

Review

A Top-Down Approach to Building Battery-Less Self-Powered Systems for the Internet-of-Things

Farah B. Yahya ^{1,2,*}, Christopher J. Lukas ^{1,2}  and Benton H. Calhoun ^{1,2}¹ Psikick, Charlottesville, VA 22902, USA; cj14hd@virginia.edu (C.J.L.); bcalhoun@virginia.edu (B.H.C.)² Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22904, USA

* Correspondence: fby5bb@virginia.edu

Received: 24 May 2018; Accepted: 12 June 2018; Published: 15 June 2018



Abstract: This paper presents a top-down methodology for designing battery-less systems for the Internet-of-Things (IoT). We start by extracting features from a target IoT application and the environment in which it will be deployed. We then present strategies to translate these features into design choices that optimize the system and improve its reliability. We look into how to use these features to build the digital sub-system by determining the blocks to implement, the digital architecture, the clock rate of the system, the memory capacity, and the low power states. We also review how these features impact the choice of energy harvesting power management units.

Keywords: Internet-of-Things; battery-less; self-powered; ultra-low power; systems-on-chip; energy harvesting

1. Introduction

In recent years, a large number of devices have been introduced as part of the Internet-of-Things (IoT) to sense new data, analyze it, and provide new insights. These devices target applications such as industrial/asset monitoring, health monitoring, environmental monitoring and many others. Each category of devices operates under different conditions and has differing constraints. For example, a wearable heart rate monitor must sample electrocardiogram (ECG) signals at a rate of 50 Hz or more [1,2], must have a small form factor to remain unobtrusive, and must have a long lifetime to enable continuous monitoring. A room temperature monitor, on the other hand, can have a much more relaxed sampling rate (once per second or less would be enough to detect changes) and form factor but would benefit from a long lifetime to reduce maintenance costs. These varying requirements and constraints make the design of systems-on-chips (SoCs) for the IoT a challenge. While many IoT systems are available in the literature [1–6], they often only offer point solutions for a single (or a class of) sensing application(s) without presenting the process followed to arrive at the different design decisions implemented, thus making it difficult for future designers to generalize their solution to other applications. To the authors' best knowledge, a methodology that uses an application's constraints to inform low-level circuit design decisions has not been presented yet. In this paper, we describe the classification of different IoT applications based on their features. Then, we show how to use these high-level features to inform design decisions when building an IoT SoC.

A common concern between many IoT devices is the need for a long lifetime. To improve the lifetime of IoT devices, SoC designers aim to both reduce the power consumption of their circuits and to harvest the required energy from the environment onto a storage element. Storage elements include batteries or super-capacitors. Batteries provide a higher storage capacity than super-capacitors, but their lifetime is limited by the number of charging cycles. While they are still the dominant energy storage element in commercial devices, the use of battery-less, self-powered systems using super-capacitors [4–7] is gaining momentum. Such systems can offer a unique advantage in terms of

lifetime and power but must offer reliability, especially under poor harvesting conditions. To improve the reliability of such systems, their power budget must be constrained based on the environment in which they are deployed. The wearable heart rate monitor, from the previous example, can harvest energy from the body's heat as well as from the surrounding light whereas the room temperature monitor relies on the surrounding light only. Thus, the design choices for the energy harvesting power management unit will change based on the high-level features of the IoT application. The available energy for each monitor will also vary based on the time of day/year, which also impacts the design of the SoC. In the proposed methodology, we address how environmental conditions impact the choice of energy harvesting units and power state management.

This paper presents the first top-down methodology for designing reliable battery-less SoCs for the IoT by using the application's features to tightly integrate components targeting that application. We start by giving an overview of the methodology. Then, we look at different IoT sensing applications and extract the main features that affect the design of the SoC. Next, we look at the different environments where the sensors will be deployed and extract the features of these environments. Then, we present a new approach to translate these features into architectural design decisions for the digital and energy harvesting power management sub-systems. Finally, we conclude with a summary of the proposed methodology.

2. Overview

Figure 1 shows the proposed top-down methodology used to design battery-less SoCs. The procedure starts by determining the target IoT sensing application. From the application, the required sensing interfaces, the required sensing rate, and the required sampling scheme are extracted. The sensing interface could be a dedicated analog front end (AFE) with an analog-to-digital converter (ADC) or a digital serial interface—such as SPI or I2C. We define the sensing rate as the rate at which the signal should be sampled to retain information relevant to the application. The sampling scheme determines how often data should be sampled for a target application. We define two sampling schemes: event-driven or continuous. Section 3 gives a few examples of IoT sensing applications and determines their features.

