^{-\frac{V_{DS}}{U}}) \end{aligned} \quad (3.16)$$

Based on equation (3.12), we have

$$\sigma(V'_{Tx}) > \sigma(V'_T) \quad (3.17)$$

Since $\mu(V'_T) = \mu(V'_{Tx})$, by comparing equations (3.13) and (3.16), we can

3.3. IMPROVING DRIVABILITY BY EXPLOITING V_T MISMATCH BETWEEN PARALLELIZED TRANSISTORS

obtain (3.18),

$$\mu(I'_{eff}) > \mu(I_{eff}) \quad (3.18)$$

As can be seen, dividing a large transistor into smaller parallelized transistors helps to increase the sub-threshold current due to larger V_T mismatch. Therefore, a larger drivability can be accomplished statistically without incurring any additional overhead such as increasing V_{DD} or transistor size.

The Monte-Carlo simulation results have confirmed the effectiveness of this approach. Assume that a Standard V_T (SV_T) nMOS transistor with aspect ratio $W/L = 0.96\mu\text{m}/0.065\mu\text{m}$ is divided as N -transistors ($N=1,2,3,4,6,8$). Its gate voltage V_{in} and drain-to-source voltage V_{DS} are set as 200mV. The simulated mean and standard deviation values of the effective driving current I_{eff} are listed in Table 3.5. As seen, the larger the N , the larger the V_T mismatch, consequently the larger sub-threshold driving current. However, Table 3.5 also shows an increasing driving current variability and larger $\sigma(I_{eff})/\mu(I_{eff})$ as the transistor becomes narrower. This is due to an increased V_T shift caused by narrow width effects. To mitigate such effect, instead of dividing all transistors into minimal width transistors, our design constrained the transistor width to be not smaller than a certain limit. By constraining a maximum $\sigma(I_{eff})/\mu(I_{eff})=20\%$, a same driving current can be achieved with approximately 10% transistor area reduction.

This interesting property can also be applied to the pass-transistor based logics, such as power-switches. Because the sub-threshold drivability is increased, the necessary transistor size can be reduced. Most importantly, since

3.3. IMPROVING DRIVABILITY BY EXPLOITING V_T MISMATCH BETWEEN PARALLELIZED TRANSISTORS

Table 3.5: Mean and standard deviation of driving current

N	$\mu(I_{eff})$ (nA)	$\sigma(I_{eff})$ (nA)
1	5.390	2.495
2	5.991	3.023
3	7.667	4.233
4	9.324	4.885
6	12.934	6.255
8	13.316	7.379

a wide transistor is thus divided into small ones, multiple-finger structured layout is therefore allowed. As a result, the layout of very huge transistors, such as power switches, avoids a strange aspect ratio and becomes much more compact which may reduce silicon area considerably. As an example, this method has been used in the design of power switches in our configurable V_T balancer. Figure 3.10 shows the layout of the configurable V_T balancer in a TSMC 65nm CMOS process. The total layout area is $25 \times 30 \mu\text{m}^2$. The balancer is surrounded by rectangular power rings on metal layers M1 and M2. As can be seen, all the power switches are divided as small transistors by using multiple-finger structured layout style, which reduces the silicon area significantly.

3.3. IMPROVING DRIVABILITY BY EXPLOITING V_T MISMATCH BETWEEN PARALLELIZED TRANSISTORS

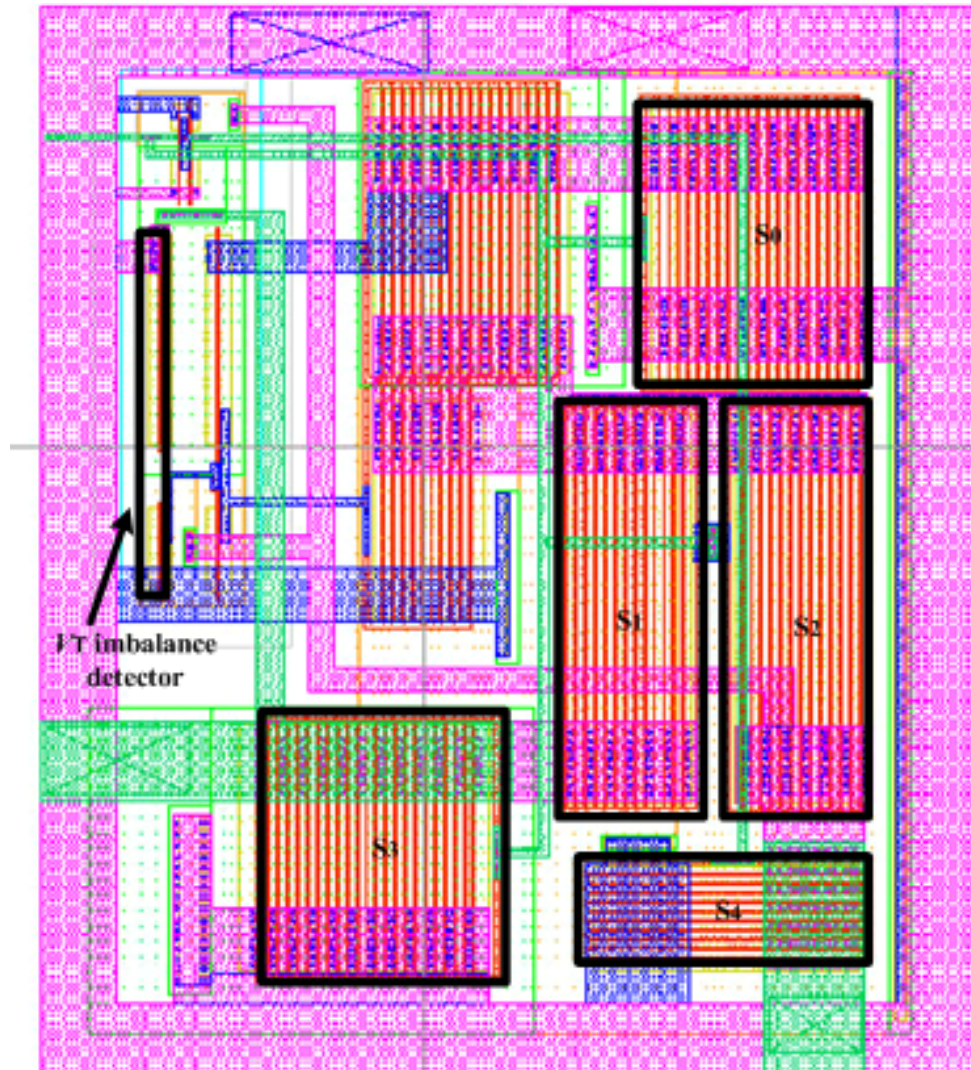


Figure 3.10: Layout of configurable V_T balancer with multiple finger structured power switch in a 65nm CMOS

3.4 Sub-threshold Library Cell Selection

As mentioned in the section 3.2, once V_{DD} goes above a certain critical value, almost all the minimum sized gates in the existing super-threshold library suffice for the functional yield constraint in the sub-threshold domains, thus no re-sizing is needed. Therefore, functional yield is not a problem for us if we operate circuits in the weak sub-threshold region. However, some of the cells have large effective driving current variability which results in remarkably deteriorated timing yield. In our work, these criminal cells are identified by Monte-Carlo simulation and filtered out before logic synthesis. As illustrated in Figure 3.11, the prohibited cells have some typical structures:

1) *More than 4 parallel transistors and more than 4 stacked transistors* (a)(b)

Parallel transistors and stacked-transistors introduce large current variability. As the number of parallel transistors and the number of stacked-transistors increase, the leakage current variability increases. We prohibit logic gates with more than 4 parallel transistors or more than 4 stacked transistors or both, such as 4-input NAND and NOR.

2) *Ratioed logic* (c)(d)

Ratioed logic can reduce the number of transistors required to implement a given logic function, but must be sized carefully so that the active current is stronger than the static current. Therefore, the correct functioning of ratioed logics depends largely on the sizing. In the sub-threshold region, the largest current variability is due to V_T variation. Even a small variation on V_T will have a heavy impact on the active or static current, so logic cells relying on

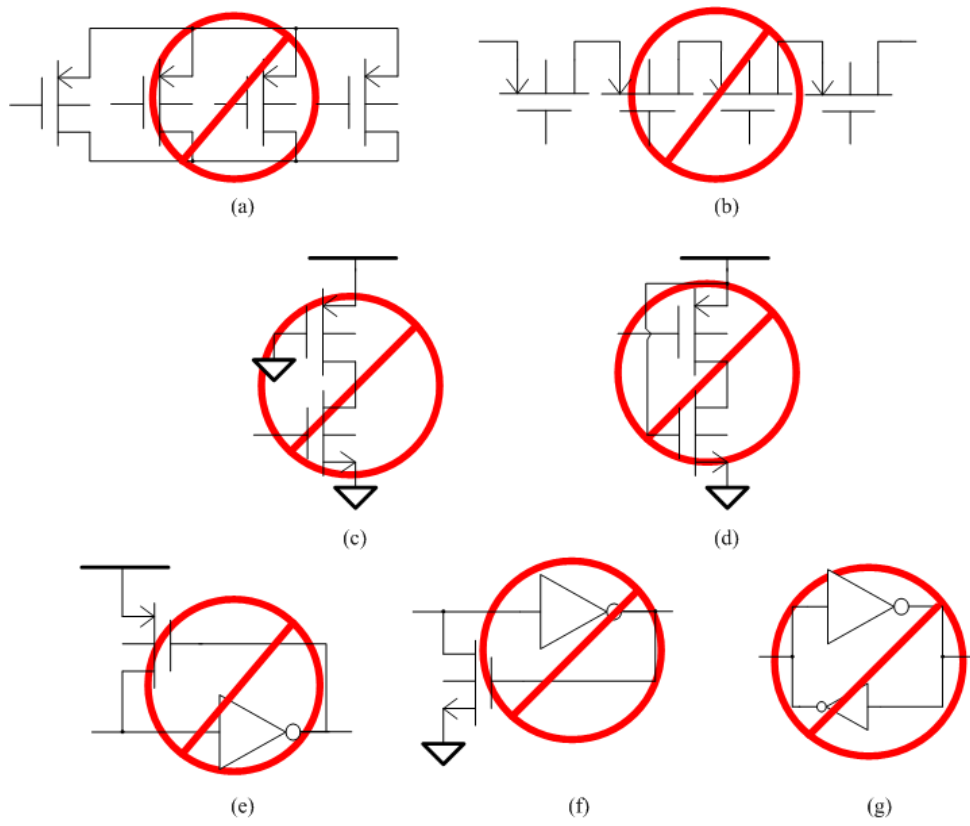


Figure 3.11: Prohibited cell structures in near/sub threshold (only parallel and stacked pMOS transistors are drawn for clarity)

transistor sizing are dangerous and should be prohibited.

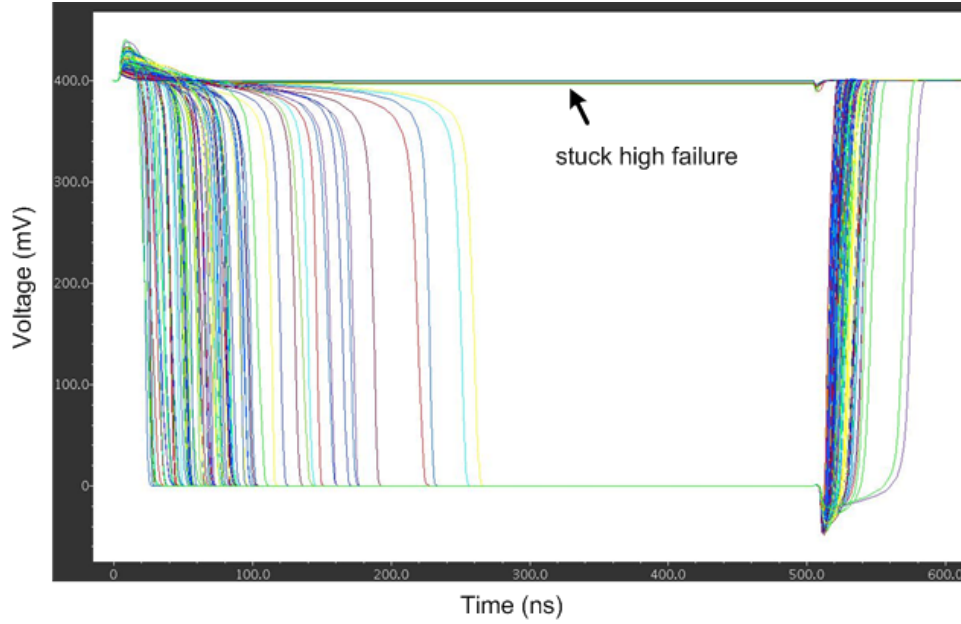


Figure 3.12: Monte-Carlo transient simulation for cross-coupling feedback inverters at $V_{DD}=400\text{mV}$

3) *Feedback logic (e)(f)(g)*

Feedback logic is a special type of ratioed logic. Such kind of logic uses positive feedback loops to help change logic values. Whether the logic value can be changed successfully and promptly depends primarily on choosing a suitable loop gain. In the super-threshold design this is achieved through proper transistor sizing. However, in the sub-threshold region, the variation of feedback loop gain is largely impacted by the V_T variation, so sizing is not useful anymore. The output can have stuck-high or stuck-low failures and thus never flip. Figure 3.12 shows the transient behavior of cross-coupled inverters

(g) at $V_{DD}=400\text{mV}$ through Monte-Carlo simulation. The malfunctioning caused by stuck-high failure in the sub-threshold region is observed. Besides, the large spread of propagation delay is unacceptable.

3.5 Turning Ratioed Logic into Non-ratioed Logic

Latches and registers are the feedback (ratioed) logic that must be used in sequential circuits. Figure 3.13 shows how to turn them into non-ratioed logic. By using the clk and \overline{clk} signals, we prevent the slave inverters (I_2, I_4) from direct cross-coupling with the master inverters (I_1, I_3). As a result, when writing into the latch, the slave inverter is always disabled, so writing to the master inverter is facilitated. After writing is done, the slave inverter is enabled to help maintain the logic value. Therefore, the race between slave and master inverters is avoided.

Figure 3.14 compares the Monte-Carlo simulation results at node X (the output from the negative latch) at $V_{DD}=400\text{mV}$ before and after turning ratioed logic into non-ratioed logic. With this work, the stuck high and stuck low failures are avoided. In addition, the propagation delay becomes more than an order tighter.

3.6 Capacitive-based Level Shifter (CBLCL)

Multiple Supply Voltages (MSV) approach allows a design to use different V_{DD} s for the sub-instances or blocks. A Level converter (LC) should be inserted to shift a signal from a low voltage domain (V_{DDL}) to a high voltage domain (V_{DDH}), to insure a correct transition. Many different types of

