UNIVERSITY OF SOUTHAMPTON

# Techniques and Circuits for Ultra-Efficient Energy Harvesting Sensor Nodes

by

Anand Savanth

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the
Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

February 2021

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND PHYSICAL SCIENCES
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Anand Savanth

Sensor nodes play increasingly important role in various enterprises like agriculture, transport, defence etc. As the number of sensor nodes increase, their energy consumption, running and maintenance costs become detrimental to their implementation. Energy harvesting is an alternative to tethered or battery-powered sensor nodes which holds the promise of long-term sustainable operation. The energy available however, is limited and decreases further as sensor nodes become smaller. A small size also calls for tighter and possibly, monolithic integration. The limited available energy and small form-factor necessitates high energy efficiency in sensor nodes that must be guaranteed at design time. While integrated circuit (IC) design uses well-characterized device models for simulation, energy harvesters rarely have accurate models upon which to draw for circuit design.

This research explores development of models for small $cm^2$ photovoltaic cells by first characterizing them in real-world conditions and develop simulation models to enable IC design. The models are then used to investigate power conversion circuits and techniques for improving energy efficiency of sensor nodes. In this thesis, a compact and low-cost characterization scheme is used to develop a simulation model for photovoltaic cell which shows good correlation with measurements. The results of this work show the potential to improve sensor node design margining by as much as $16\times$.

Holistic system solutions are then explored to maximize utilization of harvested energy with efficient power conversion resulting in a 30% increase in computation cycles in the sensor node. Ultra-low-power rail monitor and oscillator circuits are also presented. The rail monitor exploits state-awareness to provide the best-reported balance in response speed and power consumption. The oscillator uses sub-cycle comparator duty-cycling to provide the lowest energy per cycle in the smallest area while exhibiting comparable line and temperature sensitivities. This research has resulted two journals, three peer-reviewed conference papers and three granted patents.

# Contents

# List of Figures

# List of Tables

## Declaration of Authorship

I, **Anand Savanth**, declare that the thesis entitled **'Techniques and Circuits for Ultra-Efficient Energy Harvesting Sensor Nodes'** and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as listed in Section 1.4.1 of the thesis.

**Signed:**

**Date:**

# Acknowledgements

I would like to express my gratitude to Arm management for sponsoring my PhD and their continued faith in me through this long research effort. I would also like to thank my academic supervisors, Dr. Alex Weddell and Prof. Bashir Al-Hashimi for their unending patience and support with formulating the ideas central to this work, supporting with prior-art references, endless paper reviews and making my work publication worthy. I remain grateful to my industrial supervisors Dr. David Flynn and James Myers for helping me explore various ideas presented as part of this work and for the tremendous support with chip integration and testing. I would also like to acknowledge their support in helping me balance my work and academic priorities. Thanks also to Prof. Geoff Merrett and Harry Oldham for providing critical feedback on my research work. Thank you!

I am thankful also to the many anonymous reviewers of my publications and also my examiners Prof. Bernard Stark and Dr. Basel Halak for improving the quality of the data presented in the manuscripts and this thesis.

Many thanks also to my co-researchers at University of Southampton with special mention of Alberto, Ibrahim, Cai, Domenico and Vasileios for supporting me during my visits to the university labs and for all the helpful discussions. I remain thankful to the University of Southampton PGR office staff for cheerfully fulfilling my most unusual requests over the course of my PhD.

For putting up with me, motivating and cheering me up during my gloomy days, special thanks are due to all my Arm colleagues with special mention of David B, David H, John Biggs, Andy Kufel, Paul W, Emre, Sid and Pranay. For being there when I needed someone, thanks are due to Sascha, George Smart, Sheng, Jatin, Radhika, Fiona, Matt, Divya, Mahanth, Ram, Ishan and my many lovely friends without whom this work would not have been possible. I am grateful to have been part of the Arm-Liverpool gang. I remain indebted to Kauser, Balaji, Akash, Karthik, Nithya, Trupti, Tasneem, Rohan and Leo for providing the much-needed emotional nourishment from time to time.

A very special gratitude goes to Sudeep, Raj and Thiru - my friends who have guided me in life, kept me grounded and never let me lose sight of what I set out to achieve.

I cannot thank my family enough, particularly my mother, for all they have sacrificed so I can be what I am today.

*To,*

*My family,*
*far and near...*

# Abbreviations

**ADC** Analog to Digital Converter.

**CMOS** Complementary Metal Oxide Semiconductor.

**CPU** Central Processing Unit.

**DAC** Digital to Analog Converter.

**DVFS** Dynamic Voltage and Frequency Scaling.

**DVS** Dynamic Voltage Scaling.

**EDA** Electronic Design and Automation.

**FET** Field Effect Transistor.

**FIR** Finite Impulse Response.

**FOCV** Fractional Open Circuit Voltage.

**FoM** Figure of Merit.

**GPIO** General Purpose Input Output.

**IC** Integrated Circuit.

**ICT** Information and Communication Technology.

**IoT** Internet of Things.

**LED** Light Emitting Diode.

**MCU** Micro Controller Unit.

**MIM** Metal Insulator Metal.

**MOS** Metal Oxide Semiconductor.

**MPP** Maximum Power Point.

**MPPT** Maximum Power Point Tracking.

**PLL** Phase Locked Loop.

**POR** Power On Reset.

**PV** Photovoltaic.

**PVT** Process, Voltage and Temperature.

**RF** Radio Frequency.

**SoC** System on Chip.

**SPICE** Simulation Program with Integrated Circuit Emphasis.

**SRAM** Static Random Access Memory.

**TEG** Thermoelectric Generator.

**UHF** Ultra High Frequency.

**VCO** Voltage Controlled Oscillator.

# Nomenclature

$C_{bot}$ Parasitic bottom-plate capacitance.

$C_{fly}$ Flying capacitor.

$C_{gate}$ Switch-gate capacitance.

$E_{EH}$ Energy available from EH.

$E_{act}$ Active energy.

$E_{avail}$ Total available energy.

$E_{tot}$ Total energy.

$E_{wait}$ Energy expended waiting in retention mode.

$I_{load}$ Load current.

$I_{pvsc}$ Photovoltaic cell short circuit current.

$I_{pv}$ Photovoltaic cell current.

$I_{sat}$ Photovoltaic cell reverse saturation current.

$I_{sc}$ Short-circuit current.

$N_{act}$ Active cycles.

$P_{bot}$ Bottom plate loss power.

$P_{gate}$ Gate-switching loss power.

$P_{mpp}$ Power at MPP.

$P_{ret}$ Retention power.

$P_{sw}$ Switch-conduction loss power.

$R_{fsl}$ Resistance - fast switching limit.

$R_{load}$ Load resistance.

$R_{on}$ On resistance of switch.

$R_p$ Output shunt resistance.

$R_{ssl}$ Resistance - slow switching limit.

$R_s$ Output series resistance.

$T_{sw}$ Switching time period, as determined by $f_{sw}$.

$T_{wait}$ Time spent waiting in active mode.

$T$ Absolute temperature.

$V_c$ Voltage across flying capacitor.

$V_{oc}$ Open-circuit voltage.

$V_{out}$ Output voltage.

$V_{pv}$ Photovoltaic cell voltage.

$V_r$ Ripple voltage.

$V_{switch}$ Voltage across switch.

$V_{th}$ Threshold voltage.

$W_{swopt}$ Optimum switch width.

$W_{sw}$ Width of switch.

$\eta_{conv}$ Conversion power efficiency.

$\eta_{mppt}$ MPPT tracking efficiency.

$\overline{P_{out}}$ Average output power.

$f_{cpu}$ CPU clock frequency.

$f_{sw,opt}$ Optimum switching frequency.

$f_{sw}$ Switching frequency.

$k_B$ Boltzmann constant.

$q$ Elementary charge.

**VDD** Supply voltage.

# Chapter 1

# Introduction

The Internet of Things (IoT) encompasses all connected devices, ubiquitous computing, big data, little data, sensors, wireless communication, machine-to-machine communication, cloud computing and simply "everything connected, everywhere and anytime" [1]. IoT as a term was first coined by Kevin Ashton in his 1998 presentation [2] to Procter & Gamble. Researchers have subsequently provided multiple definitions as to what constitutes IoT [1] [3] [4]. McKinsey analysis lists IoT as one of the 12 disruptive technologies that will alter the socioeconomic status quo by 2025, with annual revenues predicted to be $2.7 trillion to $6.3 trillion [5]. The leading networking enterprise solutions company CISCO predicts that the value added will reach $14 trillion [6]. It is reasonable to assume that the Information and Communication Technology (ICT) industry which has thus far been driven by the internet and mobile computing, will receive a fair share of these predicted revenues.

Speculative numbers aside, it is evident that it will be necessary to gather data on a massive scale [7]. Sensors are being used to gather data while monitoring a variety of parameters from soil pH values in a large agricultural field [8] to medical diagnoses and therapy [9]. Sensors can be implanted to monitor pets, babies and less able people, packed into parcels that travel worldwide or attached to packaged food to estimate shelf life and nutritional value. There is an enormous need for data collection, creating a demand for well-designed, field-deployable sensor nodes [10].

These nodes receive sensor data (in analog or digital form), then process it or simply store it for future use. Most often, either the raw or processed data are transferred to other sensors in the network or a host through wireless communication. Sensor nodes can work on batteries or use a wired connection but the scalability of such a sensor network is limited. For a trillion sensor nodes, energy harvesting enables long-term operation by avoiding battery replacement.

Energy harvesting sensor nodes must operate at maximum energy efficiency and also be reliable, small and cost-effective. It is important to ensure that, for a given harvester

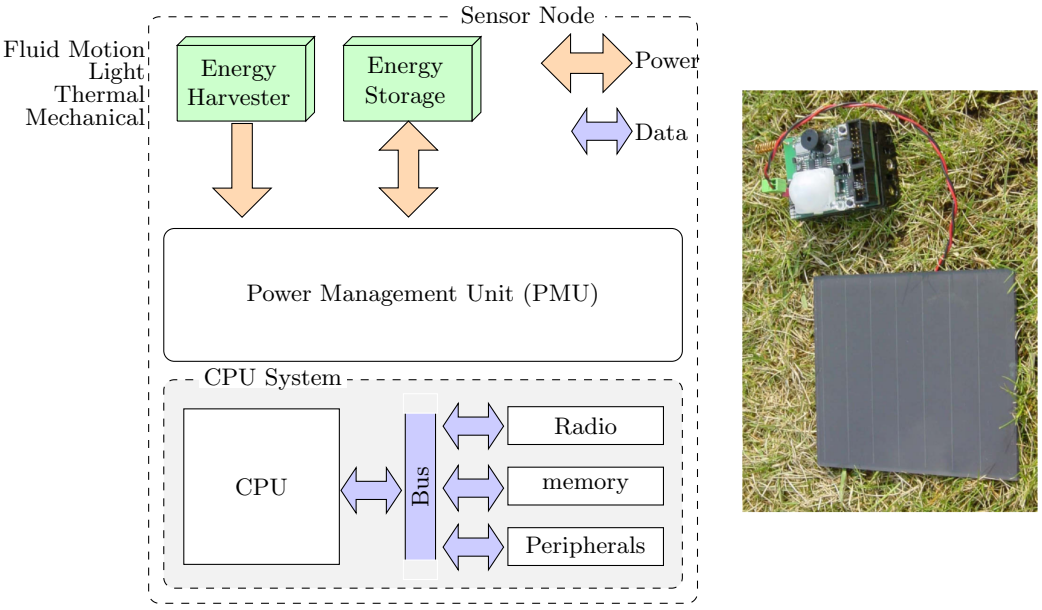



FIGURE 1.1: Block diagram of an energy harvesting WSN using discrete components block diagram (left) and actual device (right). Based on [13]

size, the sensor node can execute the specified workload. Sensor nodes can be energy neutral [11] if they can harvest as much energy from their surroundings as they expend during operation [12]. This research work aims to enable the design of such integrated energy neutral sensor systems.

Thus far, the focus on IoT has been justified. In the remainder of this chapter, the definition of a "wireless sensor node" in the context of this work (Section 1.1) is provided. The relation between energy harvesting and power conversion as applicable to sensor nodes is discussed in (Section 1.2). In Section 1.3, the main focus areas and corresponding contributions of this research are highlighted.

## 1.1 Ultra-low-power sensor nodes

Figure 1.1 shows the block diagram of a wireless sensor node that can be used in a broad range of applications. It also clarifies the definitions of sensor node, the power management unit and the central processing unit (CPU) system. Energy harvesting provides variable output depending on the ambient conditions; hence a power management unit is used to guarantee a reliable supply to the CPU system. To overcome periods of zero available energy, an energy storage element is used. Note that the power management unit and CPU system may be monolithically integrated. The CPU system in Figure 1.1 consists of the CPU core, memory, bus interface, variety of analogue and digital peripherals and a radio for wireless communication. A monolithic system on chip (SoC) implemented using industry-standard electronic design and automation

(EDA) tools with the above-mentioned features has been demonstrated in [14]. For the purposes of this thesis, the 'CPU system' will exclude the radio because radio frequency (RF) circuits have a different operating point, i.e., supply voltages >0.8 V and >1 mA current consumption compared with the digital logic which can operate at <0.4 V while consuming <10 μA current as explained below.

To achieve system level energy-efficiency, significant research has been carried over the last decade to ensure the digital CMOS logic can operate at the lowest energy point. While logic gates scale to low voltages readily, [15] show that memory can scale equally well and demonstrate that operating the CPU system at sub-threshold voltage (supply voltage reduced below the device threshold voltage) would result in minimum energy point operation. Another characteristically different behavior of wireless sensor nodes is that they have a low activity-to-sleep ratio (duty cycle) i.e., they perform the desired task for a short period of time, then switch to a low-power state [16]. These aspects of sensor nodes set challenges and open up opportunities in the design of wireless sensor nodes.

## 1.2 Energy harvesting and switched converters

**Energy harvesting:** Energy harvesting (or energy scavenging, power harvesting) involves extracting and using energy from the environment. In the wireless sensor node context, energy can be scavenged using ambient light (photovoltaic (PV)), temperature differences using thermoelectric generator (TEG), mechanical vibration using vibration energy harvesters, from ambient RF fields, fluid flow and air movement. Table 1.1 shows the power density of various energy harvesting sources. The 10 - 100 μW per $cm^3$ or $cm^2$ energy region has been estimated to be suitable for many sensor applications [19]. A $cm^3$ volume is appropriate because smaller sizes are detrimental to assembly cost (considering battery and RF antennas) and deployment [20] although ingestible and implantable sensors would benefit from smaller dimensions. Solar (light) energy has a relatively higher power density outdoors and indoors and output in DC, making it the preferred starting point for this work.

TABLE 1.1: Power density for various energy harvesting schemes. Adapted from [17] and [18]

| | |
|---|---|
| Photovoltaic (outdoors at noon) | 15 mW/$cm^3$ |
| Vibration (shoe piezoelectric inserts) | 330 μW/$cm^3$ |
| Vibration (small microwave oven) | 116 μW/$cm^3$ |
| Vibration (100db acoustic noise) | 190 nW/$cm^3$ |
| Thermoelectric (10°C gradient) | 40 μW/$cm^3$ |
| Ambient RF (GSM900) | 50 nW/$cm^2$ |

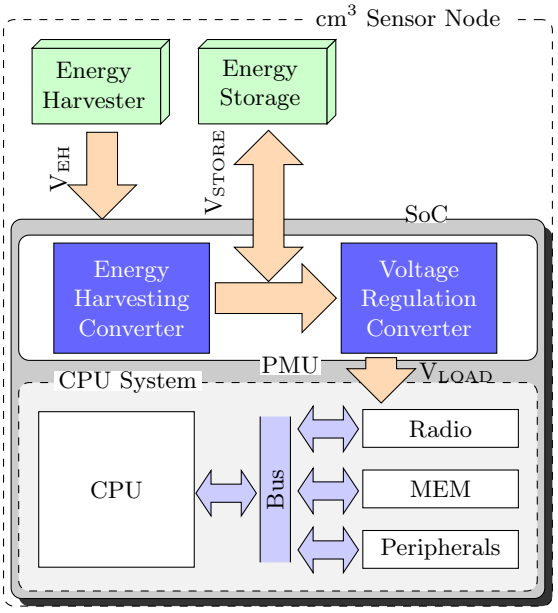



FIGURE 1.2: Two stage power conversion in WSN

**Power conversion:** Power converters are employed widely to efficiently extract harvested energy [21]. They constitute a large portion of the power management unit design effort. Figure 1.2 expands on Figure 1.1 by depicting an energy harvester to power converter interface with output of the energy harvester, $V_{EH}$, converted to $V_{STORE}$. A voltage regulator is employed to convert down from $V_{STORE}$ to $V_{LOAD}$ for the CPU system. For minimizing the overall sensor system operating energy, it is imperative that the power converters have good conversion efficiency. Other key characteristics required for the energy harvesting converter is for it to track maximum power extraction from the harvester and support the minimum energy point operation of the load.

## 1.3 Research questions

The trillion-sensor node IoT vision has spawned three research questions which motivate the work presented here. These questions and corresponding contributions from this work are discussed below.

**1) How can the real-world performance of microscale PV cells be captured to enable EDA friendly implementation of energy harvesting sensor nodes?**

Designers of integrated energy harvesting converters rely on EDA tools to ensure functionality and conversion efficiency. However, commercial energy harvesters lack simulation models and exhibit significant deviations from their specifications. This impediment has been addressed in this research by proposing a modeling methodology to generate EDA compatible energy harvesting models. The energy harvesting data is captured from

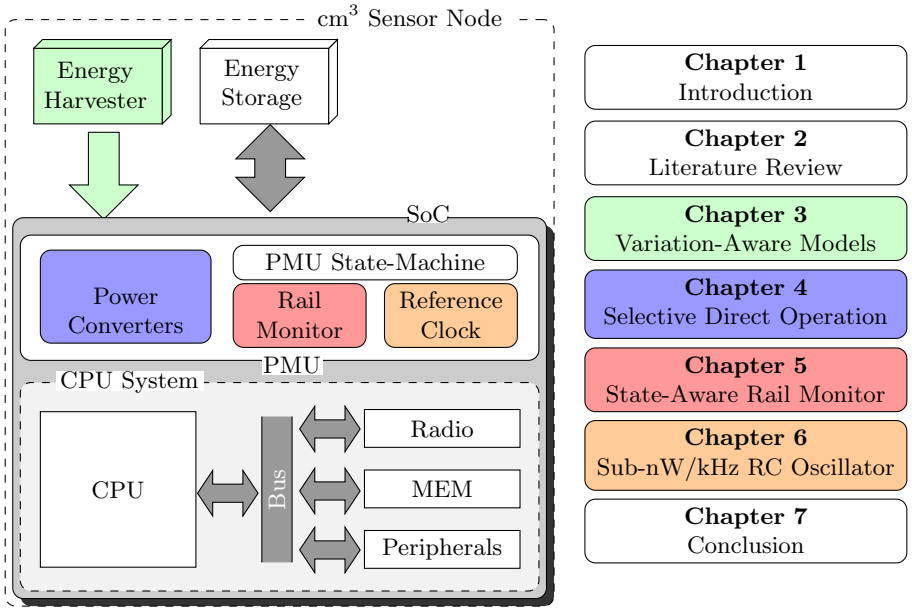



FIGURE 1.3: Focus and scope of this work. Relation of this work and a sensor system (left) and structure of this report in corresponding colors (right).

real-world behavior of the harvester under deployment conditions and variation-aware models are generated to enable worst- and best-case evaluation.

**2) How can the utilization of energy be improved through new system or circuit-level techniques?**

Utilization of energy from the harvester is reduced because of conversion inefficiencies and also due to the inability of conventional sensor node designs to function at very low voltages. The former case may be addressed by improving efficiency of the converter - an approach further bolstered in this work by the availability of energy harvester models. The latter problem has been addressed by exploiting the low-operating voltage of the sub-threshold CPU system. This work proposes system enhancements that minimize this loss and achieves up to 16% additional energy to maximize computations. The energy harvesting converter designed as part of this work demonstrates higher efficiency under low-light conditions than the state-of-the-art. Dynamic tracking is also demonstrated on a 65 nm test chip.

**3) How can the energy demand of sensor nodes be further reduced through optimization of ancillary circuits?**

To achieve the objective of microscale energy neutral sensor nodes that last a decade or longer, efficient energy harvesting and voltage regulator converters are essential. Assist circuits in the power management unit are of equal importance. Specifically, to minimize energy expended during one loop of activity, CPUs in sensor SoCs need to sleep for as long as possible and quickly wake up, execute code fast and return to sleep. This fast

transition requires fast performance mode (CPU clock frequency and voltage to match the frequency) switching to maximize energy efficiency. This justifies the need for a fast but efficient rail-monitor to enable the CPU to switch between low power modes with handshaking. Reference clock circuits are also essential to enable reliable sleep and wake times. These components need individual attention, keeping in mind the system-level impact of each design decision. Key circuit-level techniques for improving the efficiency of these analog blocks have also been demonstrated as part of this work.

Figure 1.3 depicts the focus of this work which emphasizes on the design and analysis of DC-DC converters (energy harvesting and voltage regulator), (blue blocks in Figure 1.3) and various on-chip assist circuits (rail-monitor and reference clock in Figure 1.3) used to enable energy neutral wireless sensor nodes. It should be noted that the ultra-low power energy-efficient CPU core, plus its associated memory and peripheral blocks (white blocks in Figure 1.3) are being developed as part of the IoT efforts within ARM research and are not contributions from this research work.

The structure of this document is also highlighted alongside in Figure 1.3. Switching converters and assist circuits have been explored and related state-of-the-art is surveyed in Chapter 2. Chapters 3 to 6 cover the main contributions of this research work. In Chapters 3 and 4, shortcomings in the state-of-the-art models and techniques are highlighted, and solutions that exploit the properties of the entire system to better utilize harvested energy are proposed. In Chapters 5 and 6, design constraints, simulation results and comparison with prior designs are presented for an ultra-low power voltage monitor and sub-nW/kHz clock respectively. Future plans are presented and conclusions are drawn in Chapter 7.

## 1.4 Contributions of this work

The key contributions of this work in relation to the wireless sensor node system are listed below.

1. To guarantee energy neutrality in a wireless sensor node, it is vital to understand worst-case design corners and allow for them during design. There is a need on the part of designers for harvester models that allow system co-simulations. As part of this work, a variation-aware characterization and modeling method for light harvesters has been developed [22].

2. The advantages of these models and system co-design are reflected in the measurements presented in [23]. A patent has been filed for the techniques presented here [24]. The modeling and system co-design approach has been extended to TEGs [25] as well, outlining the constraints for energy neutral operation with temperature differentials.

3. Individual circuit blocks are equally important in guaranteeing energy neutral operation. An ultra-low power voltage monitor has been designed and implemented [26] as part of this work. A response time of 6 µs was measured while expending 50 nW.

4. A sub-nW/kHz relaxation oscillator has also been demonstrated on 65 nm silicon [27]. A line sensitivity of 0.49 %/V with a 100 ppm/°C temperature stability was demonstrated. Patents have also been filed for the key techniques employed in these assist circuits.

In summary, this work demonstrates techniques that can enable an integrated energy neutral wireless sensor node system for the IoT. A complete list of publications and patents generated as part of this research is listed below and the contributions of the author by paper has been outlined in Appendix H.

### 1.4.1 Conference publications

[22] A. Savanth, A. S. Weddell, J. Myers, D. Flynn, and B. M. Al-Hashimi, "Photovoltaic Cells for Micro-Scale Wireless Sensor Nodes: Measurement and Modeling to Assist System Design," in Proceedings of the 3rd International Workshop on Energy Harvesting & Energy Neutral Sensing Systems, ENSsys 2015, Seoul, South Korea, November 1, 2015, 2015, pp. 15–20

[25] A. Savanth, M. Bellanger, A. Weddell, J. Myers, and M. Kauer, "Energy Neutral Sensor System with Micro-Scale Photovoltaic and Thermoelectric Energy Harvesting," Journal of Physics: Conference Series, vol. 1052, p. 012069, July 2018.

[26] A. Savanth, A. Weddell, J. Myers, D. Flynn, and B. Al-Hashimi, "A 50 nW Voltage Monitor Scheme for Minimum Energy Sensor Systems," in 2017 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems (VLSID), January 2017, pp. 81–86.

[27] A. Savanth, J. Myers, A. Weddell, D. Flynn, and B. Al-Hashimi, "A 0.68 nW/kHz Supply-Independent Relaxation Oscillator with ±0.49 %/V and 96 ppm/$_o$C Stability," in IEEE International Solid-State Circuits Conference (ISSCC), February 2017, pp. 96–97.

### 1.4.2 Transactions and journals

[23] A. Savanth, A. S. Weddell, J. Myers, D. Flynn, and B. M. Al-Hashimi, "Integrated Reciprocal Conversion with Selective Direct Operation for Energy Harvesting Systems," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 64, no. 9, pp. 2370–2379, September. 2017.

[28] A. Savanth, J. Myers, A. Weddell, D. Flynn, and B. Al-Hashimi, "A sub-nW/kHz Relaxation Oscillator with Ratioed Reference and sub-Clock Power Gated Comparator", IEEE Journal of Solid-State Circuits, vol. 54, no. 11, pp. 3097-3106, November. 2019

### 1.4.3 Patents

[24] A. Savanth, J. Myers, and D. Flynn, "Harvesting Power From Ambient Energy in an Electronic Device," Patent Granted - 10,651,683, May. 12, 2020.

[29] A. Savanth, S. Das, J. Myers, D. Bull, and B. Sandhu, "Integrated Oscillator Circuitry," Patent Granted - 9,831,831, November. 28, 2017.

[30] A. Savanth, B. Sandhu, J. Myers, and A. Weddell, "Monitoring Circuit and Method," Patent Granted - 10,664,031, May. 26, 2020.

# Chapter 2

# Literature Review

In the context of this research work, this chapter surveys the state-of-the-art in integrated microscale wireless sensor nodes and aims to clarify:

1. the need for energy harvesting in wireless sensor nodes (Section 2.1).
2. the common energy storage methods that can be used in wireless sensor nodes(Section 2.1.4).
3. designs and metrics for integrated power converters (Section 2.2).
4. general characteristics of state-of-the-art sub-threshold CPU systems (Section 2.3).
5. support circuits required for wireless sensor nodes and their implementation (Section 2.4).

## 2.1 Energy harvesting

Primary batteries limit the active lifetime of sensor nodes in IoT applications [31] unless the batteries are replaced on a regular basis. Sensor nodes with rechargeable batteries or super-capacitors which can be recharged without wired connections are far more attractive for wide area deployment [8], [32]. Such sensor nodes must feature ultra-low power operation, operate with high energy efficiency and also be able to recharge their batteries or super-capacitors whenever sufficient ambient energy is available. Ambient energy has zero cost but, is available sporadically and in varying levels of intensity. The output from harvesters varies from a few 100 mVs (TEGs, [33]) to a few 100 Vs (vibration harvesters, [19]). Conventional circuit design techniques need not consider this wide variation in supply voltage because they work off relatively fixed voltages regulated down from either batteries or wall sockets.

### 2.1.1 Energy density estimation

It is vital to understand the limitations with harvesting energy from the ambient environment. These losses associated with transduction establish the theoretical maximum conversion efficiency that can be achieved. For RF energy harvesting, it is well established that the available near field RF energy experiences a quadratic decay with distance of the harvesting antenna from the transmitting antenna [34]. Similarly, for large mechanical structures such as wind turbines frictional losses due to bearings, windage and gears limit the maximum possible efficiency [35]. Since this thesis focuses on photovoltaic cells, an estimation of the energy density from photovoltaic cells in outdoor conditions will be summarized in this section, based on [36].

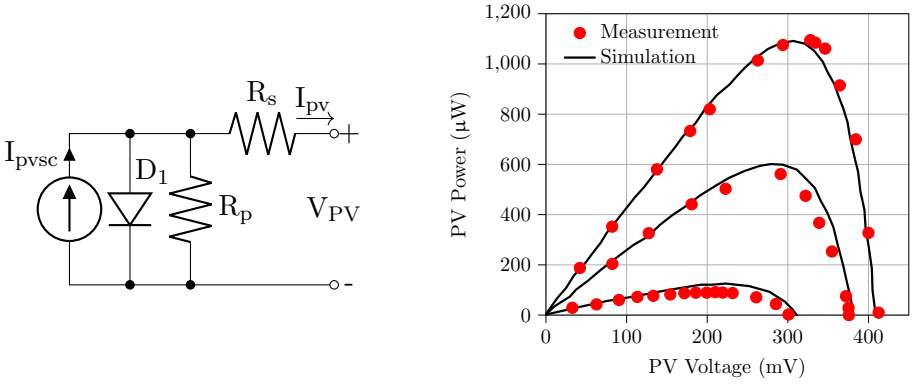
The average output of the sun is about 1360 W/m$^2$ [37]. Of this, 23% is lost due to absorption by the earth's atmosphere. Therefore, the peak received power at the earth's surface is about 1360 $\times$ (1-0.23) = 1047 W/m$^2$. For the purposes of evaluating peak power, reflection (dominated by cloud albedo [38]) will be ignored and the air-mass 1.5 (AM 1.5) standard condition will be assumed.

The silicon structure of the photovoltaic cell imposes some losses. Firstly, about 2% of the incident light is reflected back without contributing to electrical current generation. While bare Silicon is highly reflective ($\approx$ 35%), anti-reflective coating is commonly used which minimizes this loss [39]. Second, photons with energy mismatched with the band gap energy will also be lost. Photons with less than band gap energy contribute weakly to electrical current generation and photons with excess energy will lose the excess energy as heat without contributing to electrical current. These two forms of loss in the photovoltaic cell constitutes 19% and 28% loss respectively.

Photon interactions in the cross-sectional area of the photovoltaic cell are dependent on the wavelength. Shorter wavelength light is absorbed only near the surface while red light is absorbed deeper in the photovoltaic cell. Beyond a few hundred microns depth, the absorption is limited. Absorption away from the junction results in losses which is reported to be about 19%. A further loss based on reflection or absorption of photons depending on wavelength of the incident light is termed quantum efficiency of the photovoltaic cell. This loss is limited to about 5%.

TABLE 2.1: Power density estimation for solar energy harvesting. Adapted from [36]

| | |
|---|---|
| Reflection loss | 2% |
| Insufficient photon energy | 19% |
| Excess photon energy | 28% |
| Absorption away from junction | 19% |
| Quantum efficiency | 5% |
| Grid, impurity and resistances | 10% |




(a) (b)

FIGURE 2.1: Solar cell (a) simplified model and (b) Output power vs terminal voltage for illuminations of 504, 784, 1548 Lux. Adapted from [17]

Finally, as will be shown in Section 2.1.2, the photovoltaic cell has parasitic resistances which leads to ohmic losses. This combined with losses due to dust and impurities and losses encountered with interconnecting the individual cells imposes another 10% loss. The various losses from the above discussion are summarized in Table 2.1. This effectively limits the photovoltaic cell efficiency to about 17% while the theoretical limit, expressed as the Shockley–Queisser limit is 33% [40].

This effectively means that the available peak power output from outdoor sunlight would be 1047 W/m$^2$ $\times$ 0.17 = 178 W/m$^2$ or approximately 18 mW/cm$^2$. This is in line with the findings in Table 1.1. The 1047 W/m$^2$ corresponds to 1.32x10$^5$ Lux illumination. Indoor lighting on the other hand is about 650 Lux or 500 mW/cm$^2$ which translates to 85 µW output from the cm$^2$ PV cell. This is in line with the author's measurements as will be shown in Section 4.5.

### 2.1.2 Solar cell model

The PV cells transduce light energy to electrical energy by allowing photon-excited electrons to travel through an external circuit [41]. This directional current generates a DC voltage. Figure 2.1a shows the electrical model of a PV cell [17], which consists of a current source $I_{pvsc}$ in parallel with a diode $D_1$. The two resistances $R_s$ and $R_p$ are the parasitic resistances. The diode limits the maximum output voltage to a diode drop (0.7 V). The diode knee voltage, current and resistances vary with light intensity. The current output of the PV cell is given by

$$I_{pv} = I_{pvsc} - I_{sat}\left(e^{\frac{q(V_{pv}+I_{pv}R_s)}{AkBT}} - 1\right) - \frac{V_{pv} + I_{pv}R_s}{R_p} \tag{2.1}$$

In equation 2.1 $I_{sat}$ is the reverse saturation current of the diode, $q$ is the fundamental

electron charge, $k_B$ is Boltzmann's constant, $T$ is the absolute temperature and A is a fitting parameter determined by the PV cell's dimensions. The power output of the cell is the product of output current $I_{pvsc}$ and the voltage $V_{pv}$. Other key model parameters are the open circuit voltage $V_{oc}$ wherein the load current is zero and the short circuit current $I_{sc}$ wherein the load resistance is zero resulting in the terminal voltage of the photovoltaic cell to be zero.

Figure 2.1b shows the measured and simulated power output [17] for a monocrystalline $cm^2$ PV cell [42] for different illumination conditions and highlights the average power and the variation to be expected (10 µW to 1 mW) so that sensor nodes can be designed accordingly. It is worth noting that most PV cells operate at only 20-25% maximum efficiency [43]. It is possible to fabricate PV cells using the conventional complementary metal oxide semiconductor (CMOS) fabrication technique [44]. Despite this technique reporting lower efficiency (5%) it remains useful for applications that require tighter integration. PV cells exhibit reverse leakage due to shunt resistance and the diode when the PV cell is not generating enough current. Under such conditions, the cell needs electrical isolation from the energy harvesting circuitry.

Another noteworthy aspect illustrated in Figure 2.1b is that the peak power for the different illumination settings is measured at voltage which is approximately 0.7 to 0.78 times the open circuit voltage corresponding to that illumination. This fraction of open circuit voltage is helpful in establishing, approximately, the load current at which maximum power can be drawn from the PV cell. This technique is called *Fractional Open Circuit Voltage* (FOCV) method of maximum power point tracking (MPPT).

### 2.1.3 Maximum power point tracking

The output of energy harvester varies in accordance with the available ambient energy. Therefore, tracking is necessary to ensure that maximum energy is harvested under all conditions. For PV cells, MPPT can be understood using fundamental maximum power transfer theorem which states that maximum power is transferred when the source and load impedances are equal (matched) [45]. Hence, switching converters, (see Figure 2.2) must have some mechanism for determining the harvester output impedance and tuning its own input impedance for matching. On the load side, which in this case would be the sub-threshold CPU system, converter have to provide a regulated output voltage despite variations in $V_{STORE}$.

Most MPPT techniques use some form of voltage or current measurement to determine the maximum power point of the PV cell, with the exception of the technique described in [46], which utilizes a time-based power monitor has been proposed in [46]. A survey of different MPPT techniques is found in [47], which also provides a good comparison of the techniques. The fundamental sensing technique can be either analog or digital and

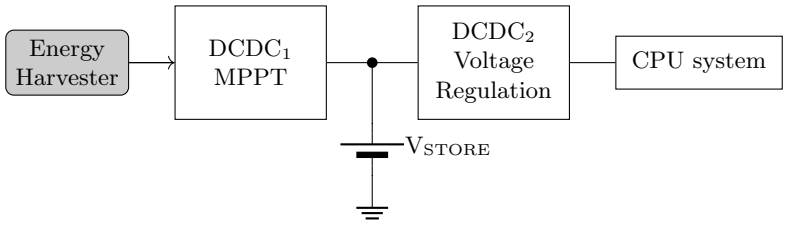



FIGURE 2.2: Block diagram of an energy harvesting wireless sensor node showing interface of energy harvester to MPPT converter and voltage regulator to load.

is a decisive factor in the choice of MPPT technique. The data presented in [48] does not, however consider complexities in MPPT when PV cells are partially shaded.

### 2.1.4 Energy storage

Energy storage is necessary in wireless sensor nodes to power the node when ambient energy is insufficient and also to sustain peak loads which the energy harvesting may not be able to support. Energy is stored in rechargeable batteries and/or super-capacitors. Although neither of them can be easily implemented or integrated with conventional CMOS fabrication techniques, they will be discussed briefly within the scope of this survey.

Batteries are gauged on the basis of weight, volume, cycle life (number of charge-discharge cycles before capacity drops to 80% of the initial value) and the cell chemical voltage or electromotive force (EMF). A lower weight and volume (for the same energy capacity) but a higher number of charge-discharge cycles is desirable. The other important metric is energy density. An excellent comparison of popular rechargeable battery types is provided in [49]. Li-ion batteries, with higher gravimetric and volumetric energy density combined with higher cycle life, do well in a variety of sensor node applications. The millimeter-scale sensor node design [44] used an all-solid-state 3.6 V rechargeable thin-film lithium battery [50] for energy storage.

Super-capacitors and ultra-capacitors are also important energy storage devices that are being increasingly adopted in sensor node designs. Super-capacitors come with several advantages, of which the lack of chemical reactions (unlike batteries) and drying-up are significant positives. However, super-capacitors suffer from charge re-distribution and self-discharge issues [51]. Another design complexity with super-capacitors is the large inrush current on the first charge. However, they fare much better in terms of charge-discharge cycles. These storage elements are ideal for providing bursts of energy, something that may be required during radio communications, and at other times, the wireless sensor node could work off a small and limited-capacity rechargeable battery.

## 2.2 Power conversion

Designing DC-DC converters is a mature subject, and these converters have been around for several decades supporting power conversions in railways and being used in other applications [52]. Integrated converters, however, are a relatively new area of research. Designing efficient converters using the same CMOS fabrication process used in the rest of the digital logic can be difficult. This section surveys integrated DC-DC converters, various topologies, design techniques, regulation schemes and common metrics for evaluating these converters.

Switched converters can use inductors or capacitors as their reactive element. [53] provides an excellent survey of switched capacitor and inductor-based converters. Some applications, for e.g., electromagnetic harvesters, which have coils built into the energy harvesting device can exploit the stray inductances to design efficient converters [54]. However, inductor-based DC-DC converters are less attractive in the context of this work because:

1. Large inductors are needed at low switching frequency $f_{sw}$. A higher $f_{sw}$ wastes power.
2. Large inductors cannot be integrated. Small on-chip inductors have low Q.
3. While off-chip inductors can be wound around magnetic materials to provide high Q, on-chip inductors must make do with nonmagnetic cores.
4. Integrated inductors require additional foundry steps during fabrication.
5. Switch voltage stresses are much higher for inductors than with switched capacitor converters.

A simple theoretical model [55] for switched capacitor converters is discussed next, which also serves as the foundation for a design methodology and eventually a comparison metric for the various switched capacitor converter topologies.

### 2.2.1 Integrated switched capacitor converters

To a first approximation, switched capacitor converter can be modeled as an ideal transformer with a conversion ratio $n$ (Figure 2.3a) and, more practically, with the loss elements included as variable impedances $R_{out}$ (Figure 2.3b). The number of turns in the transformer's primary and secondary, $m : n$, is simply the switched capacitor converters' conversion ratio. Figure 2.3c shows a simple 1:1 switched capacitor network. In this case, the series impedance represents the internal voltage drop across capacitors

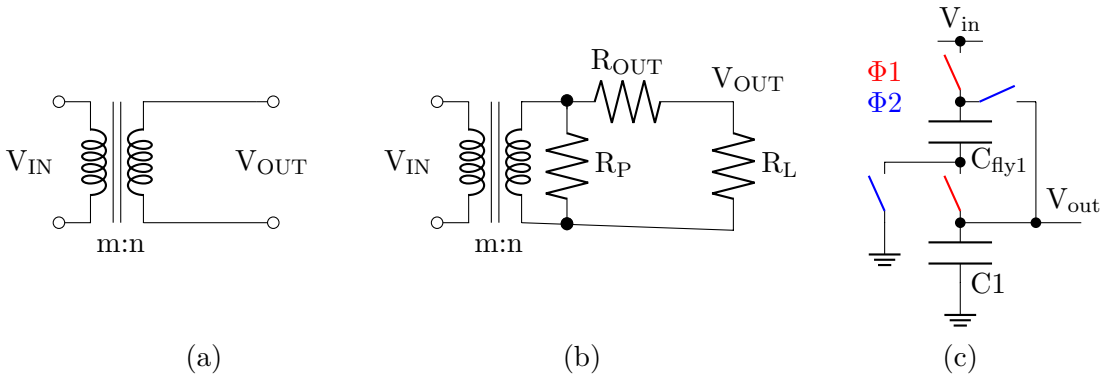



FIGURE 2.3: Switching converter: (a) Simplified 2 port model, (b) with loss elements and (c) a simple 2:1 switched capacitor. Adapted from [55] and [56]

and switches necessary for charge movement. Based on the switching frequency $f_{sw}$, switched capacitor converters can operate in two regimes [55]:

1. Slow switching: Here the switching frequency is slow enough to allow the fly capacitors to charge fully to the final steady voltage. This results in the loss being dominated by the fly capacitors and their switching frequency.

2. Fast switching: Here switching frequency is fast such that the fly capacitors cannot reach their equilibrium state. This results in a constant current through the switches and hence the loss in this case is dominated by the on resistance of the switches and the constant current.

The asymptotic limits of these regimes are the slow switching limit and the fast switching limit respectively. The switching frequency, converter topology and device sizes determine the value of the series impedance as per equations 2.2 and 2.3 below depending on the operating regime. [55].

$$R_{ssl} = \frac{1}{f_{sw}} \sum_{i \in caps} \frac{a_i^2}{2C_i} \tag{2.2}$$

$$R_{fsl} = 2 \sum_{i \in switches} R_{on,i} a_i^2 \tag{2.3}$$

In these equations, $a_i$ represents the ratio of charge transfer of the capacitor and switch [55]. The design objective is to have as low a value as possible for each of these resistances. The above equations indicate that larger capacitance, higher $f_{sw}$ and lower switch resistances $R_{on}$ are ideal. However, the requirements are inherently contradictory, limiting the design space. For a constant switch resistance, large capacitances have longer charging times meaning that $f_{sw}$ must be low. If we lower the switch resistance by increasing the switch size, then switching losses will degrade converter performance.

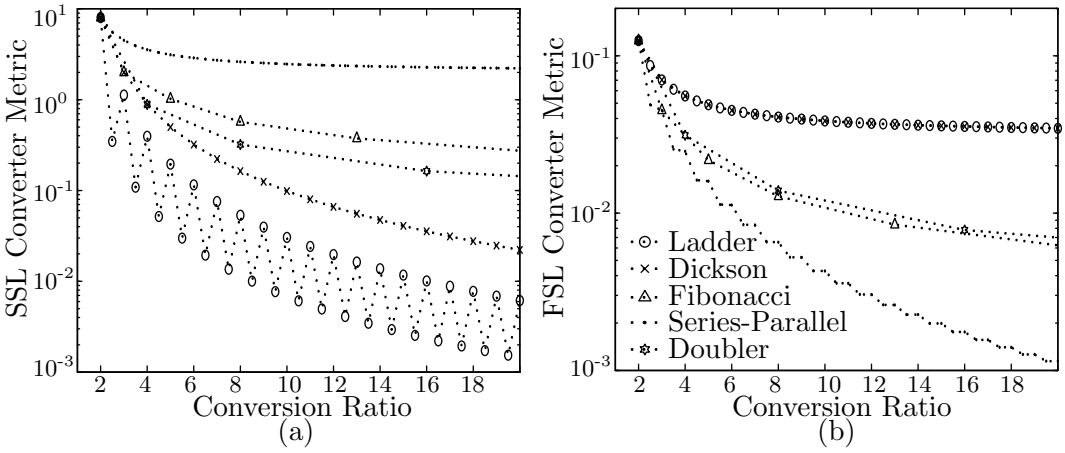



FIGURE 2.4: (a) SSL and (b) FSL metric for various switched capacitor converter topologies. Taken from [55]

It is established in [55] that the right balance is obtained when the utilization of the capacitances and switches is maximized. The authors in this work take this theoretical formulation further and verify the validity through simulations for the ladder [57], Dickson [58], Fibonacci [59] and other switched capacitor topologies. The results are reproduced in Figure 2.4.

The various switched capacitor topologies, at a conversion ratio of 2, exhibit equal switch utilization (Figure 2.4). For higher ratios, the ladder and Dickson topologies show superior switch utilization when applying the fast switching limit metric. However, for sub-Vt operation, low load currents require low $f_{sw}$ in converters, meaning that the slow switching limit metric would be more applicable. The series-parallel topology is the clear winner using the slow switching limit metric. The switched capacitor converters in micro-scale wireless sensor nodes usually operate in the slow switching limit regime to minimize losses from fast oscillators.

### 2.2.2 Designing switched capacitor converters

The simple model for switched capacitor converter in Figure 2.3a can be expanded to understand the loss mechanisms further and develop a design methodology. The evolved model is shown in Figure 2.3b, where the single series impedance is broken into a series resistance $R_s$ and a shunt resistance $R_p$. The shunt resistance represents losses that are largely independent of the load current, namely;

1. Gate loss: This is the loss in charging and discharging the gate capacitance of the MOS devices used as switches and is expended in the gate drivers

2. Bottom-plate loss: This is the loss due to repeated charging and discharging of the parasitic capacitance at the bottom plate of integrated capacitors which are invariably referenced the silicon substrate which is at ground potential [60] and

3. Control loss: This is the power loss in the additional circuitry required to effectively vary clock frequency or modify the topology of the switched capacitor network dynamically.

The series resistor represents a loss proportional to load current $I_{load}$ and voltage drop due to the switches $V_{switch}$. This power loss contributes to switched capacitor converter inefficiency and manifests as voltage ripple $V_r$ at the output node. The voltage ripple is the periodic excursion of the output voltage between a maximum and minimum voltage around it's mean with the periodicity, in the case of switched capacitor networks, being determined by the clock switching frequency. Thus for any converter, $P_{in} \approx (V_{min} + \frac{V_r}{2})(I_{load} + \frac{I_r}{2})$ and $\overline{P_{out}}$=$V_{out}I_{load}$. Although this simplified expression is derived based some assumptions [56] which may not always hold for sub-threshold designs, it does allow for a generic framework to be developed. Because $\overline{P_{out}}$ and $P_{in}$ have been derived, the design criterion is simply to minimize the difference between the two, or in other words minimize $V_r$. $V_r$ is given by

$$V_r = \frac{I_{load}}{C_{fly}} \frac{T_{sw}}{2} = \frac{I_{load}}{2C_{fly}f_{sw}} \tag{2.4}$$

where $T_{sw}$ is the switching period. The duty cycle is assumed to be 50% because this maximizes switch utilization under the SSL condition. The other loss components are switch conductance loss $P_{sw}$, bottom plate loss $P_{bot}$, and gate loss $P_{gate}$ which are expressed as:

$$P_{sw} = {I_{load}}^2 \frac{R_{on}}{W_{sw}} M_{sw} \tag{2.5}$$

$$P_{bot} = M_{bot}{V_c}^2 C_{bot} f_{sw} \tag{2.6}$$

$$P_{gate} = W_{sw}{V_{switch}}^2 C_{gate} f_{sw} \tag{2.7}$$

Here, $M_{sw}$ is a unit less factor decided by the number of switches actually conducting in a single phase, $M_{bot}$ is a factor decided by the topology that allows the combining of the different $P_{bot}$ for each capacitor in the design, $W_{sw}$ is the width of the switch and $C_{gate}$ is the unit gate capacitance decided by the technology. For a given topology, the technology and design area constraints, i.e.,unit values of $C_{fly}$, $R_{on}$, $C_{gate}$ and $C_{bot}$ are fixed parameters. The only variables that can be optimized are $W_{sw}$ and $f_{sw}$. Increasing either $W_{sw}$ or frequency $f_{sw}$ decreases $V_{switch}$ and the associated conduction loss but results in higher gate loss. Hence, these two parameters are tuned at run time to track maximum conversion efficiency [56]. The optimum values are given by

$$f_{sw,opt} = \frac{1}{\sqrt{M}} \frac{1}{C_{fly} R_{load}} \tag{2.8}$$

$$W_{swopt} = \sqrt{\frac{{V_{out}}^2}{{V_{switch}}^2} \frac{R_{on} C_{fly}}{R_{load} C_{gate}} \sqrt{M}} \tag{2.9}$$

This design methodology has been extended further to switched capacitor standard cell which has been synthesized for multiple voltages and load current ranges and is also reconfigurable at run-time [61].

This general design framework cannot be universally applied in this work because the energy harvesting converter Figure 1.2 would have to be a boost converter while the regulating converter would be a buck converter. The design constraints for boost and buck converters are slightly different. Usually, only boost converters need level-shifting in switch gate-drivers which imposes additional restrictions and power losses.

### 2.2.3 Switched converter metrics

A wide range of switched capacitor converter designs have been reported in literature ( [60], [55], [56] and [62]), each claiming novelty and best-in-class performance. In this subsection, the various metrics for switched capacitor converter designs are explained briefly.

**Efficiency:** Conversion efficiency is the most important metric and most integrated switched capacitor converters can achieve a peak of up to 80% at a conversion ratio of 2. It is defined as $P_{out}/P_{in}$.

**Power density:** The power density is the ratio of $P_{out}$ to the area of the converter. Power density can be improved by using hybrid regulators [63] or using multiphase converters [53].

**Capacitor technology:** There are multiple techniques that can be used to realize integrated capacitors and each one has an influence on power density and efficiency. The popular techniques are metal insulator metal (MIM) capacitances, metal oxide semiconductor (MOS) gate capacitance, metal side wall (MOM) capacitance and deep trench capacitances. Gate capacitance has the highest density but lowest efficiency, while MIM capacitors use up large area but improve efficiency as their leakage is relatively low and they have better temperature responses and lower bottom-plate parasitic capacitance [64].

**Phases:** Most designs use two-way interleaved phases due to the near zero overheads in implementing them. Some high-power converters use 16 and 32 phases [56] [65]. However, a higher number of phases is restrictive because each phase needs a nonoverlapping signal generation logic and an on-chip clock source.

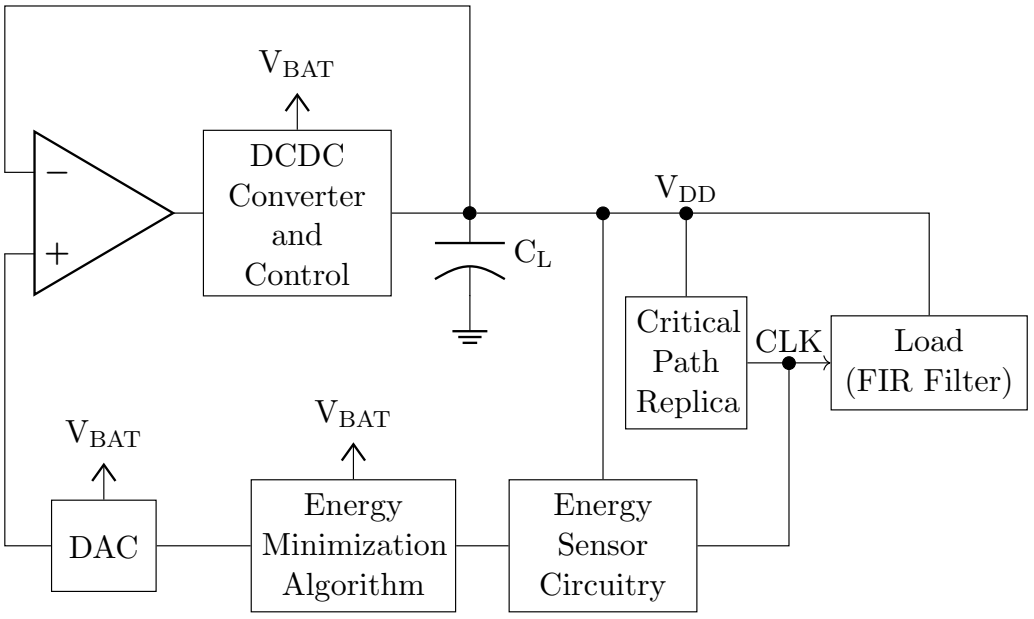



FIGURE 2.5: Minimum energy tracking using integrated converter. Adapted from [67]

**Load Current Range:** The conversion loss is better amortized at higher load currents providing better overall efficiency. With low load currents, even small leakage currents contribute a significant portion of the loss. Hence load current range is an important metric with which to compare converter designs.

**Control technique:** The state-space analysis technique [66] is the one of more commonly used control scheme for switched capacitor converters. Other techniques include dynamic control of switch sizes, dynamic capacitance modulation [62], dynamic frequency control and phase interleaving.

### 2.2.4 Minimum energy point

The design constraints of a DC-DC converter vary depending on the source and load characteristics. While an energy harvesting converter experiences wide swings in input voltage, a voltage regulation converter must support a wide range of output voltages. An energy harvesting converter is responsible for MPPT and a voltage regulation converter tracks the minimum energy point of the load circuit. Because circuits operating on $V_{th}$ exhibit minimum energy operation, DC-DC converters must track the average $V_{th}$ of the system in terms of process, voltage and temperature (PVT) variations. Two published works [68] [67] discuss such tracking methods. The more often cited [67] method uses a critical path replica (of a finite impulse response (FIR) filter block) which discharges a pre-charged capacitor during operation. The difference in the charge held by the capacitor before and after the load circuit completes its operation represents the energy consumed by the replica circuit for that operation. If this energy can be minimized via

an appropriate choice of supply voltage (converter $V_{out}$) then the rest of the system would also operate at its minimum energy point. Figure 2.5 shows the minimum energy tracking mechanism proposed by [67]. This technique is quite slow and requires the load to be disconnected during calibration.

The second method [68] uses pulse skip modulation and counts the number of pulses skipped during load voltage regulation to arrive at the energy consumed at any given voltage point. The count is minimized by altering the supply voltage and doing so tracks minimum energy point dynamically.

## 2.3 WSN CPU systems

The design constraints for VR converters and assist circuits are dictated by the CPU system. As mentioned in Section 2.2, the challenge with designing converters for integrated applications is that they must be on the same process node as the rest of the CPU system. This process node constraint also sets the operating voltage range. The minimum CPU system operating voltage defines the conversion ratio that might be needed from the converter. The minimum energy sets the load current for the corresponding conversion ratio. The retention power and dynamic energy sets boundaries for the energy harvester designs. Maximum and minimum energy point frequencies give the range of clock frequency available for the CPU system which can be reused for the converter. Hence, it is important to understand the general characteristics of CPU systems. Table 2.2 summarizes the state-of-the-art minimum energy CPU systems. Note that the comparison highlights works which have monolithically integrated switched capacitor converters as such systems are better geared to achieve low voltage ultra-efficient operation.

## 2.4 Assist circuits

Realizing a fully integrated SoC requires several assist circuits that perform critical functions in the SoC. Although some of these assist circuits have appeared in isolation in the literature, it is imperative to ensure their low-power operation as well. As will become clearer later in this section, some of these employ analog techniques which are traditionally known to be power hungry in comparison to digital circuits, especially due to the continuous current drain.

### 2.4.1 Clock generators

Processors and most microcontrollers in super-threshold operations are clocked using precision quartz or trimmed CMOS oscillators. Supporting such oscillators is inherently

TABLE 2.2: Comparison of WSN CPU systems

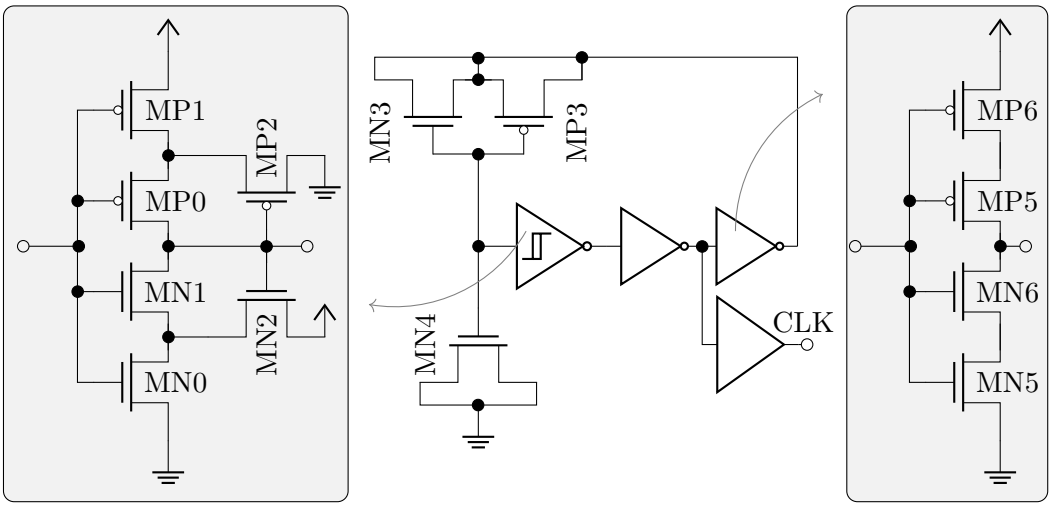
| | **MSP430 [69]** | **Subliminal [70]** | **CoreVA [71]** | **SleepWalker [72]** | **Cricket [73]** |
|---|---|---|---|---|---|
| **Technology** | 65 nm | 180nm | 65 nm | 65 nm | 65 nm |
| **$V_{DD}$** | 1.2V | 1.8V | 1.2V | 1.2V | 1.2V |
| **$P_{retention}$** | 1µW | 0.55nW | **NR** | 1.7µW | 80nW |
| **Minimum energy (pJ/cycle)** | 27.2pJ @ 500mV, 435kHz | 2.7pJ @ 400mV, 1.1MHz | 9.9pJ @ 325mV, 135kHz | 2.2pJ @ 375mV, 23MHz | 11.7pJ @ 390mV, 688kHz |
| **Minimum Voltage** | 200mV @ 10kHz | 200mV @ 200kHz | 200mV @ 10kHz | 300mV @ 10MHz | 200mV @ 27kHz |
| **Maximum Frequency** | 1.1MHz @ 600mV | 12MHz @ 0.8V | 100MHz @ 1.2V | 71MHz @ 0.5V | 66MHz @ 0.5V |
| **Int. DC-DC*** | YES | NO | NO | YES | YES |

NR: Not reported.
*: monolithic DC-DC integration.

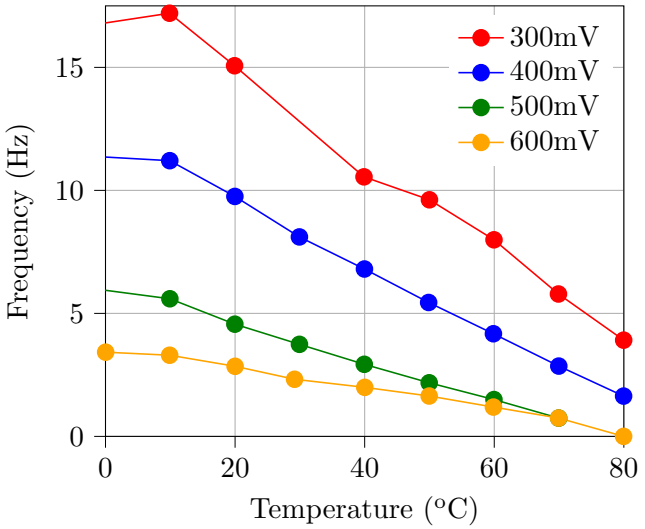
power intensive. In addition, sub-threshold CPU systems do not need precise clocks as they are unlikely to use a phase locked loop (PLL) for generating high-speed clocks. In fact, such applications could benefit from a PVT self-compensating ring oscillator based clock scheme [67]. There are however other sensor requirements that would need precision clock sources, for example timed wake up, sampling control and RF modulation. Some applications may also require oscillators with pure frequencies [77] because the signals being sensed reside in the same band as the clock frequency. From a design point of view, typical sensor SoCs require four types of clock sources.
1) A Low frequency ultra-low power oscillator
2) Programmable clock sources
3) Supply and temperature-independent clock sources
4) Sub-Hz wake-up timers.

A sub-pW timer using gate leakage [74] is shown in Figure 2.6a. Although the power expended by the circuit is the lowest reported, inaccuracy becomes a concern. The measured results are reproduced in Figure 2.6b and indicate a 25% change in frequency with 80°C change in temperature and 50% change for a 300 mV change in the supply voltage. Aging effects can also cause shifts in the oscillator center frequency.