Next, the environmental conditions in which the system will be deployed are examined to determine the viable harvesting options. Harvesting sources include light, temperature gradient, and vibration. After choosing the harvesting source(s), an analysis of the available energy from these sources throughout the day is needed to help identify the optimal harvesting unit and power budget for the system. Section 4 looks into harvesting sources and presents studies related to the available energy in a wearable health monitoring example [8,9].

The last step is to gather the extracted features from the application and environment and translate them into design decisions for the system. Section 5 elaborates on the design decisions of the SoC. We start by determining the building blocks of the system, and then we look into the constraints on the system clock frequency and memory requirements. Finally, we look into choosing an energy harvesting unit and low power states to make sure the system operates reliably.

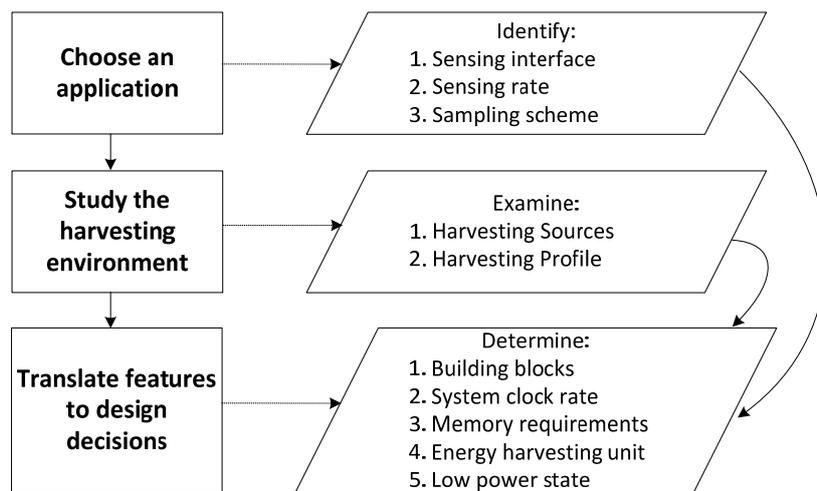


Figure 1. Overview of the proposed top-down methodology.

3. IoT Sensing Applications

Developing a reliable and energy-efficient SoC starts by determining its target applications. Once the application is chosen, the designer must find the answers to the following questions:

1. What data must be sensed?
 - a. Is a commercial sensor available to sense the data?
 - b. Or should the sensing interface be developed?
2. How fast must the data be sensed in order to not lose important information?
3. How often must the data be sampled?
 - a. Should the data be continuously sampled to not lose information?
 - b. Or is it enough to sample the data based on user request or after fixed periods of time?

In this section, we investigate the three different types of IoT applications that have become increasingly popular in recent years: health monitoring, industrial/asset monitoring, and environmental monitoring. In each of these applications, we extract three features: the sensing interface (analog/digital), the sensing rate, and the sampling scheme.

3.1. Health Monitoring

Health monitoring IoT devices have the opportunity to improve health care and reduce its cost. These devices measure heart rate, skin temperature, blood oxygen saturation, motion/activity, and other signals. Table 1 summarizes the different features of a few of these applications. Heart rate and blood oxygen are usually sensed through dedicated analog interfaces, while commercial sensors with digital interfaces are used to sense temperature and motion.

To detect heart rate variability parameters, the ECG signal is usually sampled at a rate between 50–500 Hz. Depending on the target market for such a heart rate monitor, different sampling schemes could be used. If the target market is high risk patients, continuous monitoring of the heart rate might be needed. For the regular consumer market, the monitor could be triggered by a user request. Recent blood oxygen monitors use a technique called photo-plethysmography (PPG). PPG signals are sampled at a rate between 30~700 Hz [10]. When used to detect glucose levels, this sampling scheme could be event-driven through a 15–30 min timer or by user request.

Commercial accelerometers such as the low power Analog Devices, ADXL345, are sometimes used to detect motion and activity. Such devices rely on digital serial interfaces—such as SPI and I2C—to transmit data to a microcontroller for further processing. For activity detection, a sampling rate of 100 Hz can be used [11]. Since continuous monitoring is required to detect activity, the ADXL345 implements event pins

that are triggered by different activity thresholds. These event pins allow the microcontroller to sleep during intervals of inactivity to reduce the system’s power consumption and to allow continuous monitoring.