3.6. CAPACITIVE-BASED LEVEL SHIFTER (CBLC)

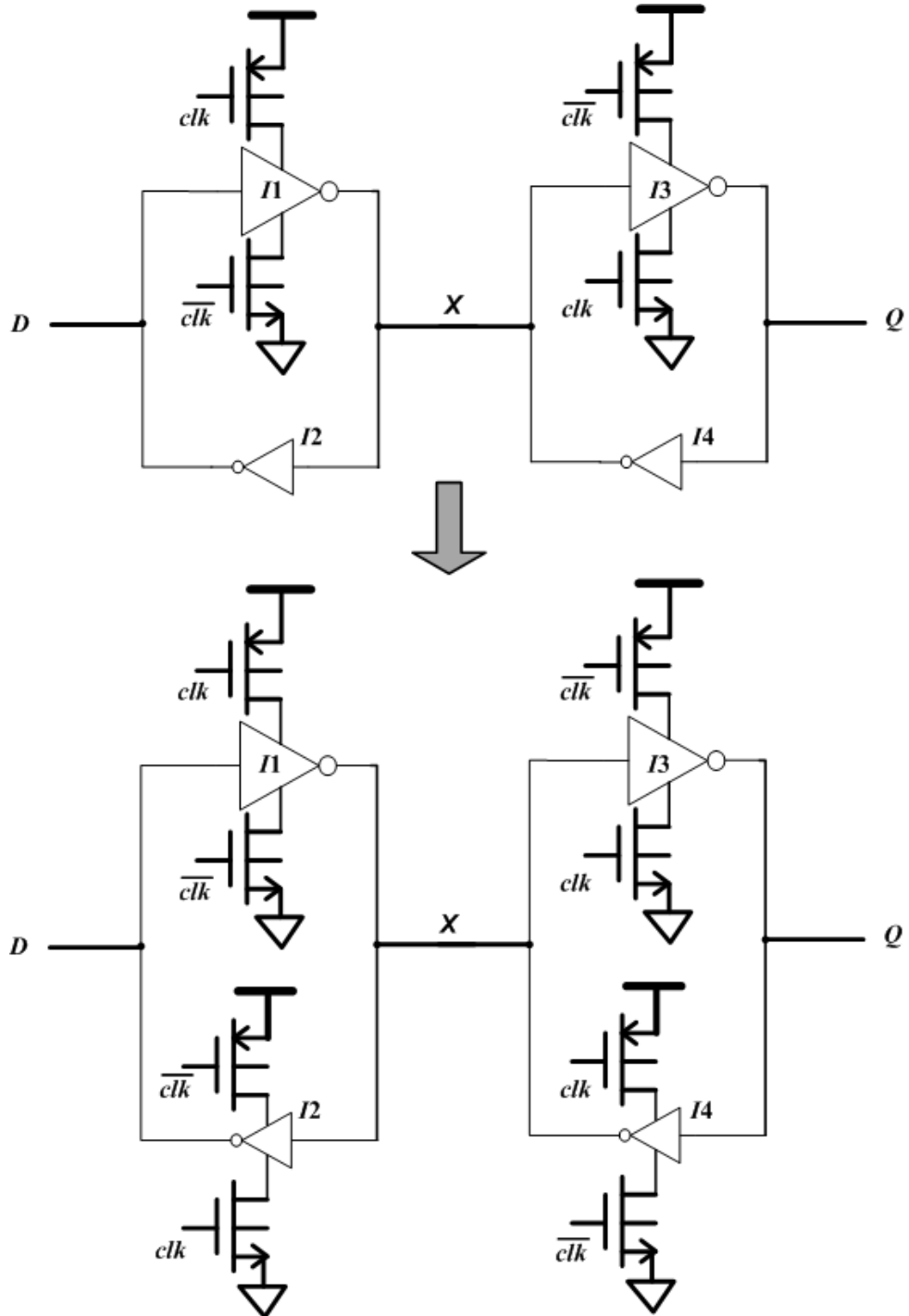


Figure 3.13: Turning ratioed logic into non-ratioed logic

3.6. CAPACITIVE-BASED LEVEL SHIFTER (CBLC)

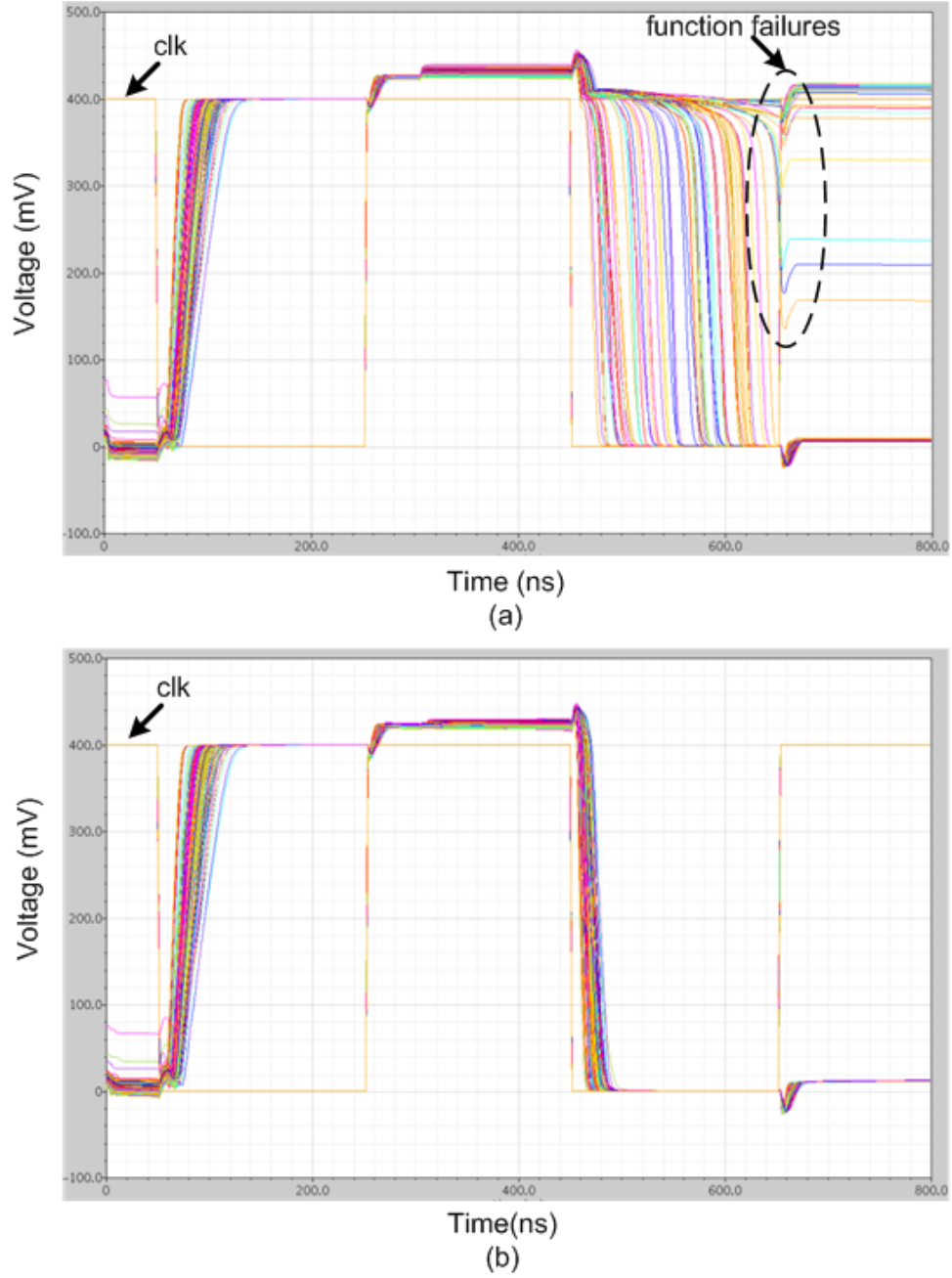


Figure 3.14: Monte-Carlo simulation results at node X at $V_{DD} = 400\text{mV}$: (a) before turning ratioed logic into non-ratioed logic (b) after turning ratioed logic into non-ratioed logic

LCs have been supported in the commercial synthesis tools. These LCs are designed for conversion between super-threshold V_{DD} s and optimized to convert a V_{DDL} around $2/3$ nominal V_{DD} to full V_{DD} . However, none of them is viable for shifting a V_{DDL} in the sub/near threshold region to a V_{DDH} that is $2\times$ higher than V_{DDL} . An analog approach has been proposed in [37], which used a differential op-amp to convert a (G_{ND}, V_{DDL}) swing into a (G_{ND}, V_{DDH}) swing. This approach needs a static biasing current is needed, introducing considerable static energy overhead.

A capacitive-based level converter (CBLC) is proposed in this work (Figure 3.15). It is capable of converting a signal from a sub/near threshold V_{DDL} to a V_{DDH} that is even higher than $2V_{DDL}$. The idea behind the CBLC is as follows: when $In=0$, the diode-connected transistor MP1 is turning on, the boosting capacitor C_L is charged to V_{DDL} . Note that at this time the node X is connected to MP1's bulk to forward-bias the bulk-source junction. The V_T of MP1 is thus decreased to facilitate charging the C_L . When In rises to V_{DDL} , MP1 turns off, the node X is floating with a potential rising much higher than V_{DDL} (near $2V_{DDL}$). This high potential turns off MP2, so that the final output reaches V_{DDH} as expected. Transistor MN1 is implemented with low- V_T device to further support this operation. The boosting capacitor can be implemented with two metal layers. It can also be implemented with C_{gs} and C_{gd} of the transistor MN2, as shown in Figure 3.15. This is possible as the needed C_L is only $1\sim 2\text{fF}$. Since no metal layer is needed to form the capacitor plates, the routing obstruct is avoided, so the routing difficulty is reduced for EDA tools. The simulated waveforms for input signal from $V_{DDL}=400\text{mV}$ to output at $V_{DDH}=800\text{mV}$ and the voltage of node X are

3.6. CAPACITIVE-BASED LEVEL SHIFTER (CBLC)

plotted in Figure 3.16.

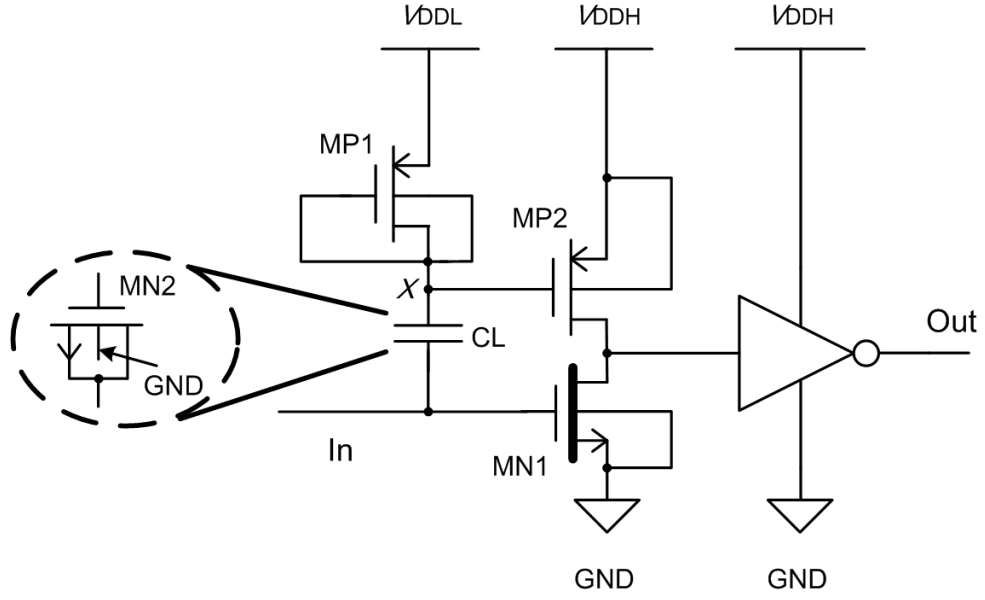


Figure 3.15: Capacitive-based level converter (CBLC)

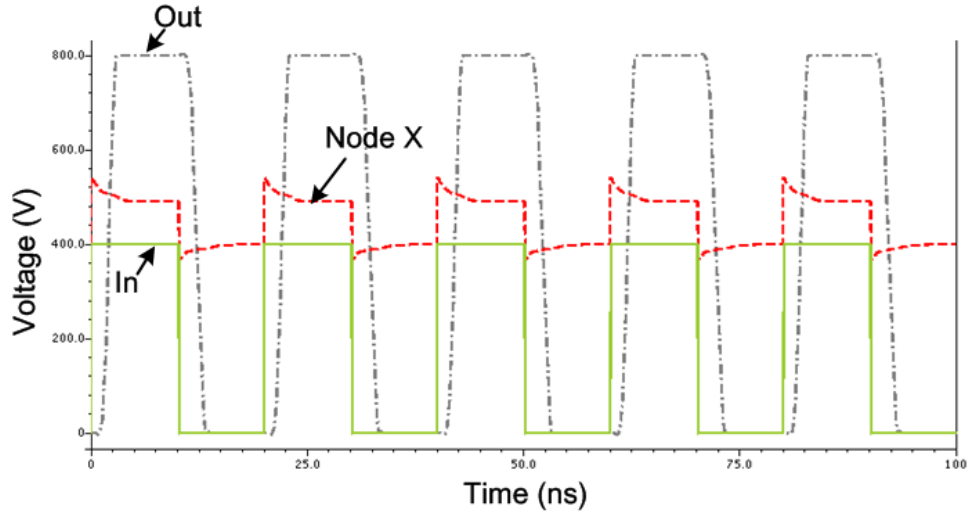


Figure 3.16: Waveforms of the CBLC ($V_{DDL}=400\text{mV}$ and $V_{DDH}=800\text{mV}$)

Chapter 4

Design of the *SubJPEG* Co-processor

In this chapter we will introduce *SubJPEG* (standing for *Sub-threshold JPEG*), a state-of-the-art 65nm CMOS ultra-low energy multi-standard JPEG encoder co-processor with a near/sub threshold power supply. We will first briefly overview the entire design flow. Then the JPEG encoding standard is introduced. Later some design and implementation issues of *SubJPEG* will be discussed in detail. Finally, the performance evaluation is provided.

4.1 Design Flow Overview

The design flow is shown in Figure 4.1. A sub-threshold standard cell library, which has been introduced in Chapter 3, is loaded before the flow starts. This library contains the .lib file (for logic synthesis), .lef file (for clock tree synthesis, placement and routing), .gds file (for final GDS collection). Apart from the standard cells, this library has a customized well-tap cell, which is needed to implement bulk-biasing.

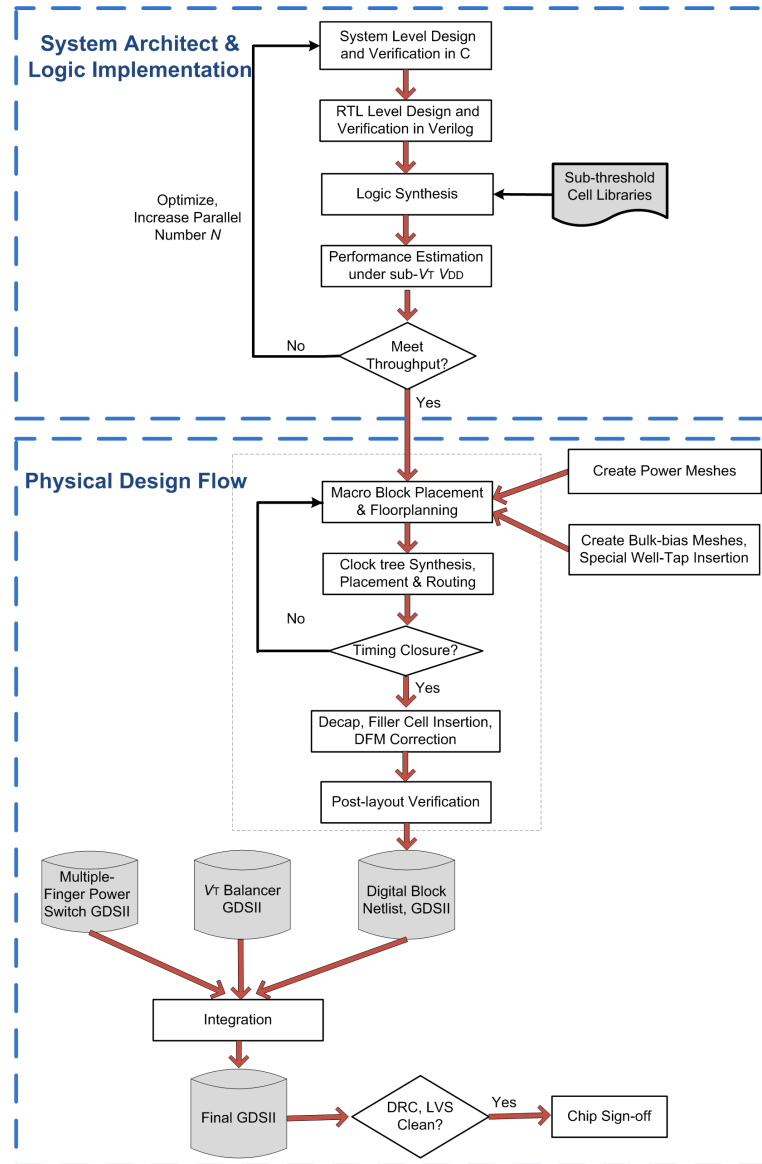


Figure 4.1: Sub-threshold design flow

The design flow starts from system architecting and logic design, which involves iterations among C-programming level architecting and verification, RTL design and verification, performance estimation. The number of parallel degree is decided to compensate throughput loss at very low V_{DD} . System partitioning, i.e., multiple clock domains, multiple voltage domains, are also optimized to reduce area and energy.

The system architecting and logic design is followed by physical design. The floor-planning takes care of the geometric locations for all the macro-blocks, multiple power domains and multiple clock domains. Separate deep n-Well regions are drawn under each macro-block to which bulk-biasing techniques are applied. In these macro-blocks, besides the regular power meshes for V_{DD} and G_{ND} nets, power meshes for n/p wells are also routed as special nets, and customized well-tap cells are inserted. Iterations from floorplanning to placement and routing are necessary to meet timing closure. The final output layout will be integrated with the V_T balancers, finger-structured power switches and other analog components. The chip is fully LVS and DRC clean before sign-off for fabrication.

In the next sections, we will use *SubJPEG* as a case study to demonstrate how the design flow has been went through.

4.2 JPEG Encoding Standard

JPEG is an international compression standard for continuous-tone still images, both grayscale and color [48]. This standard is established by the acronym JPEG, standing for Joint Photographic Experts Group. JPEG encoding is

able to greatly reduce file size with minimal image degradation by throwing away the least “important” information. As a generic and popular image compression standard, JPEG supports a wide variety of image applications.

As shown in Figure 4.2, the baseline JPEG encoding processing has three primary steps: 8×8 discrete cosine transformation (DCT), quantization, entropy encoding. These steps are detailed below.