A clever technique used to restrict the error to acceptable limits has been proposed in [75] and entails using an ultra-low power coarse uncompensated 150 nW oscillator ($OSC_{UCMP}$) and a relatively higher power, but more accurate compensated oscillator ($OSC_{CMP}$. This implementation is shown in Figure 2.7a). By operating the $OSC_{CMP}$ for a short duration only and locking the $OSC_{UCMP}$ to it the error of the $OSC_{UCMP}$ is bounded. This concept is illustrated in Figure 2.7b, with the horizontal line representing




(a) Schematic

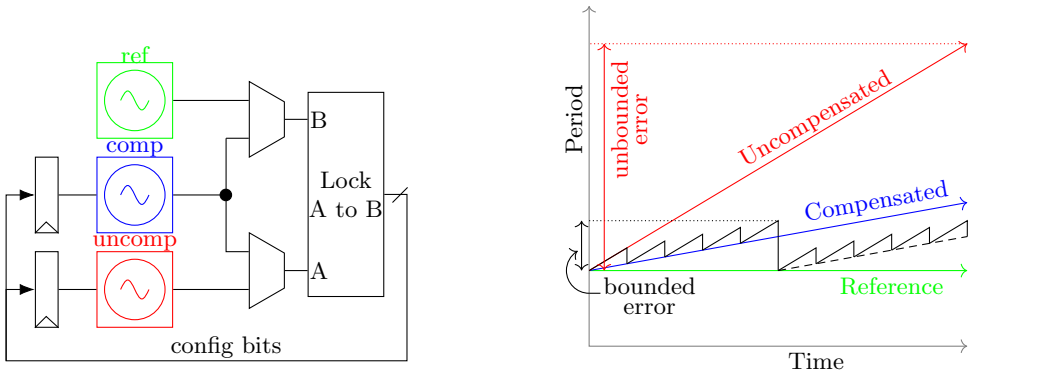



(b) Measured Results

FIGURE 2.6: Gate Leakage based sub-pW Clock. Adapted from [74]. (a) Schematic and (b) measured results with supply and temperature variation.

the period of an ideal oscillator ($OSC_{REF}$). The $OSC_{UCMP}$ has a large error that accumulates over time and is periodically locked to the $OSC_{CMP}$. Because the high-power oscillator is duty cycled, overall power consumption is minimized. The reported design was shown to work within 5 ppm/°C while consuming 150 nW.

Relaxation oscillators are attractive as an alternative PVT-independent oscillator since the oscillation time period is dictated by the RC time constant. Figure 2.8 shows common schemes for implementing relaxation oscillators [76]. Figure 2.8a uses a Schmitt inverter, which acts as a high-gain comparator with hysteresis ($V_{IL}$, $V_{IH}$). The clock period is determined by the time it takes for the capacitor to charge from $V_{IL}$ to $V_{IH}$. However, $V_{IL}$ and $V_{IH}$ vary significantly with PVT and hence the frequency is less stable.

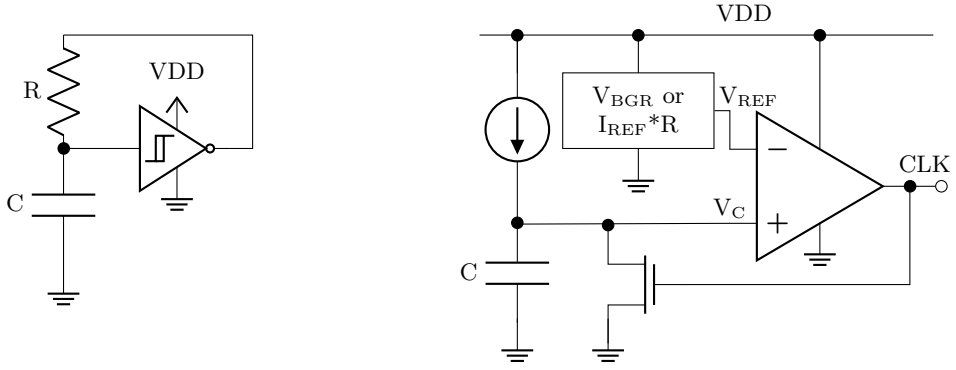



(a) Scheme. (b) Conceptual graph of period vs time.

FIGURE 2.7: Duty cycled 1.50 nW/kHz 5 ppm/°C clock scheme (a) and principle of bounding error by duty cycling (b). Adapted from [75].

Figure 2.8b shows another scheme which uses a precision comparator and a fixed reference. These designs are relatively higher power due to the comparator and PVT independent reference generators. Low-power comparators can have a significant delay which affects the frequency stability. Usually, some form of feedback is also employed to tune the reference or the comparator so as to improve frequency stability. Oscillators using fixed references essentially move the design problem from the oscillator to the reference and hence do not perform well by design in terms of stability when the supply voltage varies. This issue can be overcome using feedback to compensate for variations in reference voltage so as to achieve better stability, see [78] and [77].

An elegant solution is demonstrated in [79] which circumvents the voltage dependence problem in a correct-by-design fashion. VDD is differentially sampled to cancel out variations. Reference generators are avoided by using the op-amps' virtual ground as the reference for a trip point. In practice, the op-amps' virtual ground can move away from ideal zero due to internal offsets. Chopping is employed to average out any impact




(a) Schmitt Trigger RC Oscillator. (b) Oscillator using $V_{REF}$

FIGURE 2.8: Relaxation oscillator designs [76] using (a) device threshold and (b) using bandgaps or reference currents for comparison

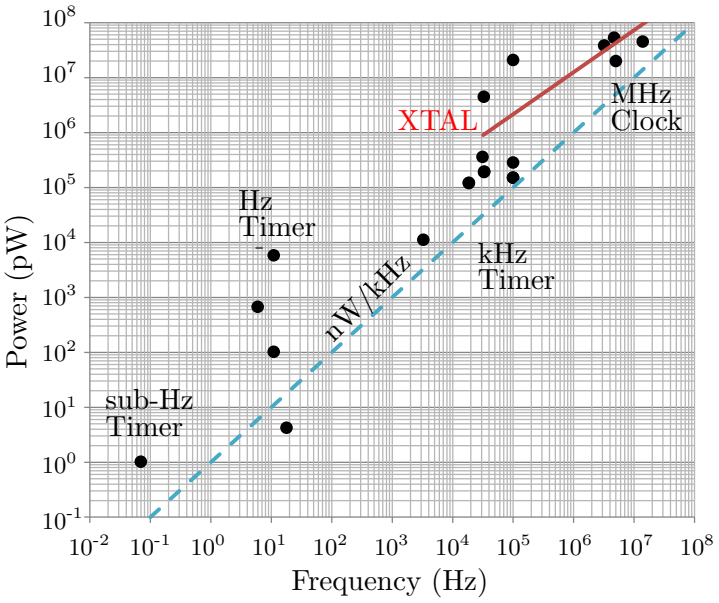



FIGURE 2.9: nW/kHz for state-of-the-art oscillator designs.

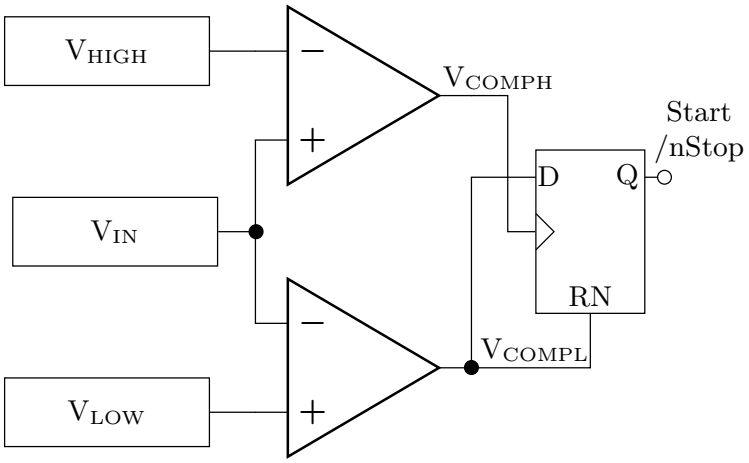



FIGURE 2.10: General scheme for voltage monitoring. Adapted from [81]

of offset thus achieving excellent stability. However, differential sampling increases the power in the timing resistor ($2V^2/R$) resulting in 11.3 nW/kHz figure of merit (FoM).

Figure 2.9 has been adapted from [80] and provides a graphical representation of the state-of-the-art clock sources for the nW/kHz metric. The graph also illustrates the performance of crystal oscillator as presented in [75]. In summary, the design of integrated sub-nW/kHz clock sources using current technology nodes remains a niche research area.

### 2.4.2 Voltage supervisory circuits

In the context of sensor SoC power management units, supervisory circuits are needed for battery and rail monitoring. The former is useful for preventing deep discharge and

TABLE 2.3: Comparison of state-of-the-art voltage monitor circuits

| | **Le [82]** | **Lee [83]** | **Mishra [81]** | **Guo [84]** | **Lee [85]** |
|---|---|---|---|---|---|
| **Node (nm)** | 180 | 180 | 180 | 90 | 180 |
| **$V_{DD}$(V)** | 1.8 | 3.6 | 1.8 | 1.0 | 3.6 |
| **Delay(s)** | 1m | 2 | 0.7 | 150n | 0.05 |
| **Power(nW)** | 3600 | 0.63 | 650 | 540 | 3.6 |
| **Area(sq.μm)** | 0.012 | 0.009 | NR | 0.088 | 0.17 |
| **Hysteresis(mV)** | NR | 200 | 66 | 432 | 77 |

NR: Not reported.

overcharging and monitoring the internal resistance, which represents battery health [85]. These circuits can be slow because battery voltages do not change in microsecond order intervals. Thus, speed can be traded with power. Rail monitoring is necessary to support dynamic transitions between sleep and active modes. Because CPU system mode changes are fast and delay can adversely affect performance and energy efficiency, rail monitors must satisfy a given response speed. The general scheme for a supervisory circuit is shown in Figure 2.10. The input voltage is compared against a reference for high and low trip points by two comparators. The decision as to the voltage being safe for operational use is made by a logical comparison of the two comparator outputs.

Note also that, in Figure 2.10, the monitor asserts *start* on the rising edge of $V_{COMPH}$ and is de-asserted on the falling edge of $V_{COMPL}$. Hence, any difference between $V_{LOW}$ and $V_{HIGH}$ establishes a hysteresis which is vital for voltage supervisory circuits. The design challenge for supervisory circuits is in limiting the power despite the two analog comparators and two voltage references while guaranteeing a certain monitoring speed. Table 2.3 provides a comparison of some of the recent works for both battery and rail voltage monitors.

# Chapter 3

# Modeling EH Sources for Circuit Co-Design

Energy harvesting enables long-term operation of wireless sensor nodes by scavenging energy from the environment. Light energy harvesting using PV cells is preferred because these cells offer relatively higher volumetric power output while allowing the nodes to be as small as possible. However, their power output can be spatially and temporally variable. This chapter reports on investigation into the performance of $cm^2$-scale PV cells, and reports on a new measurement and characterization platform. The results show that micro-PV cells perform differently from large panels: power is not simply a function of area and light levels, and manufacturing variability can be a major issue. The method presented in this chapter enables the rational design of microscale systems, including their MPPT circuits, and the evaluation of techniques for energy-neutrality (such as workload throttling) at design-time. This work was published at ENSSys 2015 [22] and the extension work related to characterising TEGs was presented as a poster at PowerMEMS 2017 [25] with the corresponding manuscript being peer reviewed.

## 3.1 Need for EH models in sensor node design

To enable mass deployment of wireless sensor node, several features are important of which size, cost, power consumption (particularly in sleep mode), and a reliable power supply are critical. For powering $cm^3$ energy-neutral wireless sensor nodes, PV energy harvesters generally offer higher volumetric power output compared to other sources [19]. Vibration energy harvesting is shown to offer higher rms output but suffers from rectification losses in the $cm^3$ wireless sensor node context. Moreover, power conversion and MPPT techniques for solar energy harvesters are relatively well understood. However, the extent to which available energy varies over time can cause problems for system designers. A sensor node design process starts with a high-level specification of the appli-

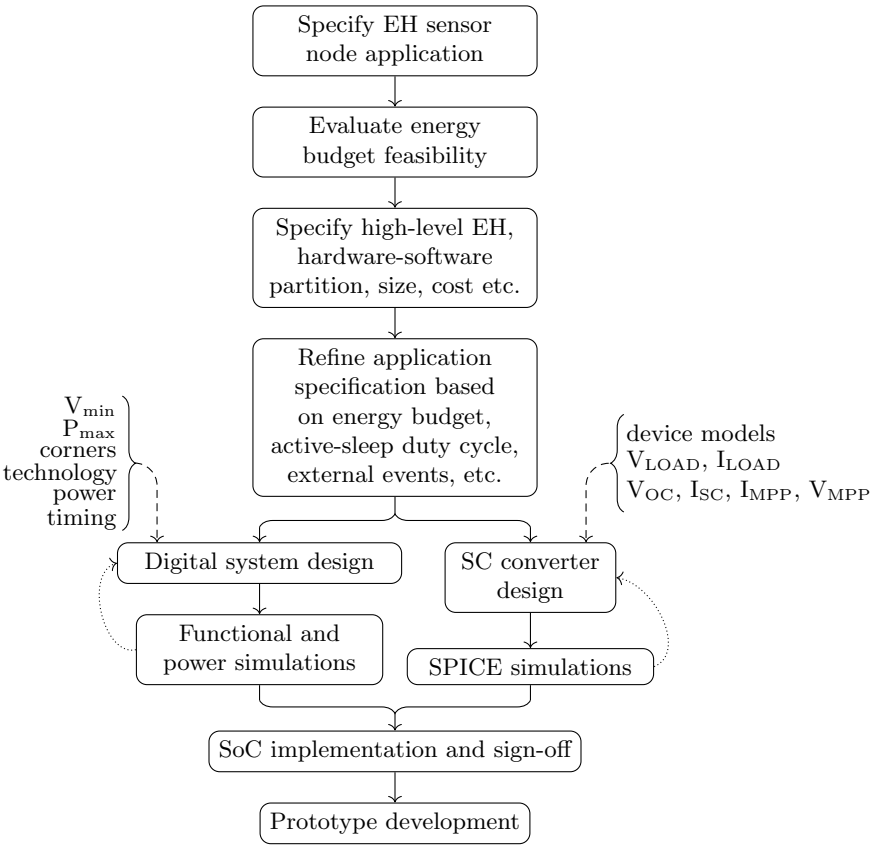



FIGURE 3.1: Conventional design flow for EH sensor nodes [88] [87]

cation outlining the nature of deployment conditions, parameters to be sensed and other aspects such as longevity etc. This application specification is distilled into a system design specification which includes the hardware-software design partitioning, functionality description, size, cost etc. The design process then, generally, includes [86] [87]:

1. Estimate the energy budget for the sensor workload,
2. choosing the energy harvester and storage device,
3. designing matching power conversion circuitry based on datasheet values,
4. designing power conversion circuitry for the digital system,
5. verifying 2 and 3 above through Simulation Program with Integrated Circuit Emphasis (SPICE) simulations
6. designing the digital system
7. functional and power-aware simulations of the digital system to estimate the power and $V_{MIN}$ of the wireless sensor node,
8. verifying 5 and 6 above through digital system simulations

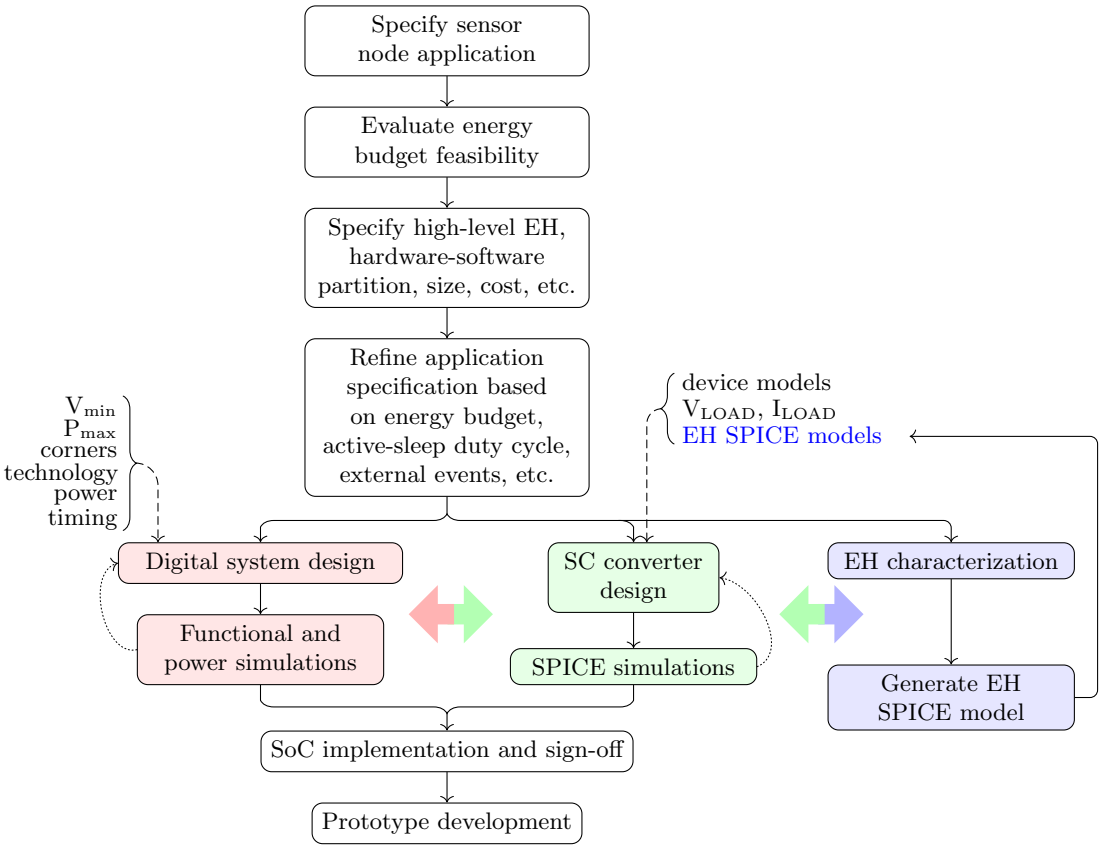



FIGURE 3.2: Design flow for EH sensor nodes with harvester characterization and modeling

9. Iterating through the above steps to refine EH, energy storage and system specifications as needed.

It is important note to that the design flow is iteratively executed to achieve the desired performance and functionality. This flow is illustrated in Figure 3.1. The complex flow adds to the cost and ability to deploy a sensor system.

Efficient and highly integrated SoC wireless sensor nodes can be implemented using current CMOS sub-threshold design techniques. Digital systems that can run on 10s of pJ per cycle have been demonstrated [69], [44]. Note that the design is aided by well characterized timing and power models for all the corner cases that the design needs to be functional.

The design of power converters for such applications can be more challenging. While the devices for power converters also benefit from such models (typically provided by the fabrication foundries), PV cell data is limited to information obtained from datasheets. The methodology proposed in this chapter allows the PV cells to be characterized under deployment conditions leading to SPICE models which eases the design of energy harvesting power converters. Figure 3.2 illustrates differences in the flow proposed compared to the conventional flow as shown in Figure 3.1. This chapter (Section 3.6) also

discusses the co-design of the power converter with the PV cells. The next chapter (Chapter 4) builds on this to also include system considerations to enable ultra-efficient energy harvesting sensor nodes.

TABLE 3.1: Comparison of schemes for automated measurement of PV cell IV curve highlighting the need for precision measurements when micro-PV cells are considered which output few 10s to 100s of μW power.

| Reference | $P_{peak}$(W) | $V_{OC}$ (V) | $I_{SC}$ (A) | Area ($cm^2$) | $\%I_{err}$ | $\%V_{err}$ | IV |
|---|---|---|---|---|---|---|---|
| [89] | 125 | 120 | 1.2 | NR | NR | NR | N |
| [90] | 80 | 20 - 80 | 1.5 - 5 | NR | NR | NR | N |
| [91] | 70m | 2.8 - 4.2 | 2.5 - 30 m | >20 | NR | NR | N |
| [92] | 245 | 35 | 3 - 8 | NR | NR | NR | Y |
| [93] | 12m | 5.5 | 3 m | NR | 28-31 | NR | Y |
| [94] | 5m | 0.02 - 10 | 430μ - 1.89 | NR | 7 | 1.5 | Y |
| This work | 600 μ | 0.05 - 2 | 90μ | 1 | <2 | <1 | Y |

*NR: Not reported.

Prior PV cell modeling efforts ( [89], [90] amongst many others) have focused on large panels with output power greater than 1 mW while, as will be presented later, the characteristics and system dynamics differ considerably when micro-PV cells are considered. Previous works exploring PV cell-based energy harvesting in the context of self-powered wireless sensor nodes [91], [95] use preliminary values from datasheets and compute the remaining model parameters iteratively. This work builds on the techniques described in [92] [94] [93] to utilize data from the continuous measurement of PV cell current-voltage (IV) characteristics to arrive at a simulation model. Note however, that the signal acquisition in this work has a better precision and is self-contained in a battery powered system. Table 3.1 provides a comparison of prior related works with the last column highlighting works that capture continuous IV sweep.

Continuous IV curve logging employed in [93] and [94] serve primarily to replay them to enable repeatable design-time experimental results via emulating energy sources. While [93] use fast 12-bit analog to digital converters (ADCs) to capture IV curves within a few milliseconds, an error of up to 70 μA is allowed during emulation, whereas the micro-PV cells characterized in this work generate much lower peak currents. Similarly, the procedure in [94] have 10s of mV and 100s of μA emulation errors, which is limiting in the context of the wireless sensor node designs considered here. This is further summarized in Table 3.1. Note that for greater than a few mW's of power output, no measurement system qualification is reported.

This work uses an embedded characterization platform to perform long-term continuous IV measurements on micro-PV cells so that post processing can be used for evaluating

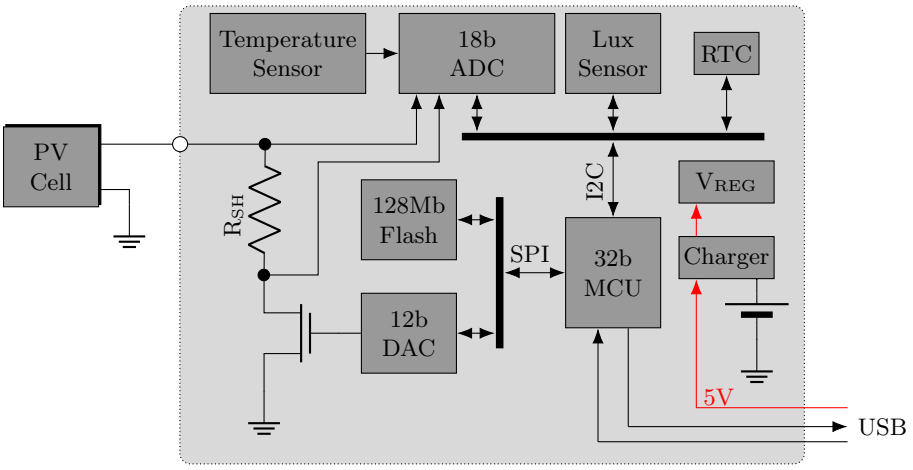



FIGURE 3.3: Characterization system.

a wide range of wireless sensor node SoC design choices. The primary objective of this work is to use the data generated to develop SPICE models for micro PV cells. The major contributions of this work are:

- Construction of a characterization system for micro-PV cells (Section 3.2), which is applied to a two-diode PV cell model (Section 3.3),
- Performance evaluation of different PV cells over extended periods (Section 3.4.1),
- Evaluation of MPPT techniques applicable to wireless sensor node SoC designs (Section 3.4.2), and consideration of the overall energy budget (Section 3.6).

## 3.2 EH characterization system

A low-cost, portable and precision measurement system is necessary to enable characterizing PV cells in large numbers. Sufficient battery and memory storage is required to support data logging over long durations. A convenient method for powering such a system would be to use batteries that can be recharged over USB because the data is eventually transferred to a PC. Thus, the desirable features for the characterization system are highly accurate measurements, large non-volatile memory, easy data transfer interface, rechargeable battery operation and small form-factor.

The block diagram for the characterisation system is shown in Figure 3.3. A 1 Ah LiPo battery, which can be charged from 5 V sources, provides sufficient energy for the characterisation system to record the performance of the device-under-test (DUT) continuously for several days unattended. A 32-bit micro controller unit (MCU) manages all the data converters and allows the PC interface for post processing of data. A light sensor, similar to that used in [96], with integrated IR and broadband spectrum detectors is used for measuring illumination levels and spectral content. A 4-channel 18-bit ADC

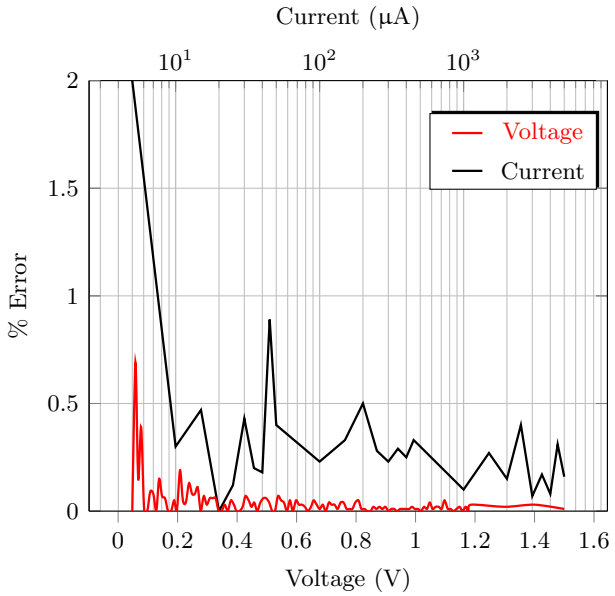



FIGURE 3.4: Percentage voltage and current error of the instrumentation in the characterisation system when calibrated against Keysight 34401A.

with internal temperature-compensated voltage reference allows ambient temperature, PV voltage and current measurements at sub-mV and μA accuracy. A 12-bit DAC is used to control the gate voltage to a power n-channel field effect transistor (FET), which in turn acts as a variable load for the DUT PV cell. All sensors, data converters and flash memory storage devices use built-in low-power sleep modes. No data is collected under extreme low-light conditions to improve battery and memory use. Lossless compression and byte-packing are used for better memory utilization. The characterisation system logs time-stamped IV and ambient sensor data and periodically writes the buffered log to the flash device. Appendix A has the complete schematic of the characterisation system and the firmware is reproduced in Appendix B.

Current measurement uses a shunt resistor. Any part-to-part variation on the value of this resistor or drift with temperature degrades the accuracy of current measurements. No compensation has been employed in the proposed setup and temperature, voltage and current measurements rely primarily on the calibration of the 18-bit ADC. The lux measurement uses the standard conversion formula provided in the device datasheet and regression is used to arrive at a correction factor. Post correction, voltage and current agree with the $6\frac{1}{2}$ digit, desktop multimeter from Keysight (34410A [97]) which was in turn periodically calibrated by test and measurement agencies. within 0.2% and 2% respectively. The relatively large (>2%) error is for currents of 5 μA or below and for higher values the error is lower (Figure 3.4). Because the light sensor covers a wide illumination range, a 20% error in measured values is observed for illumination levels of 200 lux and below while for 1000 lux or greater the error is less than 10%. The characterisation system can complete a 50-point IV sweep within 5 s. However, for measurement precision, a higher settling time is used. This can cause some of the data

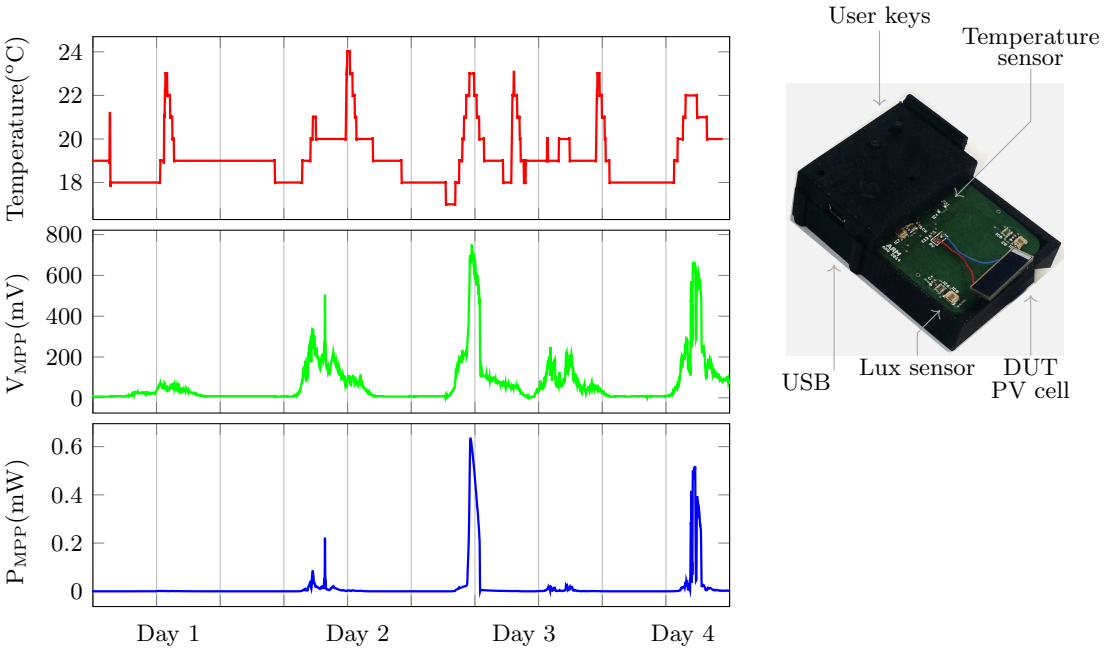



FIGURE 3.5: Plot of energy harvested over four days at an average power of 8.1 µW (left) and the 5cm x 7cm characterisation system (right) used for collecting the data.

points to be inconsistent due to the fast temporal changes in illumination. Such data points have been excluded via post processing (detailed in Section 3.3.1).

Figure 3.5 shows plots of data collected using the characterisation system from a 22 × 7 mm monocrystalline PV cell over a four-day period, at the rate of one sweep every two minutes. Indoor office illumination was used, although as in a real-world scenario, additional scattered illumination from overcast daylight was available during measurements. The cells were placed as shown in the Figure 3.5 with the photo sensitive surface pointing upwards and flat (no specific orientation angle to align with the light source was maintained). Also, the cells were laid out for testing without any cover. The cells were also not subsequently cleaned as no visible dust collection was noticed. In the Figure 3.5, $V_{mpp}$ changes are attributable solely to illumination because temperature is relatively constant (±5°C). Appendix C lists the complete script used for interfacing with the characterisation system with a computer.

## 3.3 PV cell modeling

Most PV cell modeling methods use information from datasheets to compute the parameter values for simulation models [91]. Methods that rely only on experimental data for obtaining parameters have also been explored; they are often effort-intensive, requiring several IV measurements under controlled conditions [98]. Although IV data for the entire curve is obtained from the characterisation system, an alternative modeling method [99] relies only on measurements of the 'remarkable points' - open circuit voltage,

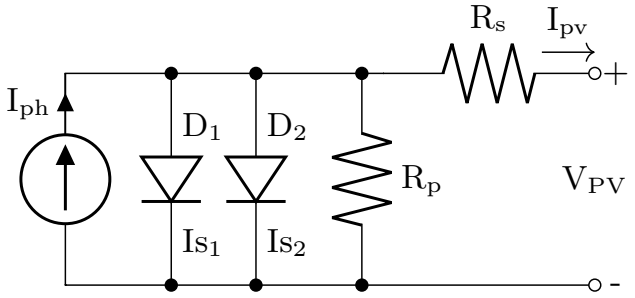



FIGURE 3.6: PV cell - two diode model ( [100]).

short circuit current and voltage and current and peak power - denoted as $V_{MPP}$ and $I_{MPP}$. The remaining model parameters are extracted iteratively based on the measurement of these remarkable points. Thus, models with better accuracy (relative to mere datasheet values) can be obtained with fewer measurements from a relatively uncontrolled environment. In addition, appreciable accuracy (<0.1% error) can be attained in the first iteration of the parameter calculation algorithm, thus reducing computation overheads. The proposed characterisation system performs with lower error if it were to be used for the measurement of only these remarkable as errors are negligible (less than 0.5%) for such high voltages and currents.

The ease of field data collection and simple computation means that the model parameters for these PV cells can be obtained without the need for relatively expensive hardware and laboratory setup and can be processed on simple micro-controller boards instead of expensive servers for computation.

### 3.3.1 PV cell parameter extraction

The two-diode model [100] is considered to be better than the single-diode model, particularly in terms of representing the behavior of PV cells under low-light conditions. The second diode ($D_2$ in Figure 3.6) which models current due to the recombination in the space-charge region ($I_{s2}$), makes the model more accurate [100]. $D_1$ models current due to ($I_{s1}$). All five model parameters, $R_s$, $R_p$, $I_{s1}$, $I_{s2}$ and $I_{ph}$ can be extracted from measured performance data, which is desirable because the modeled values are tuned for deployment conditions.

Parameter extraction relies on the following set of equations [98] which are computed in the following order. With $I_{ph} = I_{sc}$ used as the initial condition, (3.2) to (3.5) are computed over multiple iterations (depending on the desired accuracy or computational constraints). However, the logged data must first be pruned to eliminate any inconsistent sweeps. The polarity of $K_1$, which represents conductance, is an obvious indicator of inconsistent data points. The complete python script to extract the parameters from the logged data in characterisation system has been reproduced in Appendix D

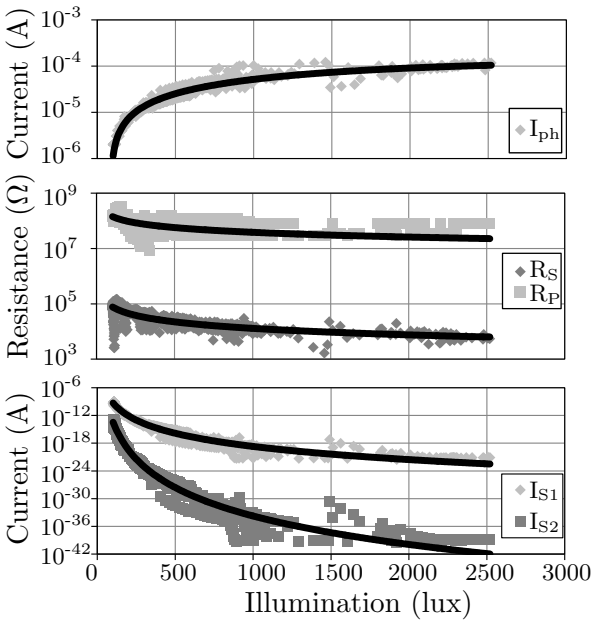



FIGURE 3.7: Extracted parameters vs illumination

$$R_{po} = -\Big(\frac{dV}{dI}\Big)_{I=I_{sc}} \tag{3.1}$$

$$K_1 = \frac{\frac{I_{mpp}}{I_{ph}-I_{mpp}} + log\Big[1 - \frac{I_{mpp}}{I_{ph}}\Big]}{2V_{mpp} - V_{oc}} \tag{3.2}$$

$$K_2 = log[I_{ph}] - V_{oc}K_1 \tag{3.3}$$

$$R_s = \frac{V_{mpp} - \frac{I_{mpp}}{[I_{ph}-I_{mpp}]K_1}}{I_{mpp}} \tag{3.4}$$

$$I_{ph} = I_{sc} + e^{(I_{sc}R_s)K_1+K_2} \tag{3.5}$$

Finally, $R_p$, $I_{s1}$, and $I_{s2}$ are computed using equations (3.6) to (3.9). In equations (3.7) to (3.9), $q$, $k$ and $T$ are the elementary charge, Boltzmann's constant and temperature in Kelvin, respectively.

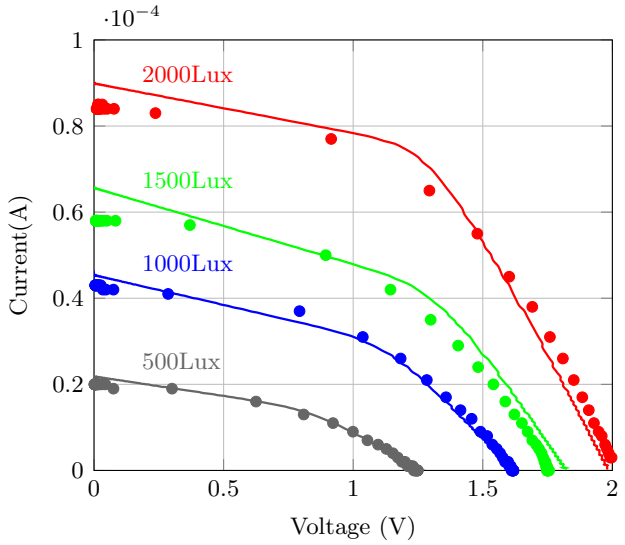
$$R_{so} = -\Big(\frac{dV}{dI}\Big)_{V=V_{oc}} \tag{3.6}$$

$$I_{s1} = \Big( - I_{sc} + \frac{V_{oc}}{R_{po}} + \frac{2kT}{q(R_{so} - R_s)}\Big)e^{-qV_{oc}/kT} \tag{3.7}$$

$$I_{s2} = 2\Big(I_{sc} - \frac{V_{oc}}{R_{po}} - \frac{kT}{q(R_{so} - R_s)}\Big)e^{-qV_{oc}/2kT} \tag{3.8}$$

$$R_p = \left(\frac{1}{R_{po} - R_s} - \frac{qI_{s1}}{kT}e^{q(I_{sc}R_s)/kT} - \frac{qI_{s2}}{2kT}e^{q(I_{sc}R_s)/2kT}\right)^{-1} \tag{3.9}$$

Figure 3.7 shows the extracted parameters plotted against illumination, with the con-




Simulated model parameters and peak power

| Lux | 500 | 1000 | 1500 | 2000 |
|---|---|---|---|---|
| $P_{peak}(\mu W)$ | 11 | 31.2 | 52.6 | 91 |
| $I_{ph}(\mu A)$ | 27 | 56 | 80 | 99 |
| $I_{s2}(A)$ | 3.3e-28 | 1.6e-34 | 5.2e-38 | 1.4e-40 |
| $I_{s1}(A)$ | 1.1e-16 | 1.6e-19 | 3.4e-21 | 2.1e-22 |
| $R_p(M\Omega)$ | 86 | 58 | 46 | 39 |
| $R_s(k\Omega)$ | 23 | 13 | 10 | 8 |

FIGURE 3.8: Results obtained from SPICE simulations (continuous lines) with measurements from characterisation system (individual points).

tinuous lines showing the fitted model. A few points are worthy of mention to illustrate how the parameters would influence SPICE simulations.

Both resistances decrease with illumination. $R_p$ is three orders of magnitude higher than $R_s$; for the micro harvester sizes considered here, $R_s$ is on the order of tens of k$\Omega$. The diode currents ($I_{s1}, I_{s2}$) also decrease with illumination as $I_{s1}$ and $I_{s2}$ influence the knee voltage of the diodes causing the PV cell output voltage to be higher at brighter illumination levels.

### 3.3.2 SPICE simulations

The SPICE model for the PV cell uses fitting functions for each parameter, then produces a function of illumination and temperature. The fitted functions from Figure 3.7 feed into the schematic shown in Figure 3.6. The resulting sub-circuit is simulated using HSPICE®, a standard EDA tool used for SoC designs. Simulated and measured values of the remarkable points are compared in Table 3.2. The $P_{mpp}$ and $V_{oc}$ readings agree within 10%, while the error for $I_{sc}$ (particularly at low light levels) is 18%.

Further, the PV model is simulated with an ideal current source as load for different illumination settings and IV curves were obtained. These are compared with the measured data in Figure 3.8. The expected use case of the model is to simulate switched capacitor converters with the PV cell model sub-circuit as the power source with illumination is set as a simulation parameter. In addition, transient simulations can be carried out by setting the illumination as a function of piece-wise-linear voltage source to evaluate metrics such as speed of MPPT convergence, power overheads and relative gains in harvested energy for different MPPT techniques. The SPICE listing of the model is listed

below.

```
.subckt pvcell p n lux=200
.param rp_scale = lux>1800?2:1
.param iph='-9e-12*lux*lux+70e-9*lux-5e-6'
.param is='5e9*pwr(lux,-9.5)'
.param is2='9e27*pwr(lux,-20.54)'
.param is_var=agauss(0,1e-11,3)
.param rp_var=agauss(0,1e5/lux,3)
.param rs_var=agauss(0,0.77e3/lux,3)
.param is1=is+is_var
.param rp='rp_scale*3e6*pwr(lux,-0.57)+rp_var'
.param rs='3e6*pwr(lux,-0.78)+rs_var'


ip n pint 'iph'
dmain pint n dpv
d2 pint n dpv2
rshunt pint n 'rp'
rseries pint p 'rs'

.model dpv  d (level=1 is = 'is1' cjo =1.09e-9  n   =1.915
   ibv =2.00e-2)
.model dpv2  d (level=1 is = 'is2' cjo =0.88e-9 n   =1.9
   ibv =1.32e-3)
.ends pvcell
```

## 3.4 Implications of EH sources on circuit design

Multiple characterisation system units were fabricated and used to evaluate several high-efficiency surface-mount monocrystalline PV cells from different manufacturers. The cells were measured both indoors and outdoors over a period of several days. The results are described first, followed by a discussion of the fractional open circuit voltage (FOCV) MPPT technique and finally a consideration of the PV cell area and wireless sensor node energy budget.

TABLE 3.2: PV cell measured and simulated remarkable points for different illumination settings

| lux | $V_{oc}$ (V) | | | $I_{sc}$ (µA) | | | $P_{mpp}$ (µW) | | |
|---|---|---|---|---|---|---|---|---|---|
| | M | S | %E | M | S | %E | M | S | %E |
| 500 | 1.25 | 1.24 | 1 | 22 | 18 | 18 | 10 | 11 | 9 |
| 1000 | 1.61 | 1.62 | 0.4 | 45 | 43 | 5 | 32 | 31 | 3 |
| 1500 | 1.84 | 1.76 | 5 | 66 | 58 | 12 | 48 | 52 | 8 |
| 2000 | 1.98 | 1.98 | 2 | 90 | 84 | 7 | 84 | 90 | 7 |

M=measured, S=simulation, E=Error

### 3.4.1 Characteristics of harvested energy

One significant advantage of the characterisation system is the availability of raw IV data (Figure 3.5) which removes ambiguity in the maximum power point location and allows potential strategies for MPPT to be assessed off-line.

Assuming an ideal MPPT technique, the maximum power that can be extracted from the DUT is shown, along with the voltage levels at which $P_{mpp}$ is obtained ($V_{mpp}$). This information is a key input for switched capacitor converter design because it enables the estimation of conversion ratios and cold-start voltages. $P_{mpp}$ integrated with time would be the ideal-case energy budget of a wireless sensor node operating for the same duration.

In practice, however, only a fraction of this ideal energy can be used. First, conventional circuits fail to use harvested energy at low voltages and about 30% of the aggregate energy is available at $V_{mpp} < 0.3V$. SoC implementations of switched capacitor converters demonstrate functional voltage limits as low as 0.14 V [101] but for voltages below 0.35 V the conversion efficiency is limited to 50%. This is because at low input voltages but relatively fixed output voltage, the converter will need a higher conversion ratio, which increases losses due to increased number of switches and fly capacitors. Thus, there is a further energy loss due to the nonideal harvester-converter interface. Secondly, power conversion losses in the conditioning circuitry will limit $P_{mpp}$ utilization even when $V_{mpp} > 0.3V$. A comprehensive review of integrated implementations of switched capacitor converters [53] shows up to 90% conversion efficiency under best-case conditions. However, the temporal variations of $V_{mpp}$ limit conversion efficiencies; this calls for careful choice of harvester and converter design parameters.

The lower voltage limit on switched capacitor converters could potentially be overcome by connecting several cells in series, to increase the PV output voltage. Denoting the number of series connected cells by $N_s$ we can evaluate the trade-offs for PV cells with $N_s$=1 and 2. These cells were measured under similar illumination and temperature conditions. The results are shown in Figure 3.9. The daily-total energy indicates that output from cells with $N_s$=2 is 14% lower than that from cells with $N_s$=1 (Day 3). The loss may be attributable to cell interconnection losses. Therefore, to increase harvested energy, $N_s$=1 would be preferred. However, for switched capacitor converters with a 300 mV lower voltage limit, the result favors cells with $N_s$=2. This is highlighted in Figure 3.9(b) which shows cells with $N_s$=2 having higher voltage $V_{mpp}$ compared to the cell with $N_s$=1, but lower current ($I_{mpp}$). In Figure 3.9(b) the aggregate is recomputed (and stated in the graph) with the integral lower limit at 300 mV (dashed line in Figure 3.9(b)) then the harvested energy from $N_s$=2 cells is 13% higher than that obtained from cells with $N_s$=1.

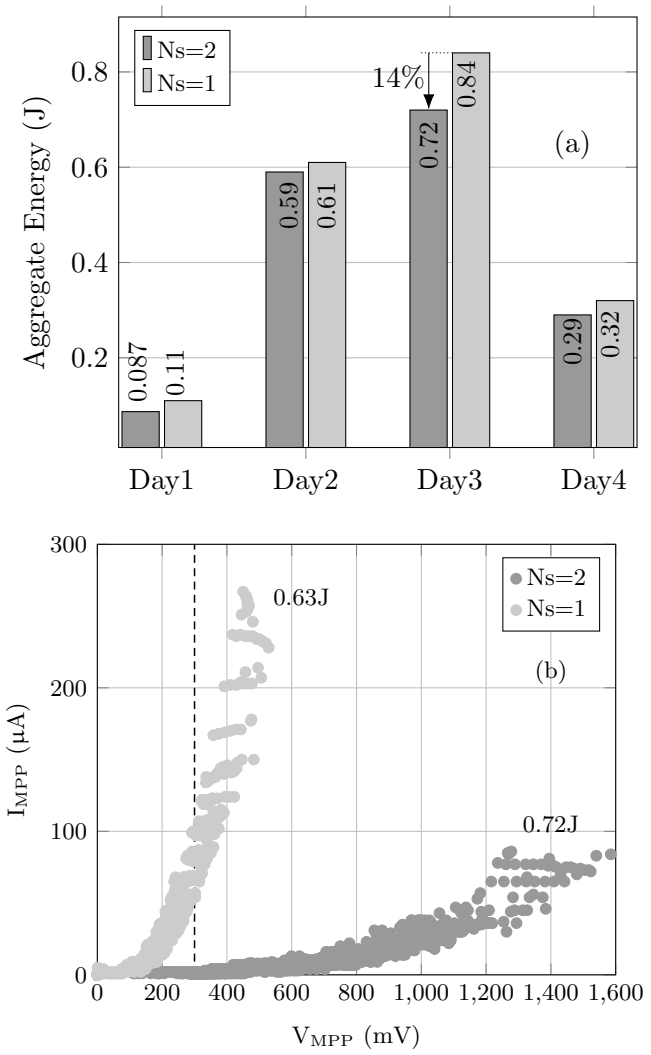



FIGURE 3.9: Aggregate energy from PV cell with $N_s$=1,2 (a) Relative energy harvesting performance and (b) $V_{mpp}$ vs $I_{mpp}$.

### 3.4.2 MPPT overheads

Several MPPT techniques have been proposed for large panels in which the power available justifies expending a small percentage of the power on MPPT circuitry, but in the context of wireless sensor node where the overall power budget is of the order of tens of µWs, the choice of MPPT technique is crucial. The FOCV technique (see Section 2.1.2), a scheme where the harvester is loaded to draw enough current until the terminal voltage is a fraction of the open circuit voltage [17], is preferred in integrated designs [102] due to the relatively lower implementation costs (power and area). The drawback with this technique is that the load needs momentary disconnection to measure the open circuit voltage $V_{oc}$ (see Section 2.1.2). Once this measurement is complete this technique relies on loading the source until the output voltage drops to a certain fraction of $V_{oc}$. The issue of momentary power loss is however, usually and simply overcome by using decoupling capacitance. The bigger challenge is in deciding the optimal value of the fraction

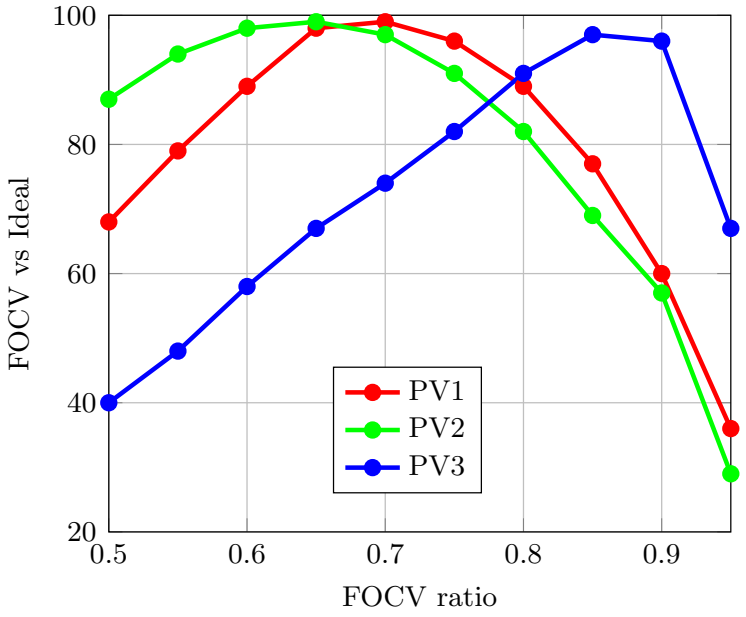
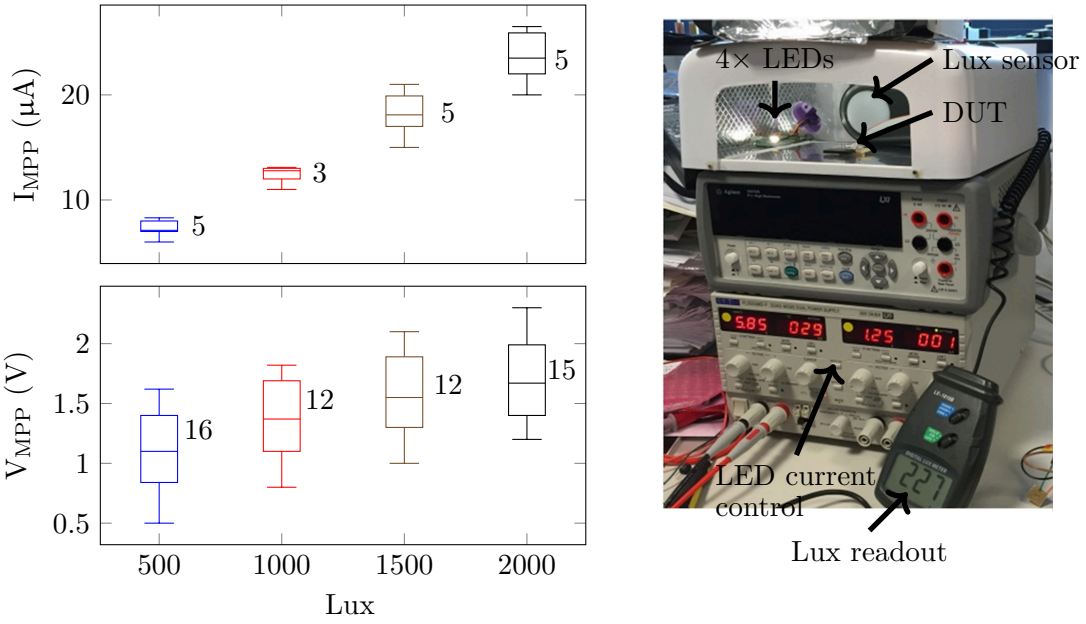



FIGURE 3.10: PV output power vs fraction in FOCV MPPT technique based on measurements from Figure 3.5. Here PV1 and 2 use $N_s$=2 while PV3 uses $N_s$=1 and has a fraction much closer to unity. This analysis highlights the need to characterise PV cells prior to implementing FOCV MPPT.

- denoted in this thesis as $K_{opt}$. This aspect has been investigated in this work.

Three different micro-PV cells were evaluated against the ideal $P_{mpp}$ obtained from the family of measured IV curves. The data collected from the characterisation system has the full IV sweep which allows the true maximum power point to be calculated. This true maximum power point (MPP) was compared with one obtained from post processing the data. Post processing involved determining the voltage and current in the IV sweep data such that the voltage is a specific fraction $K_{opt}$ of the open circuit voltage. The open circuit voltage is also obtained from the IV data. This would allow us to investigate the difference in power between the true MPP and the one obtained using $K_{opt}$. More importantly, this analysis would point out if $K_{opt}$ is consistent across different PV cells.

Figure 3.10 shows this result for various $K_{opt}$ = 0.5 to 0.95. Two of the cells used $N_s$=2 along with one which has $N_s$=1. They show contrasting fractions at which the $P_{mpp}$ is close to ideal. Cells PV1 and PV2 show $K_{opt}$=0.65 and 0.7 while PV3 has a fraction close to 0.9. This difference is attributable to the fill-factor of the PV cells. For the data shown in Figure 3.10, the power difference between the ideal and $K_{opt}$ case is about 1-3% of ideal. That is to say, if the PV3 was operated at $K_{opt}$ =0.7, then it would only provide 70% of the peak power whereas PV1 and PV2 would be closer to their true peak power. The implementation cost for the FOCV method is not accounted for. Typically, this cost is due to additional power loss in the divider resistors, comparators and other tuning circuitry. The impact of this overhead will be revisited in Section 3.6 within a system context.




(a) Measured Variation. (b) Measurement setup

FIGURE 3.11: (a) Variation for 9 mono-crystalline PV cells from the same manufacturer with numeric values indicating ratio of standard deviation over mean as a percentage and (b) automated measurement and characterisation setup.

### 3.4.3 Variability

Another design-critical issue is the variation in PV cells, which must be accounted for in simulation to evaluate best- and worst-case scenarios. Figure 3.11 shows the spread of $V_{oc}$ and $I_{sc}$ (see Section 2.1.2) for nine PV samples measured under varying indoor lighting conditions. The box indicates the spread in the measured samples along with the mean, and the whiskers indicate the expected three-sigma limits. The spread in $P_{mpp}$ is expected to follow that of $V_{oc}$ because $I_{sc}$ has a relatively tighter distribution (lower sigma-over-mean ratio). The 10-20% part-to-part variation makes margining wireless sensor node designs very difficult. However, it is possible to capture such variation in SPICE using user-defined parameters, which can later be used in Monte-Carlo sweeps for statistical yield analysis and also for evaluating worst-case operating conditions.

Another design-critical issue is the variation in PV cells, which must be accounted for in simulation to evaluate best- and worst-case scenarios. Figure 3.11 shows the spread of $V_{oc}$ and $I_{sc}$ for nine PV samples measured under varying indoor lighting conditions. The box indicates the spread in the measured samples along with the mean, and the whiskers indicate the expected three-sigma limits. The spread in $P_{mpp}$ is expected to follow that

of $V_{oc}$ because $I_{sc}$ has a relatively tighter distribution (lower sigma-over-mean ratio). The 10-20% part-to-part variation makes margining wireless sensor node designs very difficult. However, it is possible to capture such variation in SPICE using user-defined parameters, which can later be used in Monte-Carlo sweeps for statistical yield analysis and also for evaluating worst-case operating conditions.

Another design-critical issue is the variation in PV cells, which must be accounted for in simulation to evaluate best- and worst-case scenarios. Figure 3.11 shows the spread of $V_{oc}$ and $I_{sc}$ for nine PV samples measured under varying indoor lighting conditions. The box indicates the spread in the measured samples along with the mean, and the whiskers indicate the expected three-sigma limits. The spread in $P_{mpp}$ is expected to follow that of $V_{oc}$ because $I_{sc}$ has a relatively tighter distribution (lower sigma-over-mean ratio). The 10-20% part-to-part variation makes margining wireless sensor node designs very difficult. However, it is possible to capture such variation in SPICE using user-defined parameters, which can later be used in Monte-Carlo sweeps for statistical yield analysis and also for evaluating worst-case operating conditions.

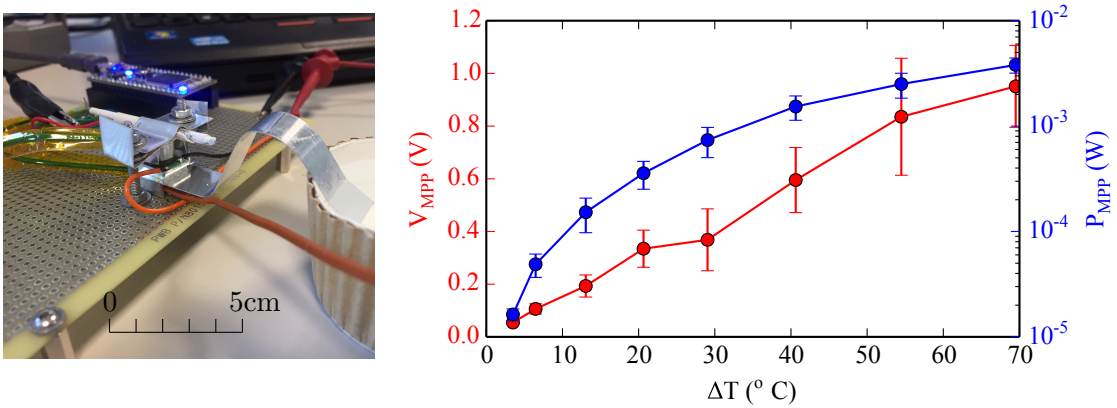
## 3.5 TEG characterization and measurements

The characterisation system was used to also characterize commercially available $cm^3$ TEGs. The differential temperature surfaces for the TEG were emulated using a heater and cooling-fin setup as shown in Figure 3.12a. The ratio of $V_{MPP}$/ $V_{OC}$ is ≈0.5. The ratio, however, shows part-to-part variation and dependence on temperature (Figure 3.12b). The error-bars on the voltage and power trace highlight the variability observed when measuring five different samples for similar temperature gradients. However, the coefficient of variance ($\sigma/\mu$) is 10 × lower compared to COTS microscale PV cells. Harvested power from ambient heat sources (hot-water dispenser and room radiator) is measured to be between 10-1000 μW (Figure 3.13).

## 3.6 WSN energy budget estimation

Unlike many computation platforms, most wireless sensor nodes benefit from a relatively fixed repetitive workload across their lifetime. Thus, knowledge of the end application and characteristics of the energy harvester can be used to optimize sensor node sizes during design. It is however challenging to estimate and optimize for the worst- and best-case scenarios.

Consider, for example, the energy reported for the example wireless sensor node SoC in [73]. While running software code the active energy is 12 pJ/cycle at 1 MHz. The sleep power is 80 nW. Assuming a 10:1 sleep:active duty cycle (D) and the CPU is active




(a) (b)

FIGURE 3.12: (a): Characterization setup for TEG showing TEG sandwiched between heating element and cooling fin for establishing temperature differential and (b): Measured power vs $\Delta$T and variability in $V_{MPP}$

for 3000 cycles, $N_{act}$ for iteration of the loop, the total energy $E_{tot}$ required for one loop of sensor activity is translates to 38.4 nJ as shown in 3.10).

$$E_{tot} = E_{act}N_{act} + P_{ret}N_{act}D \tag{3.10}$$

From Figure 3.9(a), the lowest energy daily total obtained with $V_{mpp} > 300$ mV, is 87 mJ over 24 hours. The MPPT efficiency must also be factored in to calculate obtained energy. The data in Figure 3.9(a) assumes an ideal MPPT implementation. If the MPPT implementation were to be 75% efficient with a conversion efficiency of 50%, the energy available would be 32.6 mJ (Equation (3.11)).

$$E_{avail} = E_{EH}\eta_{conv}\eta_{mppt} \tag{3.11}$$

This wireless sensor node energy requirement per day is illustrated by the lowermost bar in the graph in Figure 3.14. Compare this with the Day-1 harvested energy from Figure 3.9(a). The aforementioned workload can therefore, run continuously for 0.8 million loops or 7 hours with this limited energy. This discussion suggests that, within the limits of the assumptions made, about 4 $cm^2$ PV cell area would be necessary for the long-term continuous operation (for 24 hours) of this wireless sensor node SoC.

If the sensor hardware is fixed, then the software can be changed to throttle the duty cycle to ensure a 24 hour operation. In the above example, a duty cycle greater than 30:1 would provide a 24 hour operation for the same PV cell area or sensor volume. However, energy is also lost due to self-discharge in the energy storage devices, temperature, and other non-ideal phenomena.