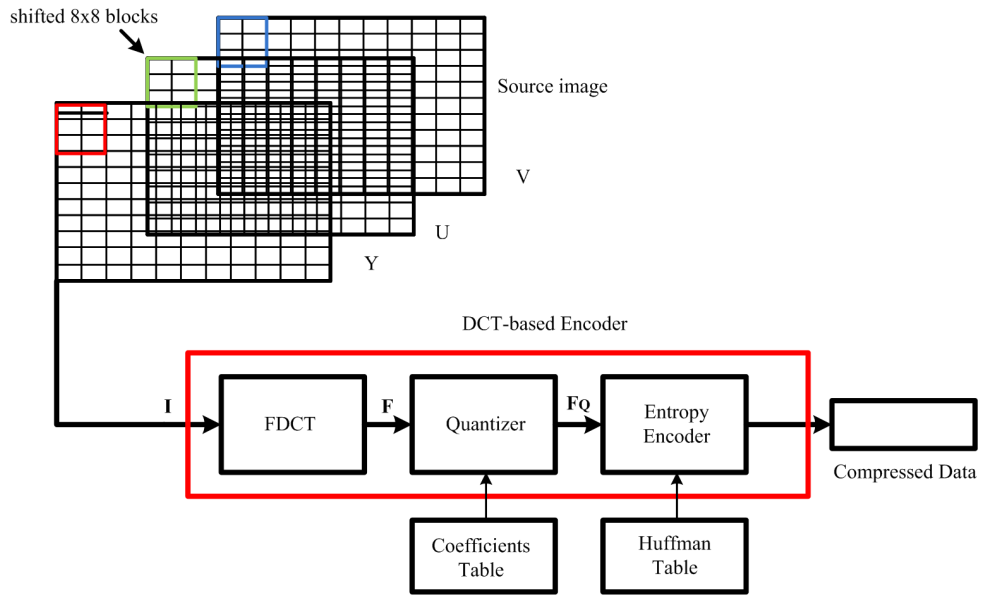


Figure 4.2: JPEG encoder processing steps

1) 8×8 DCT

At the input to the encoder, source image samples in YUV format are grouped into 8×8 blocks, shifted from unsigned integers with range $[0, 2^P - 1]$ to signed integers with range $[-2^{P-1}, 2^{P-1} - 1]$ by subtracting 2^{P-1} where P is the number of bits per value, and input to the forward 2D-DCT (2-dimensional discrete cosine transform). The mathematical definition of the

DCT is as following:

$$F(u, v) = \frac{1}{4}C(u)C(v) \sum_{x=0}^7 \sum_{y=0}^7 I(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \quad (4.1)$$

$$\text{where } C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u, v = 0; \\ 1 & \text{otherwise} \end{cases}$$

and $I(x, y)$ is the matrix of values from the source image block.

Direct implementation of 2D-DCT in ASIC requires a wide bandwidth and lots of computation resources. Note that the 2D-DCT can be decomposed by two transforms, using row-column DCT. It is more efficient to perform a 1st 1D-DCT on each row of the block,

$$F(u) = \frac{1}{2}C(u) \sum_{x=0}^7 I(x) \cos \frac{(2x+1)u\pi}{16} \quad (4.2)$$

$$\text{where } C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u = 0; \\ 1 & \text{otherwise} \end{cases}$$

then transpose the matrix and perform a 2nd 1D-DCT on each row again, in this way the column 1D-DCT is also performed.

The effect of 2D-DCT is that the output data $F(u, v)$ has been organized in terms of importance. The human eye has more difficulty discriminating between higher frequencies than low frequencies. Low frequency data therefore

carries more important information than the higher frequencies. The data in the output matrix is organized from the lowest frequency in the upper-left to the highest frequency in the lower-right. This prepares the data for the next step, quantization.

2) *Quantization*

Quantization is the step where data is “thrown away”. After output from the 2D-DCT, each of the 64 DCT coefficients is uniformly quantized in conjunction with a 64-element quantization table. The results are rounded to the nearest integers. This quantization table depends on the application and is an input to the encoder. Each element can be any integer value from 1 to 255, which specifies the step size of the quantizer for its corresponding DCT coefficient. The purpose of quantization is to reduce most of the less important high frequency coefficients to zeros. The more zeros generated the better the image will be compressed. In other words, quantization achieves further compression by representing DCT coefficients with no greater precision than is necessary to achieve the desired image quality. The mathematic description for this step is:

$$F_Q(u, v) = IntegerRound\left(\frac{F(u, v)}{Q(u, v)}\right) \quad (4.3)$$

In ASIC implementations, a division is realized through multiplication. Thus, the quantizers are inversed and stored in the quantization table. It is convenient if the user can customize the level of compression at runtime to trade off the compressed image quality and compression ratio. For example, if the user wants better quality he can lower the values in the Q matrix; or if

he wants higher compression ratio he can raise the values in the Q matrix.

3) *Entropy coding*

After quantization, the DC coefficient is treated separately from the 63 AC coefficients, as DC coefficients frequently contain a significant fraction of the total image energy. The DC coefficient is a measure of the average value of the 64 image samples. Because there is usually strong correlation between the DC coefficients of adjacent 8×8 blocks, the quantized DC coefficient is encoded as the difference from the DC term of the previous block in the encoding order. This near zero difference can be encoded with fewer bits in Huffman coding.

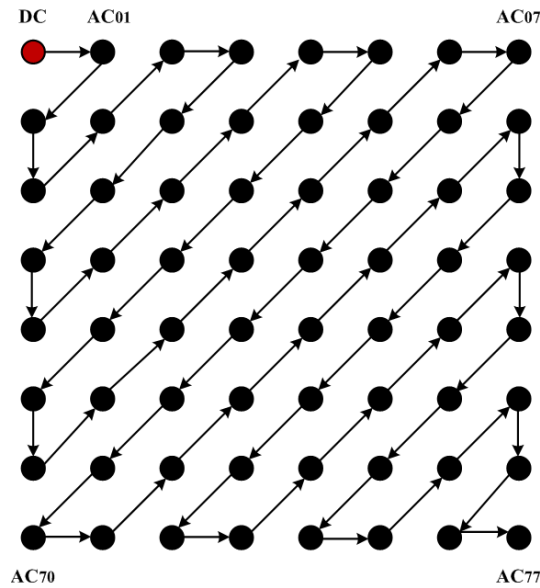


Figure 4.3: AC zig-zag sequence

Instead of reading off the AC coefficients row by row, JPEG compression reads along the diagonals. All of the quantized samples are ordered into the “zig-zag” sequence, shown in Figure 4.3. This ordering helps to place

low-frequency coefficients (which are more likely to be nonzero) before high-frequency samples and produce maximal series of 0s at the end.

The final processing step is entropy coding. This step achieves additional compression by encoding the quantized DCT coefficients more compactly based on their statistical characteristics. Entropy coding is often organized as a 2-step process. The first step converts the zig-zag sequence of quantized coefficients into an intermediate symbol sequence by applying run-length encoding (RLE). Each nonzero AC coefficient is represented in combination with the “runlength” (consecutive number of zero-valued AC coefficients which precede it in the zig-zag sequence). The second step is to assign frequently used symbols fewer bits than rare symbols through sequential coding. The Baseline sequential codec uses Huffman coding. Huffman coding requires that one or more sets of Huffman code tables be specified by the application. Huffman tables may be predefined as dictionary and referenced within an application as defaults. With the “runlength”, “amplitude” (the amplitude of non-zero coefficients), “size” (the number of bits used to encode the amplitude), the final symbol is generated by looking-up different Huffman tables. An EOI (End-of-Image) is added at the end of the final symbol representing this image.

More information about Huffman coding and JPEG compression standard can be found in [49] .

4.3 *SubJPEG* Architecture

4.3.1 Design challenge

Our purpose is to design a JPEG compression co-processor that consumes extremely low energy and thus can be widely used in the application fields such as image sensors, digital still cameras, mobile image, medical imaging, low-end image compressor, etc. The design challenge is generalized in Figure 4.4. The energy consumed by a conventional JPEG processor at the nominal supply is the energy ceiling. As discussed in Chapter 2, if the supply voltage V_{DD} can be scaled down to a near/sub-threshold level, huge energy savings could be achieved. However, parallelism needs to be used to maintain a certain throughput, so the silicon area will increase. Different V_{DD} s (for example, V_{DDA} and V_{ddb} in Figure 4.4) imply different degree of parallelism. Therefore, the main challenge is to explore an architecture with efficient parallelism to trade-off among area, throughput and energy.

Our baseline design was built from scratch to accommodate architectural changes required for sub-threshold operation in a 65nm CMOS LP-SV_T process. Its area and energy breakdown are shown in Figure 4.5. The term “engine” denotes a combined 2D-DCT and Quantization module. As seen, the engine dominates both the energy and area. At the nominal supply the engine occupies less than 50% of the total silicon area but consumes around 70% of the total energy. Obviously it is the target we need to optimize. The rest of the components, such as the Huffman encoder and the configuration logic, are of less importance. Thus minimizing the energy consumption of the engine becomes our primary target when designing the new architecture.

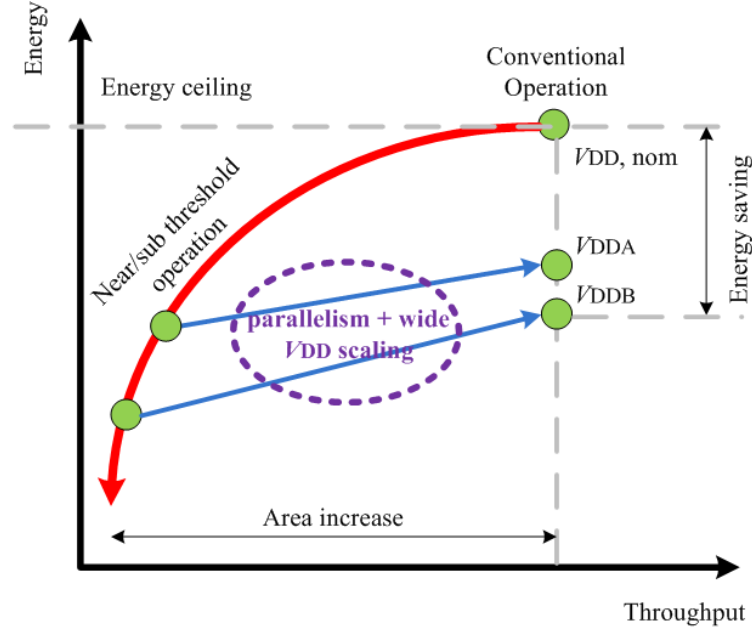


Figure 4.4: Design challenge

Therefore, instead of parallelizing the entire data-path, we decide to parallelize only the engine. Another reason for making this decision is because of the difficulty in parallelizing the Huffman encoder. As explained in section 4.2, the Huffman encoding for the DC value of an 8×8 block depends on the DC value of the previous block. If the Huffman encoder is also parallelized, additional effort must be drawn to handle this data dependency. Also it would be difficult to align the output streams from each Huffman encoder which have unpredictable lengths, a memory shuffler and many memory operations would become unavoidable.

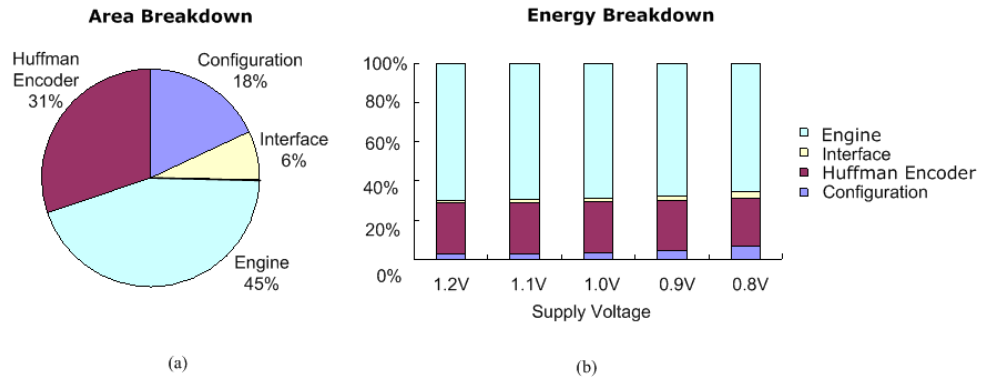


Figure 4.5: (a) Area (b) energy breakdown for conventional JPEG encoder

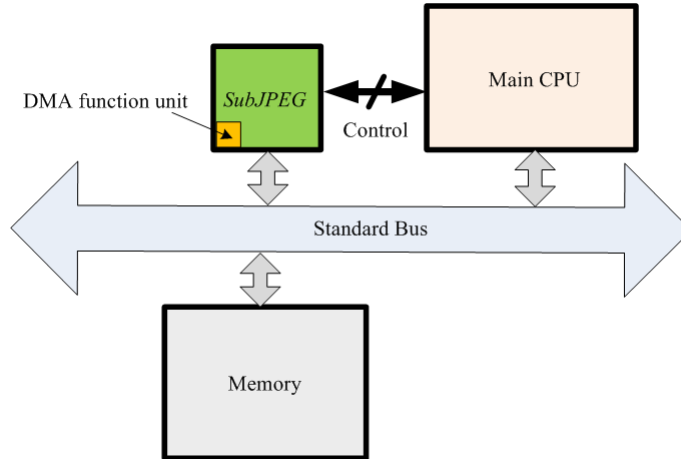


Figure 4.6: The functionality of *SubJPEG* in the system

4.3.2 *SubJPEG* Macro-Architecture

Figure 4.6 shows how *SubJPEG* functions in a system. As shown, *SubJPEG* is a co-processor hosted by a main CPU. Through the control lines, the main CPU can communicate with *SubJPEG*, issue commands and access the status registers in *SubJPEG*. *SubJPEG* interfaces with a commercial standard bus, such as PCI/PCI-X/PCI-Express. It has direct-memory-access (DMA) unit which supports fetching the image data stored in a memory without help from the main CPU.

Figure 4.7 shows the *SubJPEG* processor diagram. The final JPEG encoder processor exploits 2 supply voltage domains (V_{DDH} , V_{DDL}), 3 frequency domains (bus_clk, engine_clk, Huffman_clk), and 4 engines. The key modules are described below.

4.3.3 Control Path Design

On the control path, the configuration space, read controller (RDC), write controller (WRC) are the three main modules.

The configuration space is for the external main CPU to configure *SubJPEG* and check the running status from *SubJPEG*. It is operated with bus_clk and V_{DDH} . For each frame, the external main CPU issues a command to the configuration space of the JPEG co-processor. The configuration commands include information such as the source data start address/length, destination data start address, YUV sampling ratio, programmable quantization table coefficients, etc. In our architecture, 2 command slots are accommodated in the configuration register file, so the main CPU can is-

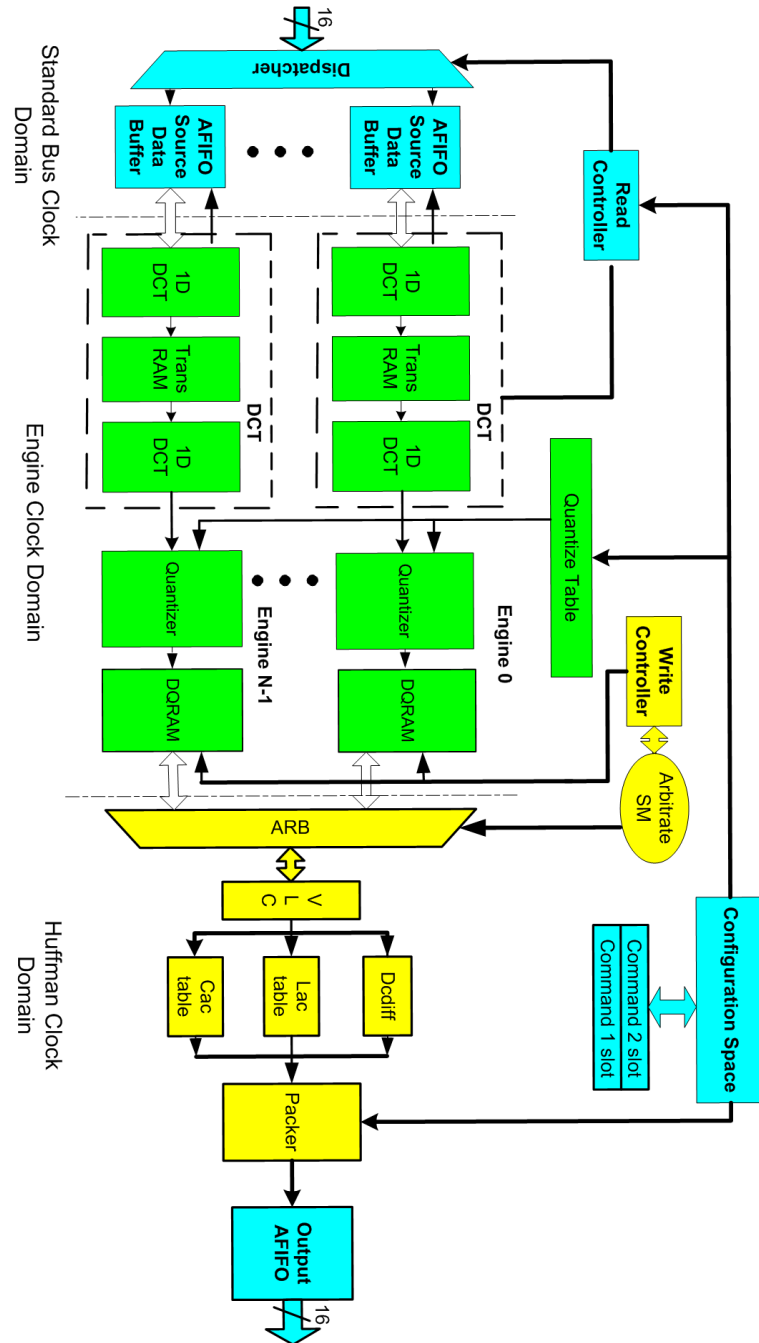


Figure 4.7: *SubJPEG* processor diagram

	31	0	No.
JPEG Base Configuration Space	JPEG Encoder Capability		0
	JPEG Control (command entry number, YUV ratio, maximum payload from the bus, request interval time)		4
	JPEG Status		8
	JPEG Quantization Table ($2 \times 64 \times 8$ bit coefficients)		12
			16
			20
			24
			28
			32
			36
			40
	JPEG Source Data Address High(command0)		44
	JPEG Source Data Address Low(command0)		48
	JPEG Source Data Length(command0)		52
	JPEG Destination Address High(command0)		56
	JPEG Destination Address Low(command0)		60
	JPEG Source Data Address High(command1)		64
	JPEG Source Data Address Low(command1)		68
	JPEG Source Data Length(command1)		72
	JPEG Destination Address High(command1)		76
	JPEG Destination Address Low(command1)		80
	Reserved		84