Alternatively, considering best-case Day-3 energy aggregate from Figure 3.9(a), 0.5 $cm^2$

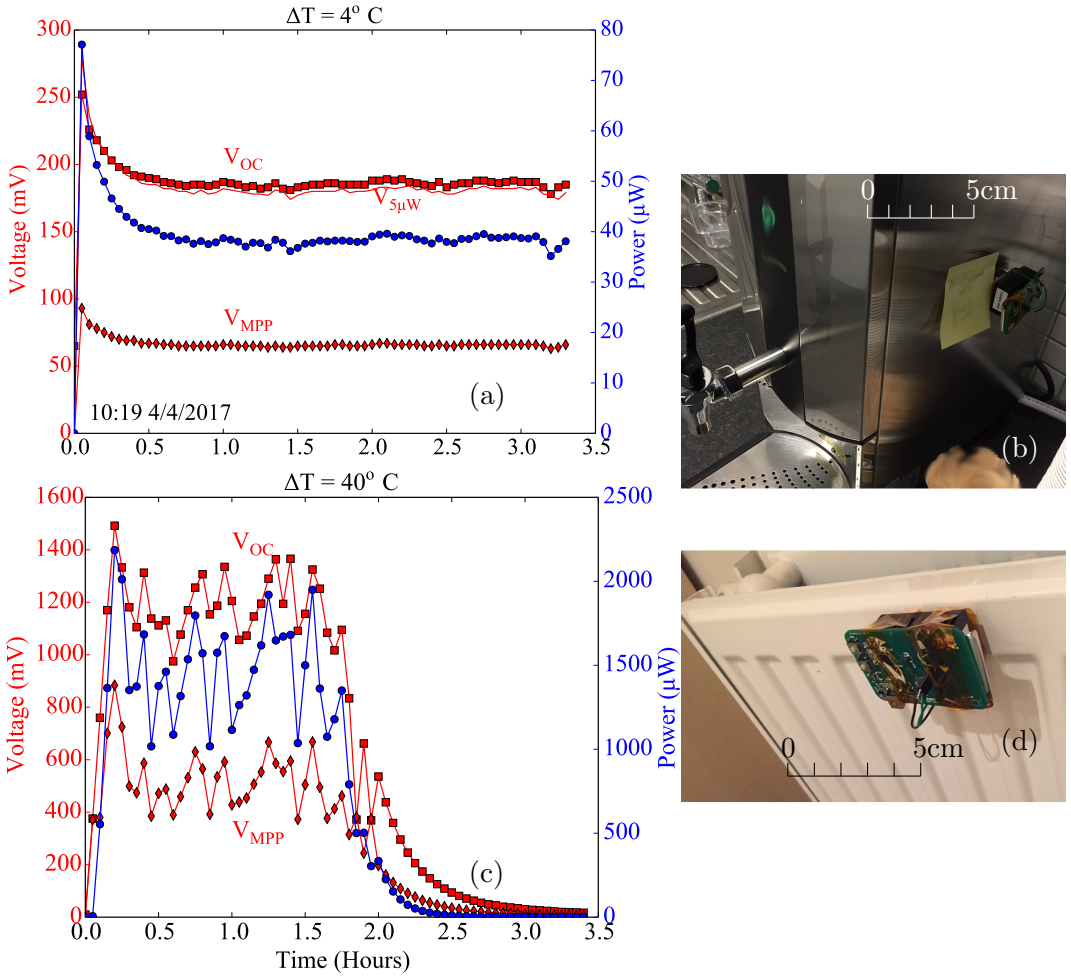



FIGURE 3.13: TEG measurements under field conditions (a,b): Low ΔT (4°C) conditions and (c,d): Large ΔT (40°C) conditions

cell area would be needed for a 24 hour energy-neutral operation. This is an 8× difference between the worst- and best- case scenarios. It is noted that the PV output is not necessarily a linear function of the area. This discussion also does not consider the part-to-part variation of PV cells. From Figure 3.11, the best PV cell has roughly 2× the output voltage of the worst cell at 1500 lux. In total, this means as much as 16× margins need to be added to the design (as shown by the topmost bar in the graph in Figure 3.14) which is unlike traditional design margining of 10%.

## 3.7 Summary

In summary, the proposed methods of EH characterisation, modeling and co-design help draw out the specifications for switched capacitor converters, choose the appropriate MPPT technique and specify design requirements for MPPT implementations. The overall view helps estimate wireless sensor node energy budgets and workloads for the

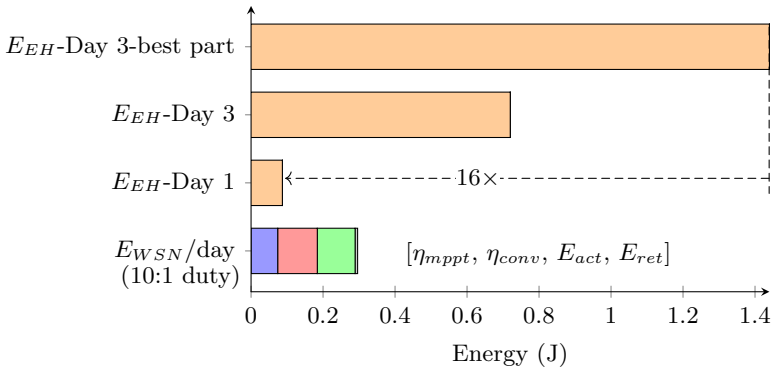



FIGURE 3.14: wireless sensor node energy budgeting in view of variation of energy harvesting. Lowermost bar shows the wireless sensor node energy budget per day as a stack of active and retention energy of the digital system multiplied by the conversion and MPPT efficiency. Day 1 and day 3 measurements are from Figure 3.9 which is compounded by part-to-part variation as shown in 3.11 to produce the topmost bar showing as much as 16× margining.

sensor. The energy estimate also has implications on the type and size of battery or super-capacitor that can be used in a sensor node design. By relying on field measurements, addressing variation and potential worst-case conditions this modeling method helps design integrated wireless sensor nodes with tighter tolerances at lower costs.

# Chapter 4

# Selective Direct Operation

This chapter applies aspects of the energy harvester and wireless sensor node power budgeting from the previous chapter to a real-world system consisting of a sub-threshold CPU. Such systems can be used in sensor node which form the extreme edge of IoT systems. As mentioned in the previous chapter, the objective of power budgeting such systems is to achieve energy neutrality, i.e., harvesting at least as much energy as is needed for sensory activities. Doing so however, is complicated by variations in environmental energy and application demands. Conventional systems use separate power converters to interface between the harvester and storage, and storage to the CPU system. Fully integrated reciprocal power conversion is known to perform both roles thereby eliminating redundancy and minimizing losses. This research proposes enhancing this topology with 'selective direct operation', which bypasses the converter completely when appropriate. The integrated system, with an 82% bidirectional conversion efficiency, was validated in 65 nm CMOS with only the harvester, battery and decoupling capacitors being off-chip. Optimized for operation with $cm^2$ photovoltaic cell and a 32-bit sub-threshold processor, the scheme enables up to 16% otherwise wasted energy to be utilized to provide >30% additional compute cycles under realistic indoor lighting conditions. Measured results show 84% peak conversion efficiency although conversion losses can be eliminated during direct operation. Energy-neutral execution of benchmark sensor software (ULPBench) with 260 lux cold-start capability has also been demonstrated. This work was published as a journal paper in transactions on circuits and systems [23].

## 4.1 Energy neutrality and two-stage power conversion

Designing for energy neutrality in wireless sensor node SoCs presents a complex optimization problem for circuit and system designers: the available energy depends on ambient conditions and is limited by power conversion efficiency, while the energy expended depends on run-time conditions and software workloads. These problems are

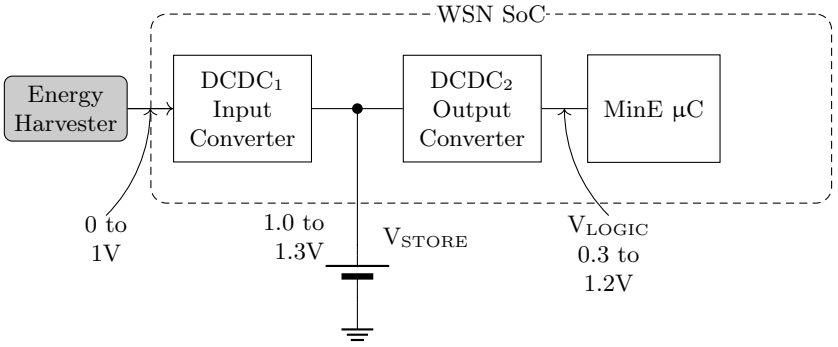



FIGURE 4.1: Conventional two stage power conversion showing input converter to track MPPT and output converter to track minimum energy point

exacerbated in form-factor constrained applications, making do with small energy harvesting and storage devices. Sensor systems with $cm^2$ form factors are attractive because they offer a good balance between harvesting and storage capacity, and overall cost.

A typical energy harvesting sensor system is illustrated in Figure 4.1 [86]. For this work, the CPU and its associated elements are referred to as the "CPU system". Energy is stored in a supercapacitor or battery, which then acts to decouple the CPU system from the dynamics of the energy harvesting. An input converter allows the spatiotemporally variable harvested energy to charge the storage device. The design challenge here is to ensure that the harvester and input converter in combination can *maximize harvested energy*, meaning that the converter must be designed to minimize conversion losses while ensuring maximum power transfer by MPPT. An output power converter provides a regulated supply to the CPU system. In many cases, the output converter is part of the integrated voltage regulator. The challenge for the combination of CPU system and the output converter is to *expend minimum energy* while undertaking sensory activities.

This work focuses on the aforementioned design challenges and demonstrates a $cm^2$ system that achieves energy neutrality while running the ULPBench software benchmark [103]. The design is centered on a highly efficient *reciprocal power converter*, which can perform both input and output power conversion, and system optimization steps for *selective direct operation* of the CPU system in certain modes, bypassing the conversion stages entirely. The key contributions are:

1. An integrated energy harvesting scheme that allows otherwise wasted energy to be used for computation.
2. A reciprocal converter with the highest bidirectional conversion and area efficiency.
3. The demonstration of a $cm^2$ energy-neutral system executing an industry-standard IoT software benchmark at very low indoor light levels (160 Lux).

Real-world measurements of $cm^2$ PV cells were presented in the previous chapter. This work first builds a case for power convertors which use the output of such micro-scale

energy harvesting devices and analyzes the impact of such conversion on state-of-the-art minimum energy CPU systems (Section 4.2). The analysis reveals an opportunity to exploit redundancies and improve energy utilization. The proposed design is presented (Section 4.3), along with corresponding measured block-level results (Section 4.4). Overall system performance is then presented (Section 4.5). The work focuses on $cm^2$ PV cells, but the techniques presented can be applied to other forms of energy harvesting e.g., thermoelectric generators.

## 4.2 Power requirements of IoT devices

To identify the design and operating requirements of energy harvesting IoT devices, a good understanding of the characteristics of both minimum energy CPU systems and microscale harvesters is essential. Table 2.2 summarizes the properties of leading minimum energy CPU systems. Minimum energy operation is possible at lower supply voltage [104] which is on the order of 300-500 mV for current CMOS technology nodes. Note that (from Table 2.2) minimum energy CPU systems feature nW to µW order sleep/retention power and 8∼10× higher power in active mode at the minimum energy point. Although the minimum energy point in most systems is achieved at ≈370 mV, the minimum functional voltage ($V_{min\text{-}LOGIC}$) is about 200 mV. Notable exceptions are the MSP430 clone [69] where minimum energy point of 500 mV is dictated by the large static random access memory (SRAM) array, and SleepWalker [72], where LP/GP process mix contributes to an increase in the minimum functional voltage (300 mV).

### 4.2.1 Voltage conversion requirements

Conventional two-stage conversion in wireless sensor nodes, as shown in Figure 4.1, consists of input and output power converters. The output converter performs the important task of converting the energy available at the $V_{STORE}$ to $V_{LOGIC}$ levels. Therefore, the output converter must be very efficient at conversion and also track the minimum energy point of the CPU system. In contrast, the input converter must be designed to support a range of conversion ratios to charge the energy storage device efficiently from available ambient energy ($V_{EH}$). Tightly coupled fully-integrated converters are desirable to help with fast dynamic voltage and frequency scaling (DVFS) when enabling minimum energy operation with switched capacitor converters being the appropriate choice in low-cost digital CMOS processes.

The Figure 3.5 has been annotated with the required conversion ratio for the input converter used in this application which illustrates one of the design criteria for the switched capacitor converters. The wide input voltage variation requires a ratio ranging between 1.5 and 12. However, switched capacitor converter loss depends on its conversion

ratio [66]. The power conversion efficiency of switched capacitor converters is given as (4.1), and (4.2) gives a breakdown of the conversion loss ($P_{LOSS}$).

$$\eta_{CONV} = \frac{P_{OUT}}{P_{OUT} + P_{LOSS}} \tag{4.1}$$

$$P_{LOSS} = P_{SW} + P_{CAP} + P_{SSL} \tag{4.2}$$

Here, $P_{SW}$ and $P_{CAP}$ are switching loss and bottom-plate loss, respectively. $P_{SSL}$ is the $I^2R$ conduction loss due to the inevitable drop across the output impedance of the switched capacitor converter. Each of these loss components increase at higher conversion ratios [66] due to the increased number of switching and reactive elements (capacitors). This increase in conversion loss at lower input voltages sets an artificial limit on the converter input voltage ($V_{min\text{-}DCDC}$).

If the input converter ratio is fixed at 2 (so as to maximize conversion efficiency) then the $V_{min\text{-}DCDC}$ is approximately 0.6 V. This high value can be detrimental in the case of $cm^2$ PV cells where the $V_{MPP}$ rarely exceeds 0.6 V, even under bright light. For microscale sensor systems this constraint means that either a larger PV cell or an array with multiple cells is required; or the system throughput will require throttling. Note, however, that for sub-threshold systems $V_{min\text{-}LOGIC}$ is well below 0.6 V and has the potential of utilizing part of the energy available at sub-$V_{min\text{-}DCDC}$ levels, provided the CPU system can be managed carefully.

Apart from this additional energy utilization, conversion losses can be minimized by eliminating redundancy. Note that most integrated output converters in state-of-the-art CPU systems (Table 2.2) include a voltage doubler (conversion ratio of 2). While some works do include additional ratios [69], [73], peak efficiency is reported for a ratio of 2. Input converters also prefer this ratio, as described in recent published works [105] [101]. Hence, designs can eliminate multiple converters performing the same function.

### 4.2.2 Two-stage vs reciprocal conversion

The need for minimizing lossy conversion stages has been recognized in prior works [46], [44] [106] and various improvement schemes have been proposed. A recent work [106] (scheme redrawn in Figure 4.2 (a)) explores stacking of PV cells to increase energy harvesting output voltage which partly alleviates the problem of $V_{min\text{-}DCDC}$. Powering the load (a video monitoring SoC) helps reduce conversion losses directly. However, the observation of a flat $V_{MPP}$ (based on simulation results) was used to preclude MPPT techniques and associated overheads. This is contrary to measurements carried out (as described in Chapter 3) which show wide $V_{MPP}$ variation with ambient energy (Figure 3.5).

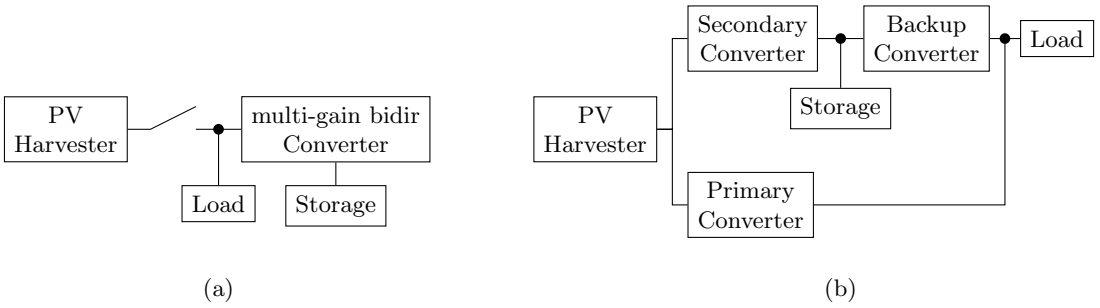



FIGURE 4.2: Prior works implementing single converter operation during periods of low ambient energy with (a) [106] showing direct and bi-directional conversion but without MPPT and (b) [46] showing multiple converters with single converter during periods of low ambient energy.

Some implementations [46] (scheme redrawn in Figure 4.2 (b)) have employed multiple converters where a primary converter is used when available energy is limited and two-stage conversion becomes excessively lossy. If sufficient energy is harvested, a secondary converter is enabled to charge a storage device. A third backup converter is employed to power the load using stored energy at times when no ambient energy is available. This modular approach is useful because each converter can be optimized for its specific purpose, but the area overhead is significant. Further, an off-chip inductor is used (although it is time-shared among all converters) which poses integration challenges.

In contrast, [44] uses a single, fully-integrated switched capacitor converter. However, the load uses a further regulation stage (low drop-out regulator) which limits conversion efficiency to 66%. Neither of these approaches overcomes the harvesting limit imposed by $V_{\text{min-DCDC}}$.

## 4.3 Proposed single converter and direct operation scheme

The work described in this chapter:

1. Avoids two-stage conversion (as in Figure 4.2) by using a single reciprocal converter (Figure 4.4) with high bidirectional conversion efficiency which can adapt to varying light levels.

2. Enables the use of an optimum conversion ratio and overcomes the harvesting limit imposed by $V_{\text{min-DCDC}}$ by using selective direct operation to exploit the ultra-low $V_{\text{min-LOGIC}}$ offered by state-of-the-art minimum energy CPU systems.

To present the proposed scheme of selective direct operation, $V_{\text{MPP}}$ measurement in Figure 3.5 is approximated in Figure 4.3 as $V_{\text{EH}}$ which varies depending on ambient light. In the conventional two-converter scheme, the battery would only be charged when $V_{\text{EH}}$ exceeds $V_{\text{min-DCDC}}$ and drained by the CPU system during software execution. In

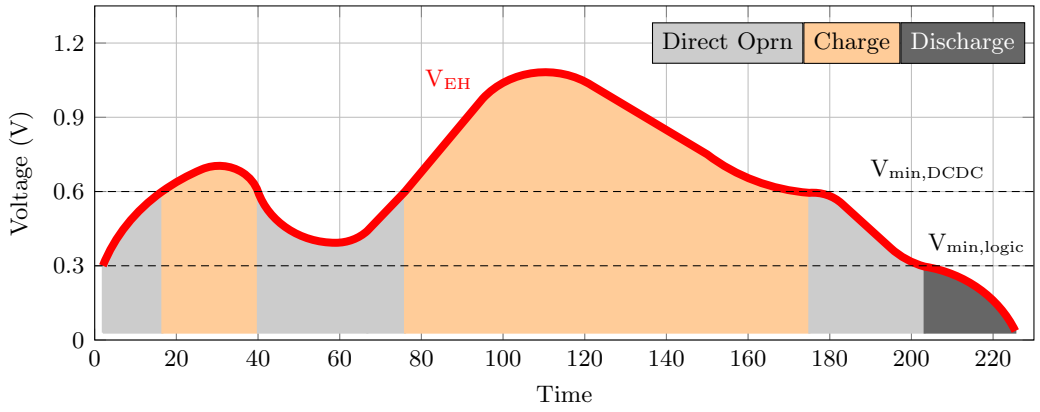


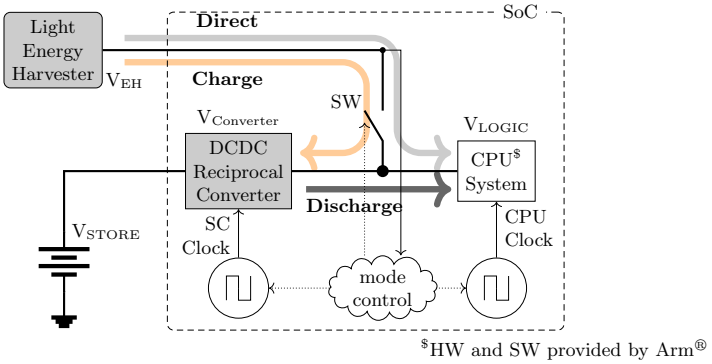

FIGURE 4.3: Conceptual representation of energy harvesting through converter-less operation of CPU system.

contrast, the proposed method allows the battery discharge to be limited to the region $V_{EH}<V_{min\text{-}LOGIC}$. Such a scheme is made possible by allowing the CPU system to operate directly without a series converter during phases in which $V_{min\text{-}LOGIC}<V_{EH}<V_{min\text{-}DCDC}$. Thus, three operational modes are possible - 1 and 2 here may overlap in operation:

1. **Charging:** $V_{EH} > V_{min\text{-}DCDC}$.
2. **Direct Operation:** $V_{min\text{-}LOGIC} < V_{EH} < V_{min\text{-}DCDC}$.
3. **Discharging:** $V_{EH} < V_{min\text{-}LOGIC}$.

The converter and CPU system interface used in implementing the three modes is illustrated in Figure 4.4. Note that the interface between the harvester, load and the storage device is through a single converter which is enabled only during the charging and discharging phases. The use of this single reciprocal converter eliminates the losses associated with a two-stage conversion process. The energy paths during the three modes of operation are highlighted. The switching frequency for the reciprocal converter (switched capacitor clock) allows MPPT during the charging phase. During direct operation the CPU clock is varied in such a manner that the rail impedance presented by the CPU matches the harvester output impedance.

The measured results presented in Section 4.5.5 show that the CPU impedance varies between 2 kΩ and 200 kΩ during direct operation. During the discharge phase the switched capacitor clock targets maximum conversion efficiency. Thus MPPT, minimum energy during direct and charge modes and maximum efficiency or minimum energy point tracking during discharge may be achieved. The frequency requirements for the switched capacitor converter ($F_{SCC}$) and the CPU system ($F_{CPU}$) during charging, direct operation and discharging modes is summarized in equations (4.3), (4.4) and (4.5) respectively.




| $V_{EH}$ | MODE | SC Clock | CPU Clock | SW |
|---|---|---|---|---|
| $V_{EH} > V_{Converter}$ | Charge | MPPT | MPPT | Close |
| $V_{LOGIC} < V_{EH} < V_{Converter}$ | Direct | 0 | MPPT | Close |
| $V_{EH} < V_{LOGIC}$ | Discharge | MinE | MinE | Open |

FIGURE 4.4: Functional diagram based on the conceptual diagram in Figure 4.3 highlighting the proposed scheme to avoid two-stage conversion and enable selective direct operation.

$$F_{SCC}, F_{CPU} = F_{MPPT} \tag{4.3}$$

$$F_{SCC} = 0, F_{CPU} = min(F_{MPPT}, F_{CPUmax}) \tag{4.4}$$

$$F_{SCC} = F_{EFFmax}, F_{CPU} = F_{MinE} \tag{4.5}$$

Here $F_{CPUmax}$ is the maximum frequency of the CPU (based on logic path timing) at a given voltage beyond which CPU timing violations lead to software execution failures. To implement a system with these modes, reciprocal converters with high bidirectional conversion efficiencies are needed, along with a low-power programmable clock generator for MPPT. Because the scheme relies on $V_{EH}$ for determining the mode of operation, ultra-low power comparators are also necessary. The next section details the design of these circuits.

## 4.4 System design and modular results

The energy harvesting sensor system was implemented in 65 nm along with the minimum energy CPU system. As indicated in Figure 4.5, apart from the 0.88 $cm^2$ PV cell, the 0.68 × 0.23 cm ($\phi$ × h) 6 mAh battery and decoupling capacitors, all other features required to implement the proposed scheme are included in the SoC. The minimum energy CPU system includes an internal clock generator which can be tuned by the control logic to match the CPU $F_{CPUmax}$ or $F_{MPPT}$ during direct operation. Under optimal settings the CPU clock generator tracks the PV output for MPPT achieving a near 99% tracking efficiency, similar to recent works [106]. The reciprocal converter

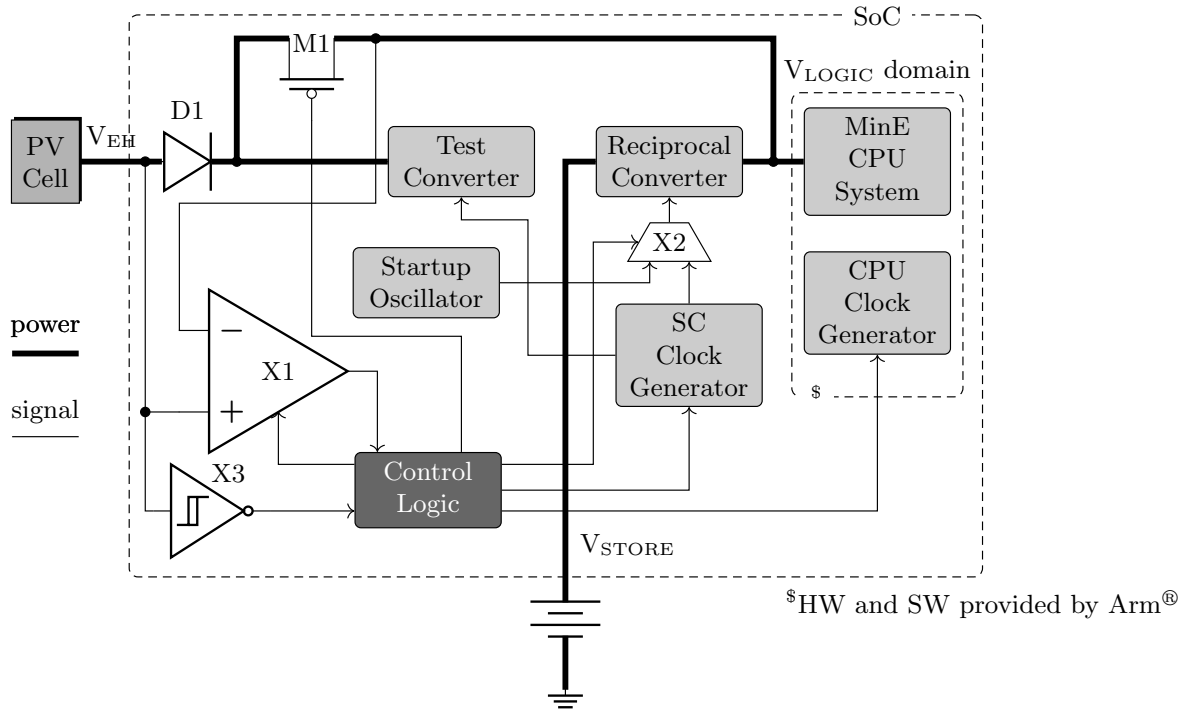



FIGURE 4.5: Energy harvesting sensor SoC implementation based on functional block diagram shown in Figure 4.4 with proposed reciprocal conversion and minimum energy CPU system. Power lines are shown in thick and signal lines are thin. Note also that the MinE CPU, its clock generator and the software [103] have been provided by Arm®

.

allows the battery to be recharged from harvested energy and also battery powers the CPU system when harvested energy is insufficient. It should be noted that although the CPU system is at its minimum energy point of 0.39 V, the minimum energy point for CPU + integrated voltage regulator is at ≈0.52 V. This shift in minimum energy point is due to converter overheads [73] which better matches with the converter output when the conversion ratio is 2.

A wide-range programmable clock generator is used for the switched capacitor converter which can be programmed for MPPT while harvesting or to achieving maximum efficiency when in discharge mode. The multiplexer, X2, allows the reciprocal converter to be clocked from the programmable clock generator or a start-up oscillator to enable cold-start. Cold-start is a critical feature of energy harvesting systems because it allows the system to boot-up with zero initial energy. The Schmitt inverter, X3, allows coarse low-voltage detection to start the control algorithm. Alternatively, a discrete solution [107] can be used which has a programmable detection down to (10 mV). When sufficient harvester output is available, the control logic uses a clocked comparator, X1, for MPPT while using the fractional $V_{OC}$ method [48].

If the harvester output is insufficient for either direct operation or charging, the PV cell is disconnected from the internal regulated node ($V_{LOGIC}$) using a MOS switch, M1. For the control algorithm to be able to sense the actual $V_{EH}$ in isolation from $V_{LOGIC}$

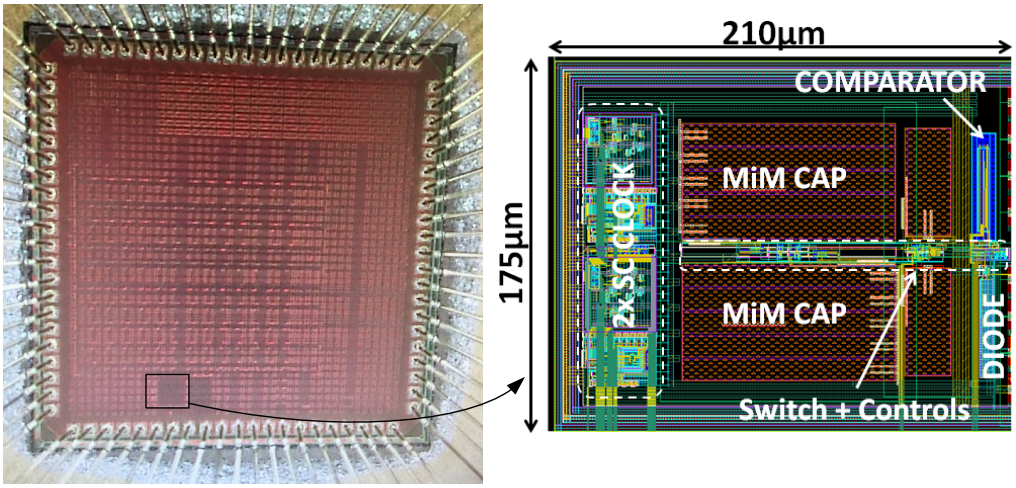



FIGURE 4.6: Die photo and annotated layout showing converter fly-caps, programmable clock generator, low power comparator and protection diode. .

when the switch is closed, the two nodes require isolation. A programmable diode D1, built using native (also known as zero $V_T$) devices provides this isolation and prevents a reverse current from flowing into the PV cell under low-light conditions.

The die photo is shown in Figure 4.6 and illustrates annotated converter fly-caps, programmable clock generator, low power comparator and protection diode. This macro was implemented in an area of 210 × 175 µm.

### 4.4.1 Reciprocal converter

switched capacitor converters can be modeled as two-port reciprocal networks assuming ideal switches [66]. Many techniques can be used to achieve near-ideal behavior in switches. The transmission gate-like switch implementation is a popular approach [105]. Other methods use gate over-drive and well-biasing. Switches designed using transmission gates ease the design of drivers and nonoverlapping clock generators, but they increase switching losses because they use up to 2× more gate capacitance per driver. Using well-biasing requires triple-well processes or large N-well isolation which costs area. Gate over-drive is easy to implement provided the over-drive voltage can be generated without significant overheads (charge-pumps and so forth).

This design relies on the battery voltage being 1.5-2× that of both the harvester voltage and the converter output voltage to ensure that the switches turn off reliably during normal operation. This super cut-off (switching off a device using higher than on working drain-source voltages) technique has the advantage of allowing low $V_{TH}$ devices to be used which have low on-resistance. Alternative techniques like use of high $V_{TH}$ or thick gate-oxide devices would compromise on resistance for reliable off state.

The converter schematic is shown in Figure 4.7a. Two phase-interleaved converters work

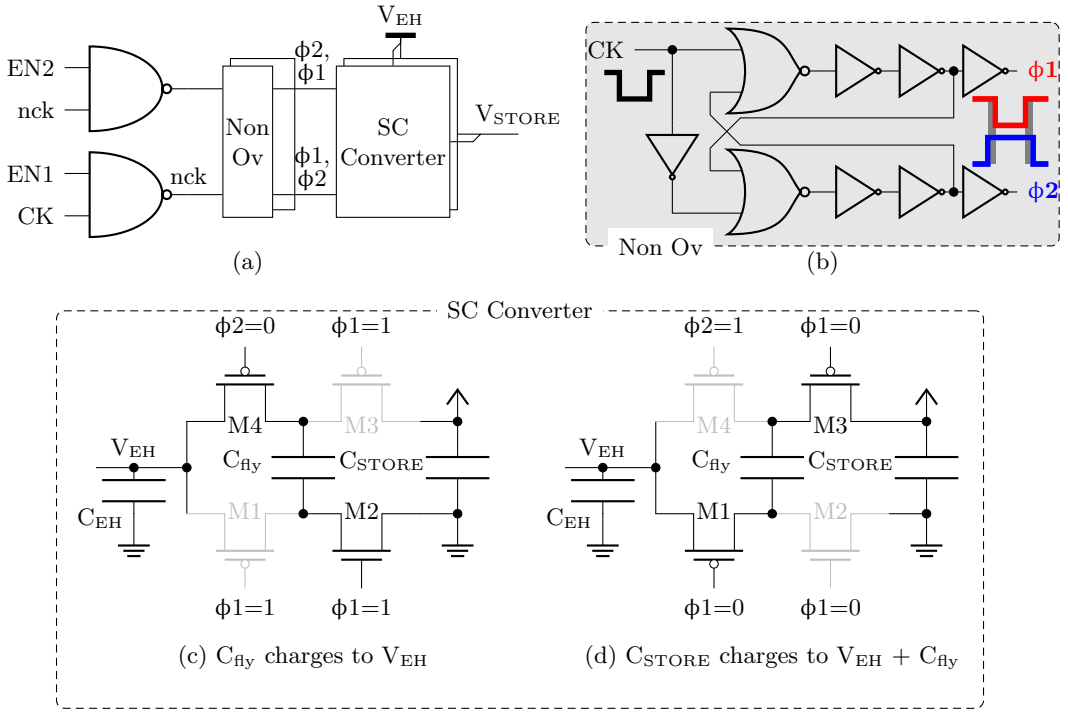



FIGURE 4.7: Proposed reciprocal switched capacitor converter scheme with (a) showing a high-level functional scheme, (b) showing the implementation details of the non-overlapping clock generator with the dead zone between ϕ1 and ϕ2 highlighted in dark gray (between the corresponding red and blue clock pulse) and (c) and (d) showing the implementation and working of the SC converter for the two phases of the clock (ϕ1=1 and ϕ2=0) and ϕ1=0 and ϕ2=1).

on complementary phases of the clock. The implementation scheme of the switched capacitor converter is shown in Figure 4.7c and d. When ϕ1=1 and ϕ2=0, $C_{fly}$ charges from $V_{EH}$ and in the next phase when ϕ1=0 and ϕ2=1, $C_{fly}$ appears in series with from $V_{EH}$ transferring charge at twice $V_{EH}$ to $C_{STORE}$. The choice of sufficiently wide PMOS switches for M3 and M4 allow reliable turn-on even if the battery voltage is lower than the harvester voltage, meaning that M3 and M4 have significant off-leakage, especially during a cold-start. However, this scenario resolves quickly when sufficient charge is transferred from the harvester to the storage device.

A single non-overlapping clock generator is sufficient since shoot-through currents are only possible through M3 and M4. M1 and M2 are never continuously on simultaneously as they are complementary devices and gated by the same phase. Note the waveforms for clock, ϕ1 and ϕ2 are inset in Figure 4.7b with the gray overlapping area showing the dead-zone when both M4 and M3 remain off ensuring a break-before-make switching.

### 4.4.2 Low-power comparators

High-power analog assist components degrade the benefits of minimum energy CPU systems. Designing power-matched mixed-signal blocks, high-efficiency converters and

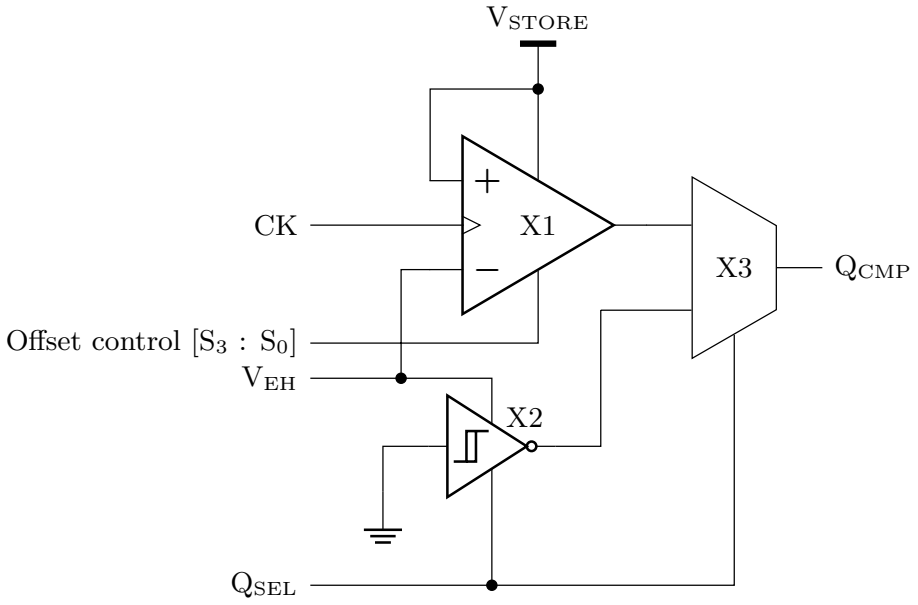



FIGURE 4.8: Proposed low power comparator scheme with coarse and fine comparators

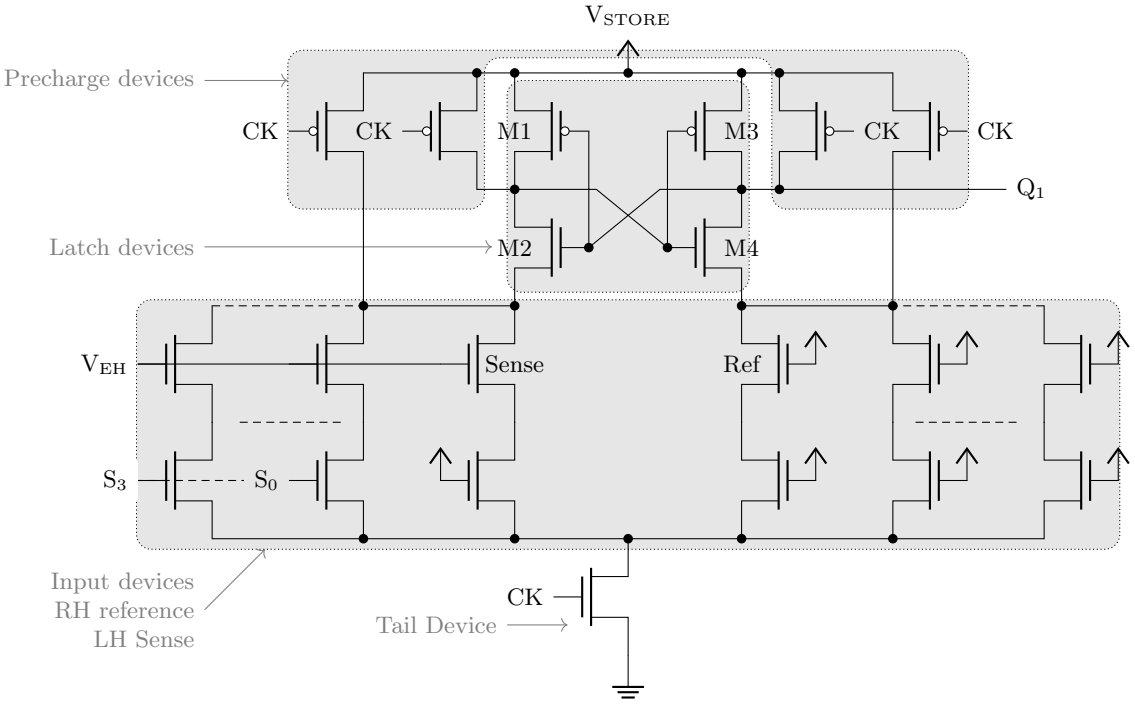



FIGURE 4.9: Programmable offset clocked comparator in Figure 4.8 using supply as reference.

low-power CPU systems are essential. Conventional comparator designs suffer from significant quiescent current making system-level energy neutrality a difficult objective. In this work, as illustrated in Figure 4.8a, two comparators are employed to match available power better: 1) a low-power coarse Schmitt comparator and 2) a programmable clocked comparator.

Clocked comparators are typically used for low quiescent power comparison. However, most designs [69], [46] employ reference voltage generators or resistive dividers with significant continuous power. To avoid this power overhead, reference generators are avoided in this design, and instead, the inherent offset in the comparator is tuned to achieve variable trip points (Figure 4.9). When clock is low, the internal nodes of the latch are precharged high using the precharge devices. Note that that tail device remains off. As clock rises and the precharge devices turn off, the tail device and input devices form a differential input stage which cause the latch devices' output to swing high or low depending on the bias at the inputs. The effective bias is the sum effect of the input voltages and the setting on the offset control transistors.

The input and offset control transistors (devices with gate connected to $S_0$ - $S_3$) are matched using common-centroid layout techniques with large channel lengths and widths, thus minimizing on-chip variation and helping with linearity. The reference input for the comparator is derived from $V_{STORE}$ and the sense input is connected to $V_{EH}$. The input devices on the sense side are sized in a binary fashion to provide 15 linearly increasing trip points for corresponding settings of S[3:0].

Figure 4.10 shows the measured comparator trip points from multiple dies. The observed linearity of a trip point vs. the offset control bit setting ($S_0$ to $S_3$ in Figure 4.9) is sufficient to sense $V_{EH}$ reliably at 50 mV steps. The control logic varies the offset control bits ($S_0$ to $S_3$ incrementally until the output of the comparator is at asserted to logic 1. When asserted, the offset-control value ($S_0$ to $S_3$ represents $V_{EH}$.

For $V_{EH}$ below 400 mV, this comparator is disabled and the Schmitt inverter, with a fixed trip point of 350 mV, is used for comparison. The Schmitt inverter has negligible quiescent power but uses a device $V_T$-based threshold to enable a coarse comparison at low voltages (Figure 4.8c). Further, the Schmitt inverter is powered from the harvester output; hence no stored energy is expended for this comparison. To ensure that the control logic can read the low-voltage output of the Schmitt, a wide-range level-shifter [108] is used. The level-shifter is disabled when the control logic is not sampling the Schmitt inverter output.

### 4.4.3 Programmable switched capacitor clock generators

This energy harvesting switching converter was developed over two iterations with the first revision using off-chip clock sources to understand the required range of frequencies under dynamically varying light conditions. Corresponding measurements revealed the need for a 10 kHz $\sim$ 30 MHz range for the switched capacitor converter to ensure MPPT. It is imperative that the overheads of clock sources should be minimal. Low-cost programmable clock generators can be designed using ring oscillators with dividers. However, for low frequencies the power expended in the initial stages of the divider

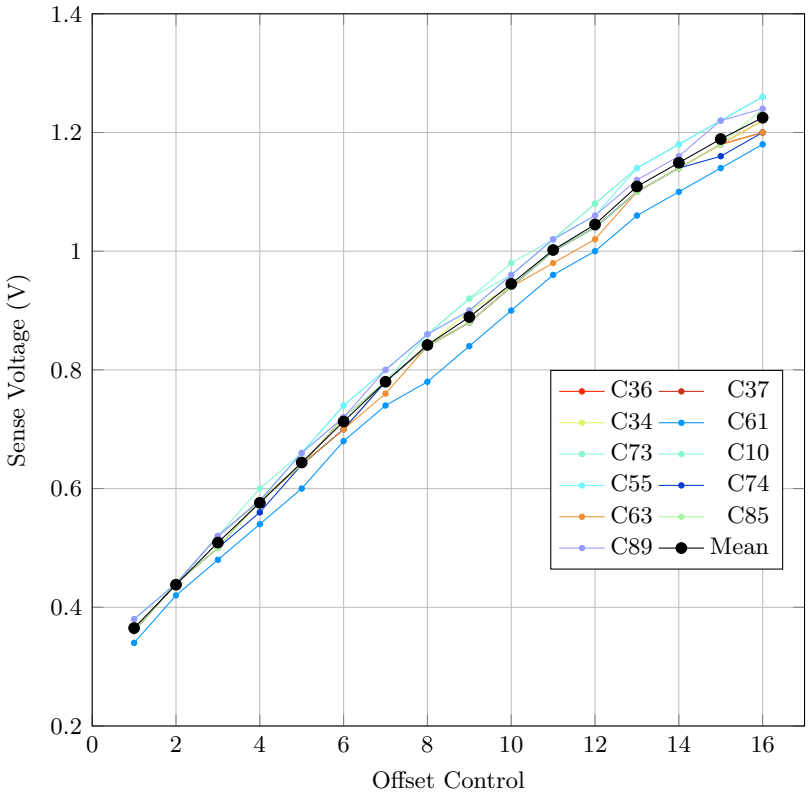



FIGURE 4.10: Measured comparator trip-points vs offset-control for multiple chips.

clocked from a fast-ring oscillator would defeat the objective of system energy-neutrality. Slow-ring oscillators on the other hand would require very long chains, costing area.

An alternate method is to use voltage controlled oscillators (VCOs) or current starved oscillators. Both VCOs and current starved oscillators, however, suffer from excessive quiescent current in the error amplifier. The proposed scheme exploits three key ideas as illustrated in Figure 4.11 :

1. Vary the voltage to the delay chain using VSEL[8:0]
2. Vary the quiescent power of the error amplifier in the local voltage regulator using PSEL[1:0]
3. Vary the length of delay chain with TAPSEL[17:0] for the desired frequency using table based look-up.

This scheme allows dynamic power-bandwidth tuning to minimize quiescent power in the error amplifier. Further dynamic power in the delay chain is reduced to obtain an integrated programmable clock generator operating at sub-nW/kHz over a wide range of frequencies.

Figure 4.11 shows the schematic of the programmable clock generator using 18 bits for tap selection (TAPSEL[17:0]) and 9 bits for VCO voltage selection (VSEL[8:0]). The

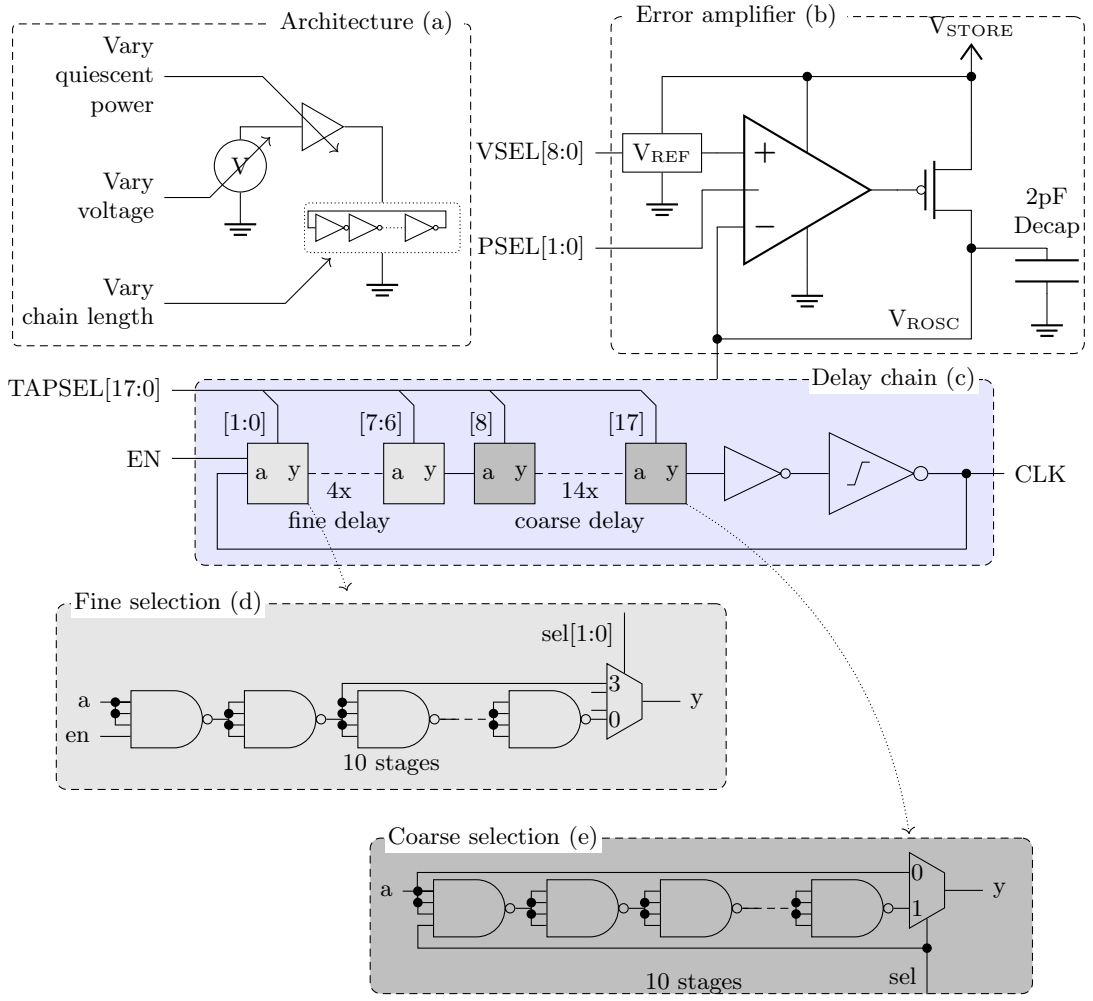



FIGURE 4.11: Proposed scheme for low-power programmable clock source with (a) showing the architecture with the three key design variables for controlling frequency and minimizing power. (b) showing implementation details of the local voltage regulator, (c) showing the arrangement of delay elements and level shifter, (d) showing multiplexer and nand gates for fine delay adjustment and (e) showing the coarse delay adjustment scheme.

quiescent power is controlled using PSEL[1:0] bits. These bits are exercised in a manner that allows the error amplifier to have a high bandwidth for deterministic settling of the oscillator, each time settings are changed. Once settled, the ring oscillator presents a relatively static load making the high bandwidth redundant and allowing the quiescent power to be reduced gradually using PSEL bits.

The delay chain is designed using 14 coarse and four fine delay stages with each stage using 10, four-input NAND gates. The stack-effect in the four-input NAND limits the dynamic short circuit current, thus reducing power. Further, the tap selection multiplexers gate the edge from propagating needlessly when a specific stage is excluded from the ring. Figure 4.12 shows the measured frequency range (using only VSEL and coarse selection bits) vs nW/kHz. The measured energy of <0.65 nW/kHz ensures low power overheads due to programmable clock generator. The lowest energy point of 0.42

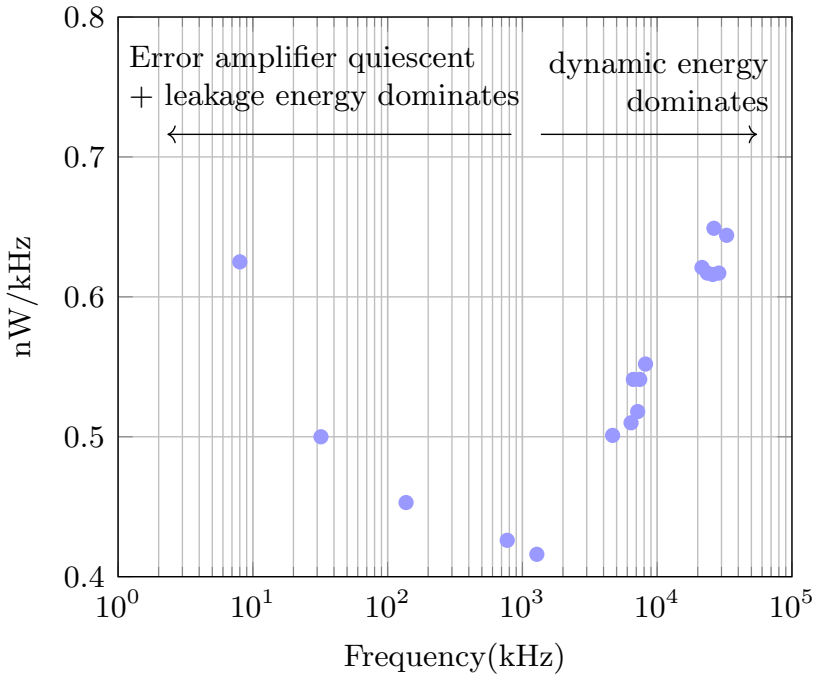



FIGURE 4.12: Measured nW/kHz for the programmable switched capacitor clock generator.

nW/kHz is observed at 1.2 kHz. Below this optimal frequency point the leakage of the programmable clock generator delay chain and the quiescent current of the error amplifier (Figure 4.11b) dominates the total power consumption. The dynamic energy of the delay chain dominates the total power for frequencies greater than 1.2 kHz.

## 4.5 Measured system results

This section presents the measurement setup and results obtained for converter efficiency, dynamic tracking and overall system performance.

### 4.5.1 Measurement setup

The measurement setup is shown in Figure 4.13. The light levels are read out using a commercial lux meter (Di-Log® DL7030 [109]) and the ambient temperature from an integrated temperature sensor (Microchip® MCP9700 [110]). The PV cells were placed in such a way that there is little/no direct light incident. The internal surface of the plastic enclosure was covered with reflective foil to ensure uniform diffusion. The light intensity is controlled by regulating the current through 4 light emitting diodes (LEDs) manufactured by Cree® (PN: CLN6A-MKW-CH0K0133 [111]). The current source meter was used to control the LED current. The exact positions of the LEDs and the sensors on the base of the enclosure had negligible impact on the test results.

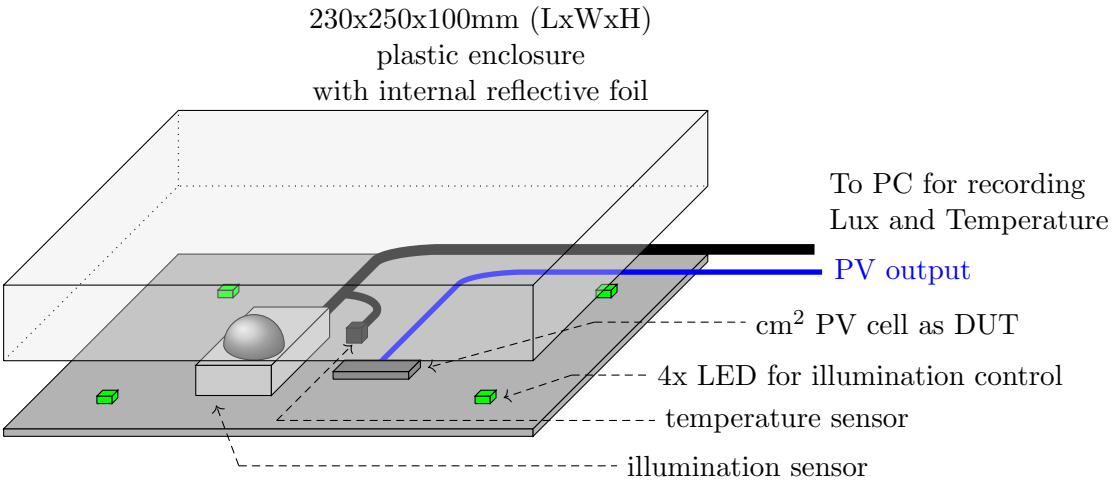



FIGURE 4.13: Measurement setup for controlled illumination tests

This LED was chosen so that majority of radiant power spectrum matches that of indoor fluorescent lighting (600nm) [112] with the exception of a small sideband at 450nm [111]. Four $cm^2$ area PV cells were used for measurements, three commercial off-the-shelf cells from IXYS® [42] and one high efficiency prototype cell from SHARP®. The converter was co-designed to better match the characteristics of the SHARP® prototype cell. Once setup, all the tests may be automated using a PC running python test scripts.

### 4.5.2 Reciprocal converter efficiency

Figure 4.14 shows the conversion efficiency of the switched capacitor converter, described in Section 4.4.1 (Figure 4.7), for both the charging and discharging modes. The charging efficiency was measured with a 6 mAh, 1.2 V NiMH battery while discharge efficiency measurement used a variable resistive load. The switched capacitor converter frequency was tuned to maximize efficiency. The measurements were recorded for both low (0 °C) and high temperatures (50 °C). Under normal operating conditions the peak conversion efficiency in discharge mode is 84% while the charging efficiency is 82%. Over the measured temperature range, the charging current remains relatively fixed because it depends largely on the battery's initial voltage. The discharge current, however, increases because the charge transfer capability of the converter increases at higher temperatures (due to better device conductance). However, increased switching losses degrade the discharge efficiency.

### 4.5.3 Dynamic tracking

To observe the dynamic tracking capability of the proposed system, the control flow was set up as shown in Figure 4.15. When no harvested energy is available, the control logic is in a slow (0.1Hz sampling frequency) loop monitoring the Schmitt output. If the

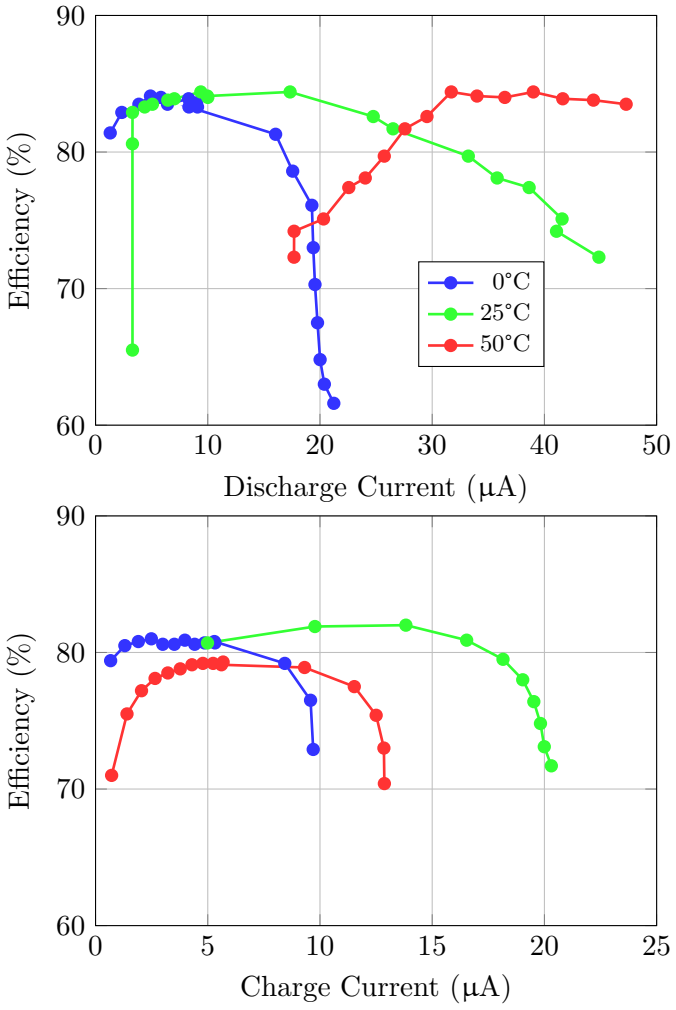



FIGURE 4.14: Measured reciprocal conversion efficiency across temperature. (a) Discharge and (b) Charge

Schmitt output indicates the availability of sufficient energy, the clocked comparator is turned on to evaluate the harvester $V_{OC}$. If the $V_{OC}$ exceeds 0.6 V, then the reciprocal converter is turned on, else direct operation mode is enabled. The comparator is read through a function call, as shown in Figure 4.16.

The MPPT was tested by using a variable-intensity light source with the wavelength of the test lamp chosen to closely match that of indoor fluorescent/LED lighting. Figure 4.17 shows the illumination variation over the 20-minute test period. The specific levels of intensity were arbitrarily chosen between 4 and 1118 lux. Multiple $cm^2$ area PV cells were tested (three from IXYS® viz. PV1-3 and one from SHARP® viz. PV4) and the harvested power level is shown in the lower pane of Figure 4.17. The minimum light levels at which the control loop decides to enable harvesting depends on the PV cells, as does the magnitude of harvested power. Note that the peak power harvested at about 20 lux is in excess of 1 µW, with 100 µW at 1000 lux. These values are in line with prior findings [19].