Figure 4.8: Configuration space overview

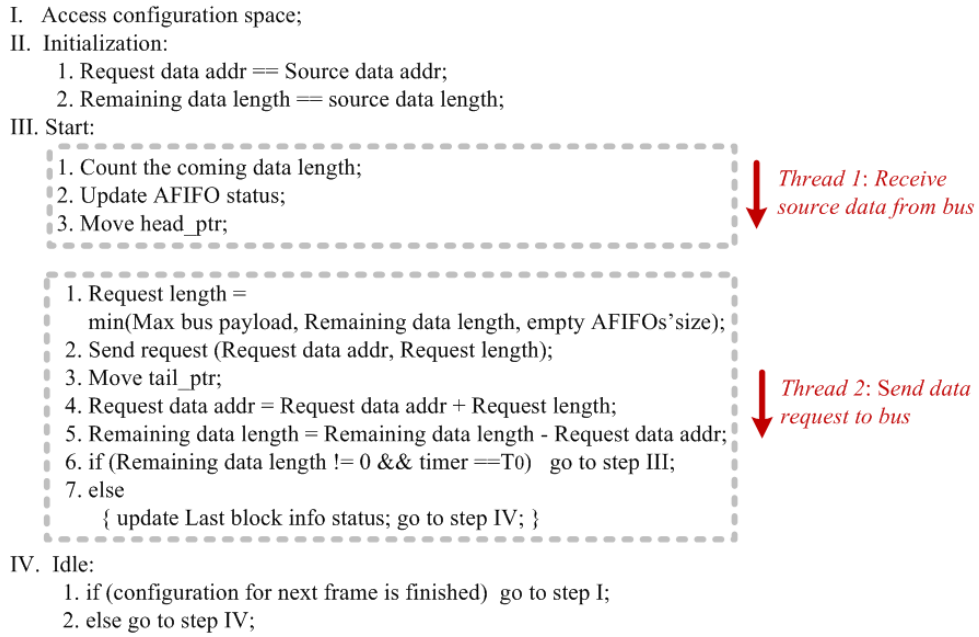
The read controller (RDC) works with `bus_clk` and V_{DDH} . Its main function is to read blocks of source data from the standard bus according to the configuration information. Figure 4.9 shows how the RDC works. A status table is maintained to record the status of the asynchronous FIFOs (AFIFOs) and the last block information. Once the new data coming from the bus has been fed into the AFIFOs, the source data counter will count the coming data length and update the AFIFOs' status in the table, as well as move the head pointer. The RDC issues data requests periodically according to the configured interval time T_0 . The requested data length is based on the minimal among remaining data length (this is initialized as the source data length at start run), maximum bus payload size and AFIFOs' empty size (how many AFIFOs are empty). As soon as the requested data length is calculated, the tail pointer will jump to AFIFO where the latest request source data block will be stored. Then the requested data address and the remaining data length are also updated. If the remaining data length is zero, meaning that the last requested data block is the ending block of the current frame, the column of last block information in the status table will be updated. Figure 4.10 is the pseudo code algorithm for RDC. Table 4.1 lists some signals that interact from the data path and control path in the RDC.

The write controller (WRC) works with `Huffman_clk` and V_{DDH} . It checks the status of DCT-Quantization RAMs (DQRAMs) from each engine and controls writing data from DQRAMs to the arbitrator. Similar to the RDC, the WRC also maintains a status table to log the DQRAMs' status and the last block information, as shown in Figure 4.11. Once a DQRAM

¹The x indicates the engine index in the multi-instance design.

Table 4.1: Some DP-CP interactive signals in RDC

Signal	I/O	Description
Safifo_rdc_full_x ¹	I	Notify the source AFIFO is full
Safifo_rdc_empty_x	I	Notify the source AFIFO is empty
Rdc_1ddct_last_x	O	Notify the 1 st 1D-DCT the current data block is the last block of an image frame. 1D-DCT will forward this signal to the 2 nd 1D-DCT

**Figure 4.10: Pseudo code algorithm for RDC**

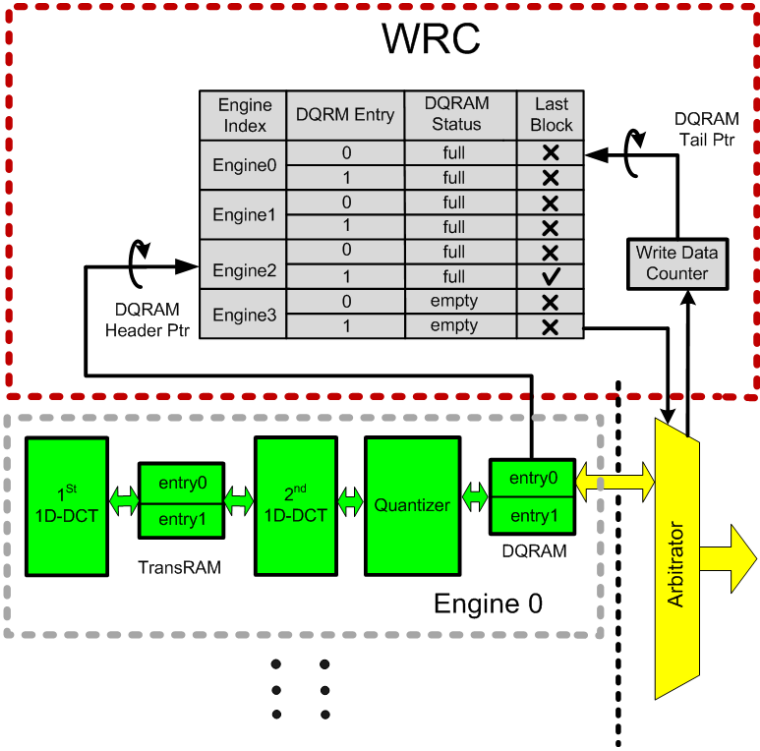


Figure 4.11: Write controller diagram


x : Engine index [0,1,2,3]

y : DQRAM entry [0,1]

```

for (  $0 \leq y \leq 1$  )
{
    for (  $0 \leq x \leq 3$  )
    {
        if (Engine_x.DQRAM_y is full)
        {
            update DQRAM status column in table;
            move head_ptr;
        }
        x++;
    }
    y++;
}

```

Thread 1:
 log the status of writing
from quantizers to
engines' DQRAMs

```

for (  $0 \leq y \leq 1$  )
{
    for (  $0 \leq x \leq 3$  )
    {
        if (Engine_x.DQRAM_y is full && entropy encoder is idle)
        {
            push data from Engine_x.DQRAM_y to arbitrator;
            update DQRAM status column in table;
            move tail_ptr;
        }
        x++;
    }
    y++;
}

```


Thread 2:
 control reading data
from engines' DQRAMs
to arbitrator

Figure 4.12: Pseudo code algorithm for WRC

Table 4.2: Some DP-CP interactive signals in WRC

Signal	I/O	Description
Dgram_wrc_valid_x_0	I	Notify the WRC that engine x's first entry of DGRAM is full
Dgram_wrc_valid_x_1	I	Notify the WRC that engine x's second entry of DGRAM is full
Dgram_wrc_last_x_0	I	Notify the WRC that engine x's first DGRAM entry contains the last data block. This signal is forwarded from 2D-DCT engine
Dgram_wrc_last_x_1	I	Notify the WRC that engine x's first DGRAM entry is last data block. This signal is forwarded from 2D-DCT engine
Encoder_dgram_rd_en_x	O	Notify the arbitrator that one of engine x's DGRAM entries is ready to push out data
Encoder_dgram_rd_fin_x	O	Notify the arbitrator that one of engine x's DGRAM entries has been pushed out
Encoder_dgram_last	O	Notify the arbitrator that current DGRAM entry contains the last data block for the current frame

entry of an engine is full, the header pointer will move to the next engine's DGRAM entry and the DGRAMs' status will be updated. If the entropy encoder is idle, the WRC will indicate to the arbitrator to push the data out of a DGRAM entry of an engine, which is pointed to by the tail pointer. Once the data in the entry is completely pushed out, the DGRAMs' status will be updated and the tail pointer will jump to the next engine's DGRAM entry. In this way the engines' DGRAMs are circulated for writing and reading. Figure 4.12 is the pseudo code algorithm for WRC. Table 4.2 lists some signals that interact from the data path and control path in the WRC.

4.3.4 Data Path Design

Figure 4.13 shows the data path in *SubJPEG*. AFIFOs are located at the front and back of the data-path to enable a flexible interface to a commercial standard bus interface. The AFIFOs are connected with `bus_clk`, `engine_clk` and operated with V_{DDH} . The intermediate results produced from the 1st 1D-DCT are stored in the “Transpose Memory” (TransRAM). The TransRAM is a dual ported RAM. While the TransRAM is written in row-major order, the 2nd 1D-DCT processing reads data from the TransRAM in a column-major order, effectively performing a transposition of the intermediate results. The TransRAM of each engine contains 2 block RAM entries, which enables a macro-level pipelined processing to enhance throughput. That is, the 1st 1D-DCT can start processing and writing intermediate output into one entry while the 2nd 1D-DCT is still reading data from the other entry. The pipeline latency for 1D-DCT is 80 `engine_clk` cycles. The output from the 2nd 1D-DCT goes to the quantizer. After the quantization process, the data is stored in DQRAM. For the same reason as with the TransRAM, the DQRAM of each engine also contains two block RAM entries. The engines are connected with `engine_clk` and V_{DDL} . Finally, the arbitrator selects data from each entry, and sends the data to the Huffman coder for entropy coding. The Huffman encoder is connected with `Huffman_clk` and V_{DDH} .

The size of each TransRAM bank is 768bits ($12b \times 64$). It can be implemented with either DFF-based register file or fast dual-ported SRAM-based register file. To compare the two design choices, we analyzed their power consumptions with a nominal 1.2V supply. The SRAM-based register file is

generated by tools from a commercial low-power SRAM vendor. The DFF-based register file is synthesized with standard cells. The results are shown in Table 4.3. In terms of leakage energy, the SRAM-based register file is better than the DFF-based register file, due to SRAM's high layout density, small silicon area and the use of HV_T process. However, in terms of dynamic energy consumption, the SRAM-based register file is worse than the DFF-based register file. This is because the energy overhead from SRAM's peripheral read-out circuitry, such as the sense-amplifiers, dominates the dynamic energy when the memory size is very small¹. In addition, since the voltage scalability of DFF-based register file is superior to that of the SRAM-based register file, we decided to implement TransRAMs with DFFs. We did not adopt the existing sub-threshold memory solutions [23]-[30], because all these solutions severely degrade speed and energy efficiency when compared to conventional SRAMs in the super-threshold mode. For the same reason, the DQRAMs are also implemented with DFFs in *SubJPEG*.

The DFF-based register files used for data storage on the data path are summarized in Table 4.4.

The engines can operate in two modes: the sub/near threshold mode and super-threshold mode. When in the sub/near threshold mode, the V_T balancers are activated. The performance estimation is given in Figure 4.14 and Figure 4.15. In Figure 4.14, the gaining factor is obtained by simulating the propagation delay and energy/cycle of a critical path on transistor level. The energy/(engine-cycle) can be reduced by $6\times$ and $9\times$ when

¹It is confirmed by NXP product team that with C065 processes a register file less than 1kb is normally implemented with standard cells.

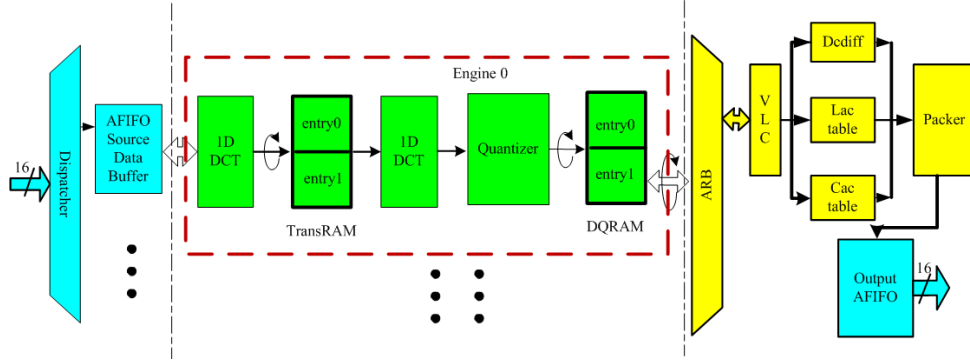


Figure 4.13: Data path diagram

Table 4.3: Memory design choices

Design choices	Dynamic power (uW/MHz) ($\max(\text{RD}, \text{WR})$, 50% address and data lines are switching)	Leakage (uA)	V_{DD} scalability
Commercial SRAMs	5.0	0.5	Very bad
DFF-based	4.4	1.8	Good

Table 4.4: Register files used in *SubJPEG* data path

Register files	V_{DD} , clk(s)	Description
AFIFO Source Data	V_{DDH} , bus_clk, engine_clk	Input buffer, 8×64 bits for each engine
TransRAM	V_{DDL} , engine_clk	12×64 bits per entry, 2 entries per engine
DQGRAM	V_{DDL} , engine_clk	8×64 bits per entry, 2 entries per engine
Output AFIFO	V_{DDH} , bus_clk, engine_clk	Output buffer, 16×4 bits

the engines are powered with a V_{DD} of 0.5V and 0.4V, respectively. The term $\text{energy}/(\text{engine}\cdot\text{cycle})$ denotes the energy consumed per cycle by a single engine during the DCT-Quantization processing. Figure 4.15 indicates the throughput vs. area tradeoff for the engines. Note that the horizontal axis represents the number of engines but not the actual total chip area. Some achievable multi-standard applications are annotated. If the application has no hard real-time constraint, such as for a digital still image camera, the V_{DD} of the engines can be scaled to a value very close to the V_{opt} which is the V_{DD} point leading to the optimal $\text{energy}/(\text{engine}\cdot\text{cycle})$.

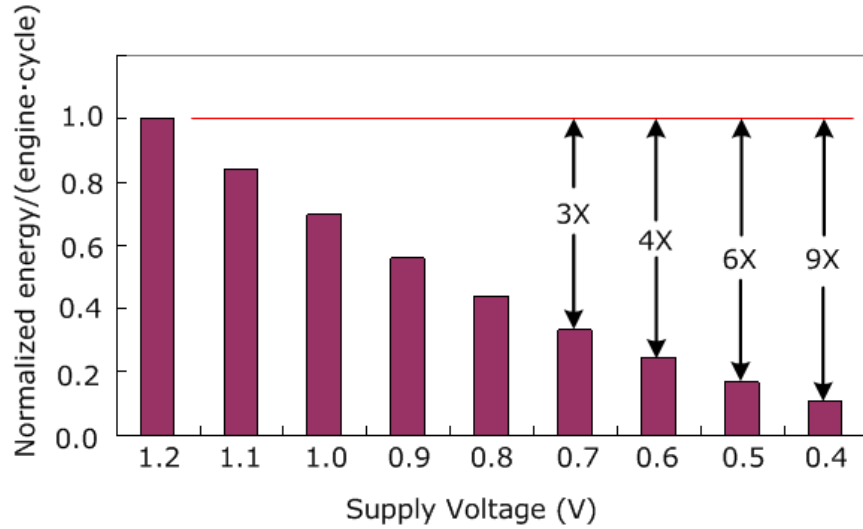


Figure 4.14: Normalized energy per cycle for each engine [$\text{energy}/(\text{engine}\cdot\text{cycle})$]

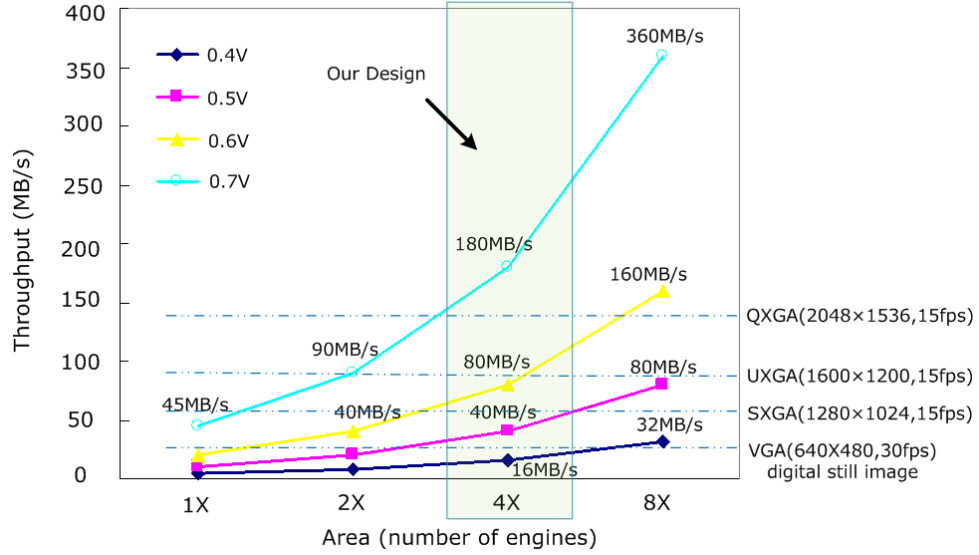


Figure 4.15: Area vs. throughput for the engines and possible real-time image applications

4.4 Implementation Issues

4.4.1 Logic Design

The IP is fully compliant with the JPEG encoder baseline standard. The synthesizable scripts include 17,000 Verilog and 1,000 Perl lines. The Perl lines are used to generate the Verilog lines for parameterized AFIFOs. Signals across different clock domains are hand-shaked to increase communication robustness. The logic synthesis is carried out with Synopsys Design Compiler. We use hierarchical synthesis approach: the engines are synthesized with a dedicated sub-threshold library, as mentioned in Chapter 3 already. The other blocks are synthesized with a conventional CMOS65 standard cell library. According to the synthesis results, the engines and the Huffman coder can

operate easily beyond 250MHz frequency with a 65nm LP-SV_T CMOS process at the nominal 1.2V supply.