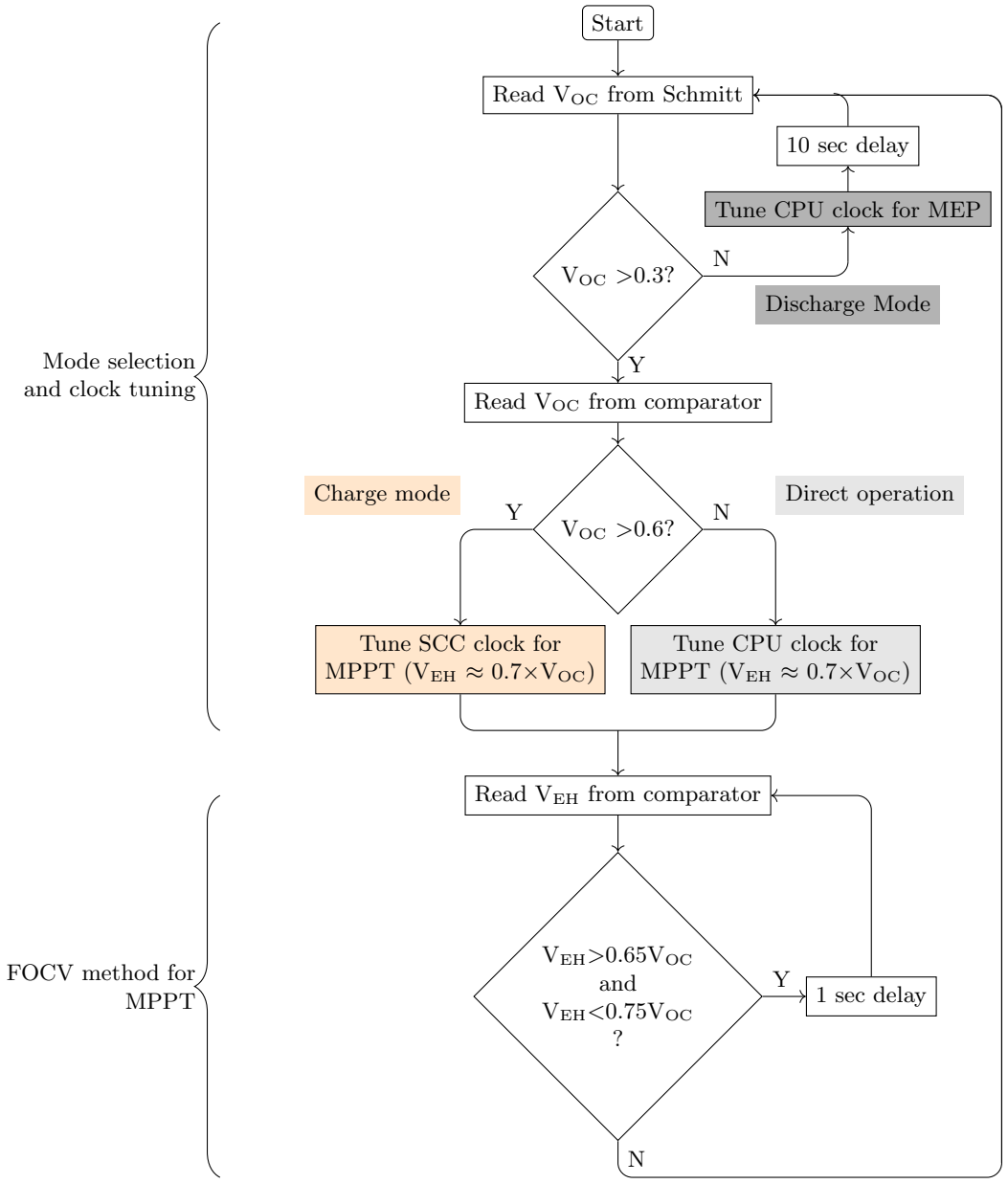



FIGURE 4.15: Control algorithm for MPPT implemented in software showing mode selection and corresponding clock tuning and fractional $V_{OC}$ (fraction=0.7). Colors correspond to modes in figures 4.3 and 4.4 which depict the principle of operation.

The control algorithm autonomously detects changes in light levels according to the flowchart in Figure 4.15 and tunes the converter frequency for MPPT. When the light level changes significantly, the loop restarts, resulting in zero harvested power for a short duration. However, for moderate-intensity changes, the loop self-adjusts as though a minor MPPT perturbation was observed. There are cases where the algorithm loses its tracking, potentially due to comparator jitter, despite ambient conditions remaining static, but it manages to reconverge to the optimal setting successfully (highlighted in Figure 4.17). Note that this loss of convergence does not affect execution of the ULPBench benchmark software as the loss of harvested energy is limited to at most 1

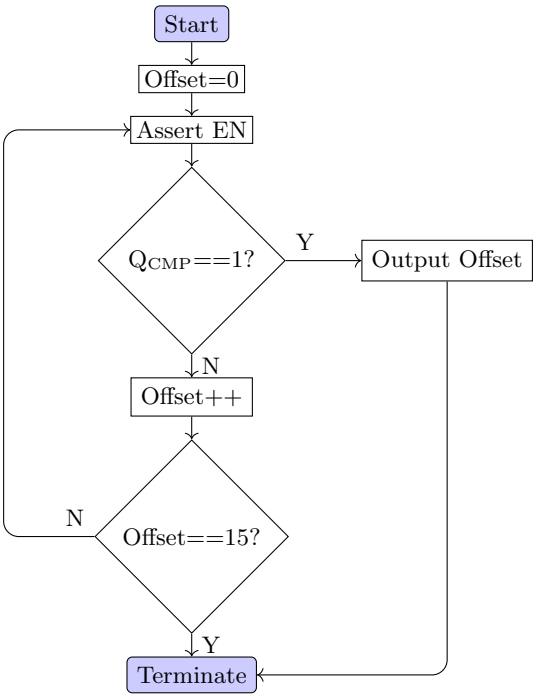



FIGURE 4.16: Control algorithm of function call for reading comparator.

second (loop speed), which is easily covered by the decoupling capacitance used in the system.

### 4.5.4 Cold-start

Cold-start is necessary for autonomous energy harvesting systems so that they can quickly boot-up from zero initial energy [113]. Converters and control logic should, therefore, be designed to operate using very low voltages. At low voltages, converter switches have poor conductance but will remain functional as will the fly-capacitors. The bigger challenge in cold-start is a reliable clock source. To overcome this problem, prior works [101] have used a self-oscillating switched capacitor converter, while others [105] have employed a short-chain current starved oscillator. For the latter case, the oscillator speed at higher voltages is limited by loading the internal nodes of the current starved oscillator with large (1.2 pF) capacitors costing dynamic power. This design uses a 96-stage ring oscillator with start-up voltages as low as 90 mV. The frequency may be nonoptimal [105] for the converter at higher input voltages but as long as the $V_{STORE}$ can charge to sufficient levels to allow the control logic to take over, a deterministic boot-up is possible.

Figure 4.18 (a) illustrates the start-up oscillator frequency vs illuminance. Oscillations start reliably at very low light levels ($<$ 100 lux) and for indoor light levels (250-500 lux), the frequency can be in excess of 16 MHz. To accelerate testing and to overcome the effects of battery self-recovery [114] (an increase in battery output voltage when idle),

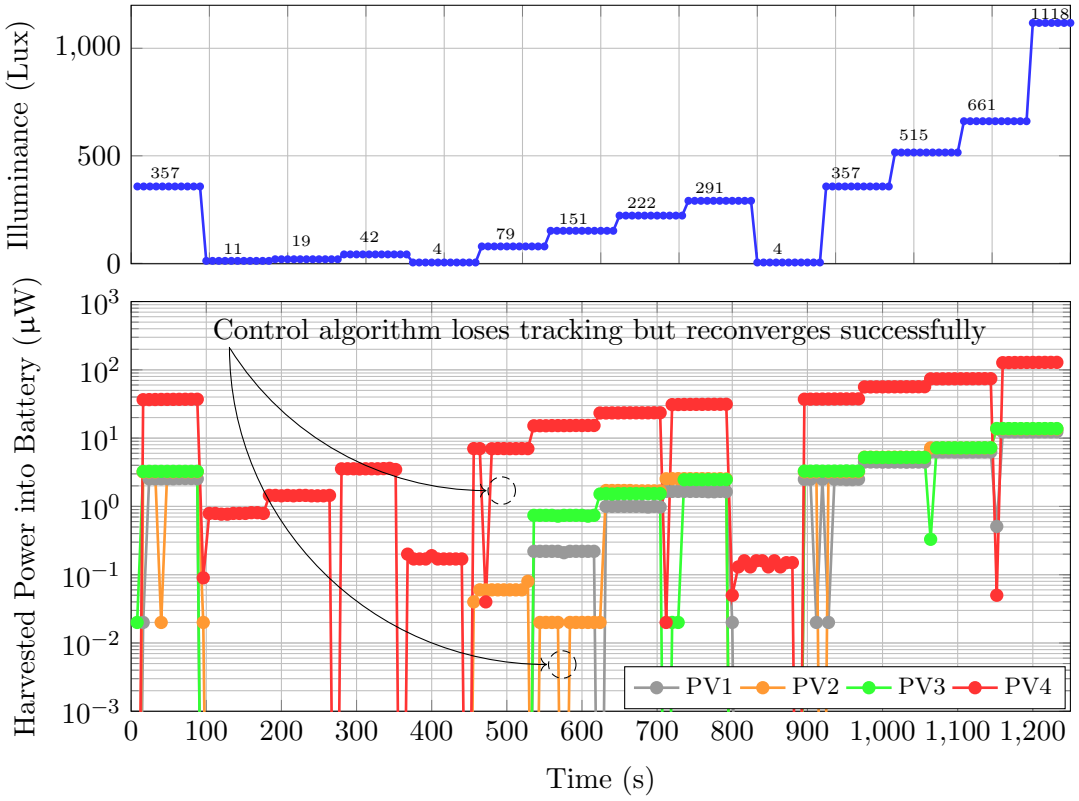



FIGURE 4.17: Runtime performance of control algorithm for varying light levels measured for multiple harvesters.

this measurement was carried out with a fully discharged 33 µF electrolytic capacitor. The results in Figure 4.18 (b) show the capacitor voltage at 260 lux. Once oscillations start, $V_{STORE}$ charges to 400 mV and then saturates because the start-up oscillator is too fast for the switched capacitor converter. Once $V_{STORE}$ reaches ≈450 mV, the control logic enables the programmable clock generator at 8 MHz switching frequency (from Figure 4.19) which is more optimal for the converter at ≈200 Lux than the 17 MHz of the start-up oscillator (from Figure 4.18). $V_{STORE}$ reaches ≈1.25 V before saturating.

### 4.5.5 System performance

Sensor workloads are heavily duty cycled, and the energy requirements depend on the active:sleep ratio. The results here are captured for both CPU sleep and 100% activity so that any real-world application would lie between these two extremes depending on the active:sleep ratio dictated by the software at run time. Figure 4.19 shows the overall system behavior and performance vs incident light levels. Discharge, direct operation and charge modes are shown.

When the CPU is active, at low light levels the reciprocal converter is disabled and only power drawn by the CPU system for computation (checksum) is shown. At higher light levels (>220 lux), the converter is enabled and some of the energy is used to charge the

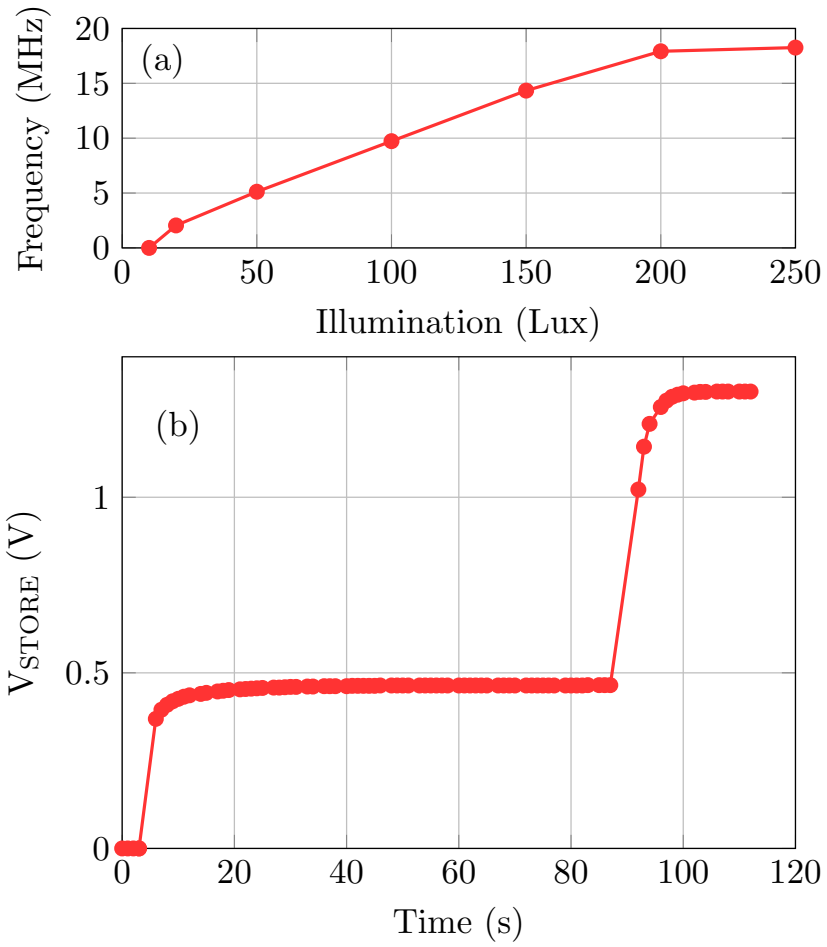



FIGURE 4.18: Measured cold-start results showing (a) start-up oscillator frequency vs. illuminance and (b) voltage build up on a capacitor at 260 lux with zero initial voltage vs. time.

battery. This is indicated by the 'Direct + Charge' trace. When no ambient energy is available, the CPU discharges the battery while tracking the minimum energy point. This is highlighted by the 'Discharge' trace. Note that for most indoor and dim light conditions the utilized power matches the ideal PV harvested power. This matching is evidence of CPU rail-impedance matching PV cell's output impedance during direct operation. The ideal output power trace was obtained using IV sweeps to identify the maximum power point.

The corresponding CPU system and converter frequencies are shown in Figure 4.19b. During direct CPU operation and during charge, the CPU and the converter frequency is set to track maximum power point. When the ambient energy is insufficient, CPU system discharges the battery and the hence the frequency is chosen for minimum energy point tracking.

Based on the measurements (Figure 3.5 and Figure 3.9b), 16% additional harvested energy is used by the CPU system without associated conversion losses when the voltage of

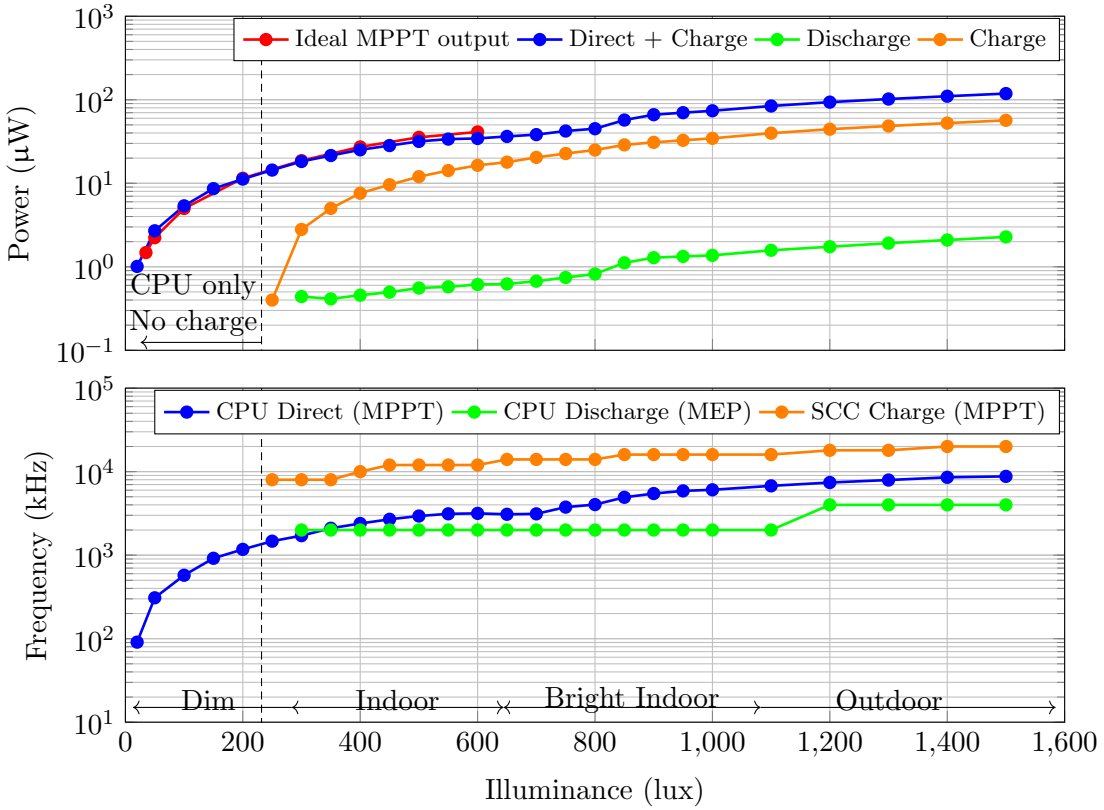



FIGURE 4.19: Measured results showing benefits of direct-operation with (a) Extracted power in different modes and (b) corresponding CPU-system and switched capacitor converter frequency vs. light levels.

harvested energy is below 0.6 V. Even with the converter retains peak efficiency of 88% in both directions over the entire load current range, this 16% additional energy translates to 30% additional computation for the same $cm^2$ harvester and ambient conditions.

### 4.5.6 Energy-neutrality and operation of EEMBC benchmark

Energy-neutral operation is possible if the system can survive the worst-case energy drain on the 6 mAh battery. The discharge mode of operation exhibits the worst case energy drain compared with both charging and direct operation modes. During the

TABLE 4.1: System power breakdown in discharge mode.

| | |
|---|---|
| CPU system | 0.6 µW |
| Converter overhead (worst-case 60%$\eta$) | 0.4 µW |
| PCG + mux at 14MHz for 5ms | 0.05 µW |
| Control logic (worst-case, same as CPU) | 0.6 µW |
| **Total** | **1.65 µW** |

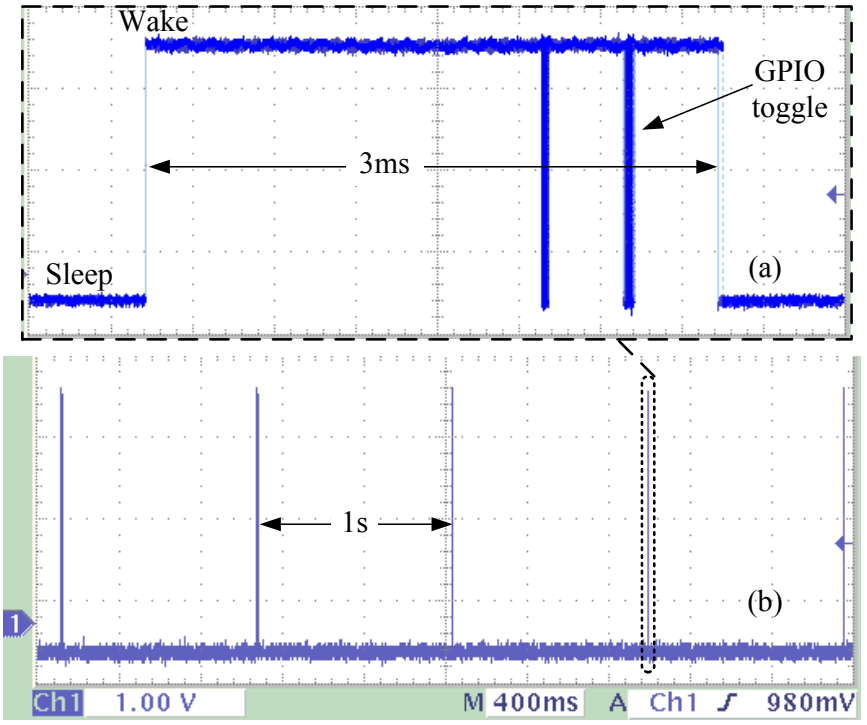



FIGURE 4.20: Oscilloscope waveform demonstrating ULPBench execution. (a) one cycle of execution with GPIOtoggles confirming successful completion of compute tasks and (b) periodic wake at 1 Hz with 997 ms sleep time.

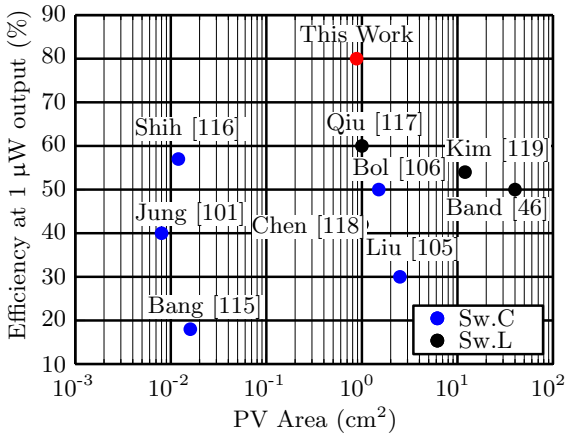



FIGURE 4.21: Comparison with prior-works for low-light efficiency (1 µW).

discharge mode of operation, the control logic, programmable clock generator and the multiplexer (Figure 4.5) add to the power overheads. The slow Schmitt comparator is used at 10 second intervals, but this is powered from the PV cell and hence does not drain the battery. Therefore, the available 24-hour power budget from the 6 mAh, 1.2 V battery is 144 µW (assuming a non-ideality factor of 0.48 [49] to account for voltage drop during battery discharge and other temperature-related effects). The power breakdown for the system in discharge mode (Table 4.1) indicates that the system can operate for

TABLE 4.2: Harvester performance compared with related works

| | **Node (nm)** | **$V_{in,pv}$ (V)** | **$V_{out}$ (V)** | **$P_{out}$ (μV)** | **$\eta_{peak}$ (%)** | **$\eta_{1\mu W}$ (%)** | **$A_{macro}$ ($mm^2$)** | **$A_{pv}$ ($cm^2$)** | **Reactance** |
|---|---|---|---|---|---|---|---|---|---|
| Liu [105] | 180 | 1.1-1.5 | 3.3 | <21 | 86.4 | <30 | 2.25 | 2.5 | Integrated C |
| Bol [106] | 65 | 0.95-2.7 | 3 | 5-10000 | 80 | - | 0.48 | 1.5 | Integrated C |
| Chen [118] | 65 | >0.8 | 1.3 | <80 | 72 | <42 | 0.245 | Ideal | External L |
| Bang [115] | 180 | 0.36-0.8 | 3.3 | 0.013-20 | 39.8 | <18 | 0.95 | 0.016 | Integrated C |
| Shih [116] | 350 | 1.8 | 1.4 | <10 | 58 | <57 | 0.42 | 0.012 | Integrated C |
| Kim [119] | 350 | 1.5-5 | 4 | 800 | 84 | <54 | 4.71 | 12 | External L |
| Qiu [117] | 250 | 0.5-2 | 5 | 5-1000 | 70 | <60 | 10.34 | Ideal | - |
| Band [46] | 350 | 0.15-0.75 | 1.8 | 2500 | 87 | - | 25 | 40 | External L |
| Jung [101] | 180 | 0.14-0.5 | 2.2-2.5 | 0.005-5 | 50 | 40 | 0.86 | 0.008 | Integrated C |
| **This work** | 65 | 0.58-1.5 | 1.2 | 0.02-100 | 84[$] | >80 | 0.037 | 0.88 | Integrated C |

[*] To support charge+compute. Minimum $V_{in,pv}$ depends on ambient light.
[$] Obtained from Figure 4.14

87 days before requiring a full recharge

Figure 4.20 shows the minimum energy CPU system running ULPBench benchmark code. The measurement was carried out at 160 lux. Figure 4.20b shows the system waking up at 1 s intervals and performing sensor activities. Figure 4.20a shows the active duration, during which the system performs initialization sequences and the general purpose input output (GPIO) toggles during code execution [103]. The results in Figure 4.17 demonstrate that the system can be energy-neutral when exposed to 50 lux continuously or 250 lux (indoor lighting) for as little as 2 hours per day. Figure 4.21 compares state-of-the-art energy harvesting converters vs harvester area and reveals an 80% conversion efficiency while harvesting into a 1.2 V battery from a $cm^2$ PV cell.

Alternatively, based on measurements in Chapter 3, wherein TEGs are characterized under realistic conditions - on a home radiator and on a hot water dispenser show that the system achieves energy neutrally when mounted on a radiator for 1 minute/day. The water dispenser has a differential of 4°C which provides sufficient power (40μW) but the voltage levels mean that conversion ratio and corresponding losses would limit energy utilization.

Table 4.2 presents a comparison with prior works. This work presents the smallest switched capacitor converter macro with integrated clock sources. The converter offers a peak conversion of 84%, can cold-start at 260 lux and exploits sub-threshold logic operation to enable selective direct operation and achieve energy-neutral operation.

## 4.6 Summary and future directions

Energy-neutrality is a challenging objective in energy harvesting systems especially when the harvester volume is constrained to a $cm^2$ form-factor. This chapter presented selective direct operation and reciprocal conversion techniques to reduce silicon area and maximize energy utilization. These techniques achieved energy-neutral operation from a compact PV cell under indoor lighting conditions. The proof of concept prototype chip was demonstrated for a system executing industry-standard sensor benchmark software demanding approximately 20 thousand instructions per second. The reciprocal converter presented has the highest bidirectional conversion efficiency. Circuit novelties were presented for low-power analog assist blocks (comparator and programmable clock generator) used to achieve an optimal system, performing autonomous MPPT. Successful cold-starting was demonstrated at low light levels. Future work is anticipated to demonstrate successful functionality while harvesting energy from other energy sources such as thermoelectric generators. The techniques presented in this work demonstrate potential solutions to key challenges in enabling energy-neutral sensing systems.

# Chapter 5

# State-Aware Voltage Monitor

Recent works have demonstrated CPU-system designs that can operate at supply voltages below transistor threshold voltages (sub-550 mV) [69] [72] [44] [73]. The low voltages allow the system to be operated at the lowest energy per cycle point on the voltage, frequency curve (operating point). This is also called minimum energy or MinE operation. At higher voltages the dynamic power loss increases system energy and at voltages below the MinE point, leakage energy increases resulting in a loss of system efficiency. MinE systems are therefore, in relative terms, most efficient and ideal for many emerging sensor applications [120] [121] [122]. Such systems can operate with severe energy constraints and have low activity rates with relatively lower performance requirements than what is offered by current technology nodes at nominal voltages. Many of these applications can also harvest energy from their environment, resulting relatively longer lifetimes [123] [124] of unattended operation compared to battery powered sensor systems.

As shown in the previous chapter, these systems also do not operate at a fixed voltage but instead jump between operating points with different voltages. The sensor workloads generally spend maximum time in the lowest power mode or sleep mode. Transitioning between these voltages must be fast and reliable. Rail monitors are employed to signal a safe condition after each transition and it is imperative that these rail monitors have minimal power overheads.

This chapter describes and demonstrates a 50 nW voltage monitor fabricated as part of the power management unit of a 65 nm MinE wireless sensor node. Ultra-low power operation is achieved by duty-cycling the comparators. Further, dynamic power-bandwidth balancing results in lower quiescent power without loss of response speed. Measured results show a 6 µs response time, providing a superior power-delay balance compared with prior works. The design, implementation and measured results along with the system implications of the design choices are presented. This work was peer-reviewed and published at the VLSI design conference [26].

## 5.1 DVS and rail monitoring in sensor nodes

A common objective of MinE CPU systems is to minimize leakage because wireless sensor nodes spend the majority of their time in sleep modes. Leakage energy also increases exponentially at low voltages, further degrading active energy. To minimize leakage, fine-grained power gating is used with example systems having as many as 14 power domains [73]. Integrated voltage regulators are another common feature of MinE CPU systems, and they are used to:

1. obtain the low voltages required for sub-threshold operation because battery voltages are 1.2 V or higher

2. reduce latency during sleep and active mode transitions.

As an example, highlighting the fast dynamic voltage and frequency scaling (DVFS) transitions applicable to wireless sensor nodes, an 80 µs sleep-active-sleep transition is demonstrated in [125] while running a standard benchmark [103] with the supply dropping to 0.3 V for sleep and ramping up to 0.8 V for active. This fast transition is workload (software) dependent and differs from the time taken to wake from sleep as reported in commercial microcontroller datasheets.

Fast, wide-range dynamic voltage scaling (DVS) is desirable in wireless sensor nodes to enable frequent entry into sleep modes and also to maximize sleep time. Enabling the clock in the CPU system upon wake-up requires careful consideration. An early enable can cause timing violations leading to a system failure, while a delayed enable defeats ultra-low power operation. Voltage monitors are thus required to guarantee safe regulator voltage ($V_{REG}$) levels before the clock is enabled. The proposed design demonstrates an ultra-low power voltage monitor scheme using:

1. Duty cycled comparators with dynamic hysteresis tuning.

2. Run-time power-bandwidth trade-off to reduce active power.

3. State-aware tuning to provide 6 µs response speed at 50 nW.

The next section describes the system with the voltage monitor highlighting key design metrics. The design and implementation of the proposed voltage monitoring scheme are explained in Section 5.3. Measured results are compared with prior-art in Section 5.4 and conclusions are drawn in Section 5.5.

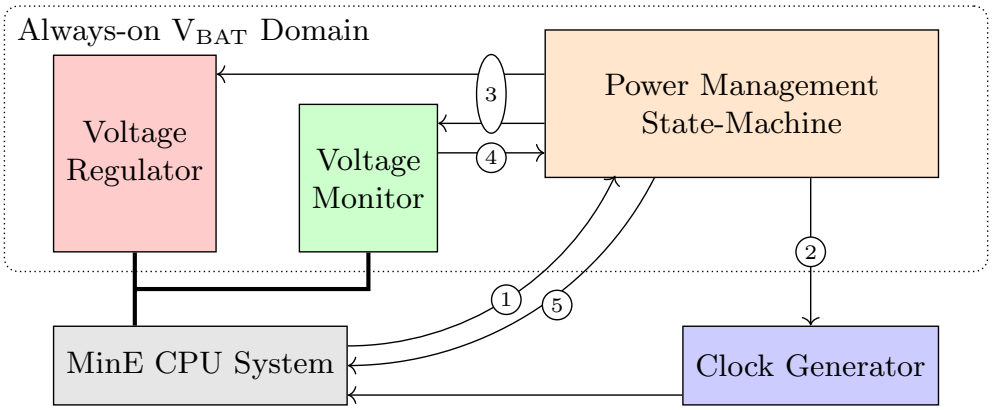



FIGURE 5.1: Voltage monitoring interface and system functional diagram.

## 5.2 System functional description

The power management unit-CPU system interface can be reliably timed by the power management unit clock if the interaction is limited to clock and/or power gating. With DVFS however, two factors introduce additional delays during mode changes.

1. Voltage settling : This is the time taken for the integrated voltage regulator to change its internal configurations and successfully charge the rail to the requested voltage.

2. PLL clock-locking This is the time taken for the PLL to change its internal dividers and charge pump settings and successfully generate a clock signal at the new frequency.

wireless sensor nodes using MinE CPU systems operate at frequencies not exceeding tens of MHz [126]. Clock settling is therefore, relatively deterministic. Voltage settling can have a greater degree of uncertainty because integrated switched capacitor converters in MinE systems can have higher output impedance compared to linear regulators. Figure 5.1 shows the power management unit interface with the CPU system and voltage monitor. Under ideal conditions, (Figure 5.2) the CPU system asserts a voltage change request (CHV) whenever a mode change is desired. Note the encircled numbers in Figure 5.1 represent the sequence of information flow in the interface and correspond to the timing diagrams in Figures 5.2 and 5.3 as well. The falling edge of CHV is captured by the power management unit state-machine and the CPU system clock is disabled (CKEN). The integrated voltage regulator setting is then changed to the requested value while de-asserting the ACK signal. Assuming the system rail voltage settles immediately, CKEN is asserted, followed by ACK. The CPU system resumes in the requested mode and CHV is de-asserted. No monitoring scheme is necessary.

In practice however, the transition time ($T_{CH}$) is much longer, because it is dominated by the voltage settling time ($T_{VS}$) and the time it takes for the monitor to detect an

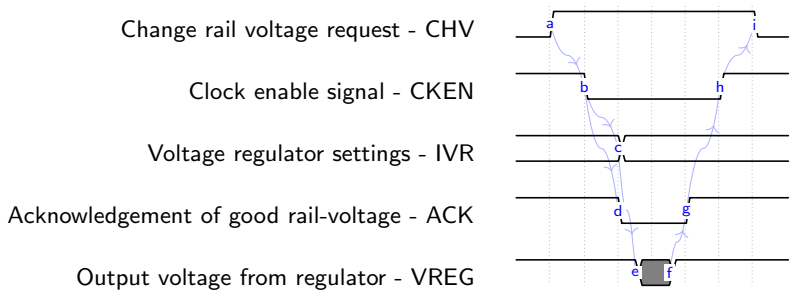



FIGURE 5.2: Ideal-case timing diagrams during dynamic voltage scaling for systems similar to Fig. 5.1.

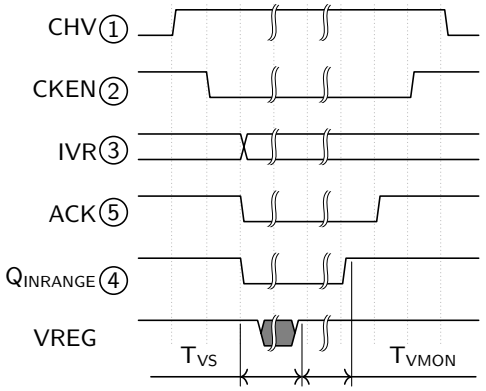



FIGURE 5.3: Mode transition waveforms in the presence of non-ideal regulator characteristics

in-range condition ($T_{VMON}$). This concept is illustrated in Figure 5.3. It is desirable to minimize $T_{VS}$ and $T_{VMON}$. $T_{VS}$ is affected by the load current which is sensitive to temperature, process and system workloads and integrated voltage regulator design characteristics such as output impedance and on/off-chip decoupling capacitance. The objective of this work is to minimize $T_{VMON}$.

Voltage monitors typically use comparators with factory-trimmed threshold voltages for detecting an unsafe rail voltage condition. Sensing slow-rising or non-monotonic rail voltages can cause oscillations as the rail voltage approaches the threshold voltage. This problem is overcome by using two comparators with slightly offset threshold voltages [127], [81]. This two-level monitoring adds hysteresis to the comparator but allows only for an unsafe low-voltage condition to be monitored. In MinE systems however, it is necessary to monitor independently for overvoltage conditions as well, because excess leakage can adversely affect MinE operation. In the conventional scheme, this would require four comparators making it an energy-expensive task.

Monitoring may be implemented with either continuous-time comparators [81] or clocked comparators [83]. Continuous-time comparators exhibit fast response speed but at the expense of higher quiescent power and having four comparators (for upper and lower thresholds) in the always-on power management unit domain can be particularly detrimental to MinE wireless sensor nodes. Clocked comparators have relatively lower quies-

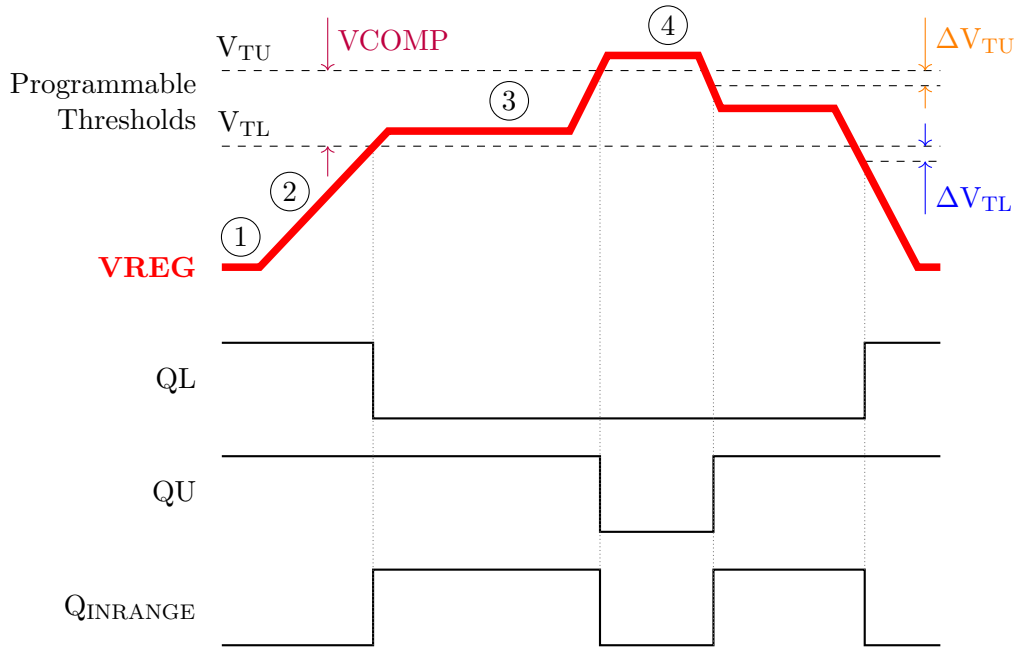



FIGURE 5.4: Conceptual voltage monitor response for varying $V_{REG}$

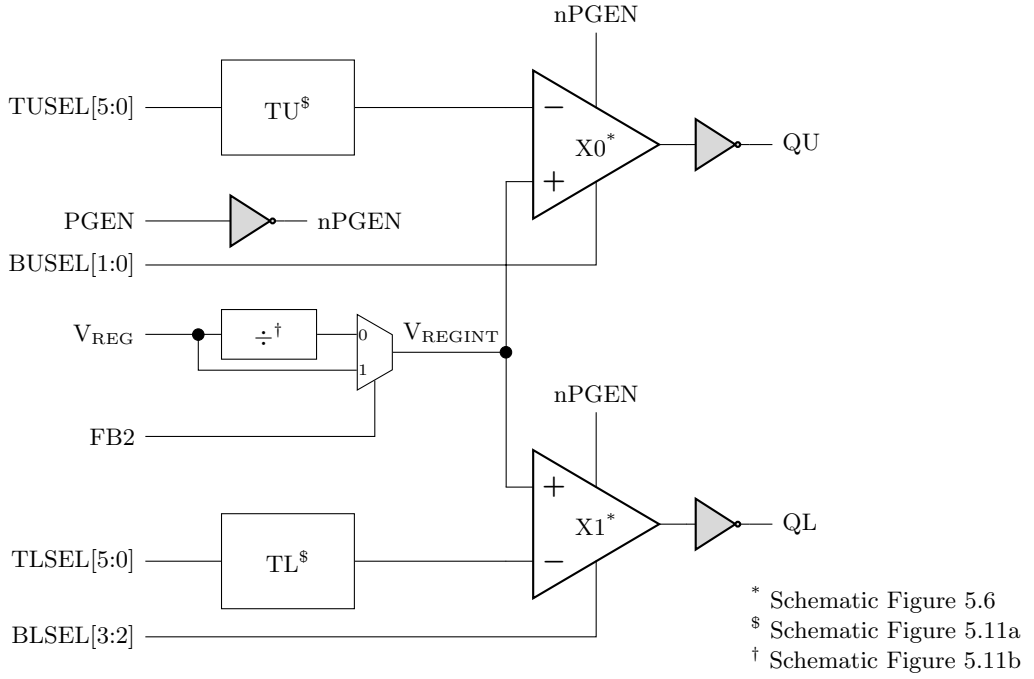
cent power but suffer from overheads of having to generate a dedicated clock. Leakage-based oscillators with thyristor-like gain stages have been used to generate comparator clock to reduce these overheads but at the expense of speed [44] [83] resulting in an undesirable increase in $T_{VMON}$.

## 5.3 Proposed design and implementation

Figure 5.4 shows ideal response of the voltage monitor for varying $V_{REG}$ levels. For the sake of simplicity $V_{REG}$ is shown to start at a voltage below $V_{TL}$. The output QU is a signal asserted to indicate that the rail voltage is within the upper threshold and QL being high indicates that the rail voltage is below the lower threshold. $Q_{INRANGE}$ is an XOR of QU and QL.

To start with QU and QL are both high and $Q_{INRANGE}$ is de-asserted indicating an out-of-range or unsafe condition for enabling CPU clock. In this state ① both comparators are off. As $V_{REG}$ starts rising (state ②), the first event is triggered when $V_{REG} > V_{TL}$. This event is signaled by the CMPL. Thus, the power management unit can power down the CMPU until this event occurs, thus reducing the voltage monitor's quiescent power. Once $V_{REG}$ is within the desired limits, state ③, both the CMPU and CMPL are turned on but in a low-bandwidth mode so as to satisfy µs order detection of an out-of-range condition. When $V_{REG}$ exceeds $V_{TU}$ in state ④, the CMPL is powered down as the trigger can be generated reliably by the CMPU when $V_{REG}$ drops below $V_{TU}$.

Thus, it is possible to exploit the power management unit 's awareness of current sys-




**Signal Definitions**

| | | | |
|---|---|---|---|
| TUSEL | : Upper threshold selection | BxSEL | : Bias selection |
| TLSEL | : Lower threshold selection | $V_{REG}$ | : Regulator output voltage to be monitored |
| PGEN | : Enable power | FB2 | : Enable divide-by-2 feedback |

FIGURE 5.5: Proposed voltage monitor scheme for the functional diagram shown in Figure 5.1.

tem to minimize the quiescent power in the voltage monitoring circuit. Figure 5.5 shows the schematic of the proposed voltage monitor which uses reference tuning to add hysteresis, allowing both upper and lower limits to be monitored using two comparators. The comparators and threshold voltage generators can be power gated using a power enable (PGEN) signal minimizing static power when the voltage monitor is power gated (system deep-sleep mode). In this mode, the integrated voltage regulator is off; hence the monitoring circuit can be powered down. The upper and lower comparison thresholds ($V_{TU}$ and $V_{TL}$) can be programmed using TUSEL and TLSEL, respectively. The tuneable range between $V_{TU}$ and $V_{TL}$ covers the entire DVS range of the MinE CPU system.

The key feature of this work is the bias current selection bits (BUSEL and BLSEL) for both upper and lower comparators (CMPU and CMPL). The bias selection bits are exercised in a manner so as to minimize the quiescent power of the voltage monitor without compromising monitoring speed.

Note that a state ③-② transition may be fatal to the system while a ③-④ transition is less critical. The CPU system remains functional in state ④ but potentially at a much higher energy cost. Therefore, in state ③, the CMPU's quiescent current is reduced further. The proposed scheme allows three bias current settings to be dialed into the

TABLE 5.1: Summary of comparator bias configurations

| State | BUSEL | BLSEL | $\Delta V_{TU}$ | $\Delta V_{TL}$ |
|---|---|---|---|---|
| **1** | 0 | 0 | - | - |
| **2** | 0 | 3 | - | $+\Delta V$ |
| **3** | 1 | 2 | $+\Delta V$ | $-\Delta V$ |
| **4** | 3 | 0 | $-\Delta V$ | - |

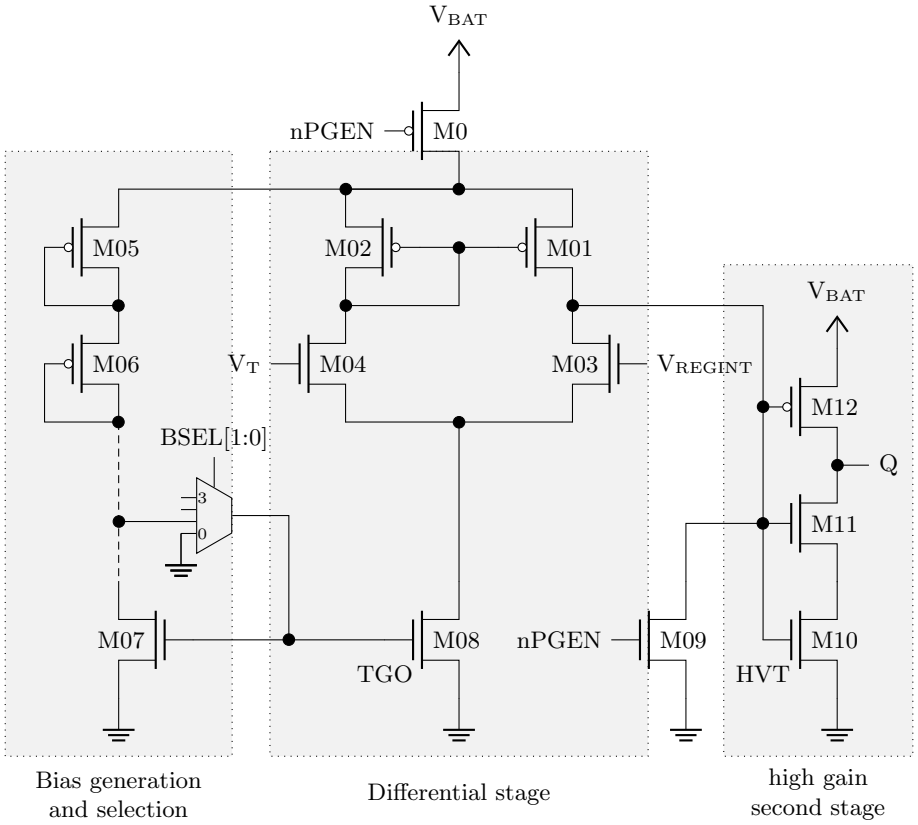



FIGURE 5.6: Comparator schematic

comparators. A BxSEL setting of '3' provides the fastest response at highest quiescent power and a setting of '1' provides the lowest power operation. Table 5.1 summarizes the bias configuration for each state as highlighted in Figure 5.4. Hysteresis may be added depending on the corresponding comparator output as described in [83]. The proposed design relies on TxSEL bits for this purpose. Thus state ②-③ transition is at $V_{TL}$ plus a small voltage ($\Delta V$) while a ③-② transition is at $V_{TL}$-$\Delta V$. Similarly, ③-④ occurs at $V_{TU}$+$\Delta V$ and ③-② at $V_{TU}$-$\Delta V$. This prevents any oscillations due to lack of hysteresis. These outcomes are summarized in Table 5.1 as $\Delta V_{TU}$ and $\Delta V_{TL}$.

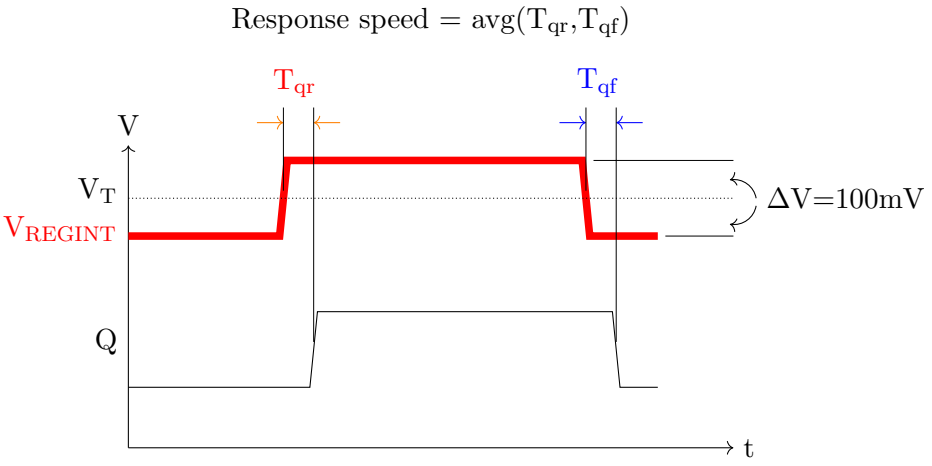



FIGURE 5.7: Comparator response speed simulation setup.

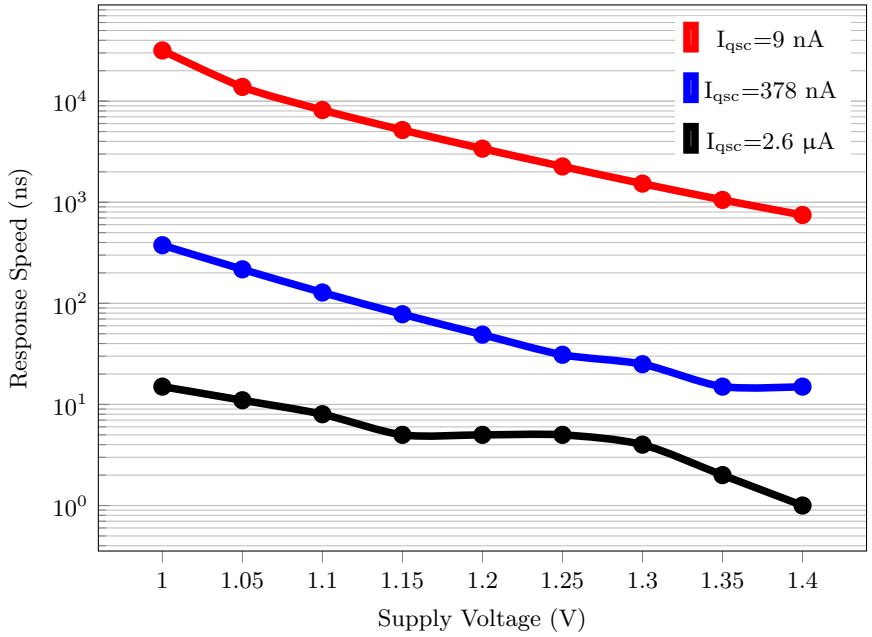



FIGURE 5.8: Comparator simulation results showing response speed vs supply voltage for different comparator tail current settings determined by BSEL bits to vary the bias setting. Bias setting 1 has the lowest tail current.

### 5.3.1 Comparator

Figure 5.6 shows the schematic of the comparator. The bias generation and selection stage is designed around a stack of six diode-connected, regular-$V_T$ transistors (M11-M17). The selection uses an analog multiplexer to choose a higher tap for higher comparator tail current. The bias selection bits effectively change the mirror ratio between M17 and M6 controlling the response speed of the comparator and its quiescent current ($I_{qsc}$). A bias setting of 1 sets the quiescent current to 9 nA while settings 2 and 3 can push it to 380 nA and 2.6 µA per comparator (at TT, 25°C and 1.2 V).

The comparator stage uses a tail current transistor built from a thick gate oxide device to

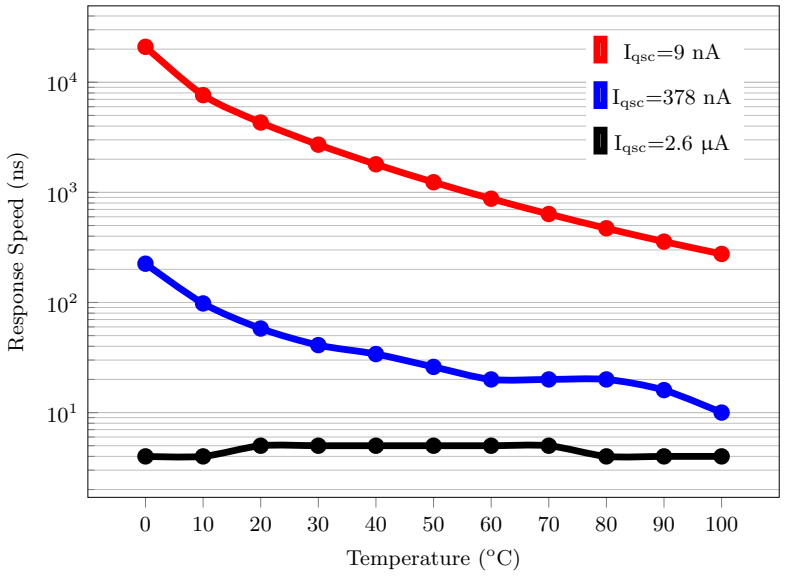



FIGURE 5.9: Comparator simulation results showing response speed vs temperature for different comparator tail current settings determined by BSEL bits to vary the bias setting. Bias setting 1 has the lowest tail current.

allow for better $V_{BIAS}$ control of the tail current. Comparators for sensing low voltages use PMOS input transistors to improve the gain in the input stage [83] which affects quiescent power. The input differential pair (M4, M5) in the proposed design use NMOS transistors and the lack of gain is compensated by using large low-$V_T$ devices, thus allowing input voltages as low as 0.2 V to be sensed reliably.

The second stage is driven from the output of the differential stage and is an inverter with stacked high-$V_T$ devices (M8-M10), which limits the short-circuit current and helps reduce power [81]. M1 and M7 allow the comparator to be power gated with the output, Q, forced high.

Figure 5.7 shows the simulation setup for measuring comparator response speed. The response speed is measured as the average delay for a correct transition on Q for $V_{REGINT}$ change from $V_T$ - 100 mV to $V_T$ + 100 mV [128]. Simulation results for supply voltage of 1.0- 1.4 V and temperature range of 0 - 100 °C are illustrated in Figure 5.8 and 5.9 respectively. At sufficiently large tail currents the comparator speed is less affected by temperature. Both speed and quiescent power increase exponentially with bias setting. Hence speed can be traded for power. Another consequence of reducing power is the increased sensitivity of the comparator speed to voltage and temperature. The simulation results (Figure 5.10) show a 20,000× increase in sensitivity with temperature and 2,000× increase with voltage. However, because the design relies on using low-bias current modes only when comparator response is less critical or is not needed, this increased sensitivity does not affect system active-sleep-active transitions.

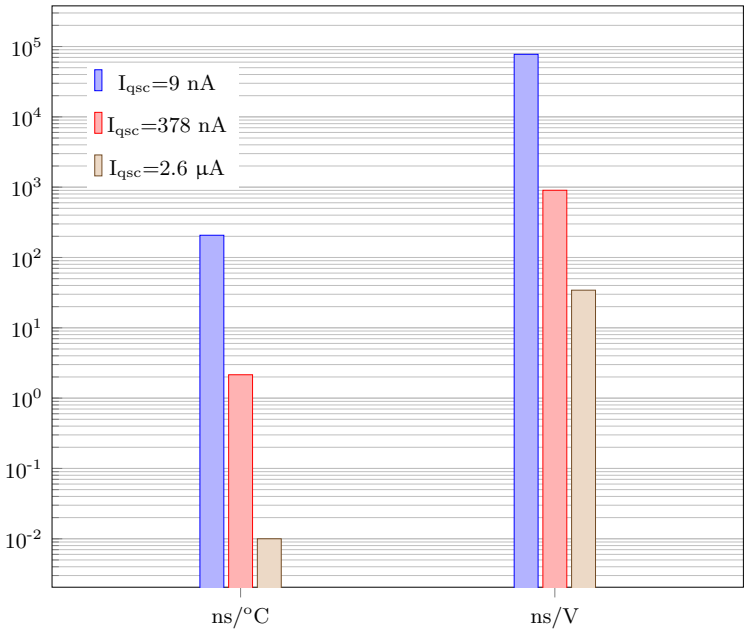



FIGURE 5.10: Comparator simulation results - sensitivity of response speed to supply voltage and temperature for different comparator tail current settings determined by BSEL bits to vary the bias setting. Bias setting 1 has the lowest tail current

TABLE 5.2: Simulated spread on $V_{COMP}$ and $\Delta V$

| **TxSEL** | | **1** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|---|---|
| $\mathbf{V_{COMP}}$ **(mV)** | µ | 54 | 48 | 44 | 42 | 41 | 42 |
| | $\sigma$ | 4 | 4 | 4 | 3 | 3 | 3 |
| $\mathbf{\Delta V_{TU}}$ **(mV)** | µ | 45 | 79 | 59 | 44 | 36 | 34 |
| | $\sigma$ | 4 | 6 | 5 | 4 | 4 | 3 |
| $\mathbf{\Delta V_{TL}}$ **(mV)** | µ | 50 | 89 | 68 | 51 | 40 | 33 |
| | $\sigma$ | 5 | 6 | 5 | 5 | 4 | 4 |

### 5.3.2 Threshold voltage generator and divider

The threshold voltage for the comparators is generated using stacked diode-connected transistors [129] as shown in Figure 5.11. Both $V_{TU}$ and $V_{TL}$ are obtained from the lower half of the stack to give identical behavior as the temperature varies. For a nominal supply voltage of 1.2 V, all transistors in the stack operate in the sub-threshold regime. PMOS devices are used in source-connected isolated N-wells to avoid body effects and ease layout. Each node in the divider stack is decoupled using 20 fF MOS capacitors to provide rejection of high-frequency supply ripple. Further, the on-resistance of the multiplexers and a 120 fF capacitance on the output node mitigates the noise on the reference node. This arrangement however is not immune to slow changes in supply voltage although the problem can be alleviated by using a native NMOS in the stack [129]

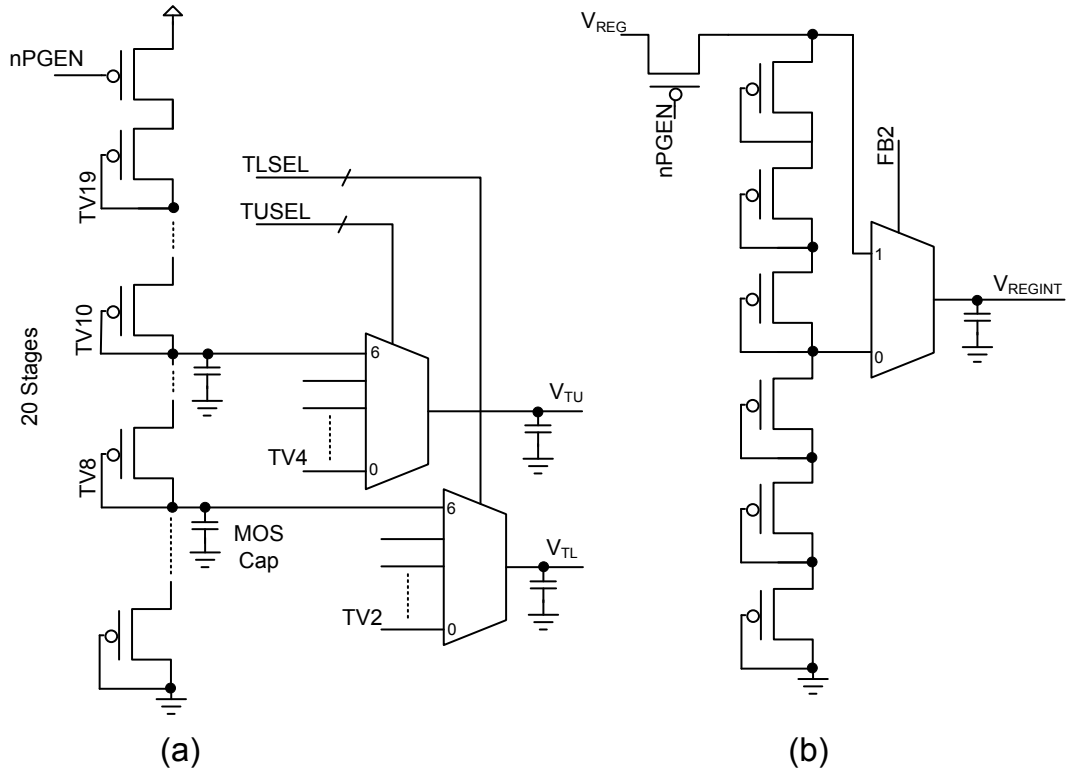



FIGURE 5.11: Schematic of (a) internal threshold voltage generator with stack of diode connected MOS and analog multiplexer (b) voltage divider using the same topology with mid-stack tap. Signal definitions are provided in Figure 5.5 which also provides functional overview.

The speed and accuracy of comparison depends on the comparator and the threshold voltage generator. The comparators use large devices, common-centroid matched layout, guard rings and dummy devices with sufficient distance between active devices and the well edges, thus minimizing the well-proximity effects. Thus, the comparator variation contributing to the variation in trip points is minimized. The threshold voltage generator on the other hand, uses devices in isolated wells which are not matched in layout. They are more prone to on-chip variation. Thus, the accuracy of comparison is largely determined by variations in the threshold voltage generator.