Some signals in the design have to cross the V_{DDL} domain and V_{DDH} domain. Therefore, a level shifting scheme is needed. In addition, the digital I/O pads in 65nm CMOS must use a reference voltage of 1.2V, so we also need a level shifting scheme to convert the signal level from the *SubJPEG* core to the I/O pads. Shown in Figure 4.16 is the 2-stage level shift scheme used in *SubJPEG*. The 1st stage level shifting is performed through simple buffers which are capable enough of pulling up signals from sub-threshold V_{DDL} to V_{DDH} . There is no need to use the CBLC introduced in Chapter 3, as the difference between V_{DDL} and V_{DDH} is less than 300mV. The 2nd stage level shifting is performed through positive feedback structured level-shifters from V_{DDH} to 1.2V I/O pads. These level shifters are added manually into the synthesized netlist.

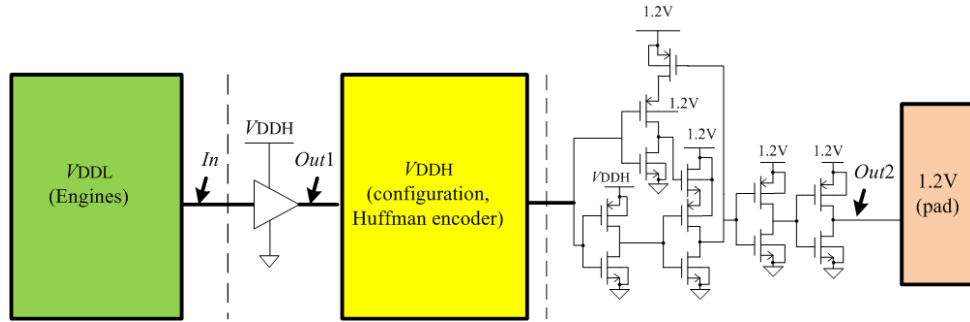


Figure 4.16: 2-stage level-shifting scheme in *SubJPEG*

Figure 4.17 shows the simulation result of the 2-stage level-shifting scheme. The first level shifter converts a signal's voltage level from 400mV to 600mV. The second level shifter converts a signal's voltage level from 600mV to 1.2V.

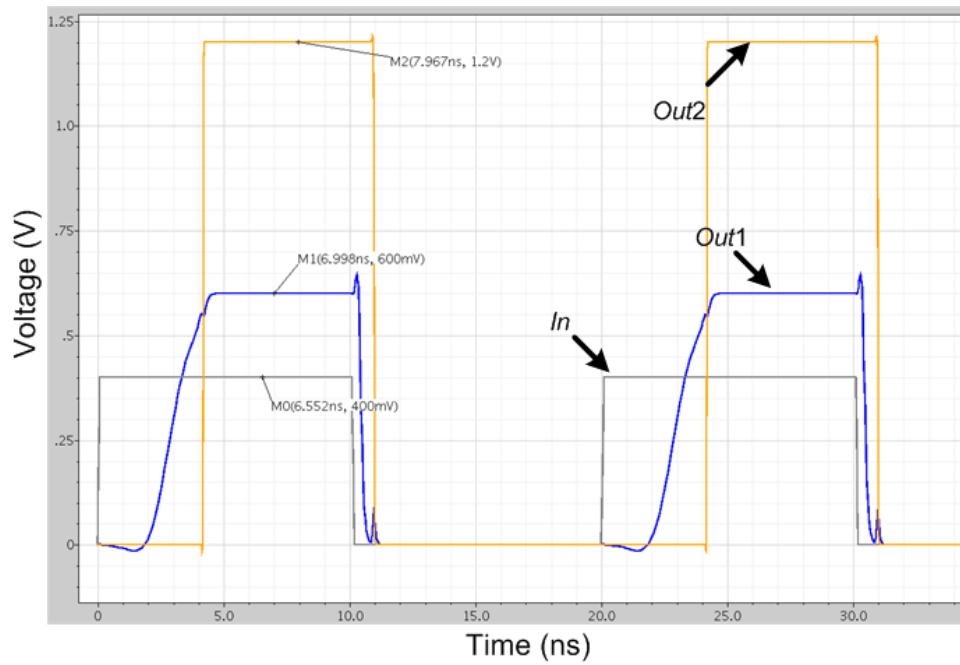


Figure 4.17: Simulation of the 2-stage level-shifting scheme (0.4V to 0.6V to 1.2V)

Compared to the delay of the 2nd stage level shifter, the delay of the 1st stage level shifter is much larger due to the fact that driving a transistor working in the super-threshold by a transistor working in the sub-threshold is far more difficult.

4.4.2 Physical Design

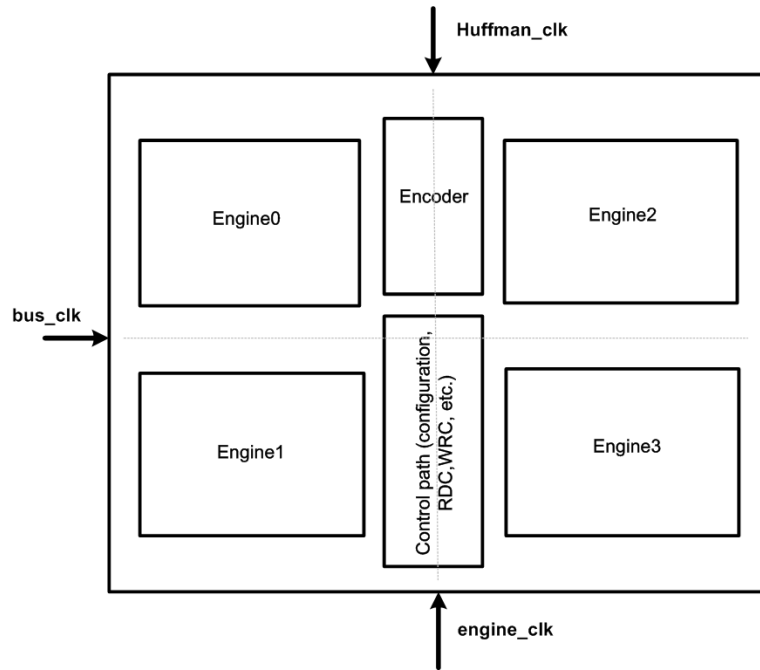


Figure 4.18: *SubJPEG* floorplan

Cadence Encounter and Virtuso tools are used for the physical design and final chip integration. After different floorplan trials, the symmetrical floorplan shown in Figure 4.18 is found to have the best timing performance after placement and routing. This is because each engine has nearly the same distance to the configuration register space and the Huffman encoder. In

addition, the three clock signals go into the core exactly at the middle points of three edges, which helps the clock tree synthesis (CKS) tool to generate well-balanced clock trees.

To implement the V_T spread control for each engine, the implementation takes four steps. First, each engine has its own underneath deep n-well to separate its bulk from the rest of the chip. Second, a special well-tap cell has been designed. Different from the traditional well-tap cell which makes connections from V_{DD} to n-well, and from G_{ND} to p-well, this special well-tap allows two individual well supply lines on n-well and p-well. Third, two power meshes in metal layer 4 have been built within each engine. The two power meshes connect to the two well supply lines of the special well-tap cells separately. The extra power meshes and deep n-wells to implement the bulk-biasing result in 9% area overhead for each engine and 2% area overhead for the entire core. Finally, each engine has its own configurable V_T balancer located at one of its corners. The two output voltages from the V_T balancer connect to the two power meshes built in step three. The reasoning why the V_T balancers are located at corners is that, the silicon realization variations are assumed planar with gradient and angle as depicted in Figure 4.19, so process corners are most likely at the chip corners.

Compared to the baseline processor, the area of SubJPEG is about $2.5\times$ larger. The area and simulated energy breakdown in the digital still image mode are shown in Fig. 4.20. The circuits that are required to parallelize the engines, i.e., dispatcher, RDC, WRC, arbiter and interface AFIFOs, occupy 8% area of the core. For digital still image processing ($V_{DDL}=0.4V$ and $V_{DDH}=0.5V$ in simulation) and $f_{Huffman_clk}=2f_{bus_clk}=4f_{engine_clk}$, these

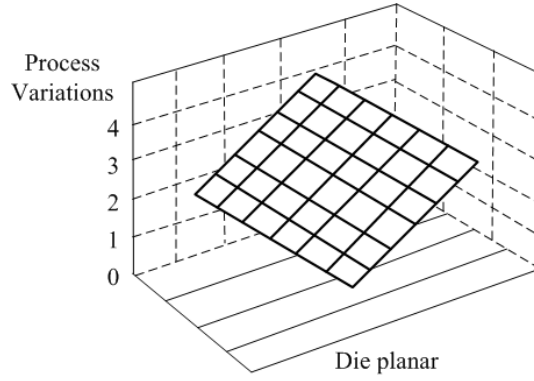


Figure 4.19: Gradient process variations

circuits would dissipate approximately 12% of the total energy.

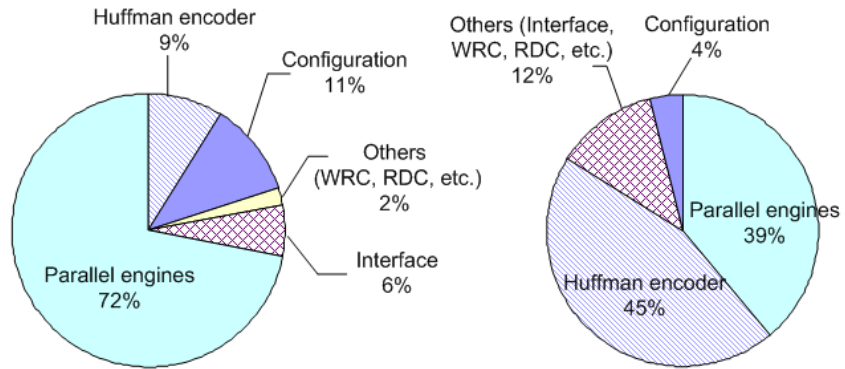


Figure 4.20: *SubJPEG* area and simulated energy breakdown in the digital still image mode

The layout of *SubJPEG* IP core integrated with the V_T balancers is shown in Figure 4.21. Since *SubJPEG* is a Direct-Memory-Access (DMA) based co-processor, it inherently has many I/O pads. To reduce the number of I/O pads, we have added additional logics to multiplex certain I/O signals. A V_{MUX} power domain working under a shift_in clock is implemented. Figure 4.22 shows the final chip layout with I/O pads. The core size is $1.4 \times 1.4 \text{ mm}^2$.

The total chip area however is no less than $2.9 \times 2.9 \text{mm}^2$ due to the large number of I/O pads. If the overall performance is not sufficient, we still can add more engines, which hardly increases the total chip area.

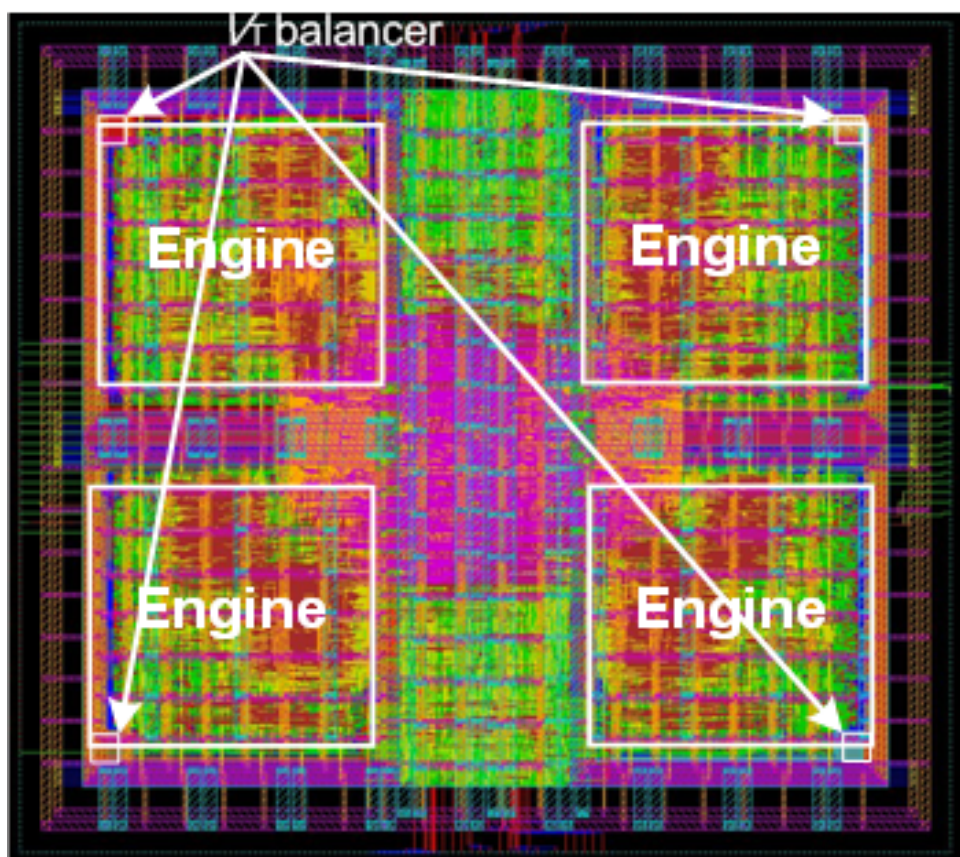


Figure 4.21: The layout of *SubJPEG* IP core integrated with the V_T balancers in Cadence Encounter view

4.5 Fabrication and Packaging

The testing chip is fabricated in TSMC 65nm 7-layer Low Power Standard V_T CMOS process. A micrograph of the prototype chip is shown in Figure

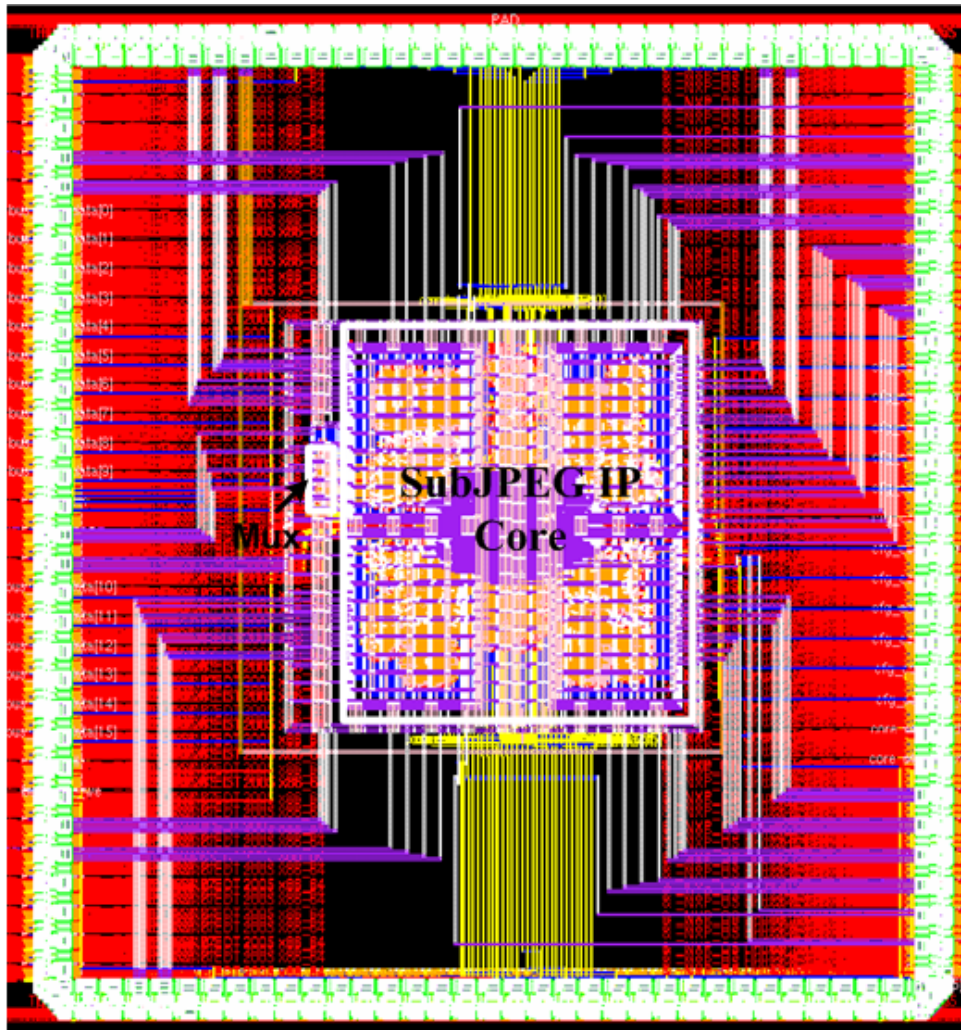


Figure 4.22: The final chip layout with I/O pads in Mentor Graphic Calibre view

4.23. The testing chip is packaged with LQFP Package/SOT486.15/W33330. Figure 4.24 shows the pin-out bonding diagram.

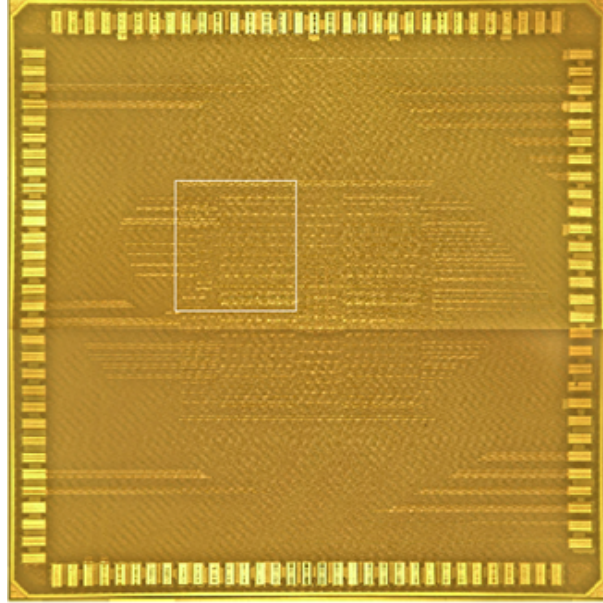


Figure 4.23: Prototype chip micrograph

4.6 Performance Evaluation

To test the functionalities of the chip, the Memec Spartan-3 LC Xilinx Development Board [50] is chosen as a mother board. A 9-layer PCB is designed as a daughter card fitted into the P160 expansion slots on top of the mother board. A testing socket is used on the PCB to accommodate the prototype chips. The 1.2V V_{ref} and 2.5V I/O voltages are generated with on-board DC-DC converters. The other supply voltages are supplied from external voltage generators. The clock signals are generated by the digital clock managers (DCM) in the Xilinx Spartan-3 FPGA chip. They can also be supplied by

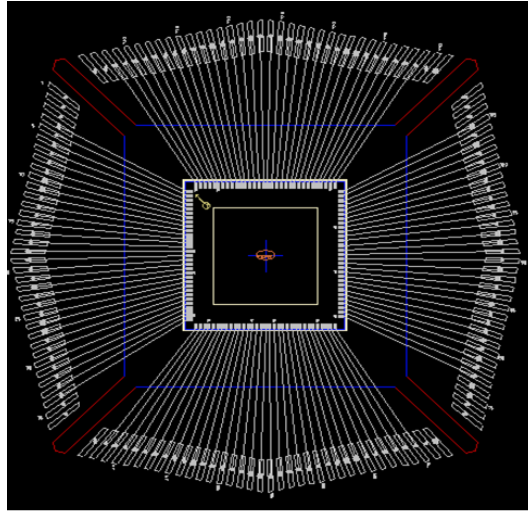


Figure 4.24: Pin-out bonding diagram

external clock generators. Figure 4.25 shows the testing boards.

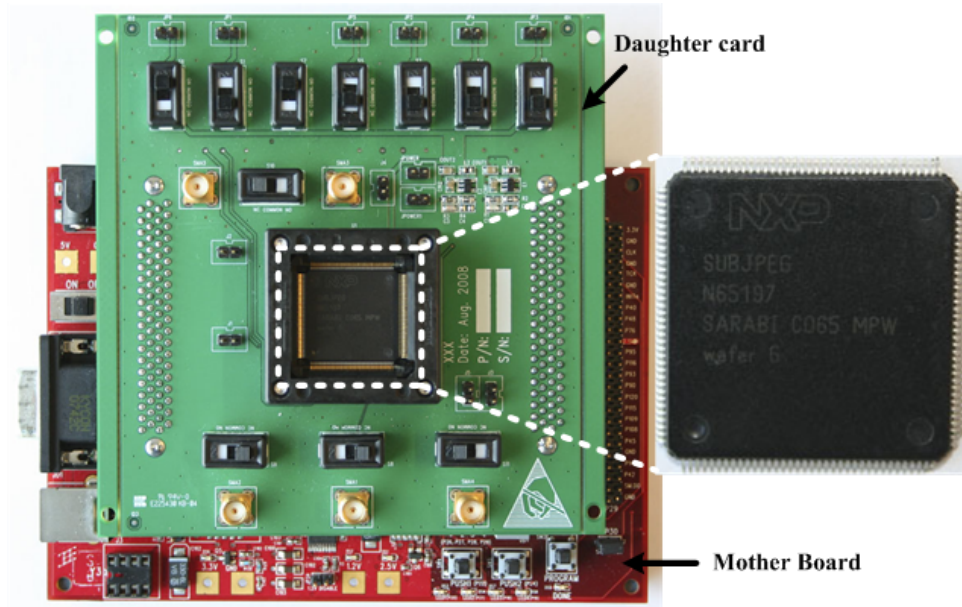


Figure 4.25: Testing boards

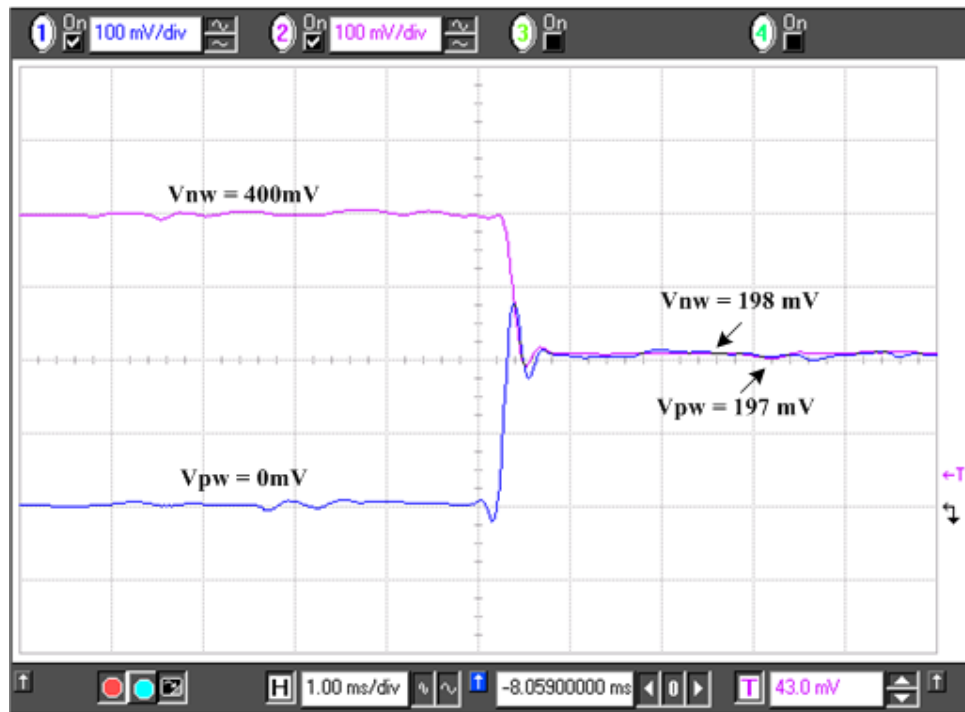
The tested items are detailed below.

1. *Configurable V_T balancer*

The behavior of the configurable V_T balancer at $V_{DD} = 400\text{mV}$ is captured by oscilloscope and shown in Figure 4.26. An off-chip capacitor is needed to mitigate ripple. As can be seen, before the V_T balancing, the n-well is connected to V_{DD} and the p-well is connected to G_{ND} . Within 1 ms after the V_T balancer is turned on, the supply voltages of both n-well and p-well converge at near $V_{DD}/2$. At $V_{DD} = 400\text{mV}$, the tested samples could not function correctly with a 2MHz engine_clk frequency without V_T balancing. With the help of V_T balancing, the samples could run at 2.5MHz. In this case, the average leakage current is increased by $2\times$. At this time, the ratio between the leakage and the dynamic energy is about $1/30$, meaning that the V_{DD} can still be further reduced to reach V_{opt} which leads to a $1/1$ ratio.

2. *Functionality verification*

On the board, *SubJPEG* is the co-processor for a Xilinx Spartan-3 FPGA chip, which functions as the main CPU. Figure 4.27 shows some logic signals analyzed by a HP logic analyzer. In this case, the bus_clk is at 20MHz, the Huffman_clk is at 10MHz and the engine_clk is at 2.5MHz. At the start, the main CPU configures *SubJPEG* by asserting “bus_core_cfg” and sending “cfg_core_data” (see Figure 4.27 (a)). When the configuration is finished, *SubJPEG* starts to issue a request signal “core_bus_req” to the bus. Each request is synchronized with the rising edge of bus_clk and lasts for 2 cycles. In the first cycle, *SubJPEG* tells the bus the starting address of the data, and in the second cycle it tells the requested length of data. The main CPU

Figure 4.26: Measurement results of switching on the V_T balancer

sends data at `bus_clk` frequency after receiving this request. The output of *SubJPEG* is synchronized with `Huffman_clk`. Once all the data is compressed successfully, *SubJPEG* appends an EOI signature “FFD9” at the end of the output stream (see Figure 4.27 (c)).

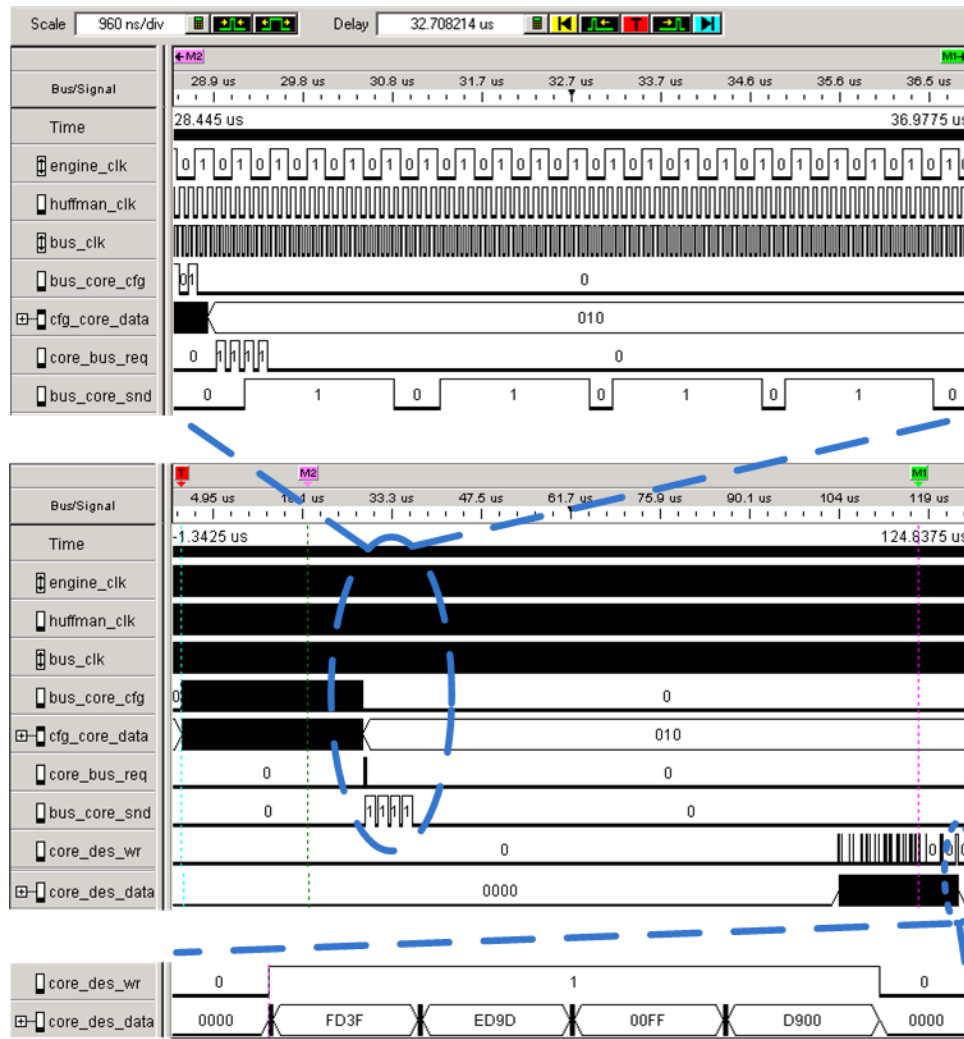


Figure 4.27: Measurement results from logic analyzer: (a)(c) are zoomed in results of (b)

3. Engine signal

The oscilloscope also captures an engine signal through an analog pad. Figure 4.28 shows the pulse trains at $V_{DDL} = 400\text{mV}$ and $V_{DDL} = 800\text{mV}$. The engine_clk is 2.5MHz for both cases.

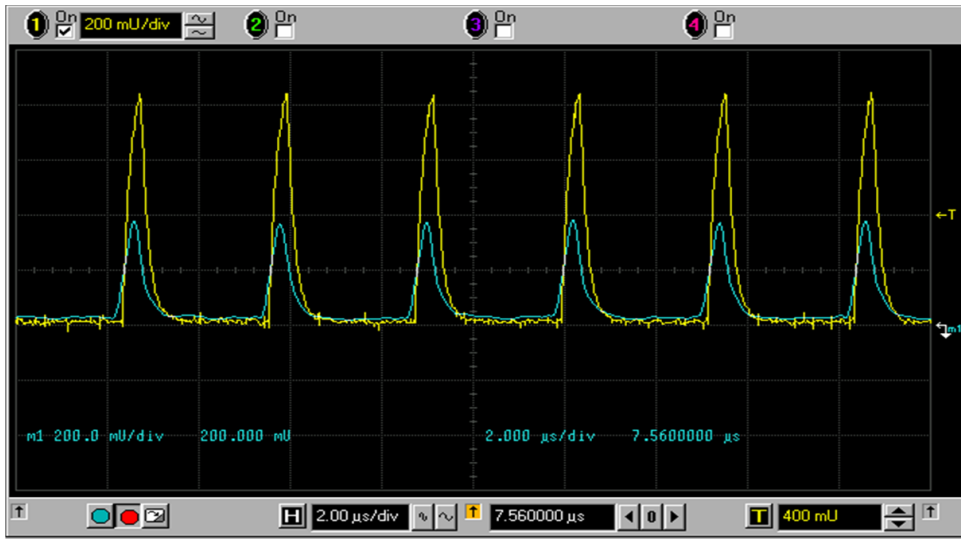


Figure 4.28: Pulse trains from engines at $V_{DDL} = 400\text{mV}$ and $V_{DDL} = 800\text{mV}$

4. Energy and throughput evaluation

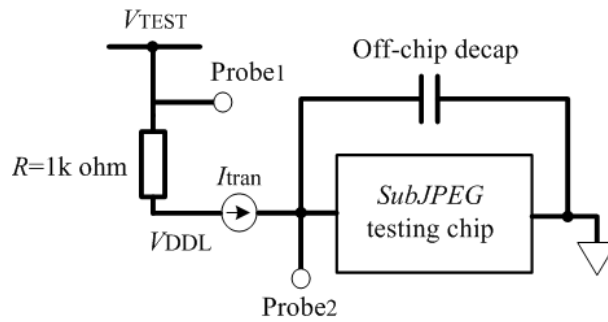


Figure 4.29: Transient current measurement scheme

Figure 4.29 illustrates how we measured the transient current for engines.

Two oscilloscope probes are attached across a $1k\Omega$ resistor. The two probes are calibrated in advance so that they have exactly the same offsets. The engines' supply V_{DDL} is adjusted through tuning the voltage source V_{TEST} . The transient current flowing into engines is therefore calculated as $I_{tran} = (V_{TEST} - V_{DDL}) \times 10^{-3}$. In the same way, the transient current for the Huffman coder can also be measured.