Figure 5.12 shows the variation in the threshold voltages ($V_{TU}$ and $V_{TL}$) for different tap settings 1000 monte-carlo runs. The worst case spread for $V_{TL}$ is about 60 mV and 64 mV for $V_{TU}$. For both $V_{TU}$ and $V_{TL}$, the box height shows the spread with center bar indicating the corresponding mean. For the same threshold voltage setting $V_{TU}$ and $V_{TL}$ do not overlap, meaning the circuit will always provide a reliable comparison window ($V_{COMP}$) . This result, however, is a pessimistic one for $V_{COMP}$ because the minimum of $V_{TU}$ and maximum of $V_{TL}$ do not occur simultaneously. The mean simulated values for $V_{COMP}$ and the corresponding hysteresis ($\Delta V$) are tabulated in Table 5.2.

Note that for $V_{REG}$ greater than $V_{BAT}/2$ (approximately), the comparator sense voltage is divided by 2 using FB2 (Figure 5.5). Since the divided version of $V_{REG}$ is obtained

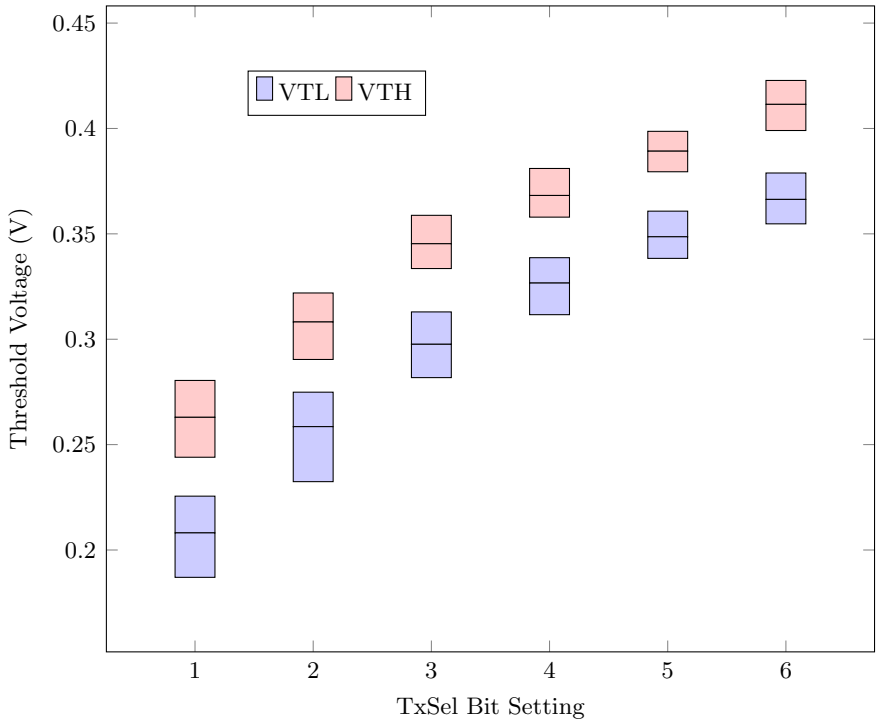



FIGURE 5.12: Spread of $V_{TU}$ and $V_{TL}$ over 1000 monte-carlo simulations.

at the midpoint of the diode stack (Figure 5.11), the ratio remains independent of temperature and $V_{REG}$.

## 5.4 Results

This section presents the measured DC and transient results of the proposed scheme. Figure 5.13 shows DC results for two cases: (a) with $V_{REG}$ increasing up to the desired range before decreasing and (b) with $V_{REG}$ increasing beyond the desired range (over-voltage). Figure 5.13a shows a $\Delta V_{TL}$ of 220 mV. However, when $V_{REG}$ exceeds $V_{TU}$ (Figure 5.13b) $\Delta V_{TL}$ is redundant and is reduced to 5 mV. A 120 mV $\Delta V_{TU}$ prevents QU from oscillating. Note that $Q_{INRANGE}$ is asserted only for $V_{TL} < V_{REG} < V_{TU}$.

Figure 5.14 shows the transient results with $V_{REG}$ transitioning from $V_{TL}$ - 30 mV to $V_{TL}$ + 30 mV. Because the voltage in this band this does not exceed $V_{TU}$, QL determines $Q_{INRANGE}$. Note that the delay in detecting an in-range condition is 6 µs (1.2 V, room temperature). Figure 5.15 demonstrates the voltage monitor for system transitions between super-, near- and sub-threshold voltages. $V_{REG}$ transitions from a 0.3 V retention voltage to 0.4 V, 0.6 V, then 0.8 V in 10 ms. In each mode the voltage monitor correctly detects an in-range and out-of-range condition (upper and lower limits). Note that FB2 is asserted for 0.4 V to bypass the divider. Mode transitions at 0.8 V are of sub-ms order, as dictated by higher CPU clock frequencies achievable at super-threshold voltages.

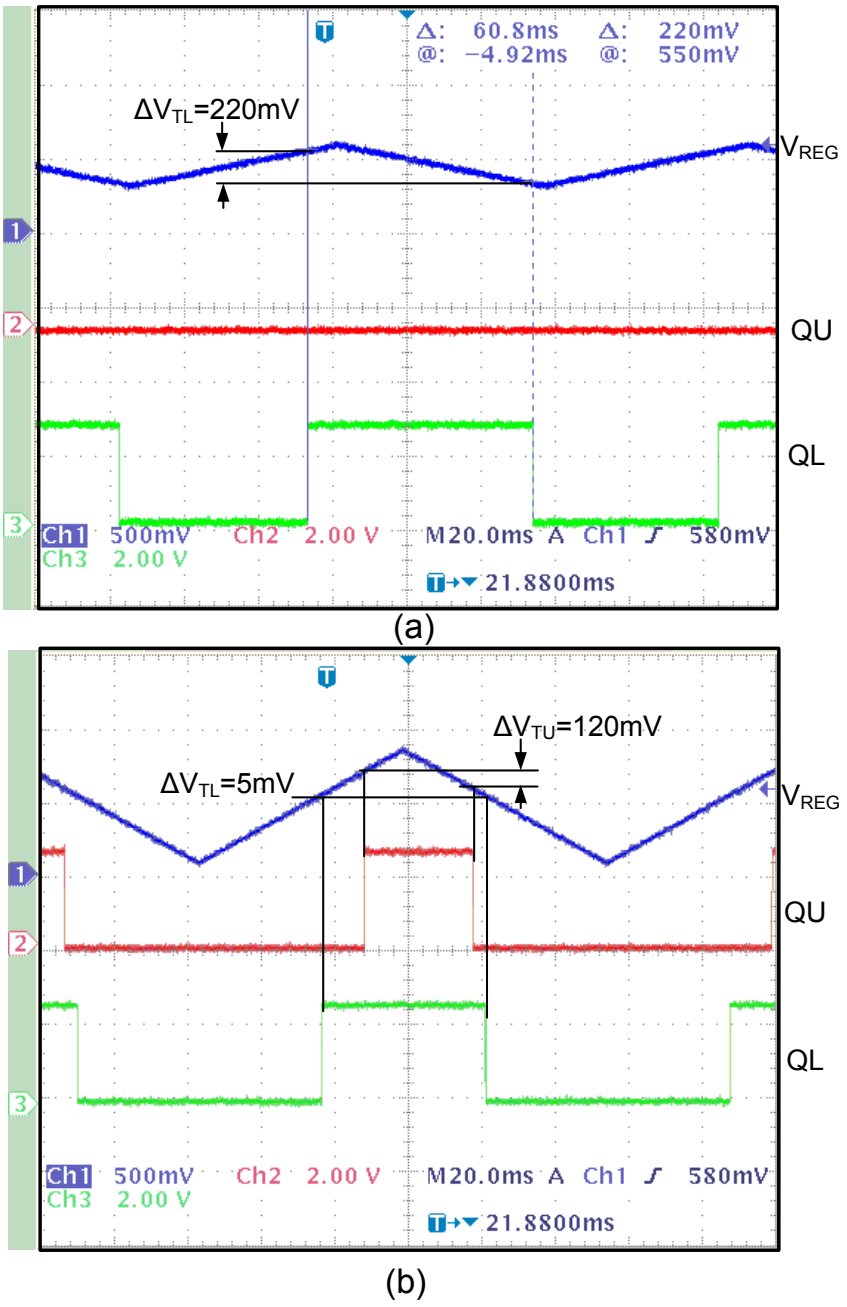



FIGURE 5.13: Measured DC results showing (a) $\Delta V_{TL}$ and (b) $\Delta V_{TU}$ for two operating states (3 and 4 in Figure 5.4).

The voltage monitor reaches its highest energy consumption (power times duration) in state ③ when CMPU and CMPL have bias settings of 1 and 2 respectively (Table 5.1). The voltage monitor consumes 50 nW in this setting at 1.2 V, as shown in Figure 5.16. The variation in the quiescent power with supply voltage and temperature is also shown. The proposed design is compared with the state-of-the-art in Table 5.3. The energy expended while waiting for a response from the monitor ($E_{wait}$) is the lowest for the proposed design. The chip plot is shown in Figure 5.17. The voltage monitor uses 58 µm x 33 µm area dominated by the two comparators.

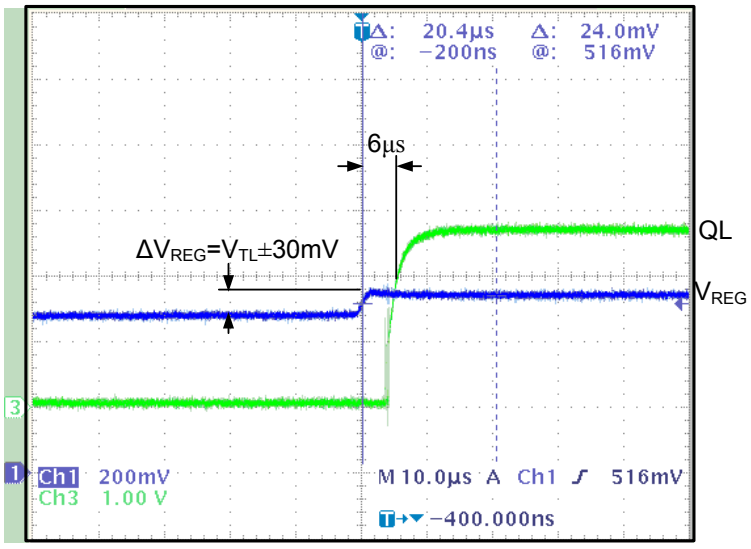



FIGURE 5.14: Measured transient results showing CMPL speed for a bias setting of 2.

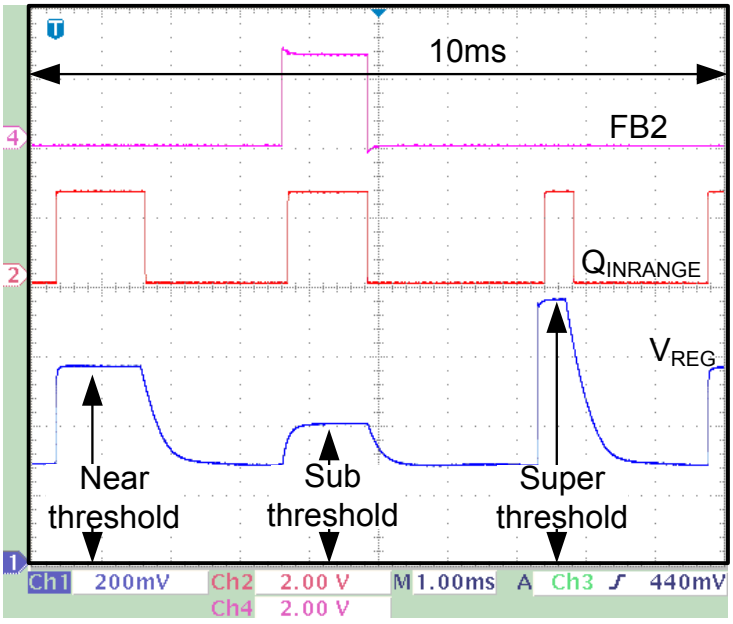



FIGURE 5.15: Voltage monitor response for $V_{REG}$ transitioning from retention to super-threshold voltages.

## 5.5 Discussion

Scaling supply voltage to sub/near-threshold level is necessary to achieve minimum energy operation in processors for wireless sensor nodes. To exploit potential energy savings best, such wireless sensor nodes need assist circuits, many of which perform analog functions. This chapter described the implementation of an ultra-low power voltage monitor circuit to assist MinE CPU systems with fast, wide-range voltage scaling. The proposed scheme achieves better balance between response speed and quiescent power as shown in Figure 5.18.

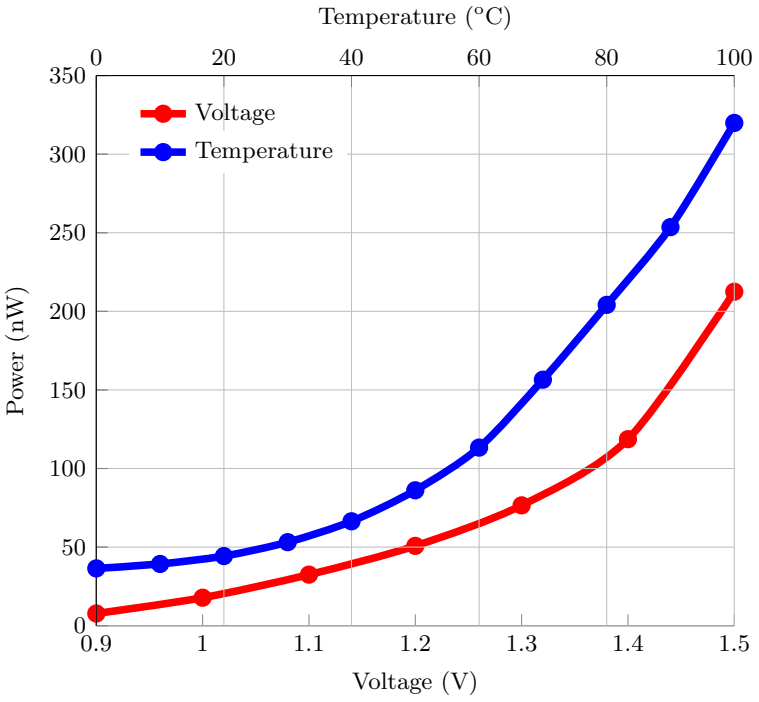



FIGURE 5.16: Measured power for voltage and temperature sweep

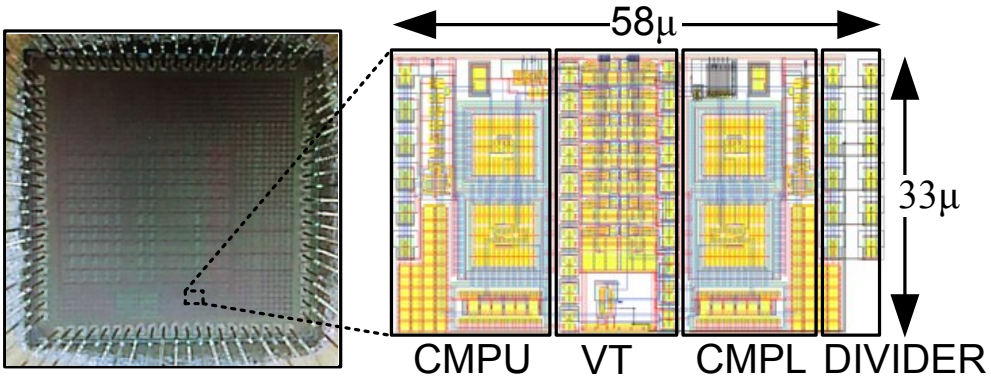



FIGURE 5.17: Chip plot

The benefits from MinE CPU systems can be easily overwhelmed by slow or high-power voltage monitors. For the CPU system described in [73], the retention power $P_{ret}$ increases 16× to 1600 nW when $V_{REG}$ changes from 0.3 V to 0.8 V in preparation for the active mode. Consider the example case from Section 3.6, where $N_{act}$=3000 and $f_{cpu}$=1 MHz. The energy spent waiting for the rail voltage to settle is given as

$$E_{wait} = P_{ret} T_{wait} \tag{5.1}$$

Here $T_{wait}$ is the time spent at the active mode voltage (0.8 V) waiting for CKEN to be asserted (Figure 5.3). Assuming a $T_{wait}$ of 1.2 ms, $E_{wait}$ = 1.92 nJ. For $D$=10, the sensor node can have 30 sleep-active cycles per second resulting in $E_{wait}$ = 57.6 nJ.

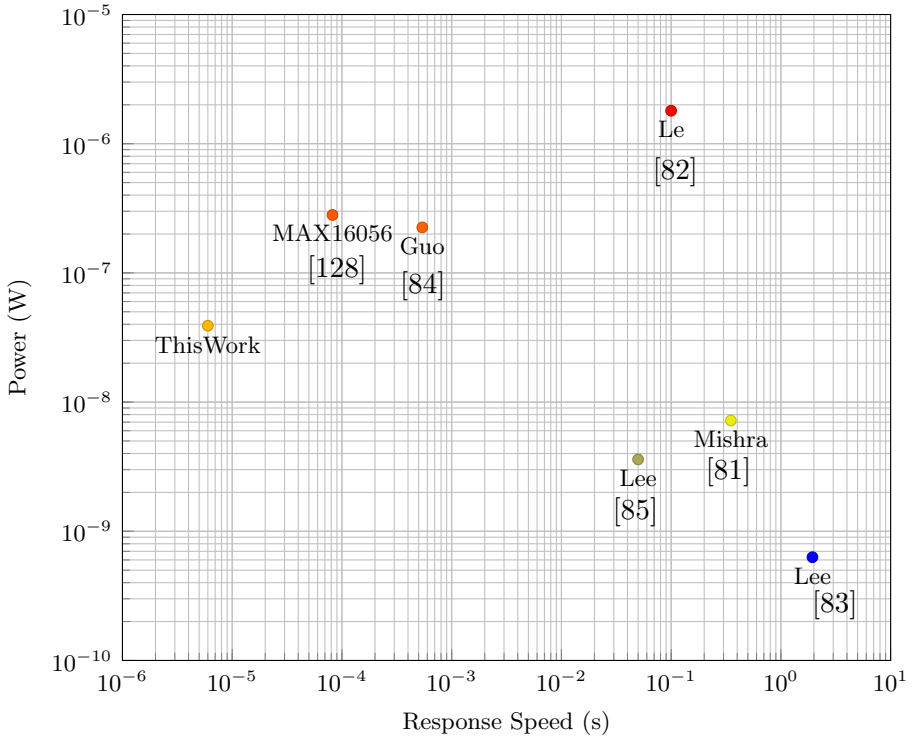



FIGURE 5.18: Power-Delay balance compared with prior works.

TABLE 5.3: Comparison of state-of-the-art voltage monitor circuits

| | **[82]** | **[83]** | **[81]** | **[84]** | **[85]** | **This Work** |
|---|---|---|---|---|---|---|
| Node (nm) | 180 | 180 | 180 | 90 | 180 | 65 |
| $V_{DD}$ (V) | 1.8 | 3.6 | 1.8 | 1.0 | 3.6 | 1.2 |
| Delay (s) | 0.1 | 1.94 | 0.35 | 500µ | 0.05 | 6µ |
| Power (nW) | 3600 | 0.63 | 650 | 540 | 3.6 | 50 |
| Area (sq.mm) | 0.012 | 0.009 | NR | 0.088 | 0.17 | 0.002 |
| ΔV (mV) | NR | 200 | 66 | 432 | 77 | Configurable |
| $E_{WAIT}$ (nJ) | 180 | 1.2 | 2.5 | 0.11 | 0.17 | 0.3m |

NR: Not reported.

The voltage monitor thus saves more energy than it expends becomes energy-neutral for sensor workloads with 30 or more wakes per second.

Note that $T_{wait}$=1.2 ms is faster than prior reported works which [83], [85], [81], [82] fail to meet the speed requirement for 30 sleep-active cycles. Duty-cycled comparators and state-aware dynamic power-bandwidth tuning limit the overheads of the proposed monitoring scheme to 1% of the CPU system's active power at minimum energy point voltage.

# Chapter 6

# RC Oscillator with Sub-Clock Power-Gated Comparator

IoT applications, such as biomedical implants [130] and environmental sensors [131], can be powered by small batteries or harvested energy [23] and must, therefore work efficiently. They are typically implemented with synchronous digital SoC devices, which rely on low-cost integrated ultra low power clock sources. The primary options are ring oscillators, crystal or LC tanks and RC relaxation oscillators.

The frequency output of ring oscillators changes significantly with voltage and temperature. Crystal and LC oscillators are fabricated with tighter tolerances and are less sensitive to temperature variation but pose integration challenges. Because their frequency depends on a resistor and capacitor relaxation oscillators are particularly attractive for integrated clock sources in general sensing applications that are not bound by radio timing requirements - such as reference timer for sensor wake up or tuning coarse CPU clocks in sensor nodes. However, relaxation oscillators (see Section 6.1) use a reference voltage which limits their line and temperature stability. In addition, comparator offsets and resistance temperature coefficients can their degrade stability.

Many of these non-idealities have a direct trade-off with power, i.e., minimizing power results in poorer stability. The trade-offs also depend on the timing needs: in high-frequency oscillators (kHz to tens of MHz), energy efficiency is limited by comparators and precision references, while, for low-frequency oscillators, (Hz to sub- Hz), it is limited by leakage due to the complex designs required for generating slow time constants. Figure 6.1 shows the state-of-the-art for oscillator frequency vs power. The region above unit nW/kHz is densely populated across the frequency range of sub-Hz to tens of MHz. It should be noted that aggressive power reduction typically degrades performance [135] [23] [79] which is not captured in Figure 6.1.

To reduce power consumption below unit nW/kHz while maintaining stability, the design

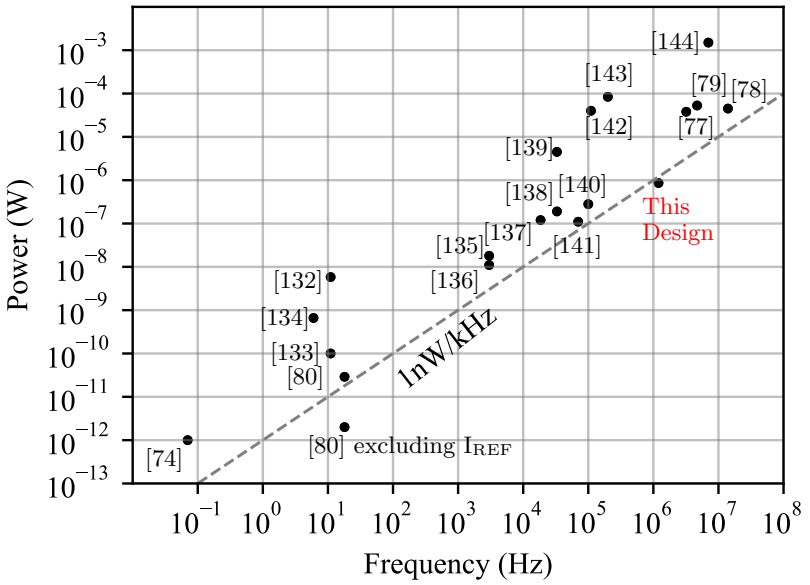



FIGURE 6.1: State-of-the-art performance of relaxation oscillators.

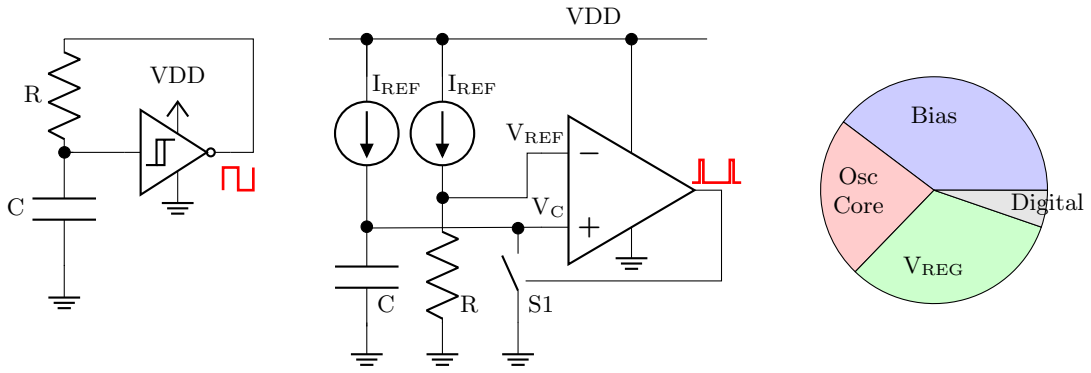
proposed in this work incorporates three key concepts that have not previously been used with relaxation oscillators:

1. Use of a switched-capacitor ratioed reference to minimize the power overheads of the reference generator and large area resistances by (Section 6.2.1).
2. Power gating the comparator and turning it on just in time for a comparison to minimize comparator power while also delivering high bandwidth performance when needed (Section 6.2.2).
3. Use digital-assist in combination with the above to improve line and temperature stability (Section 6.2.3).

A 1.2 MHz relaxation oscillator has been designed and implemented (Section 6.2) in TSMC 65 nm, validated in two silicon batches (Section 6.3). Measured results (Section 6.4) show good line (0.7 %/V) and temperature stability (100 ppm/°C) while expending 820 nW for a 1.2 MHz output frequency (∼0.68 nW/kHz). This was published first at ISSCC 2017 conference [27] and subsequently in 2019, expanded as a journal paper published in the journal of solid state circuits [28].

## 6.1 Relaxation oscillators

Relaxation oscillators are typically constructed using a high-gain inverter, or a voltage reference and comparator. In the first arrangement, the inverter [138], or a Schmitt




(a) Using Schmitt (b) Using $V_{REF}$ (c) Power Breakdown

FIGURE 6.2: Typical scheme for realizing relaxation oscillator. (a) using Schmitt trigger, (b) using a voltage reference and comparator and (c) typical power distribution in such relaxation oscillators with greater than 60% expended by comparator and reference generator combined [136].

inverter (as shown in Figure 6.2a), acts as a high-gain hysteretic comparator. The lowest input voltage at the Schmitt input to produce a logic high ($V_{IL}$) and the highest input voltage to produce a logic low ($V_{IH}$) set the trip points. As the capacitor voltage ($V_C$) crosses these trip points during charge and discharge cycles, the Schmitt output changes phase, resulting in a square wave; however, $V_{IL}/V_{IH}$ varies significantly with PVT and affects stability. The second arrangement (Figure 6.2b) improves stability by using a fixed reference, which is either a fixed current generating a reference voltage across a resistor, as shown, or a fixed voltage reference (e.g., bandgap) and a fast comparator [136], [137]. However, the high-speed comparator and reference generators that are tolerant of PVT variation consume additional power.

In essence, traditional designs suffer from a trade-off between power and stability. Two common techniques can be used to minimize the power expended in fixed references:

1. Use of a slow low-power control loop feedback to compensate for variations in reference voltage [78, 140].
2. Use of a lower internal voltage from a tracking low-drop-out regulator [138], [139].

However, both of these techniques present similar trade-off challenges. For example, control loop stability and robustness cost power and area and, with lower internal voltage, the drop-out voltage of the regulator leads to an inefficient design.

Likewise, common techniques reported in the literature to minimize comparator power are:

1. Use of a comparator-less phase-shift-based design, where the device $V_T$ sets the trip threshold [138], [80].

2. Use of a low-speed amplifier-based control loop instead of a high-speed comparator [135], [74].

3. Use of a simple (common-gate or common-source) gain-stage with a digital-logic-gate like structure to force rail-to-rail swing [145], [146].

Other critical aspects that determine the floor for low-power relaxation oscillator operation are the RC network and long-term relaxation oscillator stability. The power dissipated in the timing resistor, given by

$$P_{RC} = CV_C^2 f_{OSC} \tag{6.1}$$

is inevitably wasted. The waste can be minimized by reducing the voltage swing across the RC network. However, the smallest usable RC swing is bounded by the comparator design complexity (and its resulting power overhead), and phase noise [76]. Similarly, the long-term stability of the relaxation oscillator is limited by flicker noise in the timing resistor (comparator 1/f noise minimized by chopping or other suitable techniques) and is, again, inversely proportional to the power dissipated [147].

An elegant solution that circumvents both the reference and comparator limitations in a correct-by-design fashion is shown in Figure 6.3 [79]. It does so by:

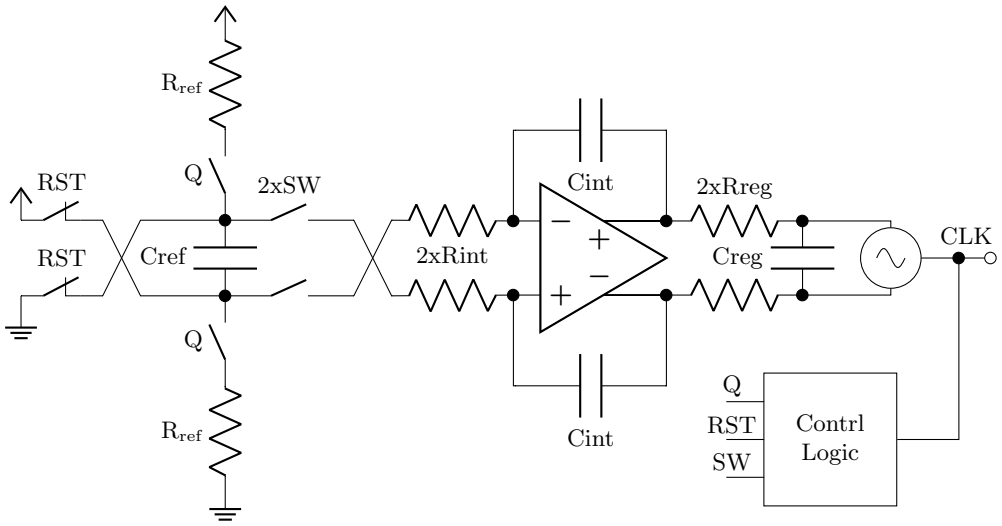
1. Sampling the supply voltage differentially using a period detector. Switches R and Q form an H-bridge allowing $C_{REF}$ to charge to ±VDD through $R_{REF}$. This differential sampling cancels out any VDD variation. The zero-crossing time is dictated only by $R_{REF}C_{REF}$.

2. Using the integrator's virtual ground as the reference and avoiding a separate reference generators. In practice, the virtual ground can move away from ideal zero due to internal offsets. So, chopping is employed to cancel the effects of internal offset.

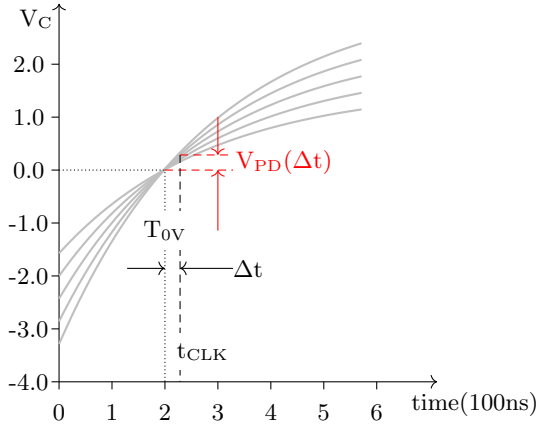
3. Using a slow integrator and a ring VCO in lieu of a high-speed comparator.

However, the requirement for differential sampling ($4V^2/R$), plus a high power integrator and ring-oscillator results in a relatively high power consumption of 53 µW (11.3 nW/kHz).

A sub-nW/kHz performance has been reported previously [80], but the design has several performance points with a varying power-stability trade-off. For example, the design consumes 2.1 nW/kHz for a stability of 8500 ppm/°C, which is an order of magnitude higher than the values reported in related works including the design proposed here. In addition, in sub-nW/kHz mode, the design delivers significantly poorer stability of 21000 ppm/°C, which would not be tolerated by many applications.




(a) Scheme.




(b) Conceptual Waveform.

FIGURE 6.3: relaxation oscillator scheme using virtual ground as the reference voltage [79]

A 3 kHz, 1.56 nW/kHz performance has been reported for a frequency-locked oscillator design [135]. The timing resistance is realized using a switched capacitor network, which helps reduce power and area. However the slow integration limits the start-up time of the oscillator (100s of ms).

As shown by the power breakdown (Figure 6.2c) for a typical relaxation oscillator design [136], the combined power of the reference generator and comparator is significant. The performance-power trade-off with reference generators and comparators limits the options available in traditional design approaches.

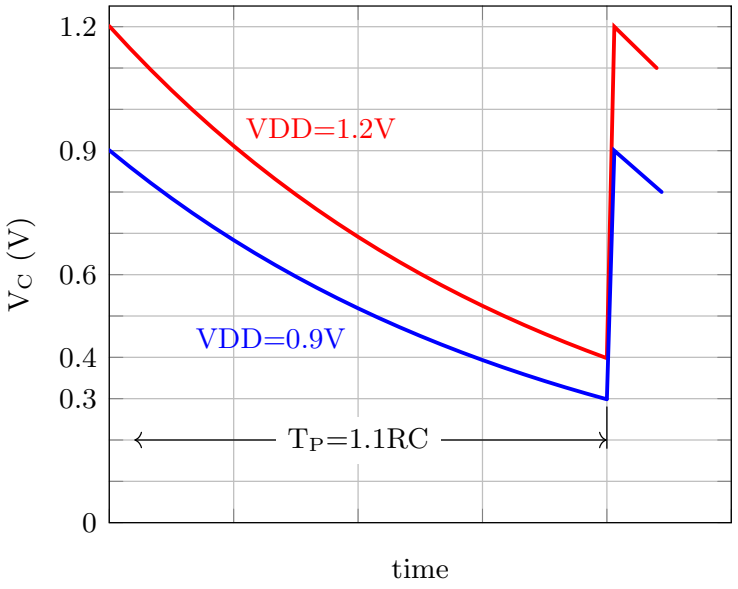



FIGURE 6.4: Conceptual RC discharge waveforms showing that period remains unaffected by supply voltage variations.

## 6.2 Proposed design

The proposed ultra-low power relaxation oscillator overcomes the limitations and trade-offs found in existing designs. It builds on the techniques highlighted in [79] (Figure 6.3) to minimize VDD sensitivity through a ratioed reference generator and saves comparator power through sub-clock duty cycling. Line and temperature stability are improved further using a digital-assist scheme.

### 6.2.1 VDD ratio reference

This work proposes minimizing VDD sensitivity by using a reference that is a ratio of VDD instead of fixed references and differential cancellation [79]. Illustrating the concept of ratioed operation, the RC discharge waveforms in Figure 6.4 show the oscillation time period ($T_P$) is the same, despite operating at different supply voltages. The starting capacitor voltage ($V_C$) is determined by the supply voltage but, irrespective of the starting point, a constant time is taken to discharge to a specific ratio of the starting voltage. $T_P$ is determined only by the timing RC network. Fixed ratio voltages can be obtained efficiently using switched capacitor converters; the proposed design uses a ratio of 1/3 switched capacitor converter for the following reasons:

1. The lower bound on the reference voltage is decided by the power dissipated in the timing resistor as $V_C$ swings between VDD and $V_{REF}$. The upper bound is limited by the comparator power which increases with input common-mode voltage because NMOS input devices are used in the comparator, as shown in Section 6.3

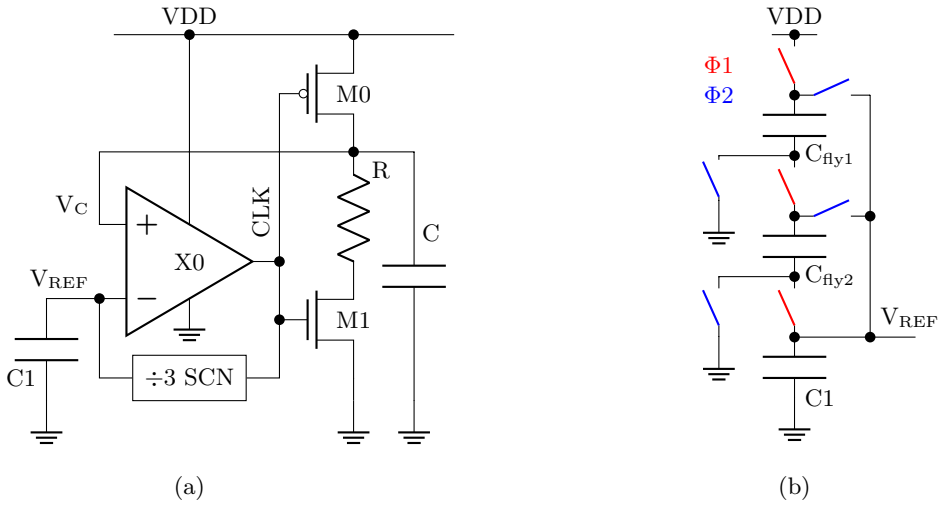



FIGURE 6.5: relaxation oscillator scheme using switched capacitor ratio-ed reference voltage.

(Figure 6.10). A 1/3 switched capacitor converter ratio provides a balance to this trade-off.

2. Larger ratios in switched capacitor converters require more switches and flying capacitors. A 1/3 ratio limits the number of switches and capacitors, reducing bottom-plate and switching losses.

The exponential decay of the capacitor voltage is given by equation (6.2), with the choice of a 1/3 ratio resulting in a $T_P$ of $1.1RC$ in equation (6.3).

$$V_C = V_{dd} e^{\frac{-t}{RC}} \tag{6.2}$$

$$T_P = ln(3)RC \tag{6.3}$$

A schematic of the proposed relaxation oscillator using the SC network is shown in Figure 6.5a. Under a steady-state, assuming $V_{REF} = VDD/3$ and $V_C$ is at VDD, the X0 output is high, which turns off the M0. The capacitor discharges via the timing resistor R and M1. As $V_C$ approaches $V_{REF}$ and equals it, X0 turns off M1 and initiates a charging cycle through M0. The capacitor charges to VDD, and the cycle repeats. The switched capacitor converter is not actively loaded by X0; hence, $V_{REF}$ is relatively unaffected by temperature, improving relaxation oscillator stability. A relatively large decoupling capacitance (C1 of 2 pF) at the switched capacitor converter's output minimizes the ripple on $V_{REF}$. The silicon implementation of this scheme allows an alternate ratio generation scheme using diode-stacks. However, the oscillator performance degrades significantly with diode stacks compared with the switched capacitor converter based ratio [27], which is in line with results from [135].

Figure 6.5b shows the switched capacitor converter scheme. In phase Φ1, $C_{fly1}$, $C_{fly2}$ and

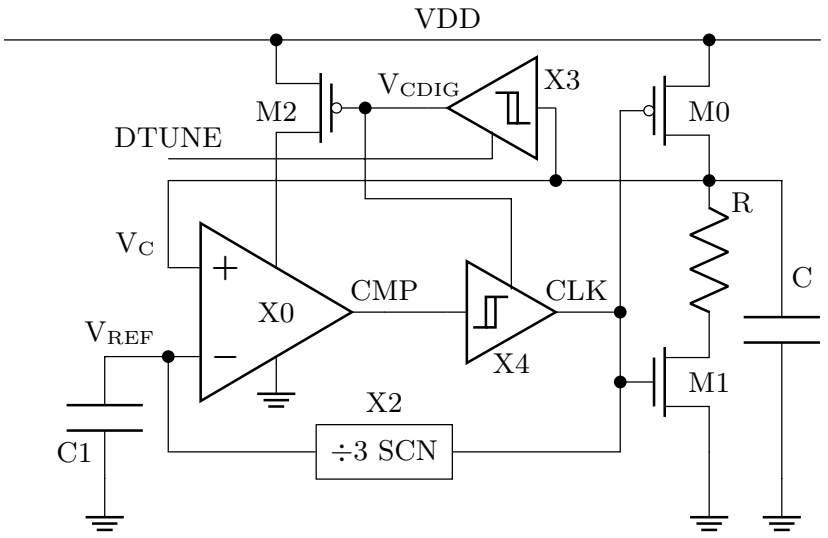



FIGURE 6.6: relaxation oscillator scheme using sub-clock duty cycled comparator and switched capacitor reference voltage

C1 connect in series between VDD and GND and each one charges to a third of the VDD. In phase Φ2, they connect in parallel and discharge into the $V_{REF}$ node. Therefore, the voltage across C1 is always at VDD/3. Note that the switched capacitor converter does not need a separate external oscillator which further reduces the power overheads and the start-up time thereof. This does not limit start-up, as will be explained later in Section 6.4.1.

### 6.2.2 Sub-clock duty cycling of comparator

The proposed scheme does not fight the comparator power-speed trade-off, as discussed in Section 6.1. Instead, it retains the full comparator bandwidth and relies on duty cycling to reduce power, i.e., the comparator is off for most of the clock time period. Sub-clock power gating has been exploited in digital designs [148] to achieve as much as a 27% reduction in power. This technique is extended here to an analog macro. Duty cycling has also been demonstrated in [80], where a higher power current reference ($I_{REF}$) is duty cycled to reduce overall system power, but the $I_{REF}$ remains off for several clock cycles. Although the attempt here is to retain as much of the comparator bandwidth as possible, the related error is still nonzero [146] as discussed further in Section 6.4.

Observing (in Figure 6.4) that the comparator X0 is only necessary when $V_C$ is close to VDD/3, this work proposes turning off X0 for $V_C \gg$VDD/3. By making sure that the high-power comparator is only powered up in time for a comparison, the overall relaxation oscillator power can be minimized greatly, allowing for as much power to be expended in the comparator as is needed to guarantee the required stability. In order to know the turn-on time, a coarse ultra-low power comparator turns the precision comparator on only for the comparison window. This duty cycling of the comparator

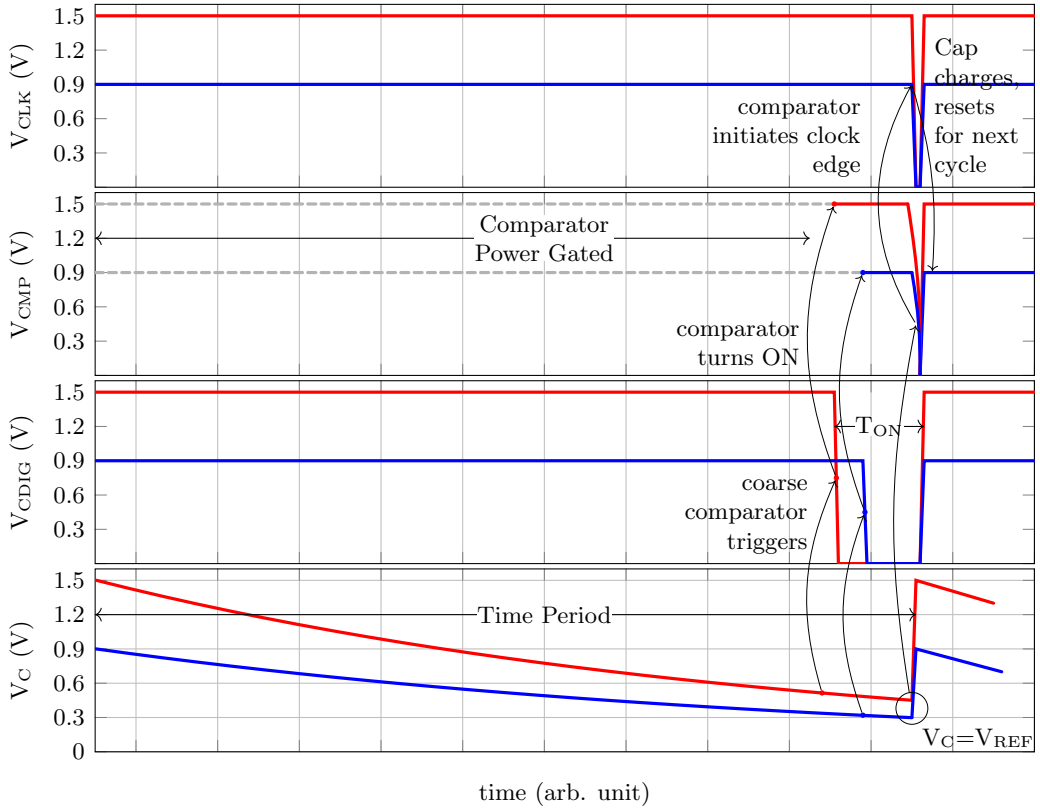
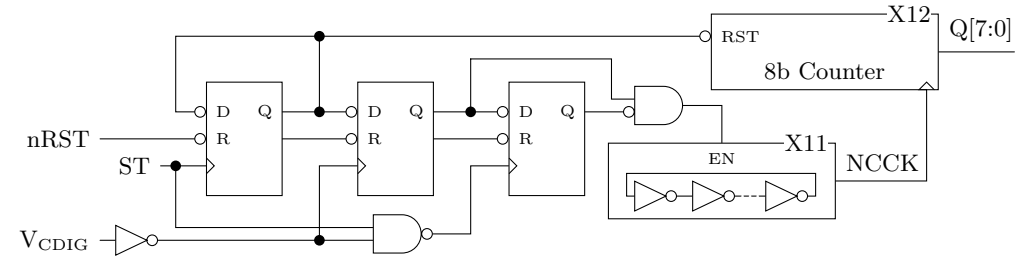



FIGURE 6.7: Conceptual waveforms showing sub-clock duty cycling of comparator.

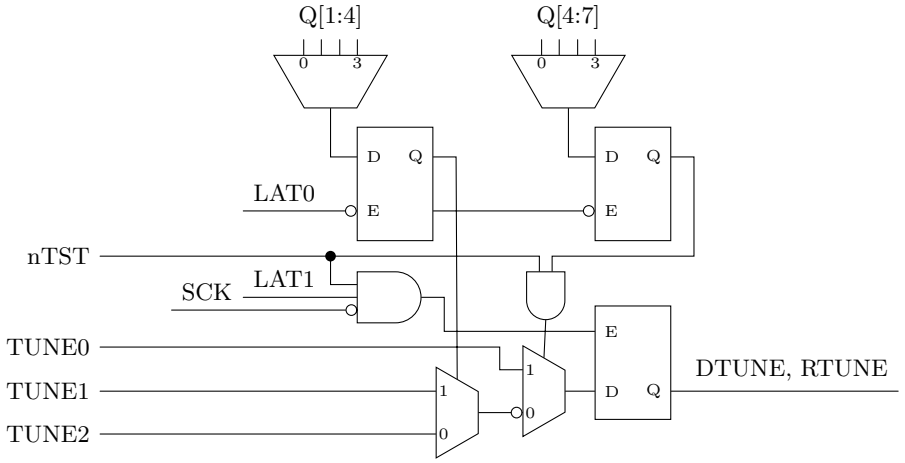
saves 80%-90% of the power. Figure 6.6 shows the conceptual implementation of this scheme.

A Schmitt trigger (X3) is used to perform the coarse comparison. When $V_C > V_{IL}$ of the Schmitt trigger, the comparator is power gated. The comparator is only turned on for $V_{IL} \geq V_C \geq VDD/3$. The $V_{IL}$ variation with PVT only affects the duty cycle ratio, thus only the relaxation oscillator power is affected and not relaxation oscillator stability. This decouples the conflicting design constraints for X3, thus allowing the $V_{IL}$ of X3 to be, relatively easily, designed in such a way that it is greater than VDD/3 across PVT. An output clamp is needed when the comparator power is gated because its output is tristated. The Schmitt trigger (X4) placed in-line with the comparator output serves this purpose.

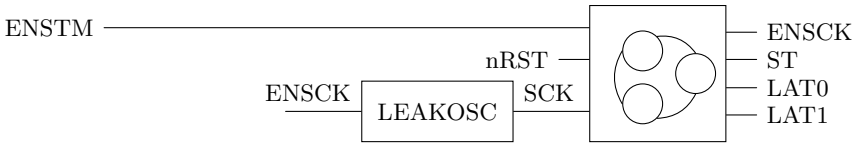
The waveforms in Figure 6.7 show the working of this topology. The lower-most panel shows $V_C$ for VDD=1.5 V and VDD=0.9 V. $V_{CDIG}$ is the power gating signal and 'CMP' is the output of the comparator. CMP is invalid for the duration when $V_{CDIG}$ is high. The 'CLK' net is clamped to VDD during this phase but allowed to transition when $V_C$ = VDD/3. Note that theoretically, the clock period remains insensitive to VDD.




(a) $T_{ON}$ Measurement




(b) Lookup




(c) Control

FIGURE 6.8: Digital-assist and system integration of the RxO showing (a) fast NAND based ring oscillator and counter for $T_{ON}$ measurement, (b) latch and multiplexer based lookup scheme and (c) slow leakage based oscillator for clocking the control state-machine.

### 6.2.3 Digital-assist

The $V_{CDIG}$ determines X0's on-time ($T_{ON}$), as shown in Figure 6.7. $T_{ON}$ increases with VDD, meaning X0 is turned ON earlier each cycle at higher voltages, which costs relaxation oscillator power. A simple counter and state-machine based all-digital support system has been implemented to minimize $T_{ON}$. This is illustrated in (Figure 6.8). This has three sub-blocks:

1. $T_{ON}$ measurement during runtime
2. A table based lookup for new tuning bit setting base on $T_{ON}$
3. A state-machine to control this digital-assist block

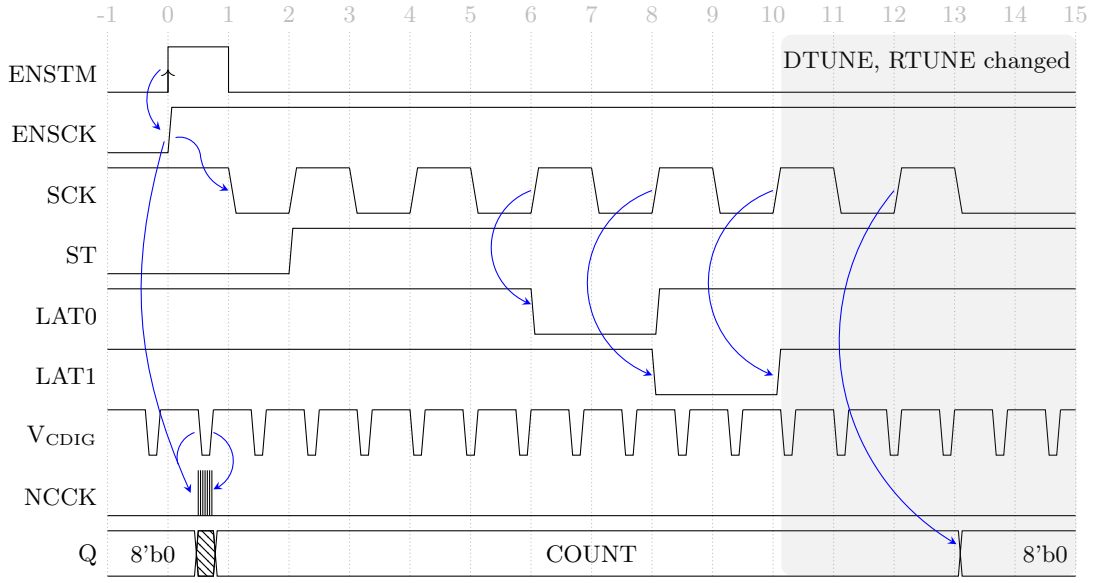



FIGURE 6.9: DAS state-machine waveforms.

**$T_{ON}$ measurement:** $T_{ON}$ is measured by enabling a counter for the low period of $V_{CDIG}$. The $T_{ON}$ measurement is relative; hence, an accurate clock is not necessary for the counter. As shown in Figure 6.8a, a simple ring oscillator (X11) is used to clock the counter (X12) and is enabled only for the duration of $T_{ON}$. The ring oscillator is implemented using NAND4 (simplified in the diagram as inverters). The falling edge of $V_{CDIG}$ arriving immediately after ENSCK is asserted by the state-machine starts the NAND4 counter clock (NCCK). On the subsequent rising edge of $V_{CDIG}$, this clock is gated. The 8 bit counter counts up with NCCK and when NCCK is gated, the counter values (Q[7:0]) reflect $T_{ON}$. The ring oscillator and the counter both run once and only for the duration of $T_{ON}$ each time a retune is initiated, thus minimizing the power overhead of the digital-assist scheme.

**Looking up new tuning settings:** This sub-block is illustrated in 6.8b. The counter's LSB is ignored to avoid noisy measurements. The remaining bits (Q[7:1]) are split to identify upper and lower thresholds. The upper threshold id determined by bits 4 to 7 and the lower limit is determined by bits 1 to 4 (4th bit allows extreme corner case to checked). If $T_{ON}$ is large then one of upper threshold bits would be high causing TUNE0 to be loaded at the end of the tuning procedure. If $T_{ON}$ is small such that all upper threshold bits are zeros then the either TUNE1 or TUNE2 settings are loaded depending on Q[4:1] bits. The tuning settings are only altered when X1 is OFF; hence, there are no false edges in the relaxation oscillator output as a result of retuning.

nTST is a test signal to override the counter values and force TUNE0 to be ignored or to allow a default tuning setting to be loaded.

**Control state-machine:** Figure 6.9 illustrates the timing of the digital-assist scheme. A retune is initiated by the rising edge of ENSTM, which sets off the state-machine

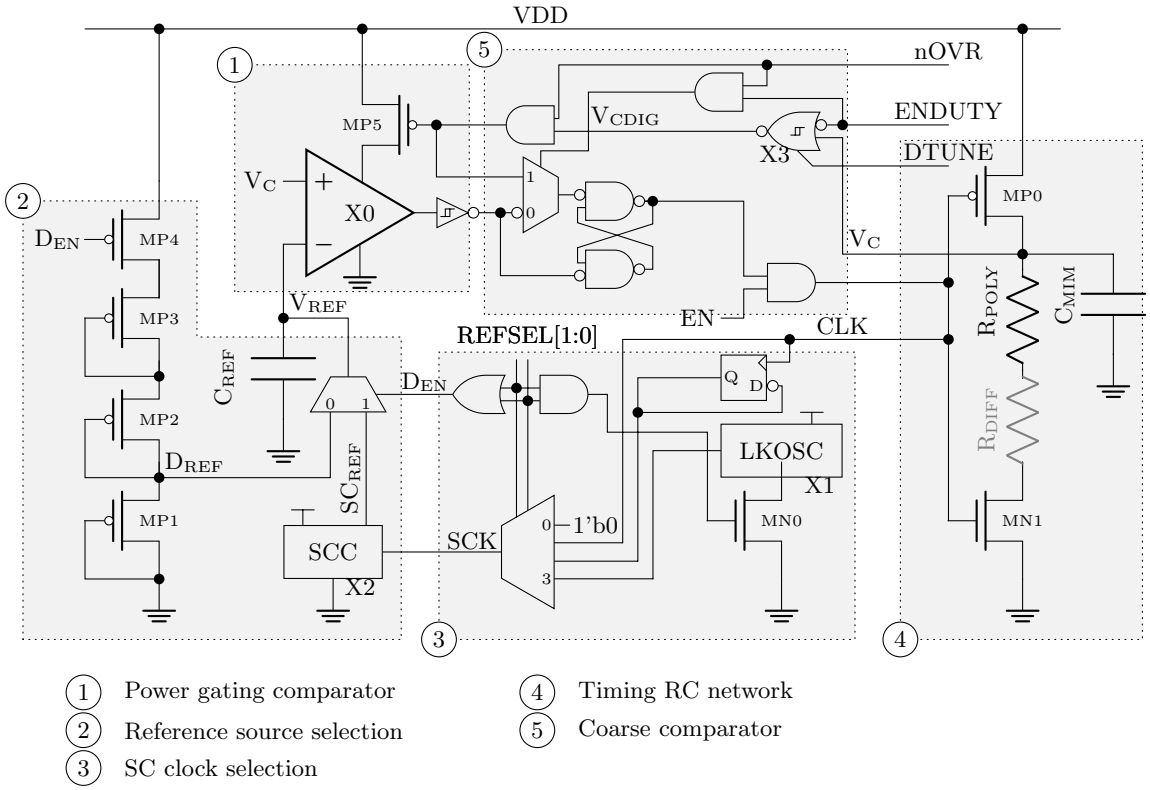



FIGURE 6.10: Relaxation oscillator scheme developed from Figure 6.7 showing test features as implemented on silicon.

clock using ENSCK. After the counter is gates the values (Q[7:0]) are latched for six SCK cycles. The third and fourth SCK rising edges latch the retuning bits based on the counter output using multiplexers as shown in Figure 6.8b. The fifth rising edge changes DTUNE and RTUNE, thus changing the $V_{IL}$ of X3. After retuning, power is minimized by clock gating the digital-assist logic by de-asserting ENSCK.

The state-machine, as shown in Figure 6.8c, is clocked from a leakage-based slow oscillator [44], thereby minimizing the power overheads of the digital-assist scheme. In practice, ENSTM may be asserted at regular intervals to initiate retuning by the ultra-low power sensor system, as in [72], [73]. As shown later in Section 6.4, the overhead of this digital-assist scheme is <0.1% of relaxation oscillator energy even when retuning is initiated at 10 ms intervals.

### 6.2.4 Digital-assist tuning

Digital-assist tuning relies on pre-loading calibration settings for preset counter threshold values in a table during post-silicon testing. During run time, the count (of X12 in Figure 6.8) is determined by $T_{ON}$ which increases with supply voltage. The temperature has a much weaker, but opposite, effect on $T_{ON}$ and hence the counter values. Measured results to this effect are shown in Section 6.4 (Figure 6.16).

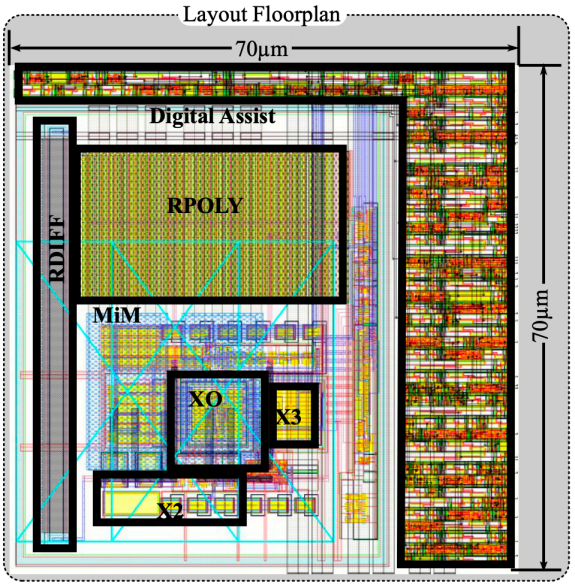



FIGURE 6.11: Layout of the proposed oscillator scheme showing the poly resistor, MiM caps and the digital assist blocks.

During run time, invoking a retune by asserting ENSTM causes the counter value to be used as an index to look up DTUNE and RTUNE settings in the preloaded values. The implemented design allows three different settings to be used for the entire operating supply voltage and temperature range although in practice more settings are possible. In addition, a default setting sets the relaxation oscillator frequency close to the nominal of 1.2 MHz at 1.2 V, 25 °C. For the measurements presented later in Section 6.4, recalibration was initiated only for line sensitivity results and not for temperature stability results.

## 6.3 Implementation and simulation results

The schematic of the proposed design as implemented in silicon is shown in Figure 6.10. The design includes test and tuning features in addition to the basic scheme shown in Figure 6.6. The following is a description of the sub-blocks of the design.

**Power gating comparator:** The output of X0 floats when it is power gated. This can lead to undesirable short-circuit currents in the multiplexer or logic gates that follow. Hence a Schmitt inverter is placed which minimizes the short-circuit current and provided clean rail-rail signal for the subsequent stages.

**Reference source selection:** $V_{REF}$ can be selected from a stack of diode-connected PMOS FETs (MP1-MP4) and the proposed switched capacitor reference (X2 in Figure 6.10). The former allows a much lower power implementation of the ratio reference but the strong temperature dependence of the bulk leakage currents of these devices make them unreliable. This alternative source of reference is included here for sake of comparison.

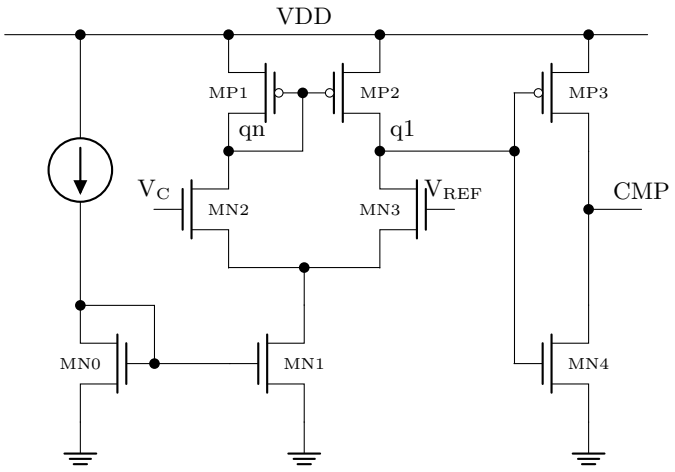



FIGURE 6.12: Comparator Schematic.