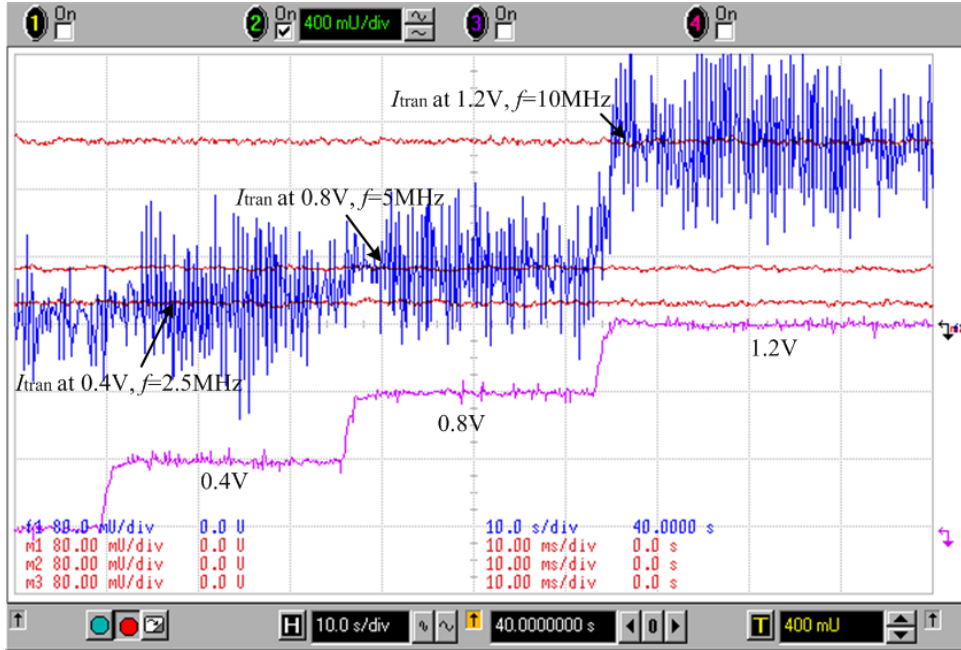


Figure 4.30: Transient and average current at (0.4V, 2.5MHz), (0.8V, 5MHz) and (1.2V, 10MHz)

Figure 4.30 shows the transient current at $V_{DDL} = 0.4\text{V}$, 0.8V , 1.2V at an engine_clk of 2.5MHz, 5MHz, 10MHz respectively. Note that 2.5MHz is the maximum operating frequency at $V_{DDL} = 0.4\text{V}$ supply, but 5MHz and 10MHz are not the maximum operating frequencies at $V_{DDL} = 0.8\text{V}$ and $V_{DDL} = 1.2\text{V}$. We also operated the engines at each $(V_{DDL}, \text{engine_clk})$

point separately for an enough long time. In this way the oscilloscope precisely measured and stored the average current data for each ($V_{DDL}, engine_clk$) point. In the snapshot, these average currents are displayed by loading them from the oscilloscope's data storage memories.

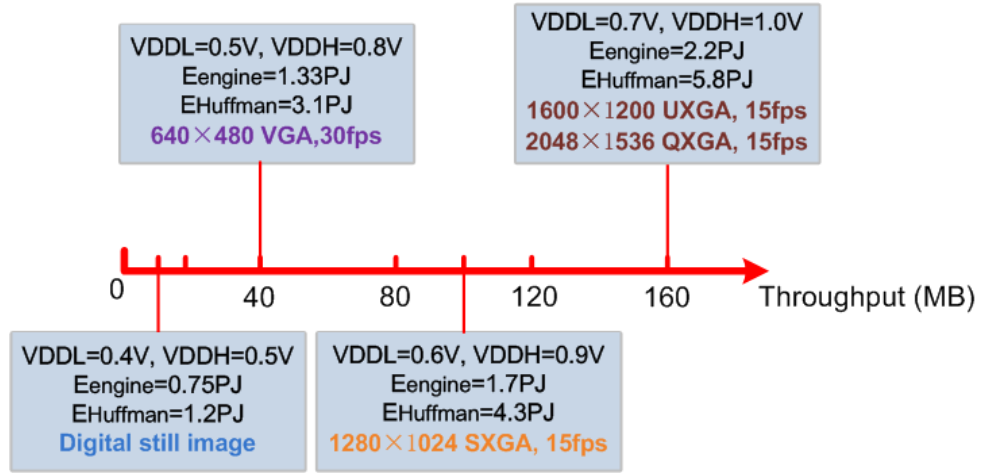


Figure 4.31: Energy per cycle for each engine [pJ/(engine-cycle)]

Figure 4.31 shows the energy/(engine-cycle) savings. More measurements of system energy and speed performance are summarized in Table 4.5 and Figure 4.32. These measurement results are close to our estimated results described in section 4.3 (refer to Figure. 4.14).

As shown in Figure 4.31 and Figure 4.32, in the sub-threshold mode the engines can operate with 2.5MHz frequency at 0.4V, with 0.75pJ/(engine-cycle). This leads to $8.3\times$ energy/(engine-cycle) reduction compared to operating at the 1.2V nominal supply. Correspondingly, the Huffman coder should be operated at 10MHz at 0.5V, with 1.2 pJ per entropy encoding cycle. In the near-threshold mode the engines can operate with 4.5MHz frequency at

Table 4.5: System throughput and possible image applications

Engine Mode	$V_{DDL}(V)$	$V_{DDH}(V)$	Throughput (MB/s)	Possible Applications
Sub-threshold	0.4	0.5-0.55	10 (2.5MHz clock)	digital still image
Near-threshold	0.45	<0.7	18 (4.5MHz clock)	VGA (640×480, 15fps)
	0.5	0.8	40 (10MHz clock)	VGA (640×480, 30fps)
Super-threshold	0.6	0.9	100 (25MHz clock)	SXGA(1280×1024, 15fps)
	0.7	1.0	160 (40MHz clock)	UXGA(1600×1200, 15fps) QXGA(2048×1536, 15fps)

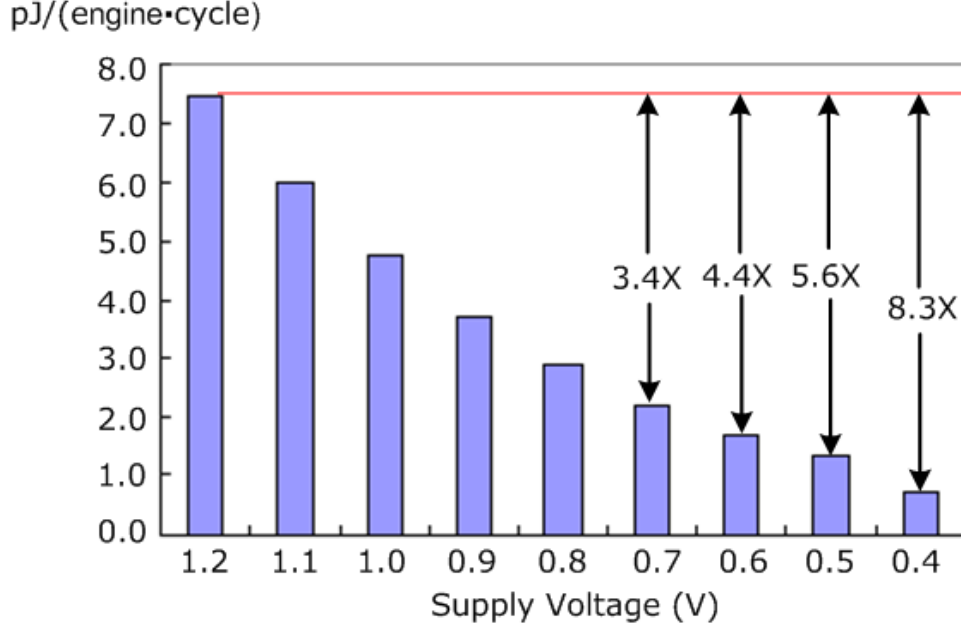


Figure 4.32: System energy and throughput

0.45V, and consume about 1.1pJ/(engine-cycle). The Huffman coder operates at 18MHz frequency with a less than 0.7V supply, and the energy/cycle dissipation is around 2.0 pJ. The overall system throughput meets the 15fps 640×480 VGA compression requirement. By further increasing both V_{DDH} and V_{DDL} , and making different (V_{DDH}, V_{DDL}) combinations, the prototype chip can achieve multi-standard image encoding.

5. Testing limitation

In total 70 packaged samples are requested and tested. Among which 8 samples burned out due to shorted bonding wires. The testing limitation for the rest of samples is the lowest V_{DDH} that the 2nd stage level shifters can tolerate. For most of the samples, the 2nd stage level shifters start to have erroneous function when V_{DDH} is lower than 0.6~0.65V. This lowest V_{DDH}

limitation affects directly the lowest V_{DDL} that the 1st stage level shifters can handle. So we cannot operate the engines with a V_{DDL} lower than 0.4V, in spite of the fact that it is quite likely that the engines still function correctly below 0.4V. This suggests future design leave more design margin.

Chapter 5

Conclusions, Future Work and Discussions

Finalizing this thesis, in this chapter we make conclusions of our work. We also introduce further research which we will explore. At the end, we will discuss the future trends and some open issues of sub/near threshold design techniques.

5.1 Conclusions

Voltage scaling is one of the most effective and straightforward means for CMOS digital circuits' energy reduction. Aggressive voltage scaling to the near or sub-threshold level helps achieving ultra-low energy consumption. However, it brings along big challenges to reach a required throughput and to have good tolerance of process variations. This thesis presents our research work in the design of robust near/sub-threshold digital circuit. Our work has two features. First, unlike the other research work that believes sub-threshold

operation is only suitable for low-frequency low-throughput applications, we use architectural-level parallelism to fix the throughput degradation, so a medium throughput can be reached. Second, several new techniques are proposed to mitigate the yield degradation due to process variations. These techniques include:

- Configurable V_T balancers to control the V_T spread. When facing process corners in the sub-threshold, our balancer will balance the V_T of p/nMOS transistors through bulk-biasing, so the yield can be boosted.
- Transistor sizing to combat V_T mismatch between transistors. This is necessary when the circuit needs to be operated with deep sub-threshold supply voltage, i.e., below 250mV for 65nm SV_T CMOS process.
- Improving sub-threshold drivability by exploiting V_T mismatch of parallel transistors. While the V_T mismatch between parallel transistors is notorious, we propose to utilize it to boost the driving current in the sub-threshold region. This interesting approach also suggests a multiple-finger layout style, which helps reducing silicon area considerably.
- Selecting reliable library cells for logic synthesis in the sub-threshold region. Standard library cells that are sensitive to process variations must be eliminated in synthesis flow. We provided the basic guideline to select reliable standard cells.
- The method that turns risky ratioed logic, such as latch and register, into non-ratioed logic.

An ultra low-energy multi-standard JPEG encoder co-processor with a sub/near threshold power supply has been designed and implemented to demonstrate all the ideas. This 8-bit resolution DMA based co-processor has multiple power domains and multiple clock domains. It uses 4 parallel DCT-Quantization engines in the data path. Instruction-level parallelism is also used. The parallelism is implemented in an efficient manner to minimize the associated area overhead. Details about this co-processor architecture and implementation are also covered in this thesis. The prototype chip is fabricated in TSMC 65nm 7-layer standard V_T CMOS process. The core area is $1.4 \times 1.4 \text{mm}^2$. Each engine has its own V_T balancer. Each V_T balancer is $25 \times 30 \mu\text{m}^2$. The measurement results show that our V_T balancer has very good balancing effect. In the sub-threshold mode the engines can operate with 2.5MHz frequency at 0.4V V_{DD} . At this time, a single engine consumes 0.75pJ energy per cycle for DCT and quantization processing, i.e., 0.75pJ/(engine-cycle). This leads to $8.3 \times$ energy/(engine-cycle) reduction compared to operating at the 1.2V nominal supply. In the near-threshold mode the engines can operate with 4.5MHz frequency at 0.45V, with about 1.1pJ/(engine-cycle). The overall system throughput still meets 15fps VGA compression requirement. By further increasing the supply, the prototype chip can satisfy multi-standard image encoding.

5.2 Future Work

We will explore an ultra-low energy near/sub-threshold DSP processor which provides good flexibility for many applications. This DSP architecture will

feature efficient parallelism and pipelining. It will also support an extended instruction set to support some biomedical applications, such as EEG, ECG, etc. In the future we will also explore the ultra-low voltage SRAMs. Currently our *SubJPEG* co-processor is fully based on standard cells, so the intermediate data is stored in DFF-based register banks. Because SRAMs have much higher transistor density, the cost of silicon area and energy can be reduced considerably if the amount of data storage required in DSP processing is large. It is also very interesting to design a DC-DC converter capable of converting the nominal supply to a near/sub threshold power supply with reasonably high conversion efficiency. To the best of our knowledge, the only DC-DC converter for such purpose has been introduced in [38] [39] , but its conversion efficiency is less than 75%, meaning that a quarter of energy has been lost during the conversion. Effort should be made to increase this energy efficiency. Even better is a programmable solution which can generate different voltages for different applications. A complete system including the DSP with embedded SRAMs and DC-DC converter will be implemented in the very near future.

5.3 Discussions: Are we ready for sub-threshold?

The topic of sub-threshold circuits has been discussed since the 1970's when the minimum supply voltage was theorized based on various sub-threshold models [57] . It has then been used only in simple designs, such as wrist watches and hearing aids. After 30 years' quiescence, it started to draw researchers' attention again from 2004 onward, when Alice Wang and Ben

Calhoun from M.I.T worked out the modeling the optimum voltage to minimize energy dissipation in deep sub-micron process and designed the first 180mV sub-threshold processor in 180nm CMOS. Their successful work has motivated investigation of ultra low energy VLSI in deep sub-micron technology, such as distributed micro-sensors or medical devices, where minimizing energy dissipation is the primary concern. Realizing the potential for sub-threshold technique to succeed in the ultra low energy market, the industrial leaders have already started competition to push it into their products as soon as possible. In 2008, IBM Research and University of Michigan have jointly released the sub-threshold sensor node processor [32] . A few months later, Texas Instruments' low power MCU group had announced the newest version MSP430 [58] , the world's lowest energy 16-bit RISC sub-threshold processor as a flexible solution for a wide range of low power and portable applications. In 2009, Intel had announced its 300mV ultra low power reconfigurable 4-way SIMD vector processor in 45nm CMOS [59] . Because of the huge market for ultra low energy applications, there is no doubt that more and more IC companies will involve in this competition.

However, compared to the super-threshold technique, designing in the sub-threshold is still facing many challenges. Overcoming these challenges needs a wide collaborated research at every design hierarchy level. Some of the necessary research work is listed below:

1. EDA support covering the entire design flow from front-end to back-end design for optimized sub-threshold designs. Ideally, the EDA tools should automate the sub-threshold digital design. It must take the

circuit's reliability and yield into consideration, and help designers to tradeoff among reliability, area, energy and speed performance. It should also support different levels of verification at design time.

2. More efficient system architectures to minimize the area cost, to relieve the integration issue and enhance the fault-tolerance capability. For example, how to make a good tradeoff among throughput, energy and degree of parallelism.
3. Customized digital modules for robust sub-threshold operation, such as the sub-threshold SRAMs. It is also nice to have an optimized sub-threshold standard cell library provided by foundries.
4. Supportive on-chip ultra-low voltage analog components such as DC-DC, DLL, etc.
5. Special CMOS process technology suitable to the sub-threshold design may also be an option. As the feature size is scaling to 45nm, 32nm and 22nm, it is also possible that the foundries may have solutions for a process with customized V_T and less V_T variation to ease sub-threshold design.
6. Special packaging that supports quick heat removal, hence stabilizing die temperature, and preventing reliability degradation due to interferences such as radiation, hotspots, etc.

I believe that the sub-threshold technique will soon become popular in the low energy market, judging from the strong momentum it has been developed

5.3. DISCUSSIONS: ARE WE READY FOR SUB-THRESHOLD?

over the past 5 years. In the next 5~10 years, I expect that the world will be ready for the sub-threshold chipsets.

Bibliography

- [1] Gordon E. Moore, “The microprocessor: engine of the technology revolution,” *Communication of the ACM*, Vol.40, No.2, pp.112-114, Feb 1997.
- [2] V. De and S. Borkar, “Technology and design challenges for low power and high performance,” *Proc. International Symposium on Low Power Electronics and Design (ISLPED)*, pp.163-168, Aug 1999.
- [3] Siva Narendra, Vivek De, Shekhar Borkar, Dimitri A.Antoniadis, Anantha P.Chandrakasan, “Full-chip subthreshold leakage power prediction and reduction techniques for sub-0.18 μ m CMOS”, *IEEE Journal of Solid-State Circuits (JSSC)*, Vol.39, No.2, pp. 501-510, Feb 2004.
- [4] T. Burd and R. Brodersen, “Design issues for dynamic voltage scaling,” *Proc. International Symposium on Low Power Electronics and Design (ISLPED)*, pp.9–14, Jul 2000.