**SC clock selection:** The clock for X2 can be chosen among the relaxation oscillator clock (CLK), a divided version of CLK or a separate leakage-based oscillator (X1). The latter was added to allow diagnosis in case of start-up failure.

**Timing RC network:** The timing resistor (1.1 MΩ) is a combination of poly with diffusion resistance with complementary temperature co-efficient, thus helping to achieve better temperature stability. The timing capacitor (600 fF) is realized using MiM layers for precision, and their tolerance to temperature and voltage variation.

**Coarse comparator:** The coarse comparator is modified to a Schmitt NOR (X3 in Figure 6.10) to allow duty cycling to be tested. The basic scheme in Figure 6.6 can become potentially unstable due to the lack of hysteresis in the comparator. As $V_C$ approaches $V_{REF}$, then equals it, X0 turns off the discharge cycle and initiates a capacitor charging cycle, causing $V_C$ to increase, which immediately forces X0 to initiate a discharge cycle. This quick action prevents $V_C$ from charging to full rail. Failure in charging to full-rail does not satisfy the ratio and greatly affects timing. X3 waits only until its $V_{IH}$ is satisfied before power-gating X0 and thus does not guarantee stability. Hence, an SR latch is included at the output of the comparator (as shown in Figure 6.10), ensuring $V_C$ always starts from the rail voltage.

Figure 6.10 also shows details of the comparator (X0), X2 and X3. The comparator (Figure 6.10) consists of a basic differential input stage followed by a high-gain inverter stage. The switched capacitor ratio-reference generator does not need a low output impedance because it only drives the gate-capacitance of the comparator input, allowing the use of complementary devices instead of pass gates for switches and eliminating the need for complementary clocks (Figure 6.10) and nonoverlapping delay generators. The flying capacitors are realized using 160 fF MOS capacitors with a 2 pF MOS decoupling capacitor on net $V_{REF}$.

The Schmitt NOR (X3) implementation is shown in Figure 6.10. Because the Schmitt

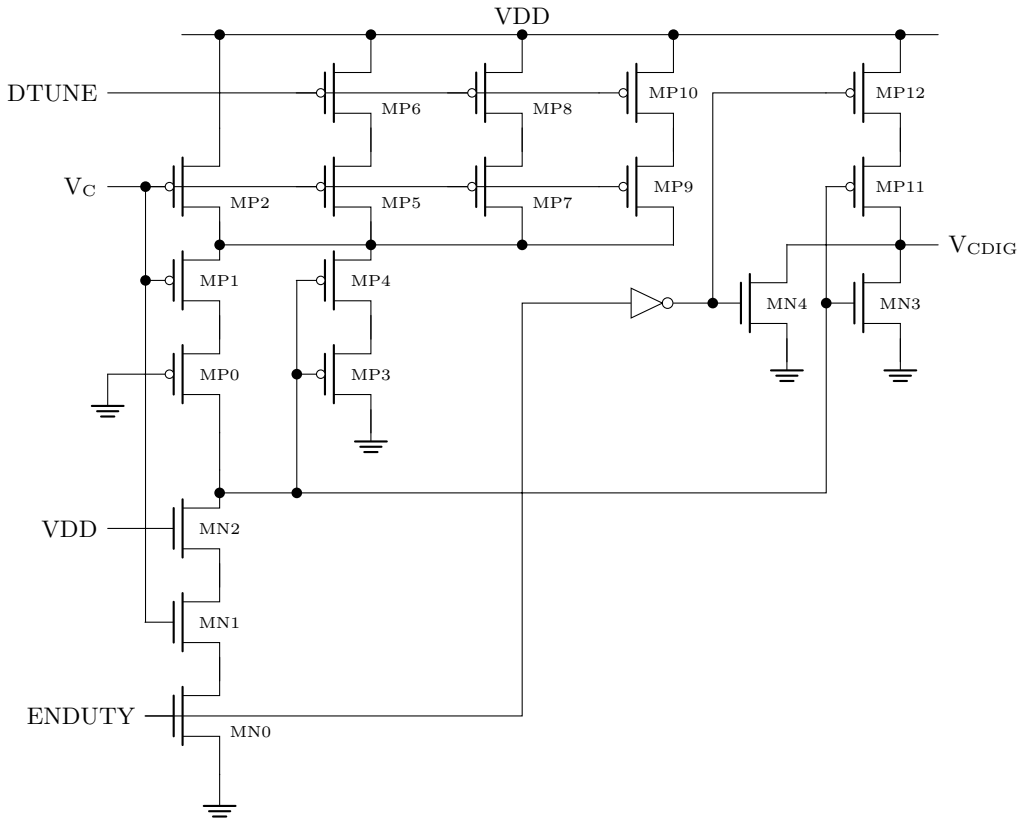



FIGURE 6.13: Schmitt NOR as a coarse comparator for duty cycling the precision comparator.

transition is only needed for the falling edge of the $V_C$, the positive feedback loop is implemented using only the PMOS FET devices MC9, MC10 and MC3. Doing so minimizes the shoot-through current as $V_C$ is slow-falling, thereby reducing the overall power overhead of the coarse comparator. Further, the effective width of MC1 can be increased using the DTUNE bits connected to MC7, MC11 and MC13. These bits shift the Schmitt trip point above $V_{REF}$ allowing control of comparator $T_{ON}$.

Figure 6.11 shows relaxation oscillator floor-plan. X0 uses separate deep N-Well guard rings for the N and P differential pairs with dummy devices to reduce well proximity effects and improve matching. The majority of the area is taken up by the poly-diffusion resistor combination. The switched capacitor network is placed adjacent to the comparator without risk of substrate noise because there is no load on the switched capacitor reference. The digital-assist logic was autoplaced and routed with EDA assistance. The entire macro, implemented on a TSMC 65LP process, was laid out in a 70 µm x 70 µm area.

In the remainder of this section, critical non-idealities of this design that affect timing will be discussed and supported with simulation results.

**Pull-up PMOS sizing:** The pull-up PMOS device (M0 in Figure 6.10) requires careful sizing because its ON resistance affects the rise time and its OFF resistance affects

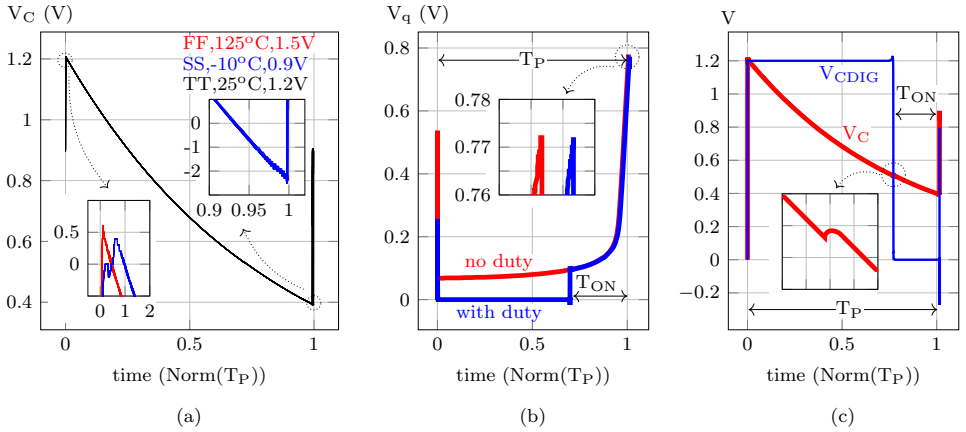
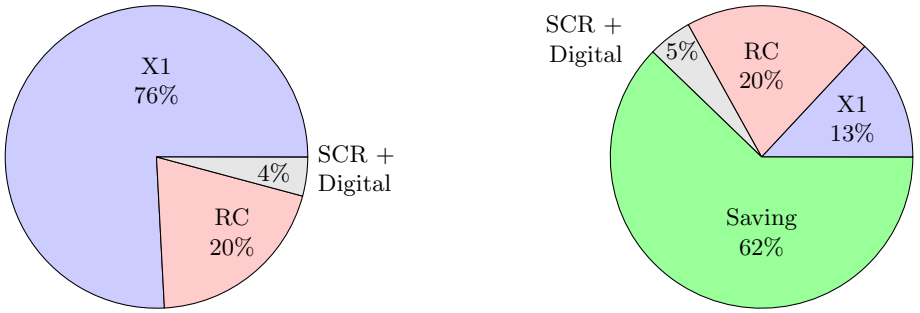



FIGURE 6.14: Simulated eye diagram of 24 cycles of $V_C$ showing (a) ideal case and timing error due to (inset left bottom) pull-up PMOS sizing, (inset right top) comparator offset, (b) sub-cycle power gating and (c) coarse comparator slew rate

leakage. The leakage places an additional burden on the pull-down device, increasing the time period. More importantly, under slow, low-voltage, low-temperature conditions, a weak pull-up may fail to charge the timing capacitor to full rail. Recall that the sensitivity to VDD is minimized in the proposed design due to the ratio reference. This aspect is defeated if the capacitor fails to charge to full-rail, or overshoots. Figure 6.14a illustrates this problem with the black trace showing the desired waveform-an eye diagram of 24 cycles of $V_C$ at TT, 1.2 V, 25 °C. Traces inset show the slow (blue) and fast (red) cases normalized to their corresponding VDD. The slow case in blue shows $V_C$ barely reaching the full rail, although the fast rising-edge on X3 eventually causes a Miller-coupled overshoot. In contrast, the fast case, in red, overshoots during charging to about 0.5% of VDD, inducing a worst-case error of 0.7% of the time period.

**Sub-cycle power gating:** Power gating X0 and the consequent fast edge at X3 output introduces further timing errors. The impact of power gating is illustrated in Figure 6.14b which shows the behavior of the comparator's internal node 'q1' in relation to the cyclical behavior of $V_C$ (Figure 6.4). The waveforms for both cases—with and without duty cycling—overlap, showing minimal variation. The comparator turn-on introduces a slight static delay (0.5% of the total time period) which can be compensated for in the RC network.

**Comparator offset:** A significant source of timing error is found in the comparator offset, which is exacerbated further by M0 sizing. As shown in Figure 6.14a inset at top right, the worst case is the slow, low-voltage, low-temperature case when $V_C$ has to drop well below the reference voltage of 33.3% of VDD ($V_{REF}$) before the comparator initiates a charge cycle. It is worth noting that the switched capacitor converter ratio reference remains immune to PVT variations. As shown in Figure 6.14c, the error is about 4% of $T_P$. This error imposes a major limitation in line and temperature stability for the proposed relaxation oscillator design. It may be alleviated by chopping [137],




(a) No duty cycling (b) with duty cycling

FIGURE 6.15: Power distribution without and with duty cycling.

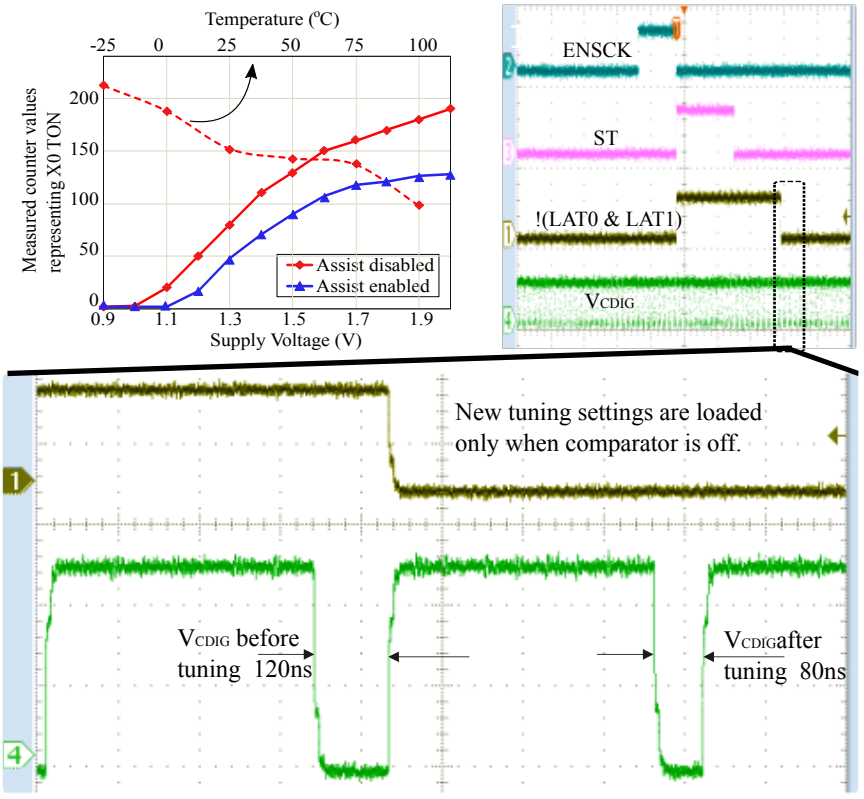



FIGURE 6.16: Oscilloscope waveforms illustrating functioning of digital-assist and the retuning of comparator on-time through tuning of $V_{CDIG}$.

which was not implemented in this design.

**Coarse comparator:** Ideally, X3 would also be a differential comparator, thus minimizing any output-to-input coupling due to parasitic capacitances. However, the proposed design uses a Schmitt-based device which has the Miller-capacitor coupling the fast-slew output back to the input. To minimize the coupling, always-on devices (MC3 and MC4) are introduced in the Schmitt NOR gate (X3 in Figure 6.10). Despite this addition, as illustrated in Figure 6.14c inset, the fast power gating $V_{CDIG}$ introduces a 0.2% timing error.

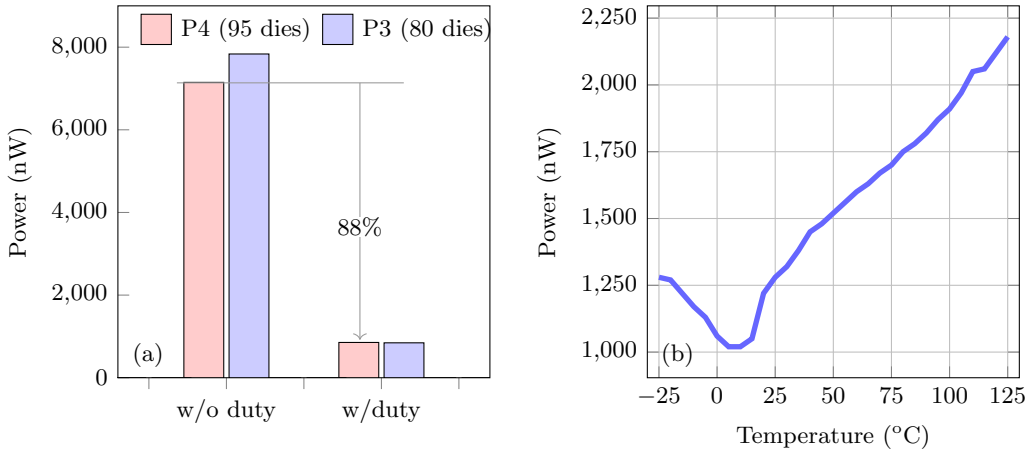



FIGURE 6.17: Measured leakage and active power for two batches of dies without and with duty cycling enabled and nW/kHz with duty cycling enabled.

Figure 6.15 shows the power expended in each sub-block of the relaxation oscillator design (as shown in Figure 6.6) with and without duty cycling. The total continuous power without duty cycling is 4.3 µW, with 76% of that power expended in the comparator, as expected. The RC network is the next most power-hungry block. Duty cycling helps balance this with both the comparator and the RC network expending nearly equal portions of power. The coarse comparator is included in "others" and expends about 16% of the total power. Despite the overheads of the coarse comparator, the power consumption with duty cycling is 62% lower. As shown in the next section, DTUNE bits can be used (as shown in Figure 6.10) to optimize duty cycling further to achieve up to 88% reduction in power.

## 6.4 Measured results

Two batches of dies were fabricated (marked P3 and P4 here, with 80 and 95 dies respectively), and all dies were functional with and without duty cycling and with digital-assist. The functional proof of digital-assist is presented in Figure 6.16 and can be related to the waveforms shown in Figure 6.9. The top-left inset graph shows the counter values without and with digital-assist showing the reduction of $T_{ON}$ when digital-assist is enabled. The variation in duty cycling with temperature indicates that $T_{ON}$ reduces at higher temperatures. When digital-assist is initiated, the rising edge of ENSCK triggers the slow-clock and the state-machine signals along with $V_{CDIG}$ are seen in the top-pane. Note the relative speeds of the oscillator and slow clock. This slowness minimizes the power-overhead of the digital-assist. The bottom pane in Figure 6.16 shows the $V_{CDIG}$ pulse width changing when the new DTUNE settings are latched in. The narrower the pulse, the lower the $T_{ON}$ of the comparator and the better the nW/kHz of the oscillator. The screenshot here shows the $T_{ON}$ reduced from 120 ns to 80 ns. This indicates the comparator is ON only for 10% of the oscillator clock period (1.2 MHz). This result can

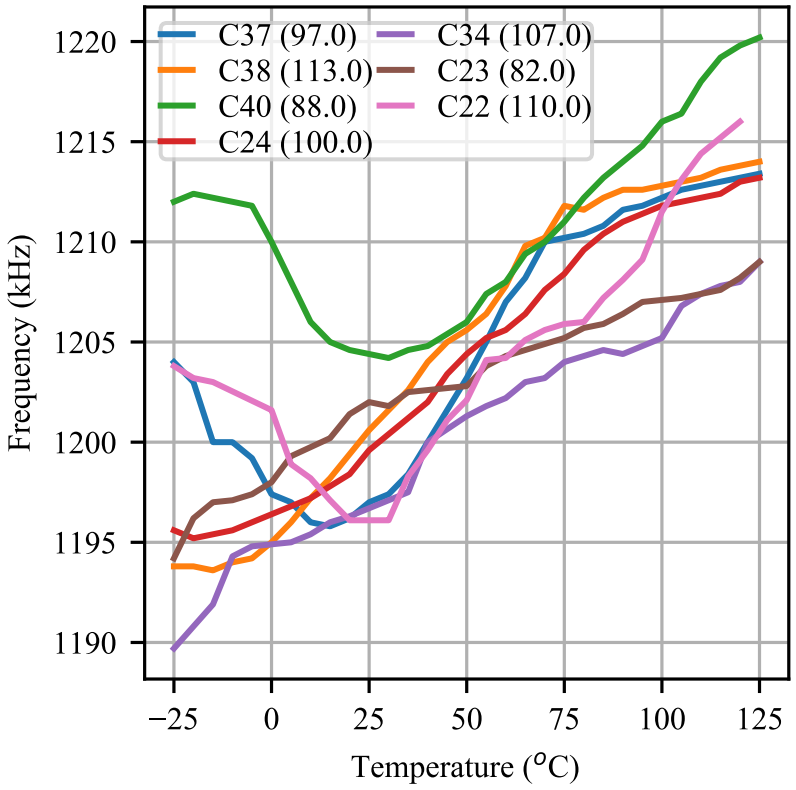



FIGURE 6.18: Measured temperature stability for seven dies.

also be confirmed through power measurements.

The average active power for the two batches of dies were 7.1 and 7.8 µW, respectively, at 0.9 V, 25 °C (Figure 6.17a). With duty cycling enabled, this value drops to 820 nW, an 88% reduction in power, in line with the pulse width reduction seen in Figure 6.16. This reduced power includes the power overheads of the coarse comparator. Figure 6.17b shows active power vs temperature depicting a trend similar to frequency vs temperature as shown in Figure 6.18. Leakage for the two batches was 4-6 nW and the measured FoM was 0.68 nW/kHz. The temperature sensitivity was measured (Figure 6.18) to be approximately 100 ppm/°C for a temperature range of -25 to 125 °C. The calculated ppm/°C is mentioned in the legend in Figure 6.18. No recalibration was applied for temperature stability measurements. Also, tuning (RTUNE) is only applied to poly resistors for setting the nominal frequency while the diffusion resistance remains fixed. This leads to the increased die-to-die temperature variation at lower temperatures. Line sensitivity for a voltage range of 0.9 V-1.8 V of 10 dies was measured to be 0.7 %/V, as illustrated in Figure 6.19.

Long-term jitter was measured using a high-speed oscilloscope (Keysight MSO9254A) to capture the frequency gap-less for 2 s at 20 GSa/s. The captured data points were post-processed to evaluate the Allan deviation. To verify the correctness of this measurement

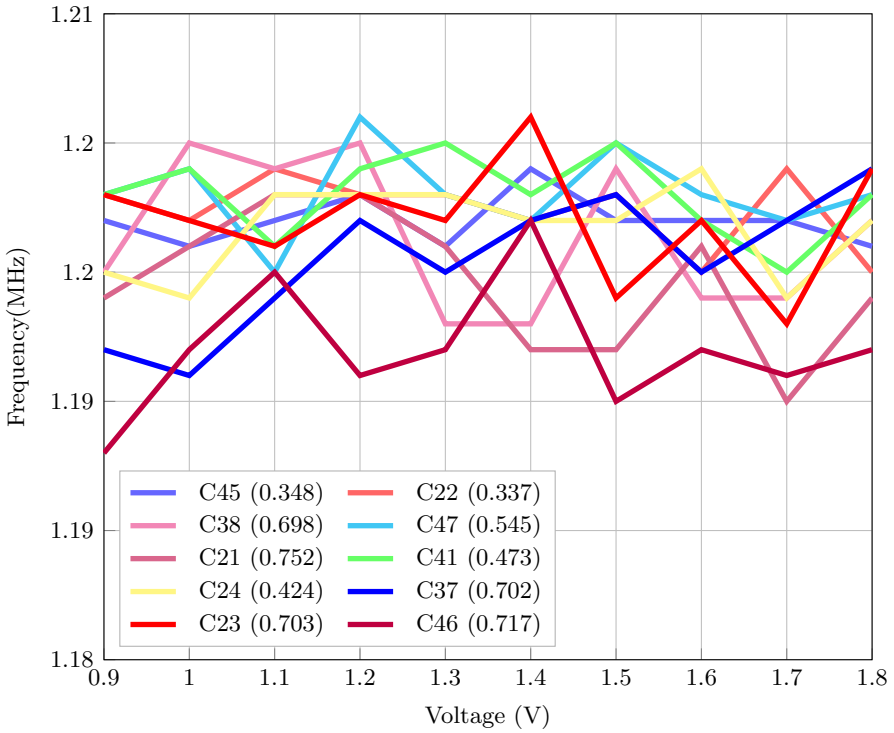



FIGURE 6.19: Measured line stability for 10 dies.

setup, a known 1 ppm external oscillator was tested and the results shown in Figure 6.20. The proposed design shows 10 ppm long-term stability for an averaging time of 1 s. Compared to [138] the proposed design uses low-value resistors which minimizes thermal noise. The small area further reduces noise coupled from the substrate. Further, keeping the comparator off for most of the time period minimizes jitter due to comparator current considering that period fluctuations at the switching instant build up from the onset of the cycle [76]. Quantifying the contribution from each of these design factors in improving long term stability requires deeper analysis.

### 6.4.1 Start-up

Without an external oscillator for the switched capacitor reference, it may appear as though the relaxation oscillator may not start-up. As explained in the following paragraphs in this section, the comparator successfully senses a differential voltage eventually leading to a clean start-up.

Even if absent at power-up, this differential manifests eventually, as follows: in the absence of a clock at start-up, the $V_{REF}$ (Figure 6.5) node is undriven and unloaded while also being capacitively decoupled/filtered with a 2 pF capacitance. So $dv/dt$ on $V_{REF}$ is much slower than the rate of change on node $V_C$, which is either driven or loaded and has a node capacitance (600 fF) that is relatively smaller than the decoupling capacitance on node $V_{REF}$.

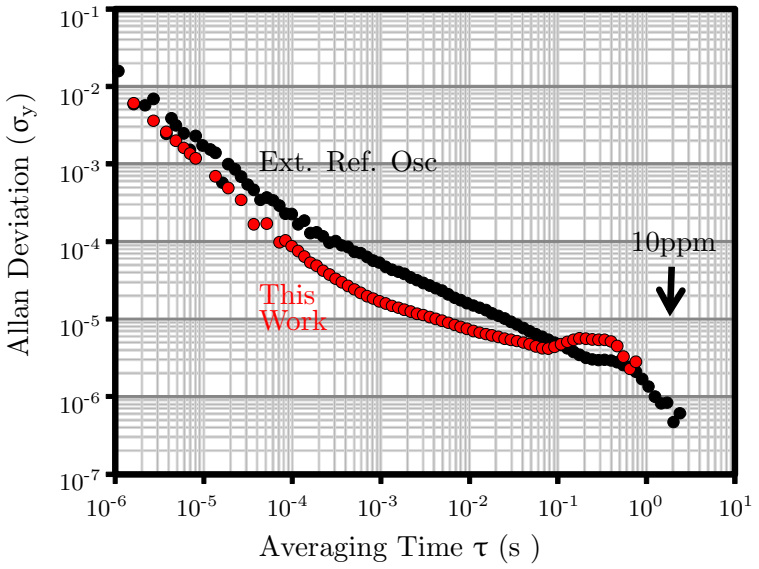



FIGURE 6.20: Measured Allan deviation (in red) in comparison with an external reference oscillator (in black).

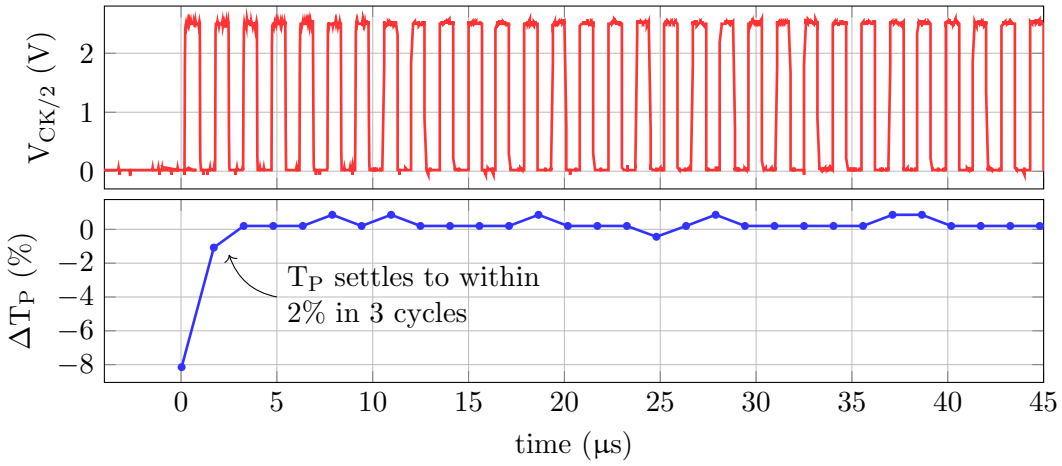



FIGURE 6.21: Measured start-up using switched capacitor reference.

Therefore, one of two scenarios can manifest at start-up: $V_{REF} = V_C$ or $V_{REF} \neq V_C$. For the latter case, if $V_C > V_{REF}$ or $V_C < V_{REF}$, then the comparator initiates a discharge or charge cycle respectively and oscillations will commence. For the rare case in which $V_{REF}=V_C$, a leakage contention or power/substrate noise combined with the relative slowness of $V_{REF}$ will introduce a voltage differential between $V_C$ and $V_{REF}$. Eventually, oscillations will kick in as described for the $V_{REF} \neq V_C$ case. The measured start-up time is reported in Section 6.4. A fail-safe mode would consist of have a power on reset (POR) or a globally enabled (EN) signal with low-leakage thick gate oxide switches which actively starts the $V_{REF}$ and $V_C$ nodes on opposite states.

The start-up of the oscillator was measured to evaluate any potential issues. The oscil-

TABLE 6.1: Comparison of proposed oscillator scheme with state-of-the-art

| Reference | Node (nm) | Power (μW) | Supply (V) | F (MHz) | Area ($mm^2$) | T (ppm/°C) | V (%/V) | Energy (nW/kHz) |
|---|---|---|---|---|---|---|---|---|
| **Denier [136]** | 350 | 0.011 | 1 | 0.003 | 0.1 | 260 | 3.5 | 3.7 |
| **Lasanen [143]** | 350 | 84 | 1.2 | 0.2 | 0.08 | 900 | 0.7 | 420 |
| **Paidimarri [137]** | 65 | 0.12 | 0.95 | 0.0185 | 0.032 | 38 | 5 | 6.5 |
| **Hsiao [139]** | 60 | 4.5 | 1.6 | 0.033 | 0.05 | 32 | 0.06 | 136 |
| **Griffith* [138]** | 65 | 0.19 | 1.2 | 0.033 | 0.05 | 38 | 0.09 | 5.75 |
| **Choe [77]** | 130 | 38 | 1.5 | 3.2 | 0.07 | 125 | 4 | 11.8 |
| **Lee [79]** | 180 | 53 | 1.4 | 4.7 | 0.09 | 42 | 0.1 | 11.3 |
| **Tokairin [140]** | 90 | 0.28 | 0.8 | 0.1 | 0.12 | 105 | 10 | 2.8 |
| **Jang [135]** | 180 | 0.018 | 1 | 0.003 | 0.2 | 14 | 0.48 | 6 |
| **Nadeau [80]** | 180 | 38e-6 | 1.8 | 18e-6 | 0.18 | 8500 | 0.9 | 2 |
| **Sebastiano [142]** | 65 | 40 | 1.2 | 0.11 | 0.1 | 220 | 1.6 | 41 |
| **Sundaresan [144]** | 250 | 1500 | 2.4 | 7 | 1.6 | 315 | 1.77 | 214 |
| **Tokunaga$ [78]** | 180 | 45 | 1.8 | 14 | 0.04 | 23 | 1.6 | 3.2 |
| **Choi [141]** | 180 | 0.11 | 1.3 | 70 | 0.26 | 34.3 | 0.75 | 1.5 |
| **This work** | 65 | 0.82 | 0.9 | 1.2 | 0.005 | 100 | 0.7 | 0.68 |

*Uses external current references.
$Uses external temperature compensation.

lator started up reliably on all 175 devices across all measured conditions. Figure 6.21 shows that the oscillator settles to within 2% of the final frequency within three cycles. The oscillator enable signal $V_{EN}$ and divide- by-2 output $V_{CK/2}$ divide-by-2 output were captured, ensuring the output was within the limits of the IOs used in the SoC. The data was post-processed to obtain the plot of $\Delta T_P$.

Table 6.1 compares the proposed design with state-of-the-art. The proposed design utilizes the least area and has the best FoM for comparable stability.

## 6.5 Summary

This work eliminates the need for fixed references, and minimizes the supply sensitivity using a reference that is a fixed ratio of VDD. Further, the proposed design demonstrates sub-clock duty cycling of the power-hungry comparator, which significantly reduces the energy/cycle. Potential enhancements for future implementations of this design include the addition of a differential coarse comparator. Further, improvements to the line and temperature stability are possible using techniques to minimize errors induced by comparator offset, such as chopping. Faster start-up could be achieved by using a separate slow oscillator, for a marginal increase in energy/cycle.

In conclusion, the design presented here was used to realize a sub-nW/kHz, 1.2 MHz relaxation oscillator in TSMC 65 nm and silicon measurements show 0.7 %/V line sensitivity and 100 ppm/°C temperature stability, while expending 0.68 nW/kHz.

# Chapter 7

# Conclusions

This thesis has addressed aspects of system, circuit and design methodology with the aim of improving the energy efficiency in integrated sensor nodes or sensor SoCs and enabling their EDA friendly implementation (Section 1.3). The circuit and system techniques explored in this thesis have been successfully demonstrated on silicon and shown to allow improvements in integrated sensor node SoCs energy efficiency while following standard EDA flows.

To begin with, in Chapter 3, characterizing and modeling of energy harvesting sources which have a bearing on aspects of SoC design methodology has been addressed. Conventional SoC design methodology relies on relatively more reliable source of power from a battery. The work in this chapter highlighted the need for specific characterization flows when powered by energy harvesting which resulted in SPICE simulation models to enable harvester-circuit co-design.

This work presented variation-aware simulation models developed from field measurements and used to co-design integrated charge-pump circuitry with an analysis of the worst- and best-case conditions. This allowed margining the SoC design appropriately. The results and discussions presented in this chapter indicated that this margining could be as high as 16×, unlike the ±10% which is typical of designs that are constrained by battery power. Using such under-margined designs for energy harvesting can result in unreliable behaviour when deployed in the field.

In Chapter 4, efficiency gains from this co-design methodology was extended to 'Selective Direct Operation' - a technique that exploits sub-threshold system characteristics to achieve a 30% increase in sensor node computation cycles. As the ambient energy falls the energy harvesting output voltage can drop too low making conventional switched capacitor power conversion prohibitively inefficient. This chapter demonstrated utilization of harvested energy without power conversion and the associated losses under low-energy conditions. The proposed methods, with a bidirectional converter of peak efficiency of

88% and a fully integrated programmable clock source, were validated on silicon and shown to provide energy-neutral operation when the sensor node gets exposed to 250 lux (indoor lighting) for as little as 2 hours per day.

The control algorithm for this system was implemented in software running on a separate host processor. While this approach eased iterative testing and validation, the power overheads of a hardware implementation, however minimal, was not accounted and the complexity of bootstrapping the system with such a controller was not addressed. One approach to realise the controller with reduced energy-overheads is by using a finite state-machine implemented using low-leakage thick-gate oxide devices and/or adopt dynamic leakage suppression logic family. This allows speed and area to be sacrificed for energy efficiency.

The subsequent two chapters explored key circuits which help with further enabling autonomous operation of the system. In Chapter 5, a scheme that exploits the state-awareness of the controller was presented which demonstrated a fast but low power voltage monitor. Ultra-efficient sensor systems frequently transition between low power sleep and active modes. Any delay in entry/exit to low power modes costs CPU system power. Therefore, fast voltage monitors are necessary which can mean keeping as many as four power hungry comparators always-on to monitor upper and lower thresholds and to introduce hysteresis during switching. The approach proposed in this chapter allowed the reference voltage to be varied and power gated redundant comparators to minimize power overheads of the voltage monitor.

A 6 µs response time was demonstrated (under nominal conditions) with a minimal power overhead 50 nW. While this scheme has the drawback of being sluggish for fast voltage droops, the response time is suitable for systems powered by small PV cells and TEGs which harvest from slow-changing ambient sources. Also, ultra-efficient CPU systems, as used in Chapter 4 rarely experience frequent sub-µs order voltage droops as would GHz order application processors. For fast applications, the power-up and settling time of the proposed comparator scheme is prohibitively slow.

Along the same topic of assist-circuits for ultra-efficient systems, Chapter 6, explored an aggressive scheme of sub-cycle power gating of the power-hungry comparators in RC relaxation oscillators. RC relaxation oscillators are amongst the most commonly used integrated timing references. The power expended in RC oscillators is overwhelmingly dominated by the reference and the comparator. As part of this research, both these challenges were addressed. A switched capacitor network was used to generate a ratio-reference minimizing related power overheads. For the latter issue, the comparator was duty cycled in each cycle. For implementing this idea, a low power coarse comparator was employed which detects when a comparison is needed and turns on the precision high-power comparator just-in-time. Further stability improvement was shown by using a fully-digital state machine leading to a 70 sq. µm sub-nW/kHz RC oscillator.

An easily correctable but major limitation of this design was the lack of chopping at the comparator leading to a systematic error due to the comparator offset. Another common critique with power cycling the comparator in an RC oscillator is the limitation on long-term stability of oscillator. However, silicon measurements of Allan-deviation showed no significant deviations as the comparator is off for most part of the cycle limiting any source of noise to that of the timing resistor.

As sensor applications demand volumetrically constrained systems with longer unattended operating lifetime, energy harvesting and system energy efficiency become the key metrics when designing such systems. The ideas presented in this work help narrow the design gap between harvester and power conversion circuit and enable autonomous operation of the system by optimizing the CPU system and key analog assist circuits.

## 7.1 Research Objectives

In the context of ultra-efficient sensor systems for IoT, this research work spans varied areas of modelling, design methodology, system and circuits. The corresponding contributions of this research work in the context of the research questions this work set out to explore are revisited below.

**1) How can the real-world performance of microscale PV cells be captured to enable EDA friendly implementation of energy harvesting sensor nodes?**

In this work, a variation-aware model of small-scale PV harvesters has been developed from extensive field measurements. The findings show as much as 16× difference between worst- and best-case designs. Circuit co-design enabled by this model allows designers to be aware of these corner cases and accordingly account for them during design. This work resulted in two conference publications [22] [25]. As a result of this approach, measured results show good bidirectional conversion efficiency (as reported in Section 4.5)

**2) How can the utilization of energy be improved through new system or circuit-level techniques?**

In order to improve utilization of harvested energy, this work proposes power management unit improvements to exploit the low operational voltages of sub-threshold systems, that are known to operate at their minimum energy point. High efficiency power conversion provides limited gains when harvested energy is very low. In such cases, allowing the CPU system to operate directly while also performing MPPT allows up to 30% more computation. This improved utilization means smaller harvesters can be used to guarantee energy neutrality for the same sensor node volume. These techniques and associated results were published in [23] and the ideas were protected in patent [24].

**3) How can the energy demand of sensor nodes be further reduced through optimization of ancillary circuits?**

As part of this work, two critical circuits have been explored that support the power management unit with autonomous operation. The state-aware rail-monitor allows power management unit to support fast mode transitions while saving as much as 23% of the sleep energy (see Section 3.6 and 5.5) for the example CPU system used for this work. The design approach presents better speed vs power balance compared with state-of-the-art. The relaxation oscillator with sub-clock duty cycled comparator presents a new approach to integrated relaxation oscillators demonstrating sub-nW/kHz operation. While the voltage monitor was published at a conference [26], the oscillator was presented at [27] and also published as a journal [28]. Both ideas were protected as patents [30] [29].

## 7.2 Future directions

The key barriers for deploying energy harvested sensors in the context of IoT have been batteries, radio (wireless communications) and non-volatile memory. In this section suggestions are made with regard to directions for further work in this area of research.

**Energy Harvesting:** EH alleviates the dependence on batteries but presents uncertainty in terms of power availability, a challenge not encountered in tethered or battery-powered systems. The temporal and spatial variability of energy harvested from light and temperature differentials was presented in this work. Future efforts along these lines involve having the ultra high frequency (UHF) RF energy harvesting exploit the ever-increasing ambient RF energy [149]. Such efforts can address wireless communication issues in battery-less sensors, a topic which was not addressed as part of this thesis. RF backscatter provides a convenient low-performance, low-power communications channel which may suffice for compute-enabled sensor nodes [150].

A key challenge with implementing the system described in this work directly for RF energy harvesting would be the relative slowness of control algorithms described in Chapter 4 and the voltage monitor due to the relatively fast changes in RF nulls and peaks based on changes in ambient environment.

**Intermittent Compute:** The intermittent availability of power in energy harvesting sensors has led to a new paradigm in computing involving software being regularly [151] or reactively [152] check-pointed so that computations may be resumed when sufficient power becomes available. This ensures progress in computational tasks despite frequent outages of supply power. Despite the lacking non-volatile memory, the system presented in Chapter 4 of this thesis is suitable for research into such compute paradigms. Off-chip non-volatile memory can be used for saving checkpoints with the host controller

taking over the boot procedures for sub-threshold processor. While this approach adds significant time and energy overheads on each power interruption, these are deterministic and can be offset theoretically during the exploration phase. This approach has the merit of allowing different non-volatile memory technologies such as resistive or ferro-electric memories to be tested and aspects such as endurance vs energy trade-offs to be explored.

**Non-volatile memory:** Conventional non-volatile memories are more power hungry because they feature charge pumps to support with high voltage write or erase functions and have a nominal voltage that is relatively higher than the sub-threshold logic system. Relying on non-volatile memory for intermittent computations exacerbates this problem further [153] as the number of access increases with supply interruption frequency. Attempts to minimize energy by optimizing the assist circuits in these sensor nodes can only bring about marginal improvements at the system level. A potential way forward for the research presented here would be to investigate emerging non-volatile memory technologies for suitability to low voltage, low power, ultra-efficient sensor systems. Alternatively, memory bit-cell topologies where area can be traded for low retention voltages can be explored so as to allow reduce frequent access to non-volatile memory.

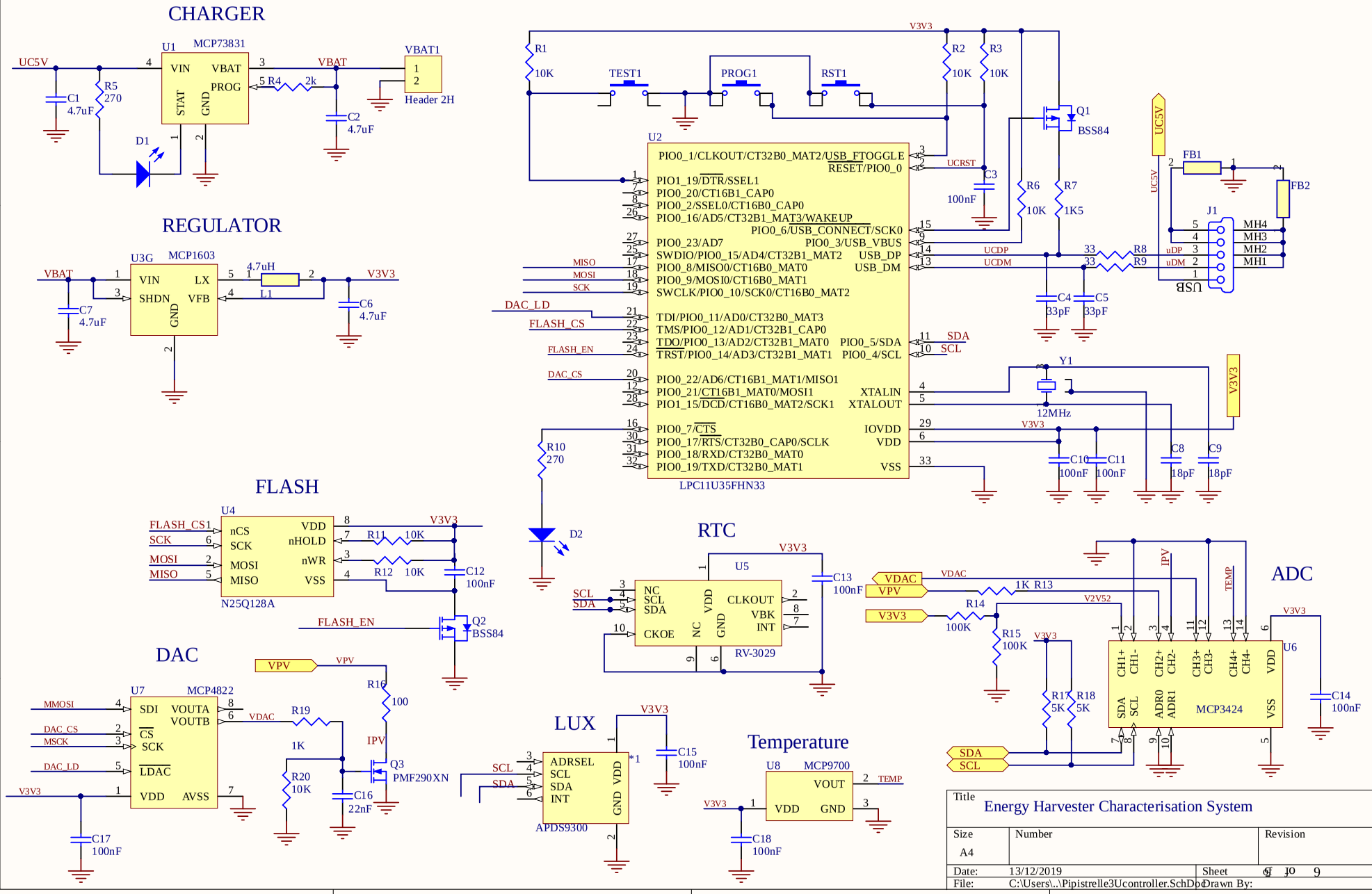
# Appendix A

# Schematic of the characterization system

The schematic for the CS, described in Chapter 3 is illustrated in this appendix. It consists of the following sections:

1. USB-LiPo battery charger (zone A1 in the schematic)
2. 3.3 V Regulator (zone B1 in the schematic)
3. 128-Mb SPI Flash (zone C1 in the schematic)
4. 12-bit digital to analog converter (DAC) (zone D1 in the schematic)
5. Temperature sensor (zone D3 in the schematic)
6. 16-bit broadband and IR spectrum lux sensor (zone D2 in the schematic)
7. Temperature compensated RTC with integrated crystal (zone C3 in the schematic)
8. 12-bit ADC (zone C4 in the schematic)
9. 32-bit MCU with USB interface (zone B3 in the schematic)

The firmware may be compiled using the online mbed compiler and transferred to the MCU using a combination of the PROG1 and RST1 buttons during power-up. This brings up the MCU as a mass storage device on the host machine allowing a drag and drop of the compiled binary. The programmed device shows up as USB Serial Device on the host machine allowing various parameters of the system iPcluding the RTC to be configured. The configured device can then be made to start/stop logging IV data by pressing PROG1 button for two seconds.


CHARGER
U1 MCP73831
UC5V
VIN
VBAT
PROG
STAT
GND
C1 4.7uF
R5 270
R4 2k
C2 4.7uF
D1
VBAT1
Header 2H
REGULATOR
U3G MCP1603
VBAT
VIN
LX
SHDN
VFB
GND
4.7uH
L1
C7 4.7uF
C6 4.7uF
V3V3
FLASH
U4
FLASH_CS1
SCK
MOSI
MISO
nCS
nHOLD
nWR
VDD
VSS
R11 10K
R12 10K
C12 100nF
N25Q128A
FLASH_EN
Q2 BSS84
DAC
U7 MCP4822
MMOSI
DAC_CS
MSCK
DAC_LD
SDI
VOUTA
VOUTB
CS
SCK
LDAC
AVSS
VPV
R16 100
R19 1K
R20 10K
C16 22nF
C17 100nF
Q3 PMF290XN
IPV
VDAC
LUX
U2
ADRSEL
SCL
SDA
INT
APDS9300
C15 100nF
Temperature
U8 MCP9700
VOUT
TEMP
C18 100nF
RTC
U5
NC
CLKOUT
VBK
INT
CKOE
RV-3029
C13 100nF
R10 270
D2
R1 10K
TEST1
PROG1
RST1
R2 10K
R3 10K
Q1 BSS84
U2
PIO0_1/CLKOUT/CT32B0_MAT2/USB_FTOGGLE
RESET/PIO0_0
PIO1_19/DTR/SSEL1
PIO0_20/CT16B1_CAP0
PIO0_2/SSEL0/CT16B0_CAP0
PIO0_16/AD5/CT32B1_MAT3/WAKEUP
PIO0_6/USB_CONNECT/SCK0
PIO0_23/AD7
PIO0_3/USB_VBUS
SWDIO/PIO0_15/AD4/CT32B1_MAT2
USB_DP
PIO0_8/MISO0/CT16B0_MAT0
USB_DM
PIO0_9/MOSI0/CT16B0_MAT1
SWCLK/PIO0_10/SCK0/CT16B0_MAT2
TDI/PIO0_11/AD0/CT32B0_MAT3
TMS/PIO0_12/AD1/CT32B1_CAP0
TDO/PIO0_13/AD2/CT32B1_MAT0
PIO0_5/SDA
TRST/PIO0_14/AD3/CT32B1_MAT1
PIO0_4/SCL
PIO0_22/AD6/CT16B1_MAT1/MISO1
PIO0_21/CT16B1_MAT0/MOSI1
XTALIN
PIO1_15/DCD/CT16B0_MAT2/SCK1
XTALOUT
PIO0_7/CTS
IOVDD
PIO0_17/RTS/CT32B0_CAP0/SCLK
VDD
PIO0_18/RXD/CT32B0_MAT0
PIO0_19/TXD/CT32B0_MAT1
VSS
LPC11U35FHN33
UCRST
UCDP
UCDM
C3 100nF
R6 10K
R7 1K5
R8 33
R9 33
C4 33pF
C5 33pF
MISO
MOSI
SCK
DAC_LD
FLASH_CS
FLASH_EN
DAC_CS
SDA
SCL
Y1
12MHz
C10 100nF
C11 100nF
C8 18pF
C9 18pF
UC5V
FB1
FB2
J1
USB
MH4
MH3
MH2
MH1
uDP
uDM
ADC
U6
MCP3424
CH1+
CH1-
CH2+
CH2-
CH3+
CH3-
CH4+
CH4-
ADR0
ADR1
R13 1K
R14 100K
R15 100K
R17 5K
R18 5K
C14 100nF
V2V52
IPV
TEMP
VDAC
VPV
Title
Energy Harvester Characterisation System
Size
A4
Number
Revision
Date: 13/12/2019
Sheet 9 of 10
File: C:\Users\..\Pipistrelle3Ucontroller.SchDoc
Drawn By:

# Appendix B

# Firmware for the characterization system

The embedded CPP code for the MCU of the CS described in Chapter 3 is provided in this appendix.

```
/*    // mbed DIP Pin Names
    p3 = P0_7,
    p4 = P0_8,
    p5 = P0_9,
    p6 = P0_10,
    p7 = P0_22,
    p8 = P0_11,
    p9 = P0_12,
    p10 = P0_13,
    p11 = P0_14,
    p12 = P0_15,
    p13 = P0_16,
    p14 = P0_23,
    p15 = P1_15,
    p16 = P0_17,
    p17 = P0_18,
    p18 = P0_19,
    p19 = P0_1,
    p20 = P1_19,
    p21 = P0_0,
    p22 = P0_20,
    p23 = P0_2,
    p24 = P0_3,
    p25 = P0_4,
    p26 = P0_5,
    p27 = P0_21,
    p28 = P0_6
    */

#include "mbed.h"
#include "USBSerial.h"
//#include "EEPROM.h"

//DEFINITIONS
#define BUF_SIZE 4096
#define EEP_SIZE 1024  //11u35 has 4096
#define EEP_VID 1025  //eeprom location for storing vid (USB)
#define DAC_MIN     800
```

```
39 #define DAC_MAX     1300
40 #define DAC_CSTEP   50
41 #define DAC_FSTEP   2
42
43 //MBED CLASSES
44 USBSerial serial;
45 EEPROM eeprom;
46 DigitalOut LED(p3);  //0p7
47 InterruptIn SW2(p19); //0p1
48 DigitalIn PB2(p19);     //0p1
49 DigitalOut FLSH_EN(p11); //0p14
50 DigitalOut FLSH_CS(p9); //0p12
51 DigitalOut DAC_CS(p7); //0p22
52 DigitalOut DAC_LDAC(p8); //0p11
53
54 Timeout timeout;  //key press timeout
55 Ticker RunStateLed; //led ticker in run mode
56 Ticker DoRunState; //IV cycle timer
57 I2C i2c(p26, p25); //SDA, SCL
58 SPI spi(p5, p4, p6); // mosi, miso, sclk
59
60 //GLOBAL VARIABLES
61 bool state = 0; //0=off, 1=on
62 bool IVcycle = 0; //0=stop, 1=log data
63 bool Tflag = 0; //time data from UART
64 char meas_data[BUF_SIZE]; //4KB buffer space
65 char usr_key=0;
66 int buffer_count = 0;  //
67 int flash_address = 0x0;
68 int session_byte_count=0;
69 int session_count = 0;
70 char timeData[16];
71 uint16_t vendor_id = 0x1f00; //default
72
73 volatile unsigned int *clkctrl=(volatile unsigned int *)0
       x40048080;
74 volatile unsigned int *iocon=(volatile unsigned int *)0x40044000
       ;
75 volatile unsigned int *ct16b1=(volatile unsigned int *)0
       x40010000;
76 volatile unsigned int *ct16b0=(volatile unsigned int *)0
       x4000C000;
77
78
79 //FUNCTION DECLARATIONS
80 bool verifyFlashPresence(); //SessionCtrl  //FlashCtrl
81 void flashWriteEn();  //FlashCtrl
82 void flashReadyTest();  //FlashCtrl
83 void sectorErase(char);  //FlashCtrl
84 void bulkErase();  //FlashCtrl
85 void flash_write();  //FlashCtrl
86 void bufferInt(int);  //FlashCtrl
87 void flashSave();  //FlashCtrl
88
89 void ControlAmbientLightSensors (bool);  //InstrumentationCtrl
90 int readRTC();  //InstrumentationCtrl
91 float readLightSensor(unsigned char); //InstrumentationCtrl
92 int readADC (char);  //InstrumentationCtrl
93 void dac_write (int);  //InstrumentationCtrl
94 void IVSweep();  //InstrumentationCtrl
95
96 void processUserKey(); //UICtrl
97
98 void startSession(); //SessionCtrl
99 void stopSession(); //SessionCtrl
100 void attimeout(); //SessionCtrl
```

```
101 void ButtonPress(); //SessionCtrl
102 void setTime(); //SessionCtrl
103 void indicateState(bool); //SessionCtrl
104 void RunStateLED(); //SessionCtrl
105 void DORunState() ; //SessionCtrl
106
107 //int read_eeprom( char, char *, int); //eeprom
108 //int write_eeprom( char *, char , int); //eeprom
109
110 USBSerial& ser()
111 {
112     static USBSerial pc(vendor_id,0x2012,0x0001,false);
113     return pc;
114 }
115
116 //ISR
117 void UartRxIrq()
118 {
119     usr_key=ser().getc();
120     //ser().printf("rxd %c\n",usr_key);
121 }
122
123 //MAIN
124 int main(void)
125 {
126
127
128     FLSH_EN=0;
129     FLSH_CS=1;
130     DAC_CS=1;
131     LED=1;
132
133     //USB VID Setting
134     //eeprom.put(0x401,0x1F);
135     //eeprom.put(0x402,0x16);
136     vendor_id = eeprom.get(0x401)<<8 | eeprom.get(0x402);
137     SW2.fall(&ButtonPress);
138     ser().attach(&UartRxIrq); //
139
140 //  setup lux measurement counters
141             iocon[20] |= 0x01;  //enable p0_20 for captures
142             iocon[2] |= 0x02;  //enable p0_2 for captures
143             clkctrl[0] |= 0x180; //below reg dont get written
    correctly without this!!
144             ct16b1[1] |= 0x03; //enable counter b1 and reset
145             ct16b0[1] |= 0x03; //enable counter b0 and reset
146 //            ct16b1[3] |= 0; //divide b1 counter by 0
147 //            ct16b0[3] |= 0; //divide b0 counter by 0
148             ct16b1[1] = 0x01; //enable counter and release reset
149             ct16b0[1] = 0x01; //enable counter and release reset
    , quite likely a 1 cycle latency?
150             ct16b1[10] = 0x01; //capture on rising edge
151             ct16b0[10] = 0x02; //capture on falling edge
152
153     //ser().printf("Starting Flash verification..\n");
154
155     if ( !verifyFlashPresence() ) {
156         ser().printf("Flash verification failed!!");
157         RunStateLed.attach(&RunStateLED,1);  //led blinks every
    1 seconds
158         while(1);
159     }
160
161     while(1) {
162         if(usr_key) processUserKey();
163         if (IVcycle) {
```

```
164             IVSweep();    // reset here. set by sampling timeout
    inDORunState
165             IVcycle = 0;
166         }
167         __WFI();
168     }
169 }
170
171
172 //FUNCTIONS
173
174 void RunStateLED()
175 {
176     LED = 1;
177     wait(0.05);
178     LED = 0;
179     wait(0.05);
180     LED = 1;
181 }
182
183
184 bool verifyFlashPresence()
185 {
186     char data[4] = {0x20, 0xBA, 0x18, 0x10}; // N25Q128 device
    ID register
187     FLSH_EN=1; //enable flash power mosfet
188     wait_ms(5);
189     FLSH_CS=0; //enable chip select
190     spi.write (0x9E); //write read_id command
191     for (int i=0; i<4; i++)
192         if (data[i] != spi.write(0x0)) {
193             FLSH_CS=1; //disable chip select
194             FLSH_EN=0;
195             return 1;
196         }
197     FLSH_CS=1; //disable chip select
198     wait_ms(1);
199     FLSH_EN=0;
200     return 1;
201 }
202
203 void flashWriteEn()
204 {
205     FLSH_CS=0; //enable chip select
206     wait_us(10);
207     spi.write (0x06); //Write Enable
208     FLSH_CS=1; //disable chip select
209     wait_us(10);
210 }
211
212 void flashReadyTest()
213 {
214     FLSH_CS=0; //enable chip select
215     wait_us(1);
216     spi.write (0x70); //Read Flag Status Register
217     while(spi.write(0x0) != 0x80); //wait till previous cycle
    finishes
218     FLSH_CS=1; //disable chip select
219 }
220
221 void indicateState(bool state)
222 {
223     int count;
224     LED=1;
225     wait(0.1);
226     if(state) count=6;
```

```
227     else count=2;
228     while(count--) {
229         LED=!LED;
230         wait(0.2);
231     }
232 }
233
234 void ControlAmbientLightSensors (bool pwr_ctl)
235 {
236     char data[2];
237     //Control Ambient light sensors
238
239     data[0]=0x80;   //command so MSB =1 and address = reg 0
240     data[1]=0x0 + pwr_ctl  + pwr_ctl  + pwr_ctl ;   //write 3
        for power up
241
242     i2c.write(0x72, data, 2);//APDS9300 IC5
243     i2c.write(0x92, data, 2);//TSL2561 IC10
244 }
245
246
247 int readRTC()
248 {
249     char data[8];
250     int dev_time = 0;
251     data[0]=0x01;    //Start reg Address
252     i2c.write(0xD0, data, 1); //RTC address = 0xD0
253     i2c.read(0xD0, data, 7); //read S,M,H,D,Dt,M,Y
254     for(int i=0; i<7; i++) {
255         if(i==1) data[i] = data[i] & 0x7F;
256         if(i==3) data[i] = data[i] & 0x0F;
257         data[i] = (((data[i] & 0xF0) >> 4 ) & 0x0F) * 10 + (data
        [i] & 0x0F); //BCD to int conversion
258     }
259     //         date, month + year, hour, min
260     dev_time = data[4] << 24 | data[5] << 20 | (data[6] - 14) <<
         16 | data[2] << 8 | data [1];
261     ser().printf("%0.2d/%0.2d/%0.2d %d %0.2d:%0.2d:%0.2d - 0x%08
        X\n\r",data[4],data[5],data[6],data[3],data[2],data[1],data
        [0],dev_time);
262     return dev_time;
263 }
264
265 float readLightSensor(unsigned char address)
266 {
267     int temp[3];
268     float sum=0;
269     int count=100;
270
271     while(count){
272         count--;
273         temp[2]=0;
274             while(temp[2]<=0){
275                 temp[0]=ct16b1[11];  //rising edge
276                 temp[1]=ct16b0[11];  //falling edge
277                 temp[2]=temp[1]-temp[0];    //delta
278                 if(temp[2]>0x8000) temp[2]=0; // remove roll-
        over error
279             }
280         sum += temp[2];
281         }
282         sum=sum/50;
283
284     return (48000/sum);
285 /*    ControlAmbientLightSensors(1);
286     wait(0.5); // wait for settling time
```

```
287     data[0]=0xAC; //channel 0
288     i2c.write(address, data, 1);
289     wait(0.5); // wait for integration time
290     i2c.read(address, data, 2);
291     Light = (data[1] << 24) + (data[0] << 16);
292     data[0]=0xAE; //channel 1
293     i2c.write(address, data, 1);
294     wait(0.5); // wait for integration time
295     i2c.read(address, data, 2);
296     Light |= (data[1] << 8) + data[0];
297     ControlAmbientLightSensors(0);
298
299     return Light;*/
300 }
301
302 void flash_write()
303 {
304     int i,count=0;
305     //int page_size;
306     //char temp_count;
307
308     FLSH_EN=1;
309
310     readRTC();
311     //ser().printf("0x%0.6X\n\r",flash_address);
312
313     while (count < buffer_count) {
314         flashWriteEn();
315         flashReadyTest();
316         flashReadyTest();
317
318         FLSH_CS=0; //enable chip select
319         wait_us(10);
320         spi.write (0x02); //Page Program
321         spi.write ( (flash_address & 0xFF0000 ) >> 16); //
    Address MS
322         spi.write ((flash_address & 0xFF00 ) >> 8); //Address
    Page
323         spi.write (flash_address & 0xFF); //Adress byte
324         //fill any left over with zeros
325         if ((buffer_count - count) < 256) for(i=(buffer_count-
    count); i<256; i++) meas_data[i]=0x0;
326
327         for (i=0; i<256; i++) spi.write(meas_data[count++]);  //
    16 * 256 data bytes to fill a sub-sector
328         FLSH_CS=1; //disable chip select
329         wait_us(10);
330         flashReadyTest();
331         flashReadyTest();
332         flash_address += 0x100;
333     }
334     session_byte_count += count; // but count is kept clean
335     FLSH_EN=0;
336     buffer_count=0;
337     readRTC();
338 }
339
340 void bufferInt(int data)
341 {
342     char indx=4; //4 bytes
343     //take int and write 4 chars
344     while (buffer_count < BUF_SIZE && indx) meas_data[
    buffer_count++] = (data >> (8* (--indx)) ) & 0xFF;
345     //ser().printf("buffer_count = %d\n\r",buffer_count);
346     if (buffer_count >= BUF_SIZE) {
347         ser().printf("Writing to Flash\n\r");
```

```
348         flash_write();
349         //buffer_count=0;
350         //while (buffer_count < BUF_SIZE && indx) meas_data[
        buffer_count++] = (data >> (8* (--indx)) ) & 0xFF; //repeat
        to finish any left over data bytes
351     }
352 }
353
354 int readADC (char channel)
355 {
356     char data[3];
357     int ADC=0;
358
359     channel-=1;
360     data[0]= 0x8C | ((channel & 0x03) << 5);
361     if (channel == 1) data[0] |= 0x03; //set gain to 8 if
        current measurement
362     i2c.write(0xDC, data, 1);
363     wait(0.4);
364     i2c.read(0xDC, data, 3);
365     ADC=(data[0] << 16) + (data[1]<<8) + data[2];
366     if (ADC & 0x800000) ADC = 0;
367     else ADC &= 0x3FFFF ;
368     return ADC;
369 }
370
371 void dac_write (int data)
372 {
373     char val;
374
375     if(data>10) data = data -3;
376     data = data & 0xFFF; //12bit data
377
378
379     DAC_CS=1;
380     DAC_LDAC=1;
381     wait(0.2);
382     DAC_CS=0;
383     wait(0.1);
384     val = 0x90 | ((data & 0xF00)>>8) ;
385     spi.write(val);
386     val = data & 0xFF;
387     spi.write(val);
388     DAC_CS=1;
389     wait(0.1);
390
391     DAC_LDAC=0;
392     wait(0.1);
393     DAC_LDAC=1;
394 }
395
396 void DORunState()
397 {
398     IVcycle = 1; //will terminate after one IV cycle and repeats
        again on timeout. Adjust timing carefully.
399 }
400
401 void flashSave()
402 {
403     int i,eeaddr=0;
404     eeaddr= eeprom.get(0); //session count
405
406     session_count++;
407     eeprom.put(0,session_count) ; //overwrite session count
408     for(i=1; i<4; i++) eeprom.put(i,(flash_address >> (8*(3-i))
        & 0xFF)); //overwrite last flash address
```

```
409
410     eeaddr = (eeaddr*3)+4; //point to last session_byte_count
411     for(i=0; i<3; i++) eeprom.put(eeaddr+i,(session_byte_count
        >> (8*(2-i)) & 0xFF)); //update session_byte_count
412     eeaddr += 3; //new data size to be written
413     session_byte_count=0;
414     buffer_count=0;
415 }
416
417 void startSession()
418 {
419     session_byte_count = 0;
420     buffer_count=0;
421     session_count = eeprom.get(0);
422     flash_address = (eeprom.get(1) << 16 | eeprom.get(2) << 8 |
        eeprom.get(3)) & 0xFFFFFF;
423
424     RunStateLed.attach(&RunStateLED,5);  //led blinks every 5
        seconds
425     DoRunState.attach(&DORunState,120);  //IV sweep every two
        minutes
426     IVcycle = 1; // and start one immediately
427 }
428
429 void stopSession()
430 {
431     RunStateLed.detach(); //
432     DoRunState.detach(); //
433     LED = 0;
434     if (buffer_count) flash_write();
435     LED = 1;
436     flashSave();
437 }
438
439 void attimeout()
440 {
441     if (!PB2) {  //If Button is still pressed after 3 seconds
442
443         state = !state;
444         if (state) startSession();
445         else stopSession();
446         indicateState(state);
447         ControlAmbientLightSensors(state); // sensor power
        control
448     }
449 }
450
451 void ButtonPress()
452 {
453     timeout.attach(&attimeout, 3);
454 }
455
456 void setTime()
457 {
458     char byteCount=0;
459     char data[8] = {0x01,0x0,0x40,0x23,0x1,0x25,0x05,0x14}; //
        data[0] = reg_address
460
461     while(byteCount < 13) {
462         timeData[byteCount] -= 0x30;
463         byteCount++;
464     }
465     data[1] = (timeData[0] & 0x0F) << 4 | (timeData[1] & 0x0F);
466     data[2] = (timeData[2] & 0x0F) << 4 | (timeData[3] & 0x0F);
467     data[3] = (timeData[4] & 0x0F) << 4 | (timeData[5] & 0x0F);
468     data[5] = (timeData[7] & 0x0F) << 4 | (timeData[8] & 0x0F);
```

```
469     data[6] = (timeData[9] & 0x0F) << 4 | (timeData[10] & 0x0F);
470     data[7] = (timeData[11] & 0x0F) << 4 | (timeData[12] & 0x0F)
        ;
471     data[4] = timeData[6] + 1;  //python to RTC day numbering
472
473     i2c.write(0xD0, data, 8);  //Dont do this every time. RTC
        keeps time.
474     readRTC();
475 }
476
477 void sectorErase(char sector)
478 {
479     FLSH_EN=1;
480     wait_ms(10);
481     flashWriteEn();
482     flashReadyTest();
483     flashReadyTest();
484
485     FLSH_CS=0; //enable chip select
486     spi.write (0xD8); //sector erase
487     spi.write (sector);
488     spi.write (0x00);
489     spi.write (0x00);
490     FLSH_CS=1; //disable chip select
491     wait_ms(1);
492     flashReadyTest();
493     flashReadyTest();
494
495     FLSH_EN=0;
496 }
497
498 void bulkErase()
499 {
500     FLSH_EN=1;
501     wait_ms(10);
502     flashWriteEn();
503     flashReadyTest();
504     flashReadyTest();
505
506     FLSH_CS=0; //enable chip select
507     spi.write (0xC7); //Chip erase
508     FLSH_CS=1; //disable chip select
509     wait_ms(1);
510     flashReadyTest();
511     flashReadyTest();
512
513     FLSH_EN=0;
514 }
515
516 void IVSweep()
517 {
518     int dac,write_val;
519     unsigned int i;
520     int Voc;
521
522     //ser().printf("%d\n\r",buffer_count);
523     i=readRTC();
524     bufferInt(i);           //4 bytes
525     ser().printf("RTC 0x%X\n\r",i);
526     i=readLightSensor(0x72);
527     bufferInt(i);  //APDS IC5 - 8 bytes
528     ser().printf("LUX 0x%X\n\r",i);
529     i=readLightSensor(0x92);
530     bufferInt(i); //TSL IC10 -12 bytes
531     ser().printf("LUX 0x%X\n\r",i);
532
```

```
533     i = 0;
534     i = (int) ((((float) readADC(4) *15.625e-6) - 0.525) / 0.01
        ); //temperature from ADC channel 4
535     i &= 0x0000FFFF;
536     i |= ((int) ((float) readADC(1) * 20 * 15.625e-6) << 16 );//
        Vbat in mV from ADC channel 0
537     ser().printf("VBT 0x%X\n\r",i);
538     bufferInt(i); //16 bytes
539     //dummy
540 //          for (i=0;i<50;i++)  bufferInt(0x55555555);
541
542     for (dac = DAC_MIN; dac < DAC_MAX; dac += DAC_FSTEP) {
543         write_val=0;
544         dac_write(dac);
545         i=readADC(2); //ADC Channel 2 - PV Current
546         write_val = ( (int) (1e6 * (((float) i * 15.625e-6) /
        800)) & 0xFFFF ) << 16;
547         i=readADC(3); //ADC Channel 3 - PV Voltage
548         Voc = ( (int) ((float) i * 15.625e-6 * 1e3) & 0xFFFF);
549         //if (dac==1000 && Voc < 100 && VocError < 5) VocError
        ++; //check for 10min
550         //if (dac==1000 && Voc > 100 && VocError > 0) VocError =
         0;
551         write_val |=  Voc;
552         bufferInt(write_val); //IV Sweep - 200 bytes
553         //if ((write_val & 0xFFFF) < 50) dac=1200;  //stop of
        voltage drops below 50mV
554     }
555     dac_write(0);   // turn off DAC at the end
556 }
557
558 void IVSweepInteractive()
559 {
560     int dac,dac_step;
561     unsigned int i;
562     float Voc;
563
564     //ser().printf("%d\n\r",buffer_count);
565     i=readRTC();
566     ser().printf("RTC 0x%X\n\r",i);
567     i=readLightSensor(0x72);
568     ser().printf("LUX 0x%X\n\r",i);
569     i=readLightSensor(0x92);
570     ser().printf("LUX 0x%X\n\r",i);
571     i = (int) ((((float) readADC(4) *15.625e-6) - 0.525) / 0.01
        ); //temperature from ADC channel 4
572     ser().printf("TEMP %0.1d\n\r",i);
573     i = (int) ((float) readADC(1) * 2 * 15.625e-6);//Vbat in mV
        from ADC channel 0
574     ser().printf("VBAT %0.1f\n\r",i);
575     for (dac = DAC_MIN; dac < DAC_MAX; dac += DAC_FSTEP) {
576         dac_write(dac);
577         i = (1e6 * (float) readADC(2) * 15.625e-6)/800; //ADC
        Channel 2 - IpV
578         Voc = (float) readADC(3) * 15.625e-6; //ADC Channel 3 -
        VpV
579         ser().printf("IV %0.3f %d\n\r",Voc,i);
580         if(Voc<0.01) break;
581     }
582     dac_write(0);   // turn off DAC at the end*/
583     ser().printf("DONE\n");
584 }
585
586
587 void processUserKey()
588 {
```

```
589     int i,j,k,word,indx;
590     static char byteCount;
591
592     /* used keys t, T, v, V, f, e, E, r, i, q, s, d, D, p, l, n,
        R, m    */
593     switch(usr_key) {
594         case 'h' :
595             ser().printf("b=read Vbat\n\r");
596             ser().printf("d=run dac coarse routine\n\r");
597             ser().printf("D=run dac fine routine\n\r");
598             ser().printf("E=erase EEprom\n\r");
599             ser().printf("e=erase falsh\n\r");
600             ser().printf("h=This help info\n\r");
601             ser().printf("i=read info\n\r");
602             ser().printf("l=report Lux values\n\r");
603             ser().printf("n=lines per sweep\n\r");
604             ser().printf("q=quick test logging\n\r");
605             ser().printf("r=read data\n\r");
606             ser().printf("R=Read EEprom data\n\r");
607             ser().printf("s=Sector erase\n\r");
608             ser().printf("t=read time\n\r");
609             ser().printf("T=set Time\n\r");
610             ser().printf("v=read Vpv\n\r");
611             ser().printf("V=Verify flash erase\n\r");
612             ser().printf("m=Get Reg\n\r");
613             ser().printf("\nEND OF HELP\n\r");
614             break;
615
616         case 'n' :
617             vendor_id = eeprom.get(0x401)<<8 | eeprom.get(0x402)
    ;
618             ser().printf("%d ID=0x%0.4X\n\r",((DAC_MAX-DAC_MIN)/
    DAC_FSTEP)+4,vendor_id);
619             break;
620
621         case 'R' :  //read eeprom contents
622             for(k=0;k<16;k++) {
623                 ser().printf("%0.2d\n\r",k);
624                 for(i=0;i<16;i++) {
625                     for(j=0;j<16;j++) ser().printf("%0.2X ",
    eeprom.get(j+i*16+k*256));
626                     ser().printf("\n\r");
627                     }
628                 ser().printf("\n\r");
629             }
630             usr_key=0;
631             break;
632
633         case 't' :
634             readRTC();
635             break;
636
637         case 'T' :
638             Tflag=1;
639             break;
640
641         case 'v' :
642             ser().printf("%0.3f\n\r",(float) readADC(3) * 15.625
    e-6); //ADC Channel 3 - VpV
643             break;
644
645         case 'b' :
646             ser().printf("%0.1f\n\r",(float) readADC(1) * 2 *
    15.625e-6); //ADC Channel 2 - Vbat
647             break;
648
```

```
649         case 'l' :
650             ser().printf("LUX= %d\n\r",8*(int)(readLightSensor(0
        x72)));
651             break;
652
653         case 'V' : //verify Flash erase
654             FLSH_EN=1;
655             wait_ms(1);
656
657             FLSH_CS=0; //enable chip select
658             wait_ms(1);
659             spi.write (0x03); //Read data Bytes
660             spi.write (0x00); //Address MS
661             spi.write (0x00); //Address Page
662             spi.write (0x00); //Adress bytei
663             for (i=0; i<0xFFFFFF; i++) if(spi.write(0x00) != 0
        xFF) {
664                     ser().printf("Flash erase failed first at 0x
        %X\n\r",i);
665                     i=0xFFFFFF;
666                 }
667             readRTC();
668             FLSH_CS=1; //disable chip select
669             FLSH_EN=0;
670             wait_ms(1);
671             break;
672
673         case 'f' :
674             ser().printf("%0.2f\n\r",(float) readADC(4) * 15.625
        e-6); //ADC Channel 4 - Temperature
675             break;
676
677         case 'e' :
678             readRTC();
679             bulkErase();
680             readRTC();
681             break;
682
683         case 'p' :
684                 IVSweepInteractive();
685             break;
686
687         case 's' :  //sector erase
688             readRTC();
689             flash_address = (eeprom.get(1) << 16 | eeprom.get(2)
         << 8 | eeprom.get(3)) & 0xFFFFFF;
690             for (i=0; i <= ((flash_address & 0xFF0000) >> 16 ) ;
         i++ ) sectorErase(i);
691             readRTC();
692             break;
693
694         case 'E' :
695             for (i=0; i<1024; i++) meas_data[i]=0;
696             eeprom.write(0,meas_data,1024);//erase eeprom
        contents too
697             readRTC();
698             break;
699
700         case 'r' :
701             FLSH_EN=1;
702             wait_ms(1);
703
704             FLSH_CS=0; //enable chip select
705             wait_ms(1);
706             flash_address = (eeprom.get(1) << 16 | eeprom.get(2)
         << 8 | eeprom.get(3)) & 0xFFFFFF;
```

```
707             spi.write (0x03); //Read data Bytes
708             spi.write (0x00); //Address MS
709             spi.write (0x00); //Address Page
710             spi.write (0x00); //Adress byte
711             for (i=0; i<(flash_address / 4); i++) {//4 reads per
        line printed
712                 indx=4;
713                 word=0;
714                 while(indx) word |= (spi.write(0x00) << (8 * (--
        indx)));
715                 ser().printf("0x%0.8X\n\r",word);  //4 data
        bytes
716             }
717             FLSH_CS=1;
718             wait_us(10);
719             FLSH_EN=0;
720             wait_ms(1);
721             break;
722
723         /*case 'R' :  //What is this??
724             read_eeprom(0,eedata,4);
725             session_count = eedata[0];
726             for(i=1; i<session_count; i++) {
727                 char addr = 3*(i-1)+1;
728                 flash_address = (eedata[addr] << 16 | eedata[
        addr+1] << 8 | eedata[addr+2]) & 0xFFFFFF;
729                 ser().printf("Session %d - Address 0x%0.8X\n\r",
        i,flash_address);  //4 data bytes
730             }
731             break;*/
732
733         case 'i' :
734             word = eeprom.get(0);
735             ser().printf("Session count: %d\n\r",word);
736             ser().printf("Flash address: 0x%0.8X\n\r",((eeprom.
        get(1) << 16 | eeprom.get(2) << 8 | eeprom.get(3)) & 0xFFFFFF
        ));
737             if(word > 0) {
738                 k=0;
739                 //leave the first four
740                 for(i=4; i<((word*3)+4); i=i+3)
741                     ser().printf("Session %d : %d bytes\n\r",k
        ++,
742                         (int) (eeprom.get(i) << 16 | eeprom.get(
        i+1) << 8 | eeprom.get(i+2)));
743             }
744             break;
745
746         case 'd' :
747             for (int dac = DAC_MIN; dac < DAC_MAX; dac +=
        DAC_CSTEP) {
748                 dac_write(dac);
749                 i=readADC(2); //ADC Channel 2
750                 ser().printf("%d %0.0f uA ",dac, 1e6 * ((float)
        i * 15.625e-6 ) / 800 );
751                 i=readADC(3); //ADC Channel 3
752                 ser().printf("%0.3f V\n\r",(float) i * 15.625e
        -6);
753             }
754             dac_write(0);   // turn off DAC at the end
755             ser().printf("\n\n\r");
756             break;
757
758         case 'D' :
759             for (int dac = DAC_MIN; dac < DAC_MAX; dac +=
        DAC_FSTEP) {
```

```
760                 dac_write(dac);
761                 i=readADC(2); //ADC Channel 2
762                 ser().printf("%d %0.0f uA ",dac, 1e6 * ((float)
    i * 15.625e-6 ) / 800 );
763                 i=readADC(3); //ADC Channel 3
764                 ser().printf("%0.3f V\n\r",(float) i * 15.625e
    -6);
765             }
766             dac_write(0);   // turn off DAC at the end
767             ser().printf("\n\n\r");
768             break;
769
770
771         case 'q' :
772             session_byte_count = 0;
773             session_count = eeprom.get(0);
774             flash_address = ((eeprom.get(1) << 16 | eeprom.get
    (2) << 8 | eeprom.get(3)) & 0xFFFFFF);
775             //fill up 4096 bytes
776             bufferInt(readRTC());
777             for (i=0; i<256; i++)  bufferInt(0x55555555);
778             bufferInt(readRTC());
779             for (i=0; i<256; i++)  bufferInt(0xAAAAAAAA);
780             bufferInt(readRTC());
781             for (i=0; i<256; i++)  bufferInt(0xCCCCCCCC);
782             bufferInt(readRTC());
783             for (i=0; i<256; i++)  bufferInt(0x33333333);
784             //bufferint will auto save to flash when full
785             //below flashsave is for flash state eeprom write
786             flashSave();
787             break;
788
789         default :
790             if (Tflag) {
791                 timeData[byteCount++]=usr_key;
792                 if (byteCount ==13) {
793                     setTime();
794                     byteCount=0;
795                     Tflag=0;
796                 }
797             } else ser().printf("%c not recognised!\n\r",usr_key
    );
798             break;
799     }
800     usr_key=0;
801 }
```

# Appendix C

# Python script for interfacing with characterization system

The Python script provided in this appendix allows the host machine to interact with the programmed CS described in Chapter 3 .

```
1  import time
2  from time import clock
3  import datetime
4  import os
5
6  import struct
7  import serial
8  import getopt
9  import time
10 import re
11 import threading
12 import msvcrt
13 import math
14 import winsound
15 import numpy as np
16 import matplotlib.pyplot as plt
17
18
19 import sys
20 from serial.serialutil import SerialException
21
22
23 import _winreg as winreg
24 import itertools
25 import re
26 import optparse
27
28 import serial.tools.list_ports
29
30 params = {'mathtext.default': 'regular' }
31 plt.rcParams.update(params)
32 #plt.rc('font',family='Times New Roman')
33
34 parser = optparse.OptionParser()
35 parser.add_option('-c', '--COM', action="store", type="int",
      help="COM port ID", dest="comPort", default="0",)
36 parser.add_option('-o', '--Options', action="store", type="
      string", help="command option (type '-o h' to see full list)"
```

```
    , dest="getData", default="t",)
#deprecated
parser.add_option('-m', '--Mode', action="store", type="string",
     help="mode: r=raw, f=full", dest="readMode", default="r",)

options, args = parser.parse_args()


COM = 'COM' + str(options.comPort);
todo=options.getData;
Mode=options.readMode;

if (options.comPort==0):    ##no com port specified
    for i in range(0 , len(list (serial.tools.list_ports.
    comports()))):
        print (list (serial.tools.list_ports.comports())[i][0])
    exit();


LinesPerIVSweep = 54;

def roundup(x):
    return int(math.ceil(x / 10.0)) * 10

def plotTEG():
    fname='IVLogger_252 ID=_170349_04042017_TEG_raw.dat'
    #fname='IVLogger_252 ID=_113807_07042017_TEG_raw.dat'
    datFile = open(fname,'r');
    voc=[]
    tArr=[]
    start=0
    pmpp_l=[]
    pmpp=[]
    vmpp=[]
    vmpp_5uw=[]
    vmpp_l=[]
    pmpp.append(0)
    vmpp.append(0)
    vmpp_5uw.append(0)
    for line in datFile:
        Volt=str.split(line,',')
        m=re.search('2017',line)
        if(m):
            start=1
            if(len(pmpp_l)>2):
                pmpp.append(max(pmpp_l))
                count=0
                for i in range(len(pmpp_l)):
                    if(pmpp_l[i]==max(pmpp_l)): max_index=i
                    if(pmpp_l[i]>4.9 and pmpp_l[i]<5.1 and count
    <1):
                        count+=1
                        index5u=i
                vmpp.append(vmpp_l[i])
                #vmpp_5uw.append(vmpp_l[index5u])
                pmpp_l=[]
                vmpp_l=[]

            mydate=line[:14]
        v=0
        try:
            v=int(Volt[0])
        except:
            pass

        if(v>5):
```

```
99              pmpp_l.append(float(str.split(Volt[2])[0]))
100             vmpp_l.append(float(str.split(Volt[0])[0]))
101
102         if(v>5 and start):
103             voc.append(v)
104             tArr.append(mydate)
105             start=0
106
107     datFile.close()
108     plot=1;
109     print len(voc), len(pmpp)
110     if(plot):
111
112         fig,axes=plt.subplots(nrows=2, ncols=1);
113         fig.set_size_inches(8, 16)
114         ax=axes[1]
115         ax2 = ax.twinx()
116         ax.set_ylabel('V$_{MPP}$ (mV)',fontsize=20,color='r')
117         ax.set_xlabel('Time (Hours)',fontsize=20)
118         ax2.set_ylabel('Power ($\mu$W)',fontsize=20,color='b')
119         xaxs=range(len(voc))
120         xaxsLbl=[]
121         print tArr[0]
122         #ax.annotate(tArr[0], xy=(0,0), xytext=(0.1,10),fontsize
        =20)
123         #ax.annotate('V$_{OC}$', xy=(0,0), xytext=(1,1400),color
        ='r',fontsize=20)
124         ax.annotate('$\Delta$T = 4$^o$C', xy=(0,0), xytext
        =(0.2,200),color='black',fontsize=20)
125         #ax.annotate('V$_{5\mu W}$', xy=(0,0), xytext=(2,170),
        color='r',fontsize=20)
126         for tick in ax.yaxis.get_major_ticks():
127                 tick.label.set_fontsize(20)
128         for tick in ax2.yaxis.get_major_ticks():
129                 tick.label2.set_fontsize(20)
130         for tick in ax.xaxis.get_major_ticks():
131                 tick.label.set_fontsize(20)
132         for xa in xaxs:
133             xaxsLbl.append(str(xa/20.0))
134         #ax.plot(xaxsLbl,voc, color='r');
135         #ax.plot(xaxsLbl,voc, marker='s',color='r');
136         ax.plot(xaxsLbl,vmpp, marker='d',color='r');
137         #ax.plot(xaxsLbl,vmpp_5uw,color='r');
138         ax.tick_params('y', colors='r')
139         ax.set_ylim([0,300])
140         ax2.plot(xaxsLbl,pmpp, marker='o',color='b');
141         ax2.tick_params('y', colors='b')
142         ax2.set_ylim([0,120])
143         #ax.set_title('$\Delta$T = 4$^o$C',fontsize=20)
144         #plt.xticks(xaxs, newTarr, rotation='vertical')
145
146
147     #fname='IVLogger_252 ID=_170349_04042017_TEG_raw.dat'
148     fname='IVLogger_252 ID=_113807_07042017_TEG_raw.dat'
149     datFile = open(fname,'r');
150     voc=[]
151     tArr=[]
152     start=0
153     pmpp_l=[]
154     pmpp=[]
155     vmpp=[]
156     vmpp_5uw=[]
157     vmpp_l=[]
158     pmpp.append(0)
159     vmpp.append(0)
160     vmpp_5uw.append(0)
```

```
161     for line in datFile:
162         Volt=str.split(line,',')
163         m=re.search('2017',line)
164         if(m):
165             start=1
166             if(len(pmpp_l)>2):
167                 pmpp.append(max(pmpp_l))
168                 count=0
169                 for i in range(len(pmpp_l)):
170                     if(pmpp_l[i]==max(pmpp_l)): max_index=i
171                     if(pmpp_l[i]>4.9 and pmpp_l[i]<5.1 and count
    <1):
172                         count+=1
173                         index5u=i
174                 vmpp.append(vmpp_l[i])
175                 #vmpp_5uw.append(vmpp_l[index5u])
176                 pmpp_l=[]
177                 vmpp_l=[]
178 
179             mydate=line[:14]
180         v=0
181         try:
182             v=int(Volt[0])
183         except:
184             pass
185 
186         if(v>5):
187             pmpp_l.append(float(str.split(Volt[2])[0])/1000)
188             vmpp_l.append(float(str.split(Volt[0])[0])/1000)
189 
190         if(v>5 and start):
191             voc.append(v)
192             tArr.append(mydate)
193             start=0
194 
195     datFile.close()
196     ax=axes[0]
197     ax2 = ax.twinx()
198     ax.set_ylabel('V$_{MPP}$ (V)',fontsize=20,color='r')
199     #ax.set_xlabel('Time (Hours)',fontsize=20)
200     ax2.set_ylabel('Power (mW)',fontsize=20,color='b')
201     xaxs=range(len(voc))
202     xaxsLbl=[]
203     print tArr[0]
204     #ax.annotate(tArr[0], xy=(0,0), xytext=(0.1,10),fontsize=20)
205     #ax.annotate('V$_{OC}$', xy=(0,0), xytext=(1,1400),color='r
    ',fontsize=20)
206     ax.annotate('$\Delta$T = 40$^o$C', xy=(0,0), xytext
    =(0.2,1.500),color='black',fontsize=20)
207     #ax.annotate('V$_{5\mu W}$', xy=(0,0), xytext=(2,170),color
    ='r',fontsize=20)
208     for tick in ax.yaxis.get_major_ticks():
209             tick.label.set_fontsize(20)
210     for tick in ax2.yaxis.get_major_ticks():
211             tick.label2.set_fontsize(20)
212     for tick in ax.xaxis.get_major_ticks():
213             tick.label.set_fontsize(0)
214     for xa in xaxs:
215         xaxsLbl.append(str(xa/20.0))
216     #ax.plot(xaxsLbl,voc, color='r');
217     #ax.plot(xaxsLbl,voc, marker='s',color='r');
218     ax.plot(xaxsLbl,vmpp, marker='d',color='r');
219     #ax.plot(xaxsLbl,vmpp_5uw,color='r');
220     ax.tick_params('y', colors='r')
221     ax.set_ylim([0,2])
222     ax2.plot(xaxsLbl,pmpp, marker='o',color='b');
```

```
223     ax2.tick_params('y', colors='b')
224     ax2.set_ylim([0,3])
225
226
227     fig.tight_layout()
228     fig.savefig('pMEMS.pdf')
229     #plt.show();
230
231
232
233     return
234
235 def get_datetime():
236   now = datetime.datetime.now()
237   return '['+str(now.year) + '/' + str(now.month) + '/' + str(
      now.day) + '] [' + str(now.hour) + ':' + str(now.minute) + ':
      ' + str(now.second) +']'
238
239
240 def printf(format, *args):
241   sys.stdout.write(format % args)
242
243 def lprintf(log, format, *args):
244     log.write(format % args)
245     sys.stdout.write(format % args)
246
247 def getTime():
248         device.write("t");
249         response = device.readline()[:-1];
250         #if(response.find('not recognised')):
251         #       printf("Device error. Unplug, Reset and retry!\
    n");
252         #       device.close();
253         #       return;
254         printf("Device time is %s\n",response);
255         return;
256
257 def getVpv():
258         device.write("v");
259         response = device.readline()[:-1];
260         printf("Vpv = %s\n",response);
261         return;
262
263 def getID():
264         device.write("n");
265         response = device.readline()[:-1];
266         printf("%s\n",response);
267         return;
268
269 def getVbat():
270         device.write("b");
271         response = device.readline()[:-1];
272         printf("Vbat = %s\n",response);
273         return;
274
275 def getTemperature():
276         device.write("f");
277         response = device.readline()[:-1];
278         printf("Device temperature (C) = %s\n",response);
279         return;
280
281 def eraseEEPDevice():
282         device.write("E");
283         response = device.readline()[:-1];
284         printf("%s\n",response);
285         return;
```

```
286 def verifyErase():
287         device.write("V");
288         printf("Erasing Device. May take upto 2 minutes. please
    wait...\n");
289         response = device.readline()[:-1];
290         printf("%s\n",response);
291         return;
292 def quickLog():
293         device.write("q");
294         response = device.readline()[:-1];
295         response = device.readline()[:-1];
296         printf("%s\n",response);
297         return;
298
299 def eraseDevice():
300         printf("Erasing Device. May take upto 2 minutes. please
    wait...\n");
301         device.write("e");
302         device.readline()[:-1];
303         response=device.readline()[:-1];
304         printf("%s\n",response);
305         return;
306
307 def secErase():
308         printf("Erasing sectors. May take upto 2 minutes. please
     wait...\n");
309         device.write("s");
310         device.readline()[:-1];
311         response=device.readline()[:-1];
312         printf("%s\n",response);
313         return;
314
315 def setTime():
316         device.write("T");
317         timeString = time.strftime("%S%M%H%w%d%m%y");
318         printf("Setting time to %s\n",timeString);
319         timeArray=list(timeString);
320         for i in range (0,13):
321                 device.write(timeArray[i]);
322                 time.sleep(0.01);
323         response = device.readline()[:-1];
324         printf("Device Time is %s\n",response);
325         return;
326
327 def getData_pv():
328         BytesPerSession = [];
329         now = datetime.datetime.now()
330         TimeStamp = str(now.hour).zfill(2) + str(now.minute).
    zfill(2) + str(now.second).zfill(2) + \
331                                                 '_' + str(now.
    day).zfill(2) + str(now.month).zfill(2) + str(now.year).zfill
    (2)
332         device.write('n');
333         DevID=device.readline()[:-1];
334         DeviceID = DevID[6:]
335         DevIDMatchArray=[];
336
337         if(os.path.exists("config.dat")) :  ##parse config file
    and prepare DevIDMatchArray
338                 printf("Reading config file...\n");
339                 ConfFile = open("config.dat",'r');
340                 DevIDPattern = re.compile("DeviceID");
341                 DevIDflag=0;
342                 DevIDlog=0;
343
344         for line in ConfFile:
```

```
345             #print line
346             if(re.match("DeviceID",line)): DevIDflag=1;
347             if(DevIDflag and re.match("{",line)):
348                 DevIDlog=1;
349             if(DevIDflag and re.match("}",line)):
350                 DevIDlog=0;
351                 DevIDflag=0;
352 
353             if(DevIDlog):
354 
355                 line = line.replace(";",""); #Strip out
    semicolon if any
356                 line = ''.join(line.split()); #Strip out white
    spaces
357                 a=line.split("=");
358                 if(len(a)==2): DevIDMatchArray.append(a);
359 
360         if(len(DevIDMatchArray)<1):
361                 printf("Possible error in config file\n");
362         else:
363 
364                 matchstr="";
365                 for i in range(0,len(DevIDMatchArray)):
366                     if(re.match(DevIDMatchArray[i][0],DeviceID))
    :
367                         matchstr = DevIDMatchArray[i][1];
368 
369                 DeviceID = DevID[6:] + '_' + matchstr;
370 
371         fname = 'IVLogger_' + DeviceID + '_' + TimeStamp + "_raw
    .dat"
372         log=open(fname ,'w')
373 
374         device.write('i');
375         response = device.readline()[:-1];
376         NumberOfSessions = int(response[15:]);
377         response = device.readline()[:-1];
378         FlashAddress = int(response[15:],16);
379         for i in range (0,NumberOfSessions):
380             response=device.readline()[:-1];
381             BytesPerSession.append(int (response[(response.find(
    ":")+1):-6]));
382 
383         log.write("\\\\%s\n" % DevID);
384 
385         device.write('r');
386 
387         for i in range (0,NumberOfSessions):
388 
389             printf("Reading %d lines of Session %d \n",
    BytesPerSession[i]/4,i);
390             log.write("Session %d\n" % i);
391 
392             for j in range (0,BytesPerSession[i]/4):
393                 response=device.readline()[:-1];
394                 if(Mode=="f"):
395                     if (j % LinesPerIVSweep == 0) : log.write("
    Time %s\n" % response);
396                     elif (j % LinesPerIVSweep == 1) : log.write(
    "Lux1 %s\n" % response);
397                     elif (j % LinesPerIVSweep == 2) : log.write(
    "Lux2 %s\n" % response);
398                     elif (j % LinesPerIVSweep == 3) : log.write(
    "Vbat,Temp %s\n" % response);
399                     else :
400                         log.write("\t\t%s\n" % response);
```

```
401                 else :
402                     log.write("%s\n" % response[1:]);
403 
404                     printf("%d / %d        \t\t\r",j,
    BytesPerSession[i]/4);
405             log.write("\n");
406 
407         log.close();
408         printf("Data saved in %s",fname);
409         return
410 
411 def getData_TEG():
412         BytesPerSession = [];
413         now = datetime.datetime.now()
414         TimeStamp = str(now.hour).zfill(2) + str(now.minute).
    zfill(2) + str(now.second).zfill(2) + \
415                                                 '_' + str(now.
    day).zfill(2) + str(now.month).zfill(2) + str(now.year).zfill
    (2)
416         device.write('n');
417         response=device.readline()[:-1];
418         LinesPerSession=int(response[:3])
419         DevID=response[:-6]
420         fname = 'IVLogger_' + DevID + '_' + TimeStamp + "
    _TEG_raw.dat"
421         log=open(fname ,'w')
422         printf("%s file opened\n" % fname)
423 
424         log.write("\\\\%s\n" % DevID);
425         log.write("\\\\%s\n" % LinesPerSession);
426         response="0x0";
427         m = re.search('0x',response)
428         device.write('r');
429         linecount=1
430         while(m):
431             response=device.readline()[:-1];
432             m = re.search('0x',response)
433             line=''
434             if(m and linecount % LinesPerSession == 1): line=
    parseTime(response)
435 
436             if(m and linecount % LinesPerSession == 2): line=
    parseTemp(response)
437             if(m and linecount % LinesPerSession >2): line=
    parseData(response)
438             log.write("%s %s\n" % (line, response)); #Print all
    lines
439             linecount+=1
440 
441         log.close();
442         printf("Data saved in %s",fname);
443         return
444 def parseTime(line, verbose=1):
445     if(verbose): print line
446     day=int(("0x" + line[3:5]),16);
447     month=int(("0x" + line[5:6]),16);
448     year=int(("0x" + line[6:7]),16) + 2014;
449     hour=int(("0x" + line[7:9]),16);
450     minute=int(("0x" + line[9:11]),16);
451     if(verbose): printf("%0.2d:%0.2d %0.2d/%0.2d/%0.4d\n",hour,
    minute,day,month,year);
452     return(str(hour) + ":" + str(minute) + " " + str(day) + "/"
    + str(month) + "/" + str(year));
453 
454 def parseTemp(line, verbose=1):
455     if(verbose): print line
```

```
456     Vbat=int(("0x" + line[3:7]),16) / 10.0;
457     Temp=int(("0x" + line[7:11]),16);
458     return(str(Vbat) + "," + str(Temp));
459
460 def parseData(line, verbose=0):
461     if(verbose): print line
462     i=int(("0x" + line[3:7]),16);
463     v=int(("0x" + line[7:11]),16);
464     return(str(v) + "," + str(i) + "," + str(v*i*1e-3));
465
466
467 def getHelp():
468     pattern = re.compile("END OF HELP");
469     response="";
470     device.write('h');
471     while(not(pattern.match(response))):
472         response=device.readline()[:-1];
473         printf("%s\n",response); #Print all lines
474     return
475
476 def getInfo():
477         device.write('i');
478         response = device.readline()[:-1];
479         printf("%s\n",response); #session count
480         NumberOfSessions = int(response[15:]);
481         response = device.readline()[:-1];
482         FlashAddress = int(response[15:],16);
483         printf("%s\n",response); #flash final address
484         for i in range (0,NumberOfSessions):
485             response = device.readline()[:-1];
486             printf("%s\n",response);
487
488         return;
489
490 def inter():
491         now = datetime.datetime.now()
492         TimeStamp = str(now.hour).zfill(2) + str(now.minute).
    zfill(2) + str(now.second).zfill(2) + \
493                                                 '_' + str(now.
    day).zfill(2) + str(now.month).zfill(2) + str(now.year).zfill
    (2)
494         fname = 'IVLogger_interactive'+ '_' + TimeStamp + ".txt"
495         global log;
496         log=open(fname ,'w')
497
498         Vpv=[];
499         Ipv=[]
500         device.write('p');
501         loop=1;
502         while(loop):
503             response = device.readline()[:-1];
504             lprintf(log,"%s\n",response);
505             arr=response.split();
506             if(len(arr)==3):
507                 Vpv.append(float(arr[1]));
508                 Ipv.append(float(arr[2]));
509             if(len(arr)<2):  loop=0
510         log.close
511
512         plot=1;
513         if(plot):
514             xmax=round(max(Vpv),1);
515             ymax=roundup(max(Ipv));
516             fig,ax=plt.subplots(1);
517             ax.set_title(fname[:-4],size=12)
518             ax.set_xlabel('Voltage (V)',size=10)
```

```
519             #ax.set_xticks(np.arange(0,xmax,round(xmax)/10))
520             #ax.set_yticks(np.arange(0,ymax,ymax/10))
521             ax.set_ylabel('Current (uA)',size=10)
522             ax.scatter(Vpv,Ipv);
523             #print ax.ylim()
524             #print ax.xlim()
525             x1,x2,y1,y2 = plt.axis()
526             plt.axis((0,x2,0,y2))
527             fig.savefig(fname[:-3]+'png',  dpi=600 )
528             plt.show();
529         return;
530
531 def main():
532     #configuration variables set here
533
534     printf("Connecting to device on %s\n",COM);
535     global device;
536     #device = serial.Serial(port=COM)
537     #device.flush()
538     #device.flushInput()
539     #device.flushOutput()
540     printf("Connected.\n");
541
542     options = {
543             't' : getTime,
544             'n' : getID,
545             'T' : setTime,
546             'i' : getInfo,
547             'v' : getVpv,
548             'q' : quickLog,
549             's' : secErase,
550             'b' : getVbat,
551             'V' : verifyErase,
552             'e' : eraseDevice,
553             'E' : eraseEEPDevice,
554             'f' : getTemperature,
555             'r' : getData_TEG,
556             'h' : getHelp,
557             'p' : inter,
558             'pTEG' : plotTEG,
559     }
560     options[todo]();
561     #device.close()
562
563     return
564
565 if __name__ == "__main__":
566     main()
```

# Appendix D

# Python script for parameter extraction from characterization data

The Python script provided in this appendix allows parsing the logged IV data to obtain the PV cell model-parameters as described in Chapter 3 .

```
1  import time
2  from time import clock
3  import datetime
4  import os
5  import struct
6  import serial
7  import getopt
8  import time
9  import re
10 import threading
11 import math
12 from numpy import *
13 import matplotlib
14 import matplotlib.dates as mdates
15 import matplotlib.pyplot as plt
16 from matplotlib.ticker import MultipleLocator,
       FormatStrFormatter, LinearLocator
17 import glob
18 import sys
19 from serial.serialutil import SerialException
20 import itertools
21 import re
22 import optparse
23 from mpl_toolkits.mplot3d import Axes3D
24 from matplotlib import cm
25 import numpy as np
26 from matplotlib.mlab import griddata
27
28 parser = optparse.OptionParser()
29 parser.add_option('-i', '--InputFileName', action="store", type=
       "string", help="Input File Name", dest="infiles", default=""
       ,)
30
31 options, args = parser.parse_args()
32
```

```
33
34 infilelist=glob.glob(options.infiles);
35
36 def printf(format, *args):
37   sys.stdout.write(format % args)
38
39 def main():
40     print infilelist
41
42     for infile in infilelist:
43         print infile
44         res=[];
45         ocv=[];
46         indx=0;
47         track=0;
48         sweep=3;
49         pmpp=0;
50         k=1.38e-23
51         q=1.6e-19
52         fileIn = open (infile,'r')
53         outfile = infile[:-8] + "_Rs.tikzdat"; #new log file
    name;
54         log=open(outfile ,'w')  #open new log file for write
55         #if not os.path.exists(outfile):
56         #    os.makedirs(outfile)
57         log.write("X Lux1 Lux2 Temp Voc Ish vmpp impp Rso Rpo\n"
    );
58  count=1
59         for line in fileIn:
60                 array=line.split(",");
61                 if(len(array) > 2 and re.search('0x', array[2]))
    :
62                     indx += 1;
63                     Pprev=0;
64                     Rp=0;
65                     ERR=0;
66
67
68                     ##Capture peak power point
69                     if(indx>1): Pprev=Pmeas
70
71                     voltage = float(array[6]);
72                     current = float(array[7]);
73                     Pmeas = float(array[8]);
74                     power = voltage*current;
75
76                     if (indx%50==1):
77                         Voc=voltage;
78                         indx_mpp=0;
79                         Temp=float(array[5])
80                         sweep += 51;
81
82
83                     if(power<pmpp):     indx_mpp=indx #reading
    where pmpp is reached
84                     if(indx == indx_mpp+2): #two readings after
    pmpp
85                         Vprev=voltage
86                         Iprev=current
87
88                     if(Pprev>0 and Pmeas==0):   #when power hits
     zero
89                         if(impp==current):
90                             current=impp-0.1;
91                             printf("Rp Current manipulated for
    sweep %d\n",sweep)
```

```
92                          Rp=(vmpp-Vprev)*1e-3 / ((impp-current)*1
    e-6)
93
94                      if(power >= pmpp):      #track max power
95                          vmpp=voltage;
96                          impp=current;
97                          pmpp=power;
98                          lux1=array[2]
99                          lux2=array[3]
100
101
102                     if (indx%50==0 ):
103                         Ish=float(array[7])
104                         track=0;
105                         if(Ish==0):ERR=1;
106                         if(impp==0):ERR+=2;
107                         if(vmpp==0):ERR+=4;
108                         if(impp>=Ish):ERR+=8;
109                         if(Rp<=0):ERR+=16;
110                         if(2*vmpp<=Voc):ERR+=32;
111                         if(Ish==0):# or impp==0 or vmpp==0 or
    impp>=Ish or Rp<=0 or 2*vmpp<=Voc): #
112                             #print Voc,sweep,line
113                             indx=0
114                             printf("Skipping sweep %d - %d\n",
    sweep,ERR)
115                             ERR=0
116                         else:
117                             Rs=(Voc/Ish)
118                             res.append(Rs)
119                             ocv.append(Voc)
120                             log.write("%d %d %d %d %0.3f %0.6f
    %0.3f %0.6f %d %d" % (count, (int(lux1,16) & 0xFFFF0000)
    >>16,(int(lux1,16) & 0x0000FFFF),Temp,Voc,Ish,vmpp,impp,Rs,Rp
    ));
121         count +=1
122      #calculate Rs
123 #                               alpha =Ish - (Voc/Rp)
124 #                               beta = Ish - impp - vmpp/Rp
125 #                               gamma = math.exp(q*(vmpp-Voc)/(2*k
    *(Temp+274.15)))
126 #                               delta = q*impp / (k*(Temp+274.15))
127 #                               log.write(",%f,%f,%f,%f" % (alpha,
    beta,gamma,delta));
128 #                               a = alpha * gamma * delta * (1-
    gamma)
129 #                               b = alpha * gamma * (2-gamma) +
    alpha * gamma * delta * Rs * (gamma-1) - beta + gamma * delta
     * (1-2*gamma) * k*(Temp+274.15)/q
130 #                               c = alpha * gamma *  Rs * (gamma
    -2) + beta * Rs + 2*gamma*(1-gamma)* k*(Temp+274.15)/q
131 #        log.write(",%f,%f,%f\n" % (a,b,c));
132                                 log.write("\n");
133 #                               I0=Ish;
134 #                               for iterN in range (0,3):
135 #                                   K1 = ((impp/(I0-impp)) + log10
    (1-impp/I0))/(2*vmpp-Voc)
136 #                                   K2 = (log10(I0) - Voc*K1)
137 #                                   #Rs = (vmpp-(impp/(I0-impp)*K1)
    )/impp
138 #                                   I0 = Ish + math.exp((Ish*Rs*K1)
     + K2)
139 #                                   Is1 = (-Ish + Voc/Rp + 2*k*Temp
    /(q*Rs))*math.exp(-q*Voc/(k*Temp))
140 #                                   Is2 = 2*(Ish - Voc/Rp - k*Temp
    /(q*Rs))*math.exp(-q*Voc/(2*k*Temp))
```

```
141 #                               log.write(",%0.3f,%0.3f,%0.3f,%f
    ,%0.3f,%f,%f\n" % (K1, K2, Rs, I0,Rp,Is1,Is2))
142
143                         indx=0;
144                         ERR=0;
145                         pmpp=0;
146         log.close();
147         fileIn.close();
148         print len(res),sweep/51;
149         plot=0;
150         if(plot):
151             fig, ax = plt.subplots()
152             ax.scatter(ocv,res)
153             ax.set_yscale('log')
154             plt.xlabel('Voc (mV)')
155             plt.ylabel('Rs (kOhm)')
156             ax.axis([0, 1000, 10, 100])
157             #fig.savefig('Rs_scatter.png',dpi=200);
158             #fig.show();
159             pngfilename= infile[:-4] + ".png"
160             plt.savefig(pngfilename, bbox_inches='tight', dpi
    =600 )
161             plt.show()
162
163 ##        #plt.figure(figsize=(8,15));
164 ##        f,(ax1,ax2,ax3) = plt.subplots(3,sharex=True,sharey=
    False);
165 ##        plt.gca().xaxis.set_major_formatter(mdates.
    DateFormatter('%d/%m %H:%M '))
166 ##        plt.gca().xaxis.set_major_locator(mdates.HourLocator(
    arange(0,25,2)))
167 ##
168 ##        ax1.set_title(ID,size=12)
169 ##        plt.xlabel('Time')
170 ##        plt.legend(loc=0,prop={'size': 7})
171 ##        ax1.plot(x,temperature);
172 ##        ax1.set_ylabel('temperature (DegC)',size=10)
173 ##        ax2.plot(x,vmpp);
174 ##        ax2.set_ylabel('Vmpp (mV)',size=10)
175 ##        ax3.plot(x,pmpp);
176 ##        ax3.set_ylabel('Pmpp (uW)',size=10)
177 ##        plt.gcf().autofmt_xdate(rotation=90)
178 ##        pngfilename= infile[:-4] + "_" + ID + ".png"
179 ##        plt.savefig(pngfilename, bbox_inches='tight', dpi=600
    )
180 ##        #plt.show();
181     return
182
183 if __name__ == "__main__":
184     main()
```

# Appendix E

# Converter-EH simulation setup

This appendix shows an example SPICE testbench which allows simulating PV cell with the EH converter for the system described in Chapter 4 .

```
1  ** 26 Sept 2014 - VPV to ESS
2
3  .TEMP 25.0
4  .OPTION post=1 probe=1 ingold=2 brief=1
5
6  .option finesim_output=tr0
7  .option finesim_resmin=0.001
8  .option finesim_enhanced_tcl_mode=0
9  .option finesim_mode="spice"
10 .option finesim_skipwarn="removed due to a dangling terminal":2
11 .option finesim_skipwarn="has only 1 connection to a MOSFET":2
12 .option finesim_subckt_dup_rule=1  $$ use first defn
13
14 .PARAM supply=1.2 simend=400u
15 .param  illum=500
16 .param ckt_settle_time='vref_settle_time+supply_settle_time+
      bias_settle_time'
17 + frequency=20e6
18
19 .include 'pvcell_model.spi'
20 .include 'VREG_V3.cdl'
21
22 .LIB "65nm_models" TT,TT_25, TT_HVT, TT_LVT, TT_RES, TT_MIM,
      TT_DIO_25
23
24 X1 OUT QCOMP VBAT VDDM VPV_MEAS VSS BYPX2P CK CL_BY2 CL_BY3 EN2
      ENBY2 ENBY3 ENBY4 ENCOMP ENLDO ENREF0 ENREF1 ENREF2 ENREF3
      ENX2B ENX2P EN_MIM PVREF0 PVREF1 PVREF2 PVREF3 PVREF4 VREG_V3
25 **X1 VDDM VSTORE VPV_MEAS VSS CK EN  X1X2_LV
26 X0 Vpv vss PVCELL lux='illum'
27 CSTORE VSTORE VSS  1n
28 CHARVEST VPV VSS 2n
29 R1 VPV_MEAS VSS pwl 0 0 400u 50u
30
31 .ic v(vstore)=0
32
33 v00 vss 0 DC=0
34 v01 vdd vss supply
35 vm vdd vddm dc 0
36 v04 ck vss dc 0 pulse (0 supply 0 100p 100p 0.5/frequency 1/
      frequency)
37 v05 EN vss dc 0 pwl 0 0 50u 0 50.001u supply
```

```
38 vvpv vpv vpv_meas dc 0
39 .print p(I1)
40
41 .tran 1p simend $$sweep frequency 2e6 40e6 2e6
42
43 .meas tran meas_vstore avg v(VSTORE) from =0.8*simend to=simend
44 .meas tran meas_ipv avg i(vvpv) from=0.8*simend to=simend
45 .meas tran meas_vpv avg v(vpv) from=0.8*simend to=simend
46
47 .meas tran meas_pin param = 'meas_vpv*meas_ipv'
48 .meas tran meas_eharvest integ p(CSTORE)
49
50
51 .END
```

# Appendix F

# SPICE testbench for rail monitor measurements

This appendix illustrates an example SPICE testbench for carrying out DC and transient measurements for the rail monitor described in Chapter 5 .

```
1  ** pgood monitor 11 jan 2016
2
3
4  .option finesim_output=tr0
5  .option finesim_resmin=0.001
6  .option finesim_enhanced_tcl_mode=0
7  .option finesim_mode="spicead"
8  .option finesim_vector_mode=1
9
10 .inc 'pgroscpgood.spice.monte'
11
12
13 .temp 25.0
14 .param supply=1.2 simend=0.4e-3 vreg=supply/2
15 + par_bh0 =1
16 + par_bh1 =1
17 + par_bl0 =1
18 + par_bl1 =1
19
20 .option
21 +    ingold=2
22 +    post=2
23 +    probe
24 +    parhier=local
25 ***+ autostop
26
27 .LIB "65nm_models" TT,TT_25, TT_HVT, TT_LVT, TT_RES, TT_MIM,
      TT_DIO_25
28
29 **.subckt pgroscpgood bhsel<1> bhsel<0> blsel<1> blsel<0> pgen
      pgood ql qu
30 ***+ rhsel<6> rhsel<5> rhsel<4> rhsel<3> rhsel<2> rhsel<1> rhsel
      <0> rlsel<6>
31 ***+ rlsel<5> rlsel<4> rlsel<3> rlsel<2> rlsel<1> rlsel<0> vdd
      vfb2 vreg vss
32 **dut instance
33 x1  bhsel1 bhsel0 blsel1 blsel0 vdd pgood ql qu  vss vss vdd vss
       vss vss vss  vss vss vss vdd vss vss vss  vdd vdd vreg vss
```

```
    pgroscpgood
cl pgood vss 2f

**signals to monitor
.probe v(q) i(v1) v(vreg) v(pgood)


**stimulus
v0 vss 0 dc=0
v1 vdd vss supply $$pwl 0 0 100n 0 10u supply
v2 vregr vss dc 0 pwl
+ 0 0.23
+ 100u 0.23
+ 100.001u 0.43
+ 300u 0.43
+ 300.001u 0.23
vbh0 bhsel0 vss 'supply*par_bh0'
vbh1 bhsel1 vss 'supply*par_bh1'
vbl0 blsel0 vss 'supply*par_bl0'
vbl1 blsel1 vss 'supply*par_bl1'


**load setup
rvreg vregr vreg 10
cvreg vreg vss 100p

**simulation analysis

**.dc v2 0 supply 0.01 sweep monte=1000
.tran 1n simend sweep supply 1.0 1.4 0.4 $$sweep monte=100

**measure statements
.meas tran vqlr find v(vreg) when v(ql)=supply/2 rise=1
.meas tran vqlf find v(vreg) when v(ql)=supply/2 fall=1
.meas tran tdlr trig v(vreg) val=0.3 rise=1 targ v(ql) val=
    supply/2 rise=1
.meas tran tdlf trig v(vreg) val=0.3 fall=1 targ v(ql) val=
    supply/2 fall=1
.meas tran tldelay param='1e9*(tdlr+tdlf)/2'

**alternate settings
.alter 10
.param
+ par_bh0 =0
+ par_bh1 =1
+ par_bl0 =0
+ par_bl1 =1

.alter 01
.param
+ par_bh0 =1
+ par_bh1 =0
+ par_bl0 =1
+ par_bl1 =0


.end
```

# Appendix G

# SPICE testbench for relaxation oscillator measurements

This appendix gives an example SPICE testbench for carrying out DC and transient measurements for the relaxation oscillator described in Chapter 6 .

```
1  ** viro 21 mar 2016
2
3  .option finesim_output=tr0
4  .option finesim_resmin=0.001
5  .option finesim_enhanced_tcl_mode=0
6  .option finesim_mode="spicead"
7  .option finesim_vector_mode=1
8
9  .inc './vir0.cdl'
10
11 **change below variables for line and temperature sensitivity
12 .temp 25.0
13 .param supply=1.4
14
15 .option
16 +    ingold=2
17 +    post=2
18 +    probe
19 +    parhier=local
20 + autostop
21 .LIB "65nm_models" TT,TT_25, TT_HVT, TT_LVT, TT_RES, TT_MIM,
      TT_DIO_25, TT_NA
22
23 x1 rcksel[1] rcksel[0] oset1[7] oset1[6] oset1[5] oset1[4] oset1
      [3] oset1[2] oset1[1]
24 + oset1[0] fcksel[1] fcksel[0] bsel[1] bsel[0] tsel[1] tsel[0]
      cxsel[1] cxsel[0] oset2[7]
25 + oset2[6] oset2[5] oset2[4] oset2[3] oset2[2] oset2[1] oset2[0]
       safe[7] safe[6] safe[5]
26 + safe[4] safe[3] safe[2] safe[1] safe[0] cnsel[1] cnsel[0] stm
      [4] stm[3] stm[2] stm[1]
27 + stm[0] q[7] q[6] q[5] q[4] q[3] q[2] q[1] q[0] ntest rn ovr
      enstm enduty en csel vcdig
28 + stop rck fck ck vss vdd viro_v0
29
30 .probe v(*ck*) v(en*) i(v1) v(vcdig) v(x1.xi0/vc) v(x1.oscset*)
      v(x1.xi0/vref)
31
```

```
32 v1 vdd vss dc supply
33 v0 vss 0 dc=0
34
35 **power terminals
36 cvdd vdd vss 100n
37
38 **signal output
39 cq0 q[0] vss 2f
40 cq1 q[1] vss 2f
41 cq2 q[2] vss 2f
42 cq3 q[3] vss 2f
43 cq4 q[4] vss 2f
44 cq5 q[5] vss 2f
45 cq6 q[6] vss 2f
46 cq7 q[7] vss 2f
47
48 cstm0 stm[0] vss 2f
49 cstm1 stm[1] vss 2f
50 cstm2 stm[2] vss 2f
51 cstm3 stm[3] vss 2f
52 cstm4 stm[4] vss 2f
53
54 cfck  fck  vss 2f
55 cstop stop vss 2f
56 cck   ck   vss 2f
57 crck  rck  vss 2f
58 cvcd  vcd  vss 2f
59
60 .ic v(x1.xi0/vc)=0 v(x1.xi0/vref)=supply/3
61 .vec pbrkdn_isscc.vec
62
63 **.dc temp 0 100 25
64 .tran 1p 10u
65
66 .meas tran meas_iavg1 avg i(v1) from = 500n to = 9u
67 .meas tran meas_iavgd avg i(x1.v0) from = 500n to = 9u
68 .meas tran meas_iavgr avg i(x1.xi0.xi6.v0) from = 500n to = 9u
69 .meas tran meas_iavgx avg i(x1.xi0.v0) from = 500n to = 9u
70 .meas tran meas_iavgc avg i(x1.xi0.v1) from = 500n to = 9u
71 .meas tran meas_iavgt avg i(x1.xi0.v2) from = 500n to = 9u
72
73
74 .meas tran meas_pwr_total param='meas_iavg1*supply*-1'
75 .meas tran meas_pwr_comparator param='(meas_iavgc-meas_iavgr)*
      supply'
76 .meas tran meas_pwr_tank param='meas_iavgt*supply'
77 .meas tran meas_pwr_schmitt param='meas_iavgx*supply'
78 .meas tran meas_pwr_ref param='meas_iavgr*supply'
79 .meas tran meas_pwr_digital param='meas_pwr_total - (
      meas_pwr_comparator + meas_pwr_tank + meas_pwr_schmitt +
      meas_pwr_ref)'
80
81
82
83 .meas tran meas_delay1 trig v(ck) val='supply/2' rise =3 targ v(
      ck) val='supply/2' rise =4
84 .meas tran meas_freq1 param='1/meas_delay1'
85
86
87 .end
```