- [5] T. Burd, T. Pering, A. Stratakos, R. Brodersen, "A dynamic voltage scaled microprocessor system," *Proc. IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 294–295, Feb 2000.
- [6] Intel XScale microarchitecture.
<http://developer.intel.com/design/intelxscale/>
- [7] Transmeta Crusoe microarchitecture.
<http://www.transmeta.com>
- [8] Nakai, M., Akui, S., Seno, K., Meguro, T., Seki, T., Kondo, T., Hashiguchi, A., Kawahara, H., Kumano, K., Shimura, M., "Dynamic voltage and frequency management for a low-power embedded microprocessor," *IEEE Journal of Solid-State Circuits (JSSC)*, Volume 40, Issue 1, pp. 28 – 35, Jan 2005.
- [9] Samsung Semiconductors.
<http://www.samsung.com/global/business/semiconductor>
- [10] Jan.M.Rabaey, and the PicoRadio Group, "Ultra-low power computation and communication enables ambient intelligence," technical report, Dept.of EECS, University California Berkeley.
- [11] P. Pentland et al., "The digital doctor: an experiment in wearable telemedicine," *Proc. 1st International Symposium Wearable Computers*, pp. 173–174, 1997.
- [12] T. Starner, "Human-powered wearable computing," *IBM Systems Journal*, vol. 35, pp. 618–629, 1996.

- [13] Bo Zhai, David Blaauw, Dennis Sylvester, Krisztian Flautner, "The limit of dynamic voltage scaling and insomniac dynamic voltage scaling," *IEEE Transactions on Very Large Scale Integration Systems (T-VLSI)*, Vol.13, No.11, Nov 2005.
- [14] Benton H. Calhoun, Anantha P. Chandrakasan, "Ultra-dynamic voltage scaling using sub-threshold operation and local voltage dithering in 90nm CMOS," *IEEE Journal of Solid-State Circuits (JSSC)*, Vol. 41, No.1, pp. 238–245, Jan 2006.
- [15] Benton H. Calhoun, Alice Wang, and Anantha Chandrakasan, "Modeling and sizing for minimum energy operation in subthreshold circuits," *IEEE Journal of Solid-State Circuits (JSSC)*, Vol.40, No.9, pp. 1778-1786, Sep 2005.
- [16] Alice Wang, Anantha P. Chandrakasan, Stephen V. Kosonocky, "Optimal supply and threshold scaling for subthreshold CMOS circuits," *Proc. IEEE Computer Society Annual Symposium on VLSI*, pp.5-9, Apr 2002.
- [17] Benton H. Calhoun, Anantha P. Chandrakasan, "Characterizing and modeling minimum energy operation for subthreshold circuits," *Proc. International Symposium on Low Power Electronics and Design (ISLPED)*, pp.90-95, Aug 2004.
- [18] Bo Zhai, Scott Hanson, David Blaauw, Dennis Sylvester, "Analysis and Mitigation of Variability in Sub-threshold Design," *Proc.*

International Symposium on Low Power Electronics and Design (ISLPED), pp.20-25, Aug 2005.

- [19] John Keane, Hanyong Eom, Tae-Hyoung Kim, Sachin Sapatnekar, Chris Kim, "Subthreshold logical effort: a systematic framework for optimal subthreshold device sizing," *Proc. Design Automation Conference (DAC)*, pp.425-428, Jul 2006.
- [20] Benton H. Calhoun, Alice Wang, Anantha P. Chandrakasan, "Device sizing for minimum energy operation in subthreshold circuits," *Proc. IEEE Custom Integrated Circuits Conference (CICC)*, pp. 95-98, Oct 2004.
- [21] Joyce Kwong, Anantha P. Chandrakasan, "Variation-Driven Device Sizing for Minimum Energy Sub-threshold Circuits," *Proc. International Symposium on Low Power Electronics and Design (ISLPED)*, pp.8-13, Oct 2006.
- [22] Hendrawan Soeleman and Kaushik Roy, "Ultra-low power digital subthreshold logic circuits," *Proc. International Symposium on Low Power Electronics and Design (ISLPED)*, pp.94-96, Aug1999.
- [23] Benton H. Calhoun, Anantha P. Chandrakasan, "Analyzing static noise margin for sub-threshold SRAM in 65nm CMOS," *Proc. IEEE European Solid State Circuits Conference (ESSCIRC)*, pp.363-366, Sept 2005.

- [24] Benton H. Calhoun, Anantha P. Chandrakasan, "A 256kb Sub-threshold SRAM in 65nm CMOS," *Proc. IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 2592–2601, Feb 2006.
- [25] Jinhui Chen, Lawrence T. Clark, and Tai-Hua Chen, "An ultra-low-power memory with a subthreshold power supply voltage," *IEEE Journal of Solid-State Circuits (JSSC)*, Vol.41, No.10, pp.2344–2353, Oct 2006.
- [26] Tae-Hyoung Kim, Jason Liu, John Keane, Chris H. Kim, "A high-density subthreshold SRAM with data-independent bitline leakage and virtual ground replica scheme," *Proc. IEEE International Solid-State Circuits Conference (ISSCC)*, pp.330–606, Feb 2007.
- [27] Bo Zhai, Scott Hanson, David Blaauw, Dennis Sylvester, "A sub-200mV 6T SRAM in 0.13nm CMOS," *Proc. IEEE International Solid-State Circuits Conference (ISSCC)*, pp.332–606, Feb 2007.
- [28] Bo Zhai, Scott Hanson, David Blaauw, Dennis Sylvester, "A Variation-Tolerant sub-200mV 6T subthreshold SRAM," *IEEE Journal of Solid-State Circuits (JSSC)*, Vol. 43, No. 10, pp. 2338 – 2348, Oct 2008.
- [29] Naveen Verma, Anantha P. Chandrakasan, "A 256 kb 65 nm 8T Subthreshold SRAM Employing Sense-Amplifier Redundancy," *IEEE Journal of Solid-State Circuits (JSSC)*, Vol. 43, No. 1, pp. 141-149, Jan 2008.

- [30] Naveen Verma, Anantha P. Chandrakasan, "A 65nm 8T sub-Vt SRAM employing sense-amplifier redundancy," *Proc. IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 328-329, Feb 2007.
- [31] Leyla Nazhandali, Bo Zhai, Javin Olson, Anna Reeves, Michael Minuth, Ryan Helfand, Sanjay Pant, Todd Austinand, David Blaauw, "Energy optimization of subthreshold-voltage sensor network processors," *Proc. International Symposium on Computer Architecture (ISCA)*, pp. 197-207, Jun 2005.
- [32] Bo Zhai, et al., "A 2.60pJ/inst subthreshold sensor processor for optimal energy efficiency", *Proc. IEEE Symposium on VLSI Circuits*, pp. 154-155, Jun 2006.
- [33] Alice Wang, Anantha P. Chandrakasan, "A 180mV FFT Processor Using Subthreshold Circuit Techniques," *Proc. IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 292-293, Feb 2004.
- [34] Alice Wang, and Anantha Chandrakasan, "A 180mV subthreshold FFT processor using a minimum energy design methodology," *IEEE Journal of Solid-State Circuits (JSSC)*, Vol.40, No.1, pp. 310-319, Jan 2005.
- [35] Vivienne Sze, Raul Blaquez, Manish Bhardwaj, Anantha Chandrakasan, "An energy efficient sub-threshold baseband processor architecture for pulsed ultra-wideband communications," *Proc. IEEE*

- International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp.14-19, May 2006.
- [36] Vivienne Sze, Anantha P. Chandrakasan, "A 0.4-V UWB baseband processor," *Proc. International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 262-267, Aug 2007.
- [37] Myeong-Eun Hwang, Raychowdhury, A. Keejong Kim Roy, K., "A 85mV 40nW process-tolerant subthreshold 8×8 FIR Filter in 130nm Technology," *Proc. IEEE Symposium on VLSI Circuits*, pp. 154-155, Jun 2007.
- [38] J. Kwong, Y. Ramadass, N. Verma, M. Koesler, K. Huber, H. Moormann, A. Chandrakasan, "A 65nm sub-Vt microcontroller with integrated SRAM and switched-capacitor DC-DC converter," *Proc. IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 318-319, Feb 2008.
- [39] Joyce Kwong, Yogesh K. Ramadass, Naveen Verma, Anantha P. Chandrakasan, "A 65 nm Sub-Vt microcontroller with integrated SRAM and switched capacitor DC-DC converter," *IEEE Journal of Solid-State Circuits (JSSC)*, vol. 44, no. 1, pp. 115-126, Jan 2009.
- [40] S.Mukhopadhyay, C. Neau, R. Cakici, A. Agarwal, C. Kim, K.Roy, "Gate leakage reduction for scaled devices using transistor stacking," *IEEE Transaction on Very Large Scale Integration Systems (TVLSI)*, Vol.11, No.4, Aug 2003.

- [41] José Pineda de Gyvez and Hans P.Tuinhout, “Threshold voltage mismatch and intra-Die leakage current in digital CMOS circuits,” *IEEE Journal of Solid-State Circuits (JSSC)*, vol.39, No.1, pp.157–168, Jan 2004.
- [42] C. F. Fang and R. A. Rutenbar and M.Puschel and T. Chen, “Towards efficient static analysis of finite precision effects in DSP applications via affine arithmetic modeling,” *Proc. Design Automation Conference (DAC)*, pp.496-501, Jun 2003.
- [43] C. F. Fang, R. A. Rutenbar, M. Puschel and T. Chen, “Fast, accurate static analysis for fixed-point finite-precision effects in DSP designs,” *Proc. of the International Conference on Computer Aided Design (ICCAD)*, pp.275-282, Nov 2003.
- [44] D-U. Lee, A. A. Gaffar, O. Mencer, W. Luk, “MiniBit: bitwidth optimization via affine arithmetic,” *Proc. Design Automation Conference (DAC)*, pp. 837-840, Jun 2005.
- [45] J. Stolfi and L.H. de Figueiredo, “An introduction to affine arithmetic,” *TEMA Tend. Mat. Apl. Comput.*, Vol. 4, No.3, pp. 297-312, 2003.
- [46] Goichi Ono, and Masayuki Miyazaki, “Threshold-voltage balance for minimum supply operation,” *IEEE Journal of Solid-State Circuits (JSSC)*, Vol.38, No.5, pp. 830-833, May 2003.

- [47] M. Pelgrom, A. Duinmaijer, A. Welbers, "Matching properties of MOS transistors," *IEEE Journal of Solid-State Circuits (JSSC)*, Vol.24, No.5, pp.1433–1439, Oct 1989.
- [48] Gregory K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions Consumer Electronics*, Vol.38, No.1, pp. xviii-xxxiv, Feb1992.
- [49] Digital Compression and Coding of Continuous Still Images, Part 1, Requirements and Guidelines. ISO/IEC JTC1 Draft International Standard 10918-1, Nov 1991.
- [50] Memec board user manual.
<http://www.em.avnet.com/>
- [51] Thomas Olivier, Valentian Alexandre, Vladimirescu Andrei, Amara Amara, "An accurate estimation model for subthreshold CMOS SOI logic," *Proc. IEEE European Solid State Circuits Conference (ESS-CIRC)*, pp. 275-278, Sept 2002.
- [52] Anantha Chandrakasan, Robert W. Brodersen, "Low-power CMOS design," Wiley-IEEE Press, January 1998.
- [53] Mohamed W. Allam, "New methodologies for low-power high-performance digital VLSI design," PhD thesis, University of Waterloo, Canada, 2000.

- [54] Junfeng Zhou, “Development of a low power digital logic family based on subthreshold currents,” master thesis, Katholieke Universiteit Leuven, 2004.
- [55] “SI-EMC: Bounce and IR-drop estimation,” Process Dependent Document for CMOS065LP (Internal design manual), NXP Semiconductors.
- [56] G. Sery, S. Borkar, V. De, “Life is CMOS: why chase the life after?” *Proc. IEEE Design Automation Conference (DAC)*, pp. 78-83, Jun 2002.
- [57] Alice Wang, Ben.H.Calhoun, Anantha P. Chandrakasan, “Subthreshold design for ultra-low power systems,” Springer, ISBN: 9780387335155
- [58] exas Instruments MSP430 MCU.
<http://focus.ti.com/mcu/docs/>
- [59] Himanshu Kaul, Mark.A.Anders, Sanu K. Mathew, Steven K.Hsu, Amit Agarwal, Ram K.Krishnamurthy, Shekhar Borkar, “A 300mV 494GOPS/W reconfigurable dual-supply 4-way SIMD vector processing accelerator in 45nm CMOS,” *Proc. IEEE International Solid-State Circuits Conference (ISSCC)*, pp.260–263, Feb 2009.
- [60] H. Soeleman, K. Roy, and B. Paul, “Sub-domino logic: Ultra-low power dynamic subthreshold digital logic ,” *IEEE International Conference on VLSI Design*,pp.211-214,Jan 2001.

- [61] V. Moalemi and A. Afzali-Kusha, "Subthreshold pass transistor logic for ultra-low power operation," *IEEE Computer Society Annual Symposium on VLSI: Emerging VLSI Technologies and Architectures*, pp. 490-491, Mar 2007.
- [62] B.C. Paul, A.t Raychowdhary, and K. Roy, "Device optimization for digital subthreshold logic operation," *IEEE Transactions on Electron Devices*, vol.52, No.2, pp. 237-247, Feb 2005.
- [63] B.C. Paul and K. Roy, "Oxide thickness optimization for digital subthreshold operation," *IEEE Transactions on Electron Devices*, vol.52, No.2, pp. 685-688, Feb 2005.
- [64] D.J.Wouters, J.P. Colinge, and H.E. Maes, "Subthreshold current in thick and thin-film SOI MOSFET transistors," *IEEE SOS/SOI Technology Conference*, pp. 21-22, Oct 1989.
- [65] P.C. Yeh and J.G. Fossum, "Subthreshold MOSFET conduction model and optimal scaling for deep-submicron fully depleted SOI CMOS," *IEEE International SOI Conference*, pp. 142-143, Oct 1993.
- [66] J. Kim and K. Roy, "Double gate MOSFET subthreshold circuit for ultralow power applications," *IEEE Transactions on Electron Devices*, vol.51, No.9, pp. 1468-1474, Sep 2004.
- [67] H. Chang, "Circuit Timing and Leakage Power Analysis Under Process Variations," *PhD Dissertation*, The University of Minnesota, Feb 2006.

Curriculum Vitae

Yu Pu was born on September 27, 1982 in Chengdu City, China. In 2000, he graduated from the High School Affiliated to Sichuan University. In 2004, he received the B.Eng degree (cum laude) from the Department of Information Science and Electronic Engineering, Zhejiang University (ZJU), Hangzhou, China. From January 2005 to February 2009, he has been a joint PhD candidate of the National University of Singapore (NUS) and the Eindhoven University of Technology (TU/e). From December 2006 to February 2009, he was with the Mixed-Signal Circuit and System Group in NXP Research Eindhoven. He is currently a research scientist in the Ultra Low-Power DSP Processor Group of the IMEC.

Yu Pu was the 1st place winner of the Huawei telecommunication software design contest (sponsored by Huawei Tech., China, 2001), the 3rd place of the international PhD students'SoC innovation design contest (Taiwan, China, 2006), the best poster award winner of the 19th ProRISC (the Netherlands, 2008) and the author of a highlighted regular paper in ISSCC (U.S.A, 2009).

List of Publications

1. Y. Pu and Y. Ha, “An Automated, Efficient and Static Bit-width Optimization Methodology towards Maximum Bit-width-to-Error Tradeoff with Affine Arithmetic Model,” *Proc. of the IEEE Asia and South-Pacific Design Automation Conference (ASP-DAC)*, pp. 886-891, Jan 2006, Yokohama, Japan.
2. Y. Pu, C. S. Lee, Y. Ha, H. Corporaal, “Power-Efficient FPGA Switch with Reconfigurable Buffers,” *Proc. of the International PhD Student Workshop on SoC*, Jul 2006, Taipei, Taiwan.
3. Y. Pu, J. Pineda de Gyvez, H. Corporaal and Y. Ha, “ V_T Balancing and Device Sizing Towards High Yield of Sub-threshold Static Logic Gates,” *Proc. of the IEEE International Symposium on Low Power Electronics and Design (ISLPED)*, pp.355-358, Aug 2007, Portland, Oregon, U.S.A.
4. Y. Pu, J. Pineda de Gyvez, H. Corporaal and Y. Ha, “Statistical Noise Margin Estimation for Sub-Threshold Combinational Circuits”, *Proc. of the IEEE Asia and South-Pacific Design Automation Conference (ASP-DAC)*, pp.176-179, Jan 2008, Seoul, South Korea.

5. Y. Pu, J. Pineda de Gyvez, H. Corporaal and Y. Ha, "Towards Reliable and Ultra Low Energy Digital Circuits with Sub/Near Threshold Supply Voltage," *Proc. of the IEEE Annual Workshop on Signal Processing, Integrated Systems and Circuits (ProRISC)*, Nov 2008, Veldhoven, the Netherlands. (Best poster award)
6. Y. Pu, J. Pineda de Gyvez, H. Corporaal and Y. Ha, "An Ultra Low-Energy/Frame Multi-standard JPEG Co-processor in 65nm CMOS with Sub/Near Threshold Power Supply," *Proc. of the IEEE International Solid State Circuits Conference (ISSCC)*, Feb 2009, San Francisco, U.S.A. (Highlighted regular paper)
7. Y. Pu, J. Pineda de Gyvez, H. Corporaal and Y. Ha, "An Ultra Low-Energy/Frame Multi-standard JPEG Co-processor in 65nm CMOS with Sub/Near Threshold Power Supply," *IEEE Journal of Solid-State Circuits (JSSC)*, under review.