## G.1 Relaxation oscillator netlist

```
1  .include standard_cells_hvt.cdl
2
3  *
       ********************************************************************************
4  * library name: tgo
5  * cell name: rlxosc
6  * view name: schematic
7  *
       ********************************************************************************
8
9  .subckt rlxosc bsel[1] bsel[0] ck csel dtune[2] dtune[1] dtune
       [0] en enduty
10 + enstck ovr rck rcksel[1] rcksel[0] rtune[4] rtune[3] rtune[2]
       rtune[1]
11 + rtune[0] stck tsel[1] tsel[0] vcdig vdd vss
12 xi13 dtune[2] dtune[1] dtune[0] enduty vc xvdd vss vcdig net025
       / rlxmitnor1
13 xi26 en q5 vdd vdd vss vss ck / rlxand2
14 xi27 vcdig ovr vdd vdd vss vss net018 / rlxand2
15 xi28 enduty ovr vdd vdd vss vss net022 / rlxand2
16 xi6 bsel[1] bsel[0] ck vref vc q2 rck rcksel[1] rcksel[0] tsel
       [1] tsel[0] cvdd
17 + vss net018 / rlxcomparator1
18 xi9 csel ck vss tvdd rtune[4] rtune[3] rtune[2] rtune[1] rtune
       [0] vc / rlxtank1
19 xi10 q3 net018 net022 vdd vss q4 / rlxbmux1
20 xi29 enstck stck vdd vss / rlxleakosc2
21 xi7 q5 net26 q4 q3 vdd vss / rlxrsff1
22 xi8 q2 vdd vss q3 / schmittconv
23 v0 vdd xvdd dc 0
24 v1 vdd cvdd dc 0
25 v2 vdd tvdd dc 0
26 .ends
27
28
29 .subckt viro_v0 rcksel[1] rcksel[0] oset1[7] oset1[6] oset1[5]
       oset1[4] oset1[3] oset1[2] oset1[1] oset1[0] fcksel[1] fcksel
       [0] bsel[1] bsel[0] tsel[1] tsel[0] cxsel[1] cxsel[0] oset2
       [7] oset2[6] oset2[5] oset2[4] oset2[3] oset2[2] oset2[1]
       oset2[0] safe[7] safe[6] safe[5] safe[4] safe[3] safe[2] safe
       [1] safe[0] cnsel[1] cnsel[0] stm[4] stm[3] stm[2] stm[1] stm
       [0] q[7] q[6] q[5] q[4] q[3] q[2] q[1] q[0] ntest rn ovr
       enstm enduty en csel vcdig stop rck fck ck vss vdd
30 xi0 bsel[1] bsel[0] ckprebuf csel oscset[2] oscset[1] oscset[0]
       en enduty enstck ovr rckprebuf rcksel[1]
31 + rcksel[0] oscset[7] oscset[6] oscset[5] oscset[4] oscset[3]
       stckprebuf tsel[1] tsel[0] vcdigprebuf rvdd vss rlxosc
32 xi1_i0_0_ i1_qmux1[0] dvdd dvdd vss vss oset1[0] oset2[0]
       i1_sel1 mx2_x1_a8th
33 xi1_i0_1_ i1_qmux1[1] dvdd dvdd vss vss oset1[1] oset2[1]
       i1_sel1 mx2_x1_a8th
34 xi1_i0_2_ i1_qmux1[2] dvdd dvdd vss vss oset1[2] oset2[2]
       i1_sel1 mx2_x1_a8th
35 xi1_i0_3_ i1_qmux1[3] dvdd dvdd vss vss oset1[3] oset2[3]
       i1_sel1 mx2_x1_a8th
36 xi1_i0_4_ i1_qmux1[4] dvdd dvdd vss vss oset1[4] oset2[4]
       i1_sel1 mx2_x1_a8th
37 xi1_i0_5_ i1_qmux1[5] dvdd dvdd vss vss oset1[5] oset2[5]
       i1_sel1 mx2_x1_a8th
38 xi1_i0_6_ i1_qmux1[6] dvdd dvdd vss vss oset1[6] oset2[6]
       i1_sel1 mx2_x1_a8th
39 xi1_i0_7_ i1_qmux1[7] dvdd dvdd vss vss oset1[7] oset2[7]
       i1_sel1 mx2_x1_a8th
```

```
40 xi1_i1_0_ i1_qmux[0] dvdd dvdd vss vss safe[0] i1_qmux1[0]
       i1_sel0 mx2_x1_a8th
41 xi1_i1_1_ i1_qmux[1] dvdd dvdd vss vss safe[1] i1_qmux1[1]
       i1_sel0 mx2_x1_a8th
42 xi1_i1_2_ i1_qmux[2] dvdd dvdd vss vss safe[2] i1_qmux1[2]
       i1_sel0 mx2_x1_a8th
43 xi1_i1_3_ i1_qmux[3] dvdd dvdd vss vss safe[3] i1_qmux1[3]
       i1_sel0 mx2_x1_a8th
44 xi1_i1_4_ i1_qmux[4] dvdd dvdd vss vss safe[4] i1_qmux1[4]
       i1_sel0 mx2_x1_a8th
45 xi1_i1_5_ i1_qmux[5] dvdd dvdd vss vss safe[5] i1_qmux1[5]
       i1_sel0 mx2_x1_a8th
46 xi1_i1_6_ i1_qmux[6] dvdd dvdd vss vss safe[6] i1_qmux1[6]
       i1_sel0 mx2_x1_a8th
47 xi1_i1_7_ i1_qmux[7] dvdd dvdd vss vss safe[7] i1_qmux1[7]
       i1_sel0 mx2_x1_a8th
48 xi1_i2_0_ oscset[0] i1_noscset[0] dvdd dvdd vss vss i1_qmux[0]
       i1_net058 tlatn_x2_a8th
49 xi1_i2_1_ oscset[1] i1_noscset[1] dvdd dvdd vss vss i1_qmux[1]
       i1_net058 tlatn_x2_a8th
50 xi1_i2_2_ oscset[2] i1_noscset[2] dvdd dvdd vss vss i1_qmux[2]
       i1_net058 tlatn_x2_a8th
51 xi1_i2_3_ oscset[3] i1_noscset[3] dvdd dvdd vss vss i1_qmux[3]
       i1_net058 tlatn_x2_a8th
52 xi1_i2_4_ oscset[4] i1_noscset[4] dvdd dvdd vss vss i1_qmux[4]
       i1_net058 tlatn_x2_a8th
53 xi1_i2_5_ oscset[5] i1_noscset[5] dvdd dvdd vss vss i1_qmux[5]
       i1_net058 tlatn_x2_a8th
54 xi1_i2_6_ oscset[6] i1_noscset[6] dvdd dvdd vss vss i1_qmux[6]
       i1_net058 tlatn_x2_a8th
55 xi1_i2_7_ oscset[7] i1_noscset[7] dvdd dvdd vss vss i1_qmux[7]
       i1_net058 tlatn_x2_a8th
56 xi1_i3 i1_sel1 i1_net033 dvdd dvdd vss vss i1_net08 i1_lat0
       tlatn_x2_a8th
57 xi1_i4 i1_qlat0 i1_net028 dvdd dvdd vss vss i1_net06 i1_lat0
       tlatn_x2_a8th
58 xi1_i5_i0 enstckprebuf i1_i5_stmtg dvdd dvdd vss vss enstm
       i1_i5_dstmtg rn dffr_x1_a8th
59 xi1_i5_i1 stm[0] i1_i5_net27 dvdd dvdd vss vss stck
       i1_i5_denstck enstckprebuf dffr_x1_a8th
60 xi1_i5_i2_0_ stm[1] i1_i5_net22[3] dvdd dvdd vss vss stck stm[0]
        enstckprebuf dffr_x1_a8th
61 xi1_i5_i2_1_ stm[2] i1_i5_net22[2] dvdd dvdd vss vss stck stm[1]
        enstckprebuf dffr_x1_a8th
62 xi1_i5_i2_2_ stm[3] i1_i5_net22[1] dvdd dvdd vss vss stck stm[2]
        enstckprebuf dffr_x1_a8th
63 xi1_i5_i2_3_ stm[4] i1_i5_net22[0] dvdd dvdd vss vss stck stm[3]
        enstckprebuf dffr_x1_a8th
64 xi1_i5_i3 i1_i5_denstck dvdd dvdd vss vss i1_i5_nenstck
       inv_x2_a8th
65 xi1_i5_i4 i1_i5_nenstck dvdd dvdd vss vss enstckprebuf
       inv_x2_a8th
66 xi1_i5_i6 i1_i5_nstmtg dvdd dvdd vss vss i1_i5_stmtg inv_x2_a8th
67 xi1_i5_i7 i1_i5_dstmtg dvdd dvdd vss vss i1_i5_nstmtg
       inv_x2_a8th
68 xi1_i5_i10 i1_i5_xn12 dvdd dvdd vss vss stm[1] stm[2]
       xnor2_x1_a8th
69 xi1_i5_i11 i1_i5_xn34 dvdd dvdd vss vss stm[3] stm[4]
       xnor2_x1_a8th
70 xi1_i5_i12 i1_i5_nst dvdd dvdd vss vss i1_i5_xn12 i1_i5_xn34
       xnor2_x1_a8th
71 xi1_i5_i13 i1_st dvdd dvdd vss vss i1_i5_nst inv_x2_a8th
72 xi1_i5_i14 i1_lat1 dvdd dvdd vss vss stm[4] i1_i5_net042
       or2_x1_a8th
73 xi1_i5_i15 i1_lat0 dvdd dvdd vss vss stm[3] i1_i5_net041
       or2_x1_a8th
```

```
74 xi1_i5_i16 i1_i5_net042 dvdd dvdd vss vss stm[3] inv_x2_a8th
75 xi1_i5_i17 i1_i5_net041 dvdd dvdd vss vss stm[2] inv_x2_a8th
76 xi1_i6_i0 i1_i6_ck[0] dvdd dvdd vss vss i1_i6_y i1_i6_y
       i1_i6_nen rn nand4_x2_a8th
77 xi1_i6_i1_0_ i1_i6_ck[1] dvdd dvdd vss vss i1_i6_ck[0] i1_i6_ck
       [0] i1_i6_ck[0] i1_i6_ck[0] nand4_x2_a8th
78 xi1_i6_i1_1_ i1_i6_ck[2] dvdd dvdd vss vss i1_i6_ck[1] i1_i6_ck
       [1] i1_i6_ck[1] i1_i6_ck[1] nand4_x2_a8th
79 xi1_i6_i1_2_ i1_i6_ck[3] dvdd dvdd vss vss i1_i6_ck[2] i1_i6_ck
       [2] i1_i6_ck[2] i1_i6_ck[2] nand4_x2_a8th
80 xi1_i6_i1_3_ i1_i6_ck[4] dvdd dvdd vss vss i1_i6_ck[3] i1_i6_ck
       [3] i1_i6_ck[3] i1_i6_ck[3] nand4_x2_a8th
81 xi1_i6_i1_4_ i1_i6_ck[5] dvdd dvdd vss vss i1_i6_ck[4] i1_i6_ck
       [4] i1_i6_ck[4] i1_i6_ck[4] nand4_x2_a8th
82 xi1_i6_i1_5_ i1_i6_ck[6] dvdd dvdd vss vss i1_i6_ck[5] i1_i6_ck
       [5] i1_i6_ck[5] i1_i6_ck[5] nand4_x2_a8th
83 xi1_i6_i1_6_ i1_i6_ck[7] dvdd dvdd vss vss i1_i6_ck[6] i1_i6_ck
       [6] i1_i6_ck[6] i1_i6_ck[6] nand4_x2_a8th
84 xi1_i6_i1_7_ i1_i6_ck[8] dvdd dvdd vss vss i1_i6_ck[7] i1_i6_ck
       [7] i1_i6_ck[7] i1_i6_ck[7] nand4_x2_a8th
85 xi1_i6_i1_8_ i1_i6_ck[9] dvdd dvdd vss vss i1_i6_ck[8] i1_i6_ck
       [8] i1_i6_ck[8] i1_i6_ck[8] nand4_x2_a8th
86 xi1_i6_i3 i1_i6_y dvdd dvdd vss vss i1_i6_ck[9] i1_i6_ck[7]
       i1_i6_ck[5] i1_i6_ck[3] fcksel[0] fcksel[1] mxi4_x2_a8th
87 xi1_i6_i4_1_ q[1] i1_i6_qn[1] dvdd dvdd vss vss i1_i6_qn[0]
       i1_i6_qn[1] i1_i6_lrn dffr_x1_a8th
88 xi1_i6_i4_2_ q[2] i1_i6_qn[2] dvdd dvdd vss vss i1_i6_qn[1]
       i1_i6_qn[2] i1_i6_lrn dffr_x1_a8th
89 xi1_i6_i4_3_ q[3] i1_i6_qn[3] dvdd dvdd vss vss i1_i6_qn[2]
       i1_i6_qn[3] i1_i6_lrn dffr_x1_a8th
90 xi1_i6_i4_4_ q[4] i1_i6_qn[4] dvdd dvdd vss vss i1_i6_qn[3]
       i1_i6_qn[4] i1_i6_lrn dffr_x1_a8th
91 xi1_i6_i4_5_ q[5] i1_i6_qn[5] dvdd dvdd vss vss i1_i6_qn[4]
       i1_i6_qn[5] i1_i6_lrn dffr_x1_a8th
92 xi1_i6_i4_6_ q[6] i1_i6_qn[6] dvdd dvdd vss vss i1_i6_qn[5]
       i1_i6_qn[6] i1_i6_lrn dffr_x1_a8th
93 xi1_i6_i4_7_ q[7] i1_i6_qn[7] dvdd dvdd vss vss i1_i6_qn[6]
       i1_i6_qn[7] i1_i6_lrn dffr_x1_a8th
94 xi1_i6_i5 q[0] i1_i6_qn[0] dvdd dvdd vss vss fck i1_i6_qn[0]
       i1_i6_lrn dffr_x1_a8th
95 xi1_i6_i8 i1_i6_lrn i1_i6_trn dvdd dvdd vss vss i1_st i1_i6_trn
       rn dffr_x1_a8th
96 xi1_i6_i9 i1_i6_start i1_i6_net051 dvdd dvdd vss vss i1_i6_vc
       i1_i6_lrn rn dffr_x1_a8th
97 xi1_i6_i10 stop i1_i6_nstop dvdd dvdd vss vss i1_i6_nvc
       i1_i6_start rn dffr_x1_a8th
98 xi1_i6_i11 i1_i6_nen dvdd dvdd vss vss i1_i6_en inv_x2_a8th
99
100 xi1_i6_i12 i1_i6_en dvdd dvdd vss vss i1_i6_nstop i1_i6_nstop
       i1_i6_start i1_i6_start nand4_x2_a8th
101 xi1_i6_i13 i1_i6_nvc dvdd dvdd vss vss i1_i6_vc inv_x2_a8th
102 xi1_i6_i16 i1_i6_vc dvdd dvdd vss vss i1_st i1_nvcdig
       and2_x2_a8th
103 xi1_i6_i24 fck dvdd dvdd vss vss i1_i6_y i1_i6_nen and2_x2_a8th
104 xi1_i7 i1_net08 dvdd dvdd vss vss q[4] q[5] q[6] q[7] cxsel[0]
       cxsel[1] mx4_x1_a8th
105 xi1_i8 i1_net06 dvdd dvdd vss vss q[1] q[2] q[3] q[4] cnsel[0]
       cnsel[1] mx4_x1_a8th
106 xi1_i9 i1_net054 dvdd dvdd vss vss ntest i1_lat1 and2_x2_a8th
107 xi1_i10 i1_sel0 dvdd dvdd vss vss ntest i1_qlat0 and2_x2_a8th
108 xi1_i11 i1_net058 dvdd dvdd vss vss i1_net054 i1_nck
       and2_x2_a8th
109 xi1_i12 i1_nck dvdd dvdd vss vss ck inv_x2_a8th
110 xi1_i13 i1_nvcdig dvdd dvdd vss vss vcdig inv_x2_a8th
111 xi2_i0 i2_na dvdd dvdd vss vss ckprebuf inv_x1_a8th
112 xi2_i1 ck dvdd dvdd vss vss i2_na inv_x3_a8th
```

```
113 xi3_i0 i3_na dvdd dvdd vss vss rckprebuf inv_x1_a8th
114 xi3_i1 rck dvdd dvdd vss vss i3_na inv_x3_a8th
115 xi4_i0 i4_na dvdd dvdd vss vss stckprebuf inv_x1_a8th
116 xi4_i1 stck dvdd dvdd vss vss i4_na inv_x3_a8th
117 xi5_i0 i5_na dvdd dvdd vss vss enstckprebuf inv_x1_a8th
118 xi5_i1 enstck dvdd dvdd vss vss i5_na inv_x3_a8th
119 xi6_i0 i6_na dvdd dvdd vss vss vcdigprebuf inv_x1_a8th
120 xi6_i1 vcdig dvdd dvdd vss vss i6_na inv_x3_a8th
121 v0 vdd dvdd dc 0
122 v1 vdd rvdd dc 0
123 .ends viro_v0
```

## G.2 Vector inputs for relaxation oscillator

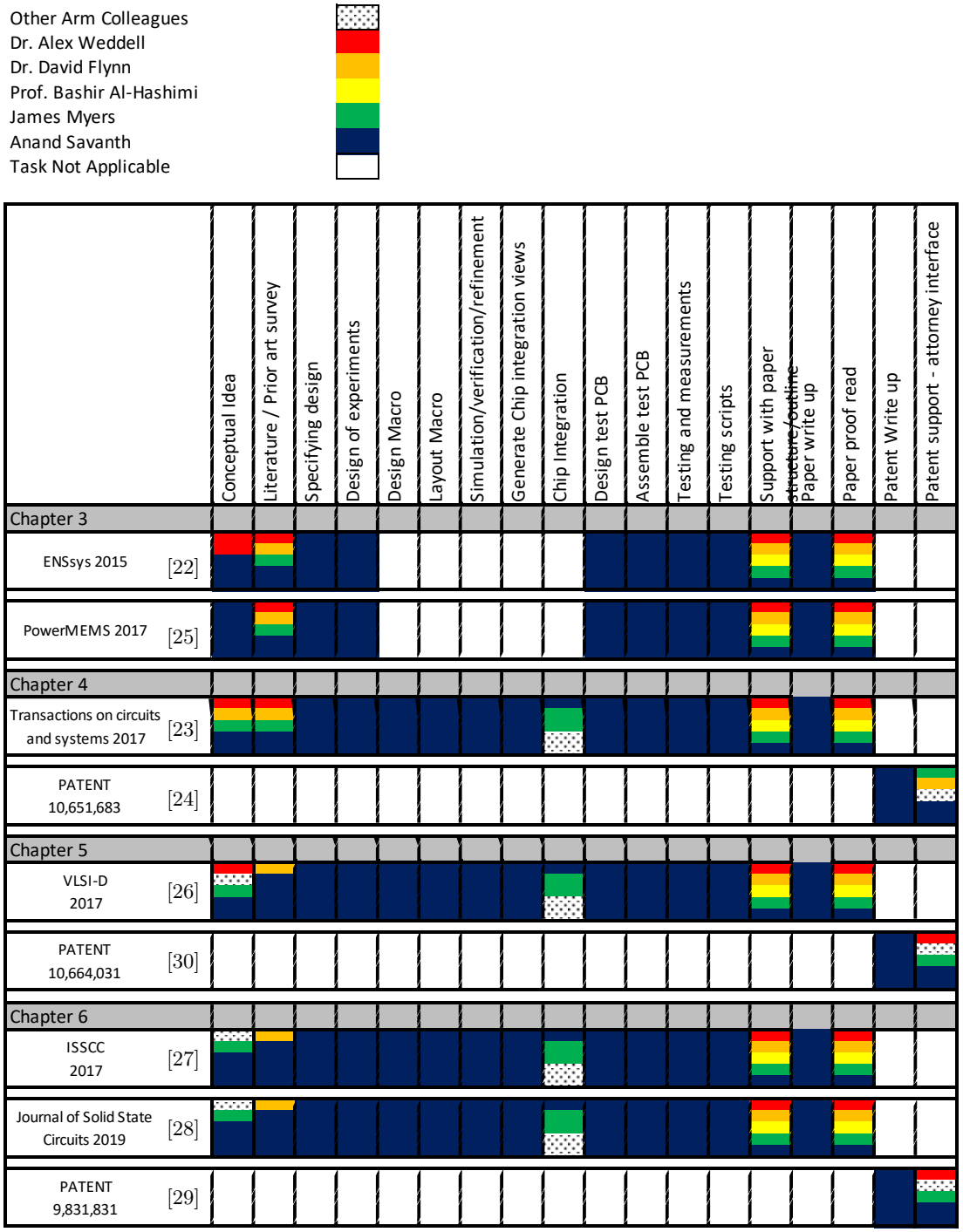
```
1  radix
2  + 1
3  + 1
4  + 1
5  + 1
6  + 1
7  + 1
8  + 1
9  + 2
10 + 2
11 + 2
12 + 2
13 + 2
14 + 2
15 + 44
16 + 44
17 + 44
18
19
20 vname
21 + csel
22 + en
23 + enduty
24 + enstm
25 + ntest
26 + ovr
27 + rn
28 + bsel[1:0]
29 + cnsel[1:0]
30 + cxsel[1:0]
31 + fcksel[1:0]
32 + rcksel[1:0]
33 + tsel[1:0]
34 + oset1[7:0]
35 + oset2[7:0]
36 + safe[7:0]
37
38 io
39 + i
40 + i
41 + i
42 + i
43 + i
44 + i
45 + i
46 + i
47 + i
48 + i
```

```
49 + i
50 + i
51 + i
52 + ii
53 + ii
54 + ii
55
56
57 tunit us
58 slope 0.001
59 vih   supply
60 vil   0
61
62 ;time csel  en  enduty  enstm  ntest  ovr  rn  bsel  cnsel
      cxsel  fcksel  rcksel  tsel   oset1  oset2  safe
63 0   0 1 1 0 0 1 0 3 0 0 0 0 3 00 00 00
```

# Appendix H

# Contribution of the author per chapter by task

The chart below lists the publications by chapter in the first column and the key tasks in subsequent columns. The colors in each box acknowledge the contributions of the author, supervisors and other colleagues at Arm as per the legend at the top left. For tasks highlighted as the author's own contributions, the supervisors' continuous guidance and contributions through discussions is acknowledged.




Dr. Alex Weddell

# Bibliography


[1] O. Vermesan, P. Friess, S. G. Patrick Guillemin, H. Sundmaeker, A. Bassi, I. S. Jubert, M. Mazura, M. E. Mark Harrison, P. Doody *et al.* (2011) Internet of Things Strategic Research Roadmap. [Online]. Available: http://www.internet-of-things-research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2011.pdf Last accessed Oct 2014.

[2] K. Ashton. (2009) That 'Internet of Things' Thing. [Online]. Available: http://www.rfidjournal.com/articles/view?4986 Last accessed Oct 2014.

[3] EPoSS. (2008) Internet of Things in 2020: Roadmap for the Future- European Commission Information Society and Media, Working Group RFID of the ETP EPoSS, Version 1.1. [Online]. Available: http://www.caba.org/resources/Documents/IS-2008-93.pdf Last accessed Sep 2020.

[4] Lu Tan and Neng Wang, "Future Internet: The Internet of Things," in *3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, vol. 5, Aug 2010, pp. 376–380.

[5] J. Manyika, M. Chui, J. Bughin, R. Dobbs, P. Bisson, , and A. Marrs. (2013) Disruptive Technologies: Advances That Will Transform Life, Business, and The Global Economy. [Online]. Available: http://www.mckinsey.com/insights/business_technology/disruptive_technologies Last accessed Sep 2020.

[6] J. Bradley, J. Barbier, and D. Handler. (2013) Embracing The Internet of Everything To Capture Your Share of $14.4 Trillion. [Online]. Available: http://www.cisco.com/web/about/ac79/docs/innov/IoE_Economy.pdf Last accessed Sep 2020.

[7] S. Higginbotham. (2010) Sensor Networks Top Social Networks for Big Data. [Online]. Available: https://gigaom.com/2010/09/13/sensor-networks-top-social-networks-for-big-data-2/ Last accessed Sep 2020.

[8] D. Anurag, S. Roy, and S. Bandyopadhyay, "Agro-sense: Precision Agriculture using Sensor-based Wireless Mesh Networks," in *First ITU-T Kaleidoscope Academic Conference - Innovations in NGN: Future Network and Services*, May 2008, pp. 383–388.

[9] D. Son, J. Lee, S. Qiao, R. Ghaffari, J. Kim, J. E. Lee, C. Song, S. J. Kim, D. J. Lee, S. W. Jun, S. Yang, M. Park, J. Shin, K. Do, M. Lee, K. Kang, C. S. Hwang, N. Lu, T. Hyeon, and D. H. Kim, "Multifunctional Wearable Devices for Diagnosis and Therapy of Movement Disorders," *Nature Nanotechnology*, vol. 9, pp. 397–404, 2014.

[10] R. Norman, J. Bolin, E. T. Powell, S. Amin, and J. Nacker, "Using Big Data Analytics to Create a Predictive Model for Joint Strike Fighter," in *IEEE International Conference on Big Data (Big Data)*, Dec 2018, pp. 3590–3596.

[11] S. Sudevalayam and P. Kulkarni, "Energy Harvesting Sensor Nodes: Survey and Implications," *IEEE Communications Surveys and Tutorials*, vol. 13, no. 3, pp. 443–461, Mar 2011.

[12] S. Peng and C. P. Low, "Throughput Optimal Energy Neutral Management for Energy Harvesting Wireless Sensor Networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Apr 2012, pp. 2347–2351.

[13] T. Voigt, H. Ritter, and J. Schiller, "Utilizing Solar Power in Wireless Sensor Networks," in *Proceedings of 28th Annual IEEE International Conference on Local Computer Networks*, Oct 2003, pp. 416–422.

[14] A. Klinefelter, N. E. Roberts, Y. Shakhsheer, P. Gonzalez, A. Shrivastava, A. Roy, K. Craig, M. Faisal, J. Boley, S. Oh, Y. Zhang, D. Akella, D. D. Wentzloff, and B. H. Calhoun, "A 6.45 μW Self-Powered IoT SoC with Integrated Energy-Harvesting Power Management and ULP Asymmetric Radios," in *IEEE International Solid-State Circuits Conference (ISSCC)*, Feb 2015, pp. 1–3.

[15] B. H. Calhoun, "Low Energy Digital Circuit Design using Sub-threshold Operation," Ph.D. dissertation, Massachusetts Institute of Technology, 2008. [Online]. Available: http://dspace.mit.edu/handle/1721.1/35527

[16] A. Chandrakasan, R. Min, M. Bhardwaj, S. Cho, and A. Wang, "Power Aware Wireless Microsensor Systems," in *Proceedings of the 28th European Solid-State Circuits Conference (ESSCIRC)*, Sep 2002, pp. 47–54.

[17] C. Lu, V. Raghunathan, and K. Roy, "Micro-Scale Energy Harvesting: A System Design Perspective," in *15th Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan 2010, pp. 89–94.

[18] M. Piñuela, P. D. Mitcheson, and S. Lucyszyn, "Ambient RF Energy Harvesting in Urban and Semi-Urban Environments," *IEEE Transactions on Microwave Theory and Techniques*, vol. 61, no. 7, pp. 2715–2726, 2013.

[19] S. Boisseau, G. Despesse, and B. A. Seddik, "Electrostatic Conversion for Vibration Energy Harvesting," in *Small-Scale Energy Harvesting*. 5 Princes Gate Court, London, SW7 2QJ: IntechOpen, 2012.

[20] J. Lu, H. Okada, T. Itoh, R. Maeda, and T. Harada, "Assembly of Super Compact Wireless Sensor Nodes for Environmental Monitoring Applications," in *IEEE Symposium on Design, Test, Integration and Packaging of MEMS/MOEMS (DTIP)*, Apr 2013, pp. 1–4.

[21] F. C. C. Manuel and N. F. Paulino, *CMOS Indoor Light Energy Harvesting System for Wireless Sensing Applications.* Springer International Publishing Switzerland, 2016.

[22] A. Savanth, A. S. Weddell, J. Myers, D. Flynn, and B. M. Al-Hashimi, "Photovoltaic Cells for Micro-Scale Wireless Sensor Nodes: Measurement and Modeling to Assist System Design," in *Proceedings of the 3rd International Workshop on Energy Harvesting & Energy Neutral Sensing Systems, ENSsys*, 2015, pp. 15–20. [Online]. Available: http://doi.acm.org/10.1145/2820645.2820653

[23] A. Savanth, A. S. Weddell, J. Myers, D. Flynn, and B. M. Al-Hashimi, "Integrated Reciprocal Conversion With Selective Direct Operation for Energy Harvesting Systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 9, pp. 2370–2379, Sep 2017.

[24] A. Savanth, J. Myers, and D. Flynn, "Harvesting Power from Ambient Energy in an Electronic Device," Patent 10,651,683, May 12, 2020.

[25] A. Savanth, M. Bellanger, A. Weddell, J. Myers, and M. Kauer, "Energy Neutral Sensor System With Micro-scale Photovoltaic and Thermoelectric Energy Harvesting," *Journal of Physics: Conference Series*, vol. 1052, p. 012069, Jul 2018. [Online]. Available: https://doi.org/10.1088/1742-6596/1052/1/012069

[26] A. Savanth, A. Weddell, J. Myers, D. Flynn, and B. Al-Hashimi, "A 50nW Voltage Monitor Scheme for Minimum Energy Sensor Systems," in *30th International Conference on VLSI Design and 16th International Conference on Embedded Systems (VLSID)*, Jan 2017, pp. 81–86.

[27] A. Savanth, J. Myers, A. Weddell, D. Flynn, and B. Al-Hashimi, "A 0.68nW/kHz Supply-Independent Relaxation Oscillator with ±0.49%/V and 96 ppm/°C Stability," in *IEEE International Solid-State Circuits Conference (ISSCC)*, Feb 2017, pp. 96–97.

[28] A. Savanth, A. S. Weddell, J. Myers, D. Flynn, and B. M. Al-Hashimi, "A Sub-nW/kHz Relaxation Oscillator With Ratioed Reference and Sub-Clock Power Gated Comparator," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 11, pp. 3097–3106, 2019.

[29] A. Savanth, S. Das, J. Myers, D. Bull, and B. Sandhu, "Integrated oscillator circuitry ," Patent 9,831,831, Nov. 28, 2017.

[30] A. Savanth, B. Sandhu, J. Myers, and A. Weddell, "Monitoring Circuit and Method," Patent 10,664,031, May 26, 2020.

[31] BlueSpark. (2011) Case Study: TempTraq Wearable Wireless Thermometer. [Online]. Available: http://www.bluesparktechnologies.com/index.php/products-and-services/temptraq Last accessed Sep 2020.

[32] A. Girolami, D. Brunelli, and L. Benini, "Low-cost and distributed health monitoring system for critical buildings," in *IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)*, 2017, pp. 1–6.

[33] Micropelt. (2015) MPG-D655 Thin Film Thermogenerator. [Online]. Available: http://micropelt.com/downloads/datasheet_mpg_d655.pdf Last accessed Sep 2020.

[34] S. Promwong and P. Panthap, "Experimental Evaluation of Complex form Friis' Transmission Formula with Indoor/Outdoor for Ultra Wideband Impulse Radio," in *International Conference on Computer and Communication Engineering*, 2008, pp. 1037–1041.

[35] R. Takahashi, H. Ichita, J. Tamura, M. Kimura, M. Ichinose, M. Futami, and K. Ide, "Efficiency Calculation of Wind Turbine Generation System with Doubly-fed Induction Generator," in *The XIX International Conference on Electrical Machines - ICEM*, 2010, pp. 1–4.

[36] GSES, *International Grid-Connected PV Systems Design and Installation: First Edition.* Global Sustainable Energy Solutions, 2018. [Online]. Available: https://books.google.co.uk/books?id=u1kIyAEACAAJ

[37] G. Myhre, D. Shindell, F. Breon, W. Collins, J. Fuglestvedt, J. Huang, D. Koch, J. Lamarque, D. Lee, B. Mendoza, T. Nakajima, A. Robock, G. Stephens, T. Takemura , and H. Zhang. (2013) Anthropogenic and Natural Radiative Forcing Supplementary Material. In: Climate Change 2013: The Physical Science Basis. Contribution of Working Group to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change. [Online]. Available: www.climatechange2013.org Last accessed Sep 2020.

[38] G. H. Yordanov, T. O. Saetre, and O. Midtgård, "1.6 suns at 58°20Ń - The Solar Resource in Southern Norway," in *IEEE 40th Photovoltaic Specialist Conference (PVSC)*, 2014, pp. 0815–0820.

[39] K. N. Nguyen, D. Abi-Saab, P. Basset, E. Richalot, F. Marty, D. Angelescu, Y. Leprince-Wang, and T. Bourouina, "Black Silicon with sub-Percent Reflectivity: Influence of the 3D Texturization Geometry," in *16th International Solid-State Sensors, Actuators and Microsystems Conference*, 2011, pp. 354–357.

[40] O. D. Miller, E. Yablonovitch, and S. R. Kurtz, "Strong internal and external luminescence as solar cells approach the shockley–queisser limit," *IEEE Journal of Photovoltaics*, vol. 2, no. 3, pp. 303–311, 2012.

[41] I. Stuart, *Solar Cells and Photovoltaics*. Springer Nature, Switzerland: Springer Handbooks, 2017.

[42] IXYS Korea Ltd. (2019) IXYS Solar Products, Product Brief. [Online]. Available: https://ixapps.ixys.com/DataSheet/IXYS%20Solar%20Product%20Brief.pdf Last accessed July 2020.

[43] S. Beeby and N. White, *PhotovoltaicEnergyHarvesting*. Artech House, 2010.

[44] M. Fojtik, D. Kim, G. Chen, Y.-S. Lin, D. Fick, J. Park, M. Seok, M.-T. Chen, Z. Foo, D. Blaauw, and D. Sylvester, "A Millimeter-Scale Energy-Autonomous Sensor System with Stacked Battery and Solar Cells," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 3, pp. 801–813, Mar 2013.

[45] M. E. V. Valkenburg, *Network Analysis*, ser. Electrical Engineering series. Prentice-Hall Incorporated, 1955.

[46] S. Bandyopadhyay and A. Chandrakasan, "Platform Architecture for Solar, Thermal, and Vibration Energy Combining with MPPT and Single Inductor," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 9, pp. 2199–2215, Sep 2012.

[47] A. N. A. Ali, M. H. Saied, M. Z. Mostafa, and T. M. Abdel- Moneim, "A Survey of Maximum PPT Techniques of PV Systems," in *IEEE Energytech*, 2012, pp. 1–17.

[48] T. Esram and P. Chapman, "Comparison of Photovoltaic Array Maximum Power Point Tracking Techniques," *IEEE Transactions on Energy Conversion*, vol. 22, no. 2, pp. 439–449, Jun 2007.

[49] D. Linden and T. Reddy, *Handbook of Batteries*, ser. McGraw-Hill Handbooks. McGraw-Hill, 2002.

[50] C. Corporation. (2009) Ultra Low Power Microcontroller Backup Using The EnerChip CC. [Online]. Available: http://www.cymbet.com/pdfs/AN-1027.pdf Last accessed Sep 2020.

[51] H. Yang and Y. Zhang, "Analysis of Supercapacitor Energy Loss for Power Management in Environmentally Powered Wireless Sensor Nodes," *IEEE Transactions on Power Electronics*, vol. 28, no. 11, pp. 5391–5403, Nov 2013.

[52] M. Rashid, *Power Electronics: Circuits, Devices, and Applications*. Pearson Education, 2009.

[53] G. Villar-Pique, H. Bergveld, and E. Alarcon, "Survey and Benchmark of Fully Integrated Switching Power Converters: Switched-Capacitor Versus Inductive Approach," *IEEE Transactions on Power Electronics*, vol. 28, no. 9, pp. 4156–4167, Sep 2013.

[54] G. D. Szarka, S. G. Burrow, and B. H. Stark, "Ultralow Power, Fully Autonomous Boost Rectifier for Electromagnetic Energy Harvesters," *IEEE Transactions on Power Electronics*, vol. 28, no. 7, pp. 3353–3362, 2013.

[55] M. Seeman and S. Sanders, "Analysis and Optimization of Switched-Capacitor DC-DC Converters," *IEEE Transactions on Power Electronics*, vol. 23, no. 2, pp. 841–851, Mar 2008.

[56] H.-P. Le, S. Sanders, and E. Alon, "Design Techniques for Fully Integrated Switched-Capacitor DC-DC Converters," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 9, pp. 2120–2131, Sep 2011.

[57] W. T. Harrigill and I. T. Myers, "Efficiency and Weight of Voltage Multiplier Type Ultra Lightweight DC-DC Converters," in *IEEE Power Electronics Specialists Conference*, 1975, pp. 31–37.

[58] J. F. Dickson, "On-chip High-Voltage Generation In MNOS Integrated Circuits Using An Improved Voltage Multiplier Technique," *IEEE Journal of Solid-State Circuits*, vol. 11, no. 3, pp. 374–378, 1976.

[59] M. S. Makowski and D. Maksimovic, "Performance limits of switched-capacitor DC-DC converters," in *Proceedings of PESC '95 - Power Electronics Specialist Conference*, vol. 2, 1995, pp. 1215–1221 vol.2.

[60] Y. K. Ramadass, "Energy Processing Circuits for Low-Power Applications," Ph.D. dissertation, Massachusetts Institute of Technology, 2009. [Online]. Available: http://dspace.mit.edu/handle/1721.1/63026

[61] H.-P. Le, M. Seeman, S. Sanders, V. Sathe, S. Naffziger, and E. Alon, "A 32nm Fully Integrated Reconfigurable Switched-Capacitor DC-DC Converter Delivering 0.55W/mm$^2$ At 81% Efficiency," in *IEEE International Solid-State Circuits Conference (ISSCC)*, Feb 2010, pp. 210–211.

[62] Y. Ramadass, A. Fayed, and A. Chandrakasan, "A Fully-Integrated Switched-Capacitor Step-Down DC-DC Converter with Digital Capacitance Modulation in 45 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 12, pp. 2557–2565, Dec 2010.

[63] M. Wieckowski, G. Chen, M. Seok, D. Blaauw, and D. Sylvester, "A Hybrid DC-DC Converter for Sub-Microwatt Sub-1 V Implantable Applications," in *Symposium on VLSI Circuits (VLSIC)*, Jun 2009, pp. 166–167.

[64] T. Andersen, F. Krismer, J. Kolar, T. Toifl, C. Menolfi, L. Kull, T. Morf, M. Kossel, M. Brandli, P. Buchmann, and P. Francese, "A Deep Trench Capacitor Based 2:1 and 3:2 Reconfigurable On-Chip Switched Capacitor DC-DC Converter in 32nm SOI CMOS," in *IEEE Applied Power Electronics Conference and Exposition (APEC)*, Mar 2014, pp. 1448–1455.

[65] T. Van Breussegem and M. Steyaert, "Monolithic Capacitive DC-DC Converter with Single Boundary Multiphase Control and Voltage Domain Stacking in 90nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 7, pp. 1715–1727, Jul 2011.

[66] S. M. Douglas, "A Design Methodology for Switched-Capacitor DC-DC Converters," Ph.D. dissertation, EECS Department, University of California, Berkeley, 2009. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-78.html

[67] Y. Ramadass and A. Chandrakasan, "Minimum Energy Tracking Loop with Embedded DC-DC Converter Enabling Ultra-Low-Voltage Operation Down to 250 mV in 65 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 256–265, Jan 2008.

[68] N. Mehta and K. Makinwa, "Minimum Energy Point Tracking for Sub-Threshold Digital CMOS Circuits Using An In-situ Energy Sensor," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2013, pp. 570–573.

[69] J. Kwong, Y. Ramadass, N. Verma, M. Koesler, K. Huber, H. Moormann, and A. Chandrakasan, "A 65nm Sub-Vt Microcontroller with Integrated SRAM and Switched-Capacitor DC-DC Converter," in *IEEE International Solid-State Circuits Conference (ISSCC)*, Feb 2008, pp. 318–616.

[70] B. Zhai, S. Pant, L. Nazhandali, S. Hanson, J. Olson, A. Reeves, M. Minuth, R. Helfand, T. Austin, D. Sylvester, and D. Blaauw, "Energy-Efficient Subthreshold Processor Design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 8, pp. 1127–1137, Aug 2009.

[71] S. Lutkemeier, T. Jungeblut, H. Berge, S. Aunet, M. Porrmann, and U. Ruckert, "A 65 nm 32b Subthreshold Processor with 9T Multi-VT SRAM and Adaptive Supply Voltage Control," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 8–19, Jan 2013.

[72] D. Bol, J. De Vos, C. Hocquet, F. Botman, F. Durvaux, S. Boyd, D. Flandre, and J. Legat, "SleepWalker: A 25-MHz 0.4-V Sub-$mm^2$ 7µW/MHz Microcontroller in 65-nm LP/GP CMOS for Low-Carbon Wireless Sensor Nodes," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 20–32, Jan 2013.

[73] J. Myers, A. Savanth, D. Howard, R. Gaddh, P. Prabhat, and D. Flynn, "An 80nW Retention 11.7pJ/Cycle Active Subthreshold ARM Cortex-M0+ Subsystem in 65nm CMOS for WSN Applications," in *IEEE International Solid-State Circuits Conference (ISSCC)*, Feb 2015, pp. 1–3.

[74] Y.-S. Lin, D. Sylvester, and D. Blaauw, "A Sub-pW Timer Using Gate Leakage for Ultra Low-Power Sub-Hz Monitoring Systems," in *IEEE Custom Integrated Circuits Conference (CICC)*, Sep 2007, pp. 397–400.

[75] A. Shrivastava and B. Calhoun, "A 150nW, 5ppm/°C, 100kHz On-Chip Clock Source for Ultra Low Power SoCs," in *IEEE Custom Integrated Circuits Conference (CICC)*, Sep 2012, pp. 1–4.

[76] A. A. Abidi and R. G. Meyer, "Noise in Relaxation Oscillators," *IEEE Journal of Solid-State Circuits*, vol. 18, no. 6, pp. 794–802, Dec 1983.

[77] K. Choe, O. Bernal, D. Nuttman, and M. Je, "A Precision Relaxation Oscillator with A Self-Clocked Offset-Cancellation Scheme for Implantable Biomedical SoCs," in *IEEE International Solid-State Circuits Conference (ISSCC)*, Feb 2009, pp. 402–403,403a.

[78] Y. Tokunaga, S. Sakiyama, A. Matsumoto, and S. Dosho, "An On-Chip CMOS Relaxation Oscillator with Power Averaging Feedback Using A Reference Proportional To Supply Voltage," in *IEEE International Solid-State Circuits Conference (ISSCC)*, Feb 2009, pp. 404–405,405a.

[79] J. Lee, P. Park, S. Cho, and M. Je, "A 4.7MHz 53 μW Fully Differential CMOS Reference Clock Oscillator with -22dB Worst-Case PSNR for Miniaturized SoCs," in *IEEE International Solid-State Circuits Conference (ISSCC)*, Feb 2015, pp. 1–3.

[80] P. M. Nadeau, A. Paidimarri, and A. P. Chandrakasan, "Ultra Low-Energy Relaxation Oscillator with 230 fJ/Cycle Efficiency," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 4, pp. 789–799, Apr 2016.

[81] B. Mishra, C. Botteron, G. Tasselli, C. Robert, and P. A. Farine, "A sub-μA Power Management Circuit in 0.18μm CMOS for Energy Harvesters," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, Mar 2013, pp. 1197–1202.

[82] H. B. Le, X. D. Do, S. G. Lee, and S. T. Ryu, "A Long Reset-Time Power-On Reset Circuit with Brown-Out Detection Capability," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, no. 11, pp. 778–782, Nov 2011.

[83] I. Lee, S. Bang, Y. Lee, Y. Kim, G. Kim, D. Sylvester, and D. Blaauw, "A 635pW Battery Voltage Supervisory Circuit for Miniature Sensor Nodes," in *Symposium on VLSI Circuits (VLSIC)*, Jun 2012, pp. 202–203.

[84] J. Guo, W. Shi, K. N. Leung, and C. S. Choy, "Power-On-Reset Circuit with Power-off Auto-Discharging Path for Passive RFID Tag ICs," in *53rd IEEE International Midwest Symposium on Circuits and Systems*, Aug 2010, pp. 21–24.

[85] I. Lee, Y. Lee, D. Sylvester, and D. Blaauw, "Low Power Battery Supervisory Circuit with Adaptive Battery Health Monitor," in *Symposium on VLSI Circuits (VLSIC)*, Jun 2014, pp. 1–2.

[86] J. Jeong and D. Culler, "A Practical Theory of Micro-Solar Power Sensor Networks," *ACM Trans. Sen. Netw.*, vol. 9, no. 1, pp. 9:1–9:36, Nov 2012. [Online]. Available: http://doi.acm.org/10.1145/2379799.2379808

[87] J. Janhunen, K. Mikhaylov, J. Petajajarvi, and M. Sonkki, "Wireless Energy Transfer Powered Wireless Sensor Node for Green IoT: Design, Implementation and Evaluation," *Sensors*, vol. 19, p. 90, 12 2018.

[88] J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.

[89] Y.-P. Huang, "A Rapid Maximum Power Measurement System for High-Concentration Photovoltaic Modules Using The Fractional Open-Circuit Voltage Technique and Controllable Electronic Load," *IEEE Journal of Photovoltaics*, vol. 4, no. 6, pp. 1610–1617, Nov 2014.

[90] U. Zimmermann and M. Edoff, "A Maximum Power Point Tracker for Long-Term Logging of PV Module Performance," *IEEE Journal of Photovoltaics*, vol. 2, no. 1, pp. 47–55, Jan 2012.

[91] D. Dondi, D. Brunelli, L. Benini, P. Pavan, A. Bertacchini, and L. Larcher, "Photovoltaic Cell Modeling for Solar Energy Powered Sensor Networks," in *2nd International Workshop on Advances in Sensors and Interface (IWASI)*, Jun 2007, pp. 1–6.

[92] M. Taciuc, "An Experimental System for Measuring the PV Panel Characteristics Curves Under Real Operation Conditions," in *International Symposium on Fundamentals of Electrical Engineering (ISFEE)*, 2016, pp. 1–6.

[93] J. Hester, T. Scott, and J. Sorber, "Ekho: Realistic and Repeatable Experimentation for Tiny Energy-Harvesting Sensors," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '14. New York, NY, USA: ACM, 2014, pp. 1–15. [Online]. Available: http://doi.acm.org/10.1145/2668332.2668336

[94] S. Bobovych, N. Banerjee, R. Robucci, J. P. Parkerson, J. Schmandt, and C. Patel, "SunaPlayer: High-Accuracy Emulation of Solar Cells," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*. ACM, 2015, pp. 59–70.

[95] J.-E. Jeong, J.-H. Bae, J. Lee, C. S. Lee, J.-H. Chun, and K.-W. Kwon, "A Photovoltaic Power Management System Using A Luminance-Controlled Oscillator for USN Applications," *Journal of Semiconductor Technology and Science*, vol. 13, no. 1, pp. 48–57, 2013. [Online]. Available: http://www.dbpia.co.kr/Article/3108507

[96] J. Sarik, K. Kim, M. Gorlatova, I. Kymissis, and G. Zussman, "More Than Meets The Eye - A Portable Measurement Unit for Characterizing Light Energy Availability," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec 2013, pp. 387–390.

[97] Keysight Technologies. (2015) 34410A and 34411A Multimeters. [Online]. Available: https://www.keysight.com/us/en/assets/7018-01326/data-sheets/5989-3738.pdf Last accessed July 2020.

[98] M. C. D. Piazza and G. Vitale, *Photovoltaic Sources - Modeling and Emulation.* Verlag, London: Springer, 2013.

[99] M. Glass, "Improved Solar Array Power Point Model with SPICE Realization," in *Proceedings of the 31st Intersociety Energy Conversion Engineering Conference (IECEC)*, vol. 1, Aug 1996, pp. 286–291 vol.1.

[100] D. Chan and J. Phang, "Analytical Methods for the Extraction of Solar-Cell Single- and Double-Diode Model Parameters From I-V Characteristics," *IEEE Transactions on Electron Devices*, vol. 34, no. 2, pp. 286–293, Feb 1987.

[101] W. Jung, S. Oh, S. Bang, Y. Lee, Z. Foo, G. Kim, Y. Zhang, D. Sylvester, and D. Blaauw, "An Ultra-Low Power Fully Integrated Energy Harvester Based On Self-Oscillating Switched-Capacitor Voltage Doubler," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 12, pp. 2800–2811, Dec 2014.

[102] K. Kadirvel, Y. Ramadass, U. Lyles, J. Carpenter, V. Ivanov, V. McNeil, A. Chandrakasan, and B. Lum-Shue-Chan, "A 330nA Energy-Harvesting Charger with Battery Management for Solar and Thermoelectric Energy Harvesting," in *IEEE International Solid-State Circuits Conference (ISSCC)*, Feb 2012, pp. 106–108.

[103] EEMBC. (2015) ULPBench. [Online]. Available: http://www.eembc.org/ulpbench/ Last accessed Oct 2017.

[104] A. Wang and A. Chandrakasan, "A 180-mV Subthreshold FFT Processor Using A Minimum Energy Design Methodology," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 1, pp. 310–319, Jan 2005.

[105] X. Liu and E.Sánchez-Sinencio, "An 86% Efficiency 12μW Self-Sustaining PV Energy Harvesting System with Hysteresis Regulation and Time-Domain MPPT for IOT Smart Nodes," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 6, pp. 1424–1437, Jun 2015.

[106] D. Bol, E. H. Boufouss, D. Flandre, and J. De Vos, "A 0.48mm 25µW-10mW Indoor/Outdoor PV Energy-Harvesting Management Unit in a 65nm SoC Based on a Single Bidirectional Multi-gain/Multi-mode Switched-cap Converter with Supercap Storage," in *41st European Solid-State Circuits Conference (ESSCIRC)*, Sep 2015, pp. 241–244.

[107] S. Adami, G. Yang, C. Zhang, P. Proynov, and B. H. Stark, "A 10 nW, 10 mV Signal Detector using a 2 pA Standby Voltage Reference, for Always-on Sensors and Receivers," in *IEEE Applied Power Electronics Conference and Exposition (APEC)*, March 2018, pp. 1065–1070.

[108] Y. Kim, Y. Lee, D. Sylvester, and D. Blaauw, "SLC: Split-Control Level Converter for Dense and Stable Wide-Range Voltage Conversion," in *European Solid-State Circuits Conference (ESSCIRC)*, Sep 2012, pp. 478–481.

[109] Di-LOG. (2009) DL7030. [Online]. Available: https://www.tlc-direct.co.uk/Technical/DataSheets/Dilog/TMDL7030_Manual.pdf Last accessed Sep 2020.

[110] Microchip Technology inc. (2009) MCP9700. [Online]. Available: http://ww1.microchip.com/downloads/en/devicedoc/21942e.pdf Last accessed Sep 2020.

[111] CREE. (2009) Cree CLN6A-WKW/MKW Data Sheet. [Online]. Available: http://docs-europe.electrocomponents.com/webdocs/0cdd/0900766b80cddec3.pdf\right Last accessed Sep 2020.

[112] J. Quill, G. Fedor, P. Brennan, and E. Everett. (2007) Quantifying The Indoor Light Environment. [Online]. Available: http://www.q-lab.com/documents/public/1e85c327-1aa2-4e87-808e-df04653e4d5e.pdf Last accessed Sep 2020.

[113] G. Yang, B. H. Stark, S. J. Hollis, and S. G. Burrow, "Challenges for Energy Harvesting Systems Under Intermittent Excitation," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 4, no. 3, pp. 364–374, Sep. 2014.

[114] Y. Cai, Z. Zhang, Y. Zhang, and Y. Liu, "A Self-Reconfiguration Control Regarding Recovery Effect to Improve the Discharge Efficiency in the Distributed Battery Energy Storage System," in *IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2015, pp. 1774–1778.

[115] S. Bang, Y. Lee, I. Lee, Y. Kim, G. Kim, D. Blaauw, and D. Sylvester, "A Fully Integrated Switched-Capacitor Based PMU with Adaptive Energy Harvesting Technique for Ultra-Low Power Sensing Applications," in *IEEE International Symposium on Circuits and Systems (ISCAS2013)*, May 2013, pp. 709–712.

[116] Y. C. Shih and B. P. Otis, "An Inductorless DC-DC Converter for Energy Harvesting with A 1.2-µW Bandgap-Referenced Output Controller," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, no. 12, pp. 832–836, Dec 2011.

[117] Y. Qiu, C. V. Liempd, B. O. het Veld, P. G. Blanken, and C. V. Hoof, "5μW-to-10mW Input Power Range Inductive Boost Converter for Indoor Photovoltaic Energy Harvesting with Integrated Maximum Power Point Tracking Algorithm," in *IEEE International Solid-State Circuits Conference*, Feb 2011, pp. 118–120.

[118] P. H. Chen, X. Zhang, K. Ishida, Y. Okuma, Y. Ryu, M. Takamiya, and T. Sakurai, "An 80 mV Startup Dual-Mode Boost Converter By Charge-Pumped Pulse Generator and Threshold Voltage Tuned Oscillator with Hot Carrier Injection," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 11, pp. 2554–2562, Nov 2012.

[119] H. Kim, S. Kim, C. K. Kwon, Y. J. Min, C. Kim, and S. W. Kim, "An Energy-Efficient Fast Maximum Power Point Tracking Circuit in an 800-μW Photovoltaic Energy Harvester," *IEEE Transactions on Power Electronics*, vol. 28, no. 6, pp. 2927–2935, Jun 2013.

[120] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*. ACM, 2002, pp. 88–97.

[121] K. Chintalapudi, E. A. Johnson, and R. Govindan, "Structural Damage Detection Using Wireless Sensor-Actuator Networks," in *Proceedings of the IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control*, Jun 2005, pp. 322–327.

[122] E. S. Biagioni and K. W. Bridges, "The Application of Remote Sensor Technology to Assist the Recovery of Rare and Endangered Species," *International Journal of High Performance Computing Applications*, vol. 16, p. 2002, 2002.

[123] S. Roundy, P. K. Wright, and J. Rabaey, "A Study of Low Level Vibrations as a Power Source for Wireless Sensor Nodes," *Computer Communications*, vol. 26, no. 11, pp. 1131 – 1144, 2003. [Online]. Available: 'http://www.sciencedirect.com/science/article/pii/S0140366402002487'

[124] S. Escolara, S. Chessab, and J. Carreteroa, "Energy-Neutral Networked Wireless Sensors," *Simulation Modelling Practice and Theory*, vol. 43, pp. 1 – 15, 2014. [Online]. Available: 'http://www.sciencedirect.com/science/article/pii/S1569190X14000033'

[125] J. Myers, A. Savanth, R. Gaddh, D. Howard, P. Prabhat, and D. Flynn, "A Subthreshold ARM Cortex-M0+ Subsystem in 65 nm CMOS for WSN Applications with 14 Power Domains, 10T SRAM, and Integrated Voltage Regulator," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 1, pp. 31–44, 2016.

[126] Y. Pu, X. Zhang, J. Huang, A. Muramatsu, M. Nomura, K. Hirairi, H. Takata, T. Sakurabayashi, S. Miyano, M. Takamiya, and T. Sakurai, "Misleading Energy

and Performance Claims in Sub/Near Threshold Digital Systems," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2010, pp. 625–631.

[127] MAX8212, Maxim Integrated. (2008) Microprocessor Voltage Monitors with Programmable Voltage Detection. [Online]. Available: https://datasheets.maximintegrated.com/en/ds/MAX8211-MAX8212.pdf Last accessed July 2016.

[128] MAX1056, Maxim Integrated. (2015) 125nA Supervisory Circuits with Capacitor-Adjustable Reset and Watchdog Timeouts. [Online]. Available: https://datasheets.maximintegrated.com/en/ds/MAX16056-MAX16059.pdf Last accessed July 2016.

[129] J. Lee and S. Cho, "A 1.4-μW 24.9-ppm/°C Current Reference With Process-Insensitive Temperature Compensation in 0.18-μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 10, pp. 2527–2533, Oct 2012.

[130] X. Zou, X. Xu, L. Yao, and Y. Lian, "A 1-V 450-nW Fully Integrated Programmable Biomedical Sensor Interface Chip," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1067–1077, Apr 2009.

[131] J. Yin, J. Yi, M. K. Law, Y. Ling, M. C. Lee, K. P. Ng, B. Gao, H. C. Luong, A. Bermak, M. Chan, W. Ki, C. Tsui, and M. Yuen, "A System-on-Chip EPC Gen-2 Passive UHF RFID Tag With Embedded Temperature Sensor," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 11, pp. 2404–2420, Nov 2010.

[132] S. Jeong, I. Lee, D. Blaauw, and D. Sylvester, "A 5.8nW, 45ppm/°C On-chip CMOS Wake-up Timer Using a Constant Charge Subtraction Scheme," in *IEEE Custom Integrated Circuits Conference (CICC)*, Sep 2014, pp. 1–4.

[133] Y. Lin, D. M. Sylvester, and D. T. Blaauw, "A 150pW Program-and-Hold Timer for Ultra-Low-Power Sensor Platforms," in *IEEE International Solid-State Circuits Conference (ISSCC)*, Feb 2009, pp. 326–327,327a.

[134] Y. Lee, B. Giridhar, Z. Foo, D. Sylvester, and D. B. Blaauw, "A Sub-nW Multi-stage Temperature Compensated Timer for Ultra-Low-Power Sensor Nodes," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 10, pp. 2511–2521, Oct 2013.

[135] T. Jang, M. Choi, S. Jeong, S. Bang, D. Sylvester, and D. Blaauw, "A 4.7nW 13.8ppm/°C Self-Biased Wakeup Timer Using A Switched-Resistor Scheme," in *IEEE International Solid-State Circuits Conference (ISSCC)*, Jan 2016, pp. 102–103.

[136] U. Denier, "Analysis and Design of An Ultra-Low-Power CMOS Relaxation Oscillator," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 8, pp. 1973–1982, Aug 2010.

[137] A. Paidimarri, D. Griffith, A. Wang, A. P. Chandrakasan, and G. Burra, "A 120nW 18.5kHz RC Oscillator with Comparator offset Cancellation for 0.25% Temperature Stability," in *IEEE International Solid-State Circuits Conference (ISSCC)*, Feb 2013, pp. 184–185.

[138] D. Griffith, P. T. Rine, J. Murdock, and R. Smith, "A 190nW 33kHz RC Oscillator with 0.21% Temperature Stability and 4ppm Long-Term Stability," in *IEEE International Solid-State Circuits Conference (ISSCC)*, Feb 2014, pp. 300–301.

[139] K. J. Hsiao, "A 32.4 ppm/°C 3.2-1.6V Self-Chopped Relaxation Oscillator with Adaptive Supply Generation," in *Symposium on VLSI Circuits (VLSIC)*, Jun 2012, pp. 14–15.

[140] T. Tokairin, K. Nose, K. Takeda, K. Noguchi, T. Maeda, K. Kawai, and M. Mizuno, "A 280nW, 100kHz, 1-Cycle Start-up Time, On-chip CMOS Relaxation Oscillator Employing a Feedforward Period Control Scheme," in *Symposium on VLSI Circuits (VLSIC)*, Jun 2012, pp. 16–17.

[141] M. Choi, T. Jang, S. Bang, Y. Shi, D. Blaauw, and D. Sylvester, "A 110 nW Resistive Frequency Locked On-Chip Oscillator with 34.3 ppm/°C Temperature Stability for System-on-Chip Designs," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 9, pp. 2106–2118, Sep 2016.

[142] F. Sebastiano, L. Breems, K. M. S. Drago, D. Leenaerts, and B. Nauta, "A Low-Voltage Mobility-Based Frequency Reference for Crystal-Less ULP Radios," in *34th European Solid-State Circuits Conference, ESSCIRC*, Sep 2008, pp. 306–309.

[143] K. Lasanen, E. Raisanen-Ruotsalainen, and J. Kostamovaara, "A 1-V, Self Adjusting, 5-MHz CMOS RC-Oscillator," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 4, 2002, pp. IV–377–IV–380 vol.4.

[144] K. Sundaresan, P. E. Allen, and F. Ayazi, "Process and Temperature Compensation in a 7-MHz CMOS Clock Oscillator," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 2, pp. 433–442, Feb 2006.

[145] S. Dai and J. K. Rosenstein, "A 14.4nW 122kHz Dual-phase Current-mode Relaxation Oscillator for Near-zero-power Sensors," in *IEEE Custom Integrated Circuits Conference (CICC)*, Sep 2015, pp. 1–4.

[146] H. Jiang, P. P. Wang, P. P. Mercier, and D. A. Hall, "A 0.4-V 0.93-nW/kHz Relaxation Oscillator Exploiting Comparator Temperature-Dependent Delay to Achieve 94-ppm/°C Stability," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 10, pp. 3004–3011, Oct 2018.

[147] R. Navid, T. H. Lee, and R. W. Dutton, "Minimum Achievable Phase Noise of RC Oscillators," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 3, pp. 630–637, Mar 2005.

[148] J. N. Mistry, J. Myers, B. M. Al-Hashimi, D. Flynn, J. Biggs, and G. V. Merrett, "Active Mode Subclock Power Gating," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 9, pp. 1898–1908, Sep 2014.

[149] T. Le, K. Mayaram, and T. Fiez, "Efficient Far-Field Radio Frequency Energy Harvesting for Passively Powered Sensor Networks," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 5, pp. 1287–1302, May 2008.

[150] D. J. Yeager, P. S. Powledge, R. Prasad, D. Wetherall, and J. R. Smith, "Wirelessly-Charged UHF Tags for Sensor Data Collection," in *IEEE International Conference on RFID*, Apr 2008, pp. 320–327.

[151] A. Colin, A. P. Sample, and B. Lucia, "Energy-Interference-Free System and Toolchain Support for Energy-Harvesting Devices," in *International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES)*, Oct 2015, pp. 35–36.

[152] D. Balsamo, A. S. Weddell, G. V. Merrett, B. M. Al-Hashimi, D. Brunelli, and L. Benini, "Hibernus: Sustaining Computation During Intermittent Supply for Energy-Harvesting Systems," *IEEE Embedded Systems Letters*, vol. 7, no. 1, pp. 15–18, Mar 2015.

[153] M. Hicks, "Clank: Architectural Support for Intermittent Computation," in *ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, Jun 2017, pp. 228–240.