Table 1. Features of common health monitoring Internet of Things (IoT) applications.

Application	Signal Measured	Sensing Interface	Sensing Rate	Sampling Scheme
Heart rate (healthy) Heart rate (high risk)	Electrocardiogram (ECG)	Analog	50–500 Hz	Event-driven Continuous
Skin temperature	Temperature	Analog/Digital	≤10 Hz	Event-driven
Blood oxygen	Photo-plethysmography (PPG)	Analog	30–700 Hz	Event-driven
Motion/activity	Acceleration	Digital	100 Hz	Continuous

3.2. Industrial/Asset Monitoring

Another class of IoT devices targets industrial and asset monitoring to reduce down-time and maintenance cost. Table 2 shows three applications in this class along with their features. Industrial machines vibrate and produce noise at different frequencies depending on their condition. Thus, analyzing vibration and audio is a powerful tool for detecting and diagnosing failures in machines. For reliability, most industrial monitoring devices use commercial sensors with digital serial interfaces to communicate the sensed data to a microcontroller for further processing. Thus, for these applications, a digital sensing interface is a must. Detecting and diagnosing machine faults require much higher sensing rates (1–20 kHz) than health monitoring devices. Once sampled, the vibration or audio data is processed through a Fast Fourier Transform (FFT) to determine the signal strength in the bands of interest. Even though these devices require a high sensing rate, machine characteristics do not change quickly; thus, the sampling scheme is usually event-driven, either through a long timer or by user request.

Asset monitoring is another class of applications that is gaining momentum. Devices targeting this class aim to track assets and report their data in an effort to improve their lifetime and utilization. This application space is wide and varied; an example application is shipping integrity tracking. In this application, shipping packages are tagged with a sensor that reports on fall/drop events. An accelerometer with a digital interface is used to wake up the SoC for further processing. To detect falls, the accelerometer sensing rate can be as low as 100 Hz.

Table 2. Features of common industrial/asset monitoring IoT applications.

Application	Signal Measured	Sensing Interface	Sensing Rate	Sampling Scheme
Machine monitoring	Vibration	Digital	1–20 kHz	Event-driven
Machine monitoring	Audio	Digital	1–20 kHz	Event-driven
Shipping integrity tracking	Acceleration	Digital	100 Hz	Continuous

3.3. Environmental Monitoring

Environmental monitors are a third class of IoT devices that sense data such as temperature, humidity, and gas. These devices are used in a variety of applications—from smart homes, to wearable sensors, to packaging applications and many others. Table 3 shows three applications in this IoT class along with their features. Temperature and humidity sensors are widely available in the market, and many include digital serial interfaces to communicate with an SoC. These signals, especially in homes, are slow-varying; thus, an event-driven sampling scheme is enough to acquire the data. Another common application is air quality monitors. These use metal oxide (MOx) sensors to detect carbon monoxide (CO) and other volatile organic compounds (VOC) [12]. These sensors usually produce analog signals that must be sampled by an ADC. The ADC sampling rate can be as slow as 100 Hz. For indoor home applications, air quality monitors can benefit from event-driven sampling schemes to reduce their power consumption.

Table 3. Features of common environmental monitoring IoT applications.

Application	Signal Measured	Sensing Interface	Sensing Rate	Sampling Scheme
Room temperature	Temperature	Digital	≤10 Hz	Event-driven
Room humidity	Humidity	Digital	≤10 Hz	Event-driven
Air quality	MOx	Analog	100 Hz	Event-driven

3.4. Analysis

After determining the different features of the application, designers can study the impacts of these features on the power consumption of their system, and an initial power budget can be determined. Designers can start by looking into available commercial sensors that can acquire the data needed. From the data sheets of these sensors, the active and sleep power consumption can be derived. By combining this data with the sampling scheme and sensing rate, the designers can determine the maximum (P_{s_max}) and average (P_{s_av}) power budgets of the sensor for the application. Next, the designers must determine the maximum (P_{h_max}) and average (P_{h_av}) amounts of power available to the system from different ambient sources in their deployment sites. If $P_{s_max} > P_{h_max}$ or $P_{s_av} > P_{h_av}$, the harvesters alone will not be able to support the system. In this case, the designers can explore the following options: (1) designing a low power sensor within the harvesters’ power budget or (2) adding an alternative backup power supply, such as a battery. If both $P_{s_max} < P_{h_max}$ and $P_{s_av} < P_{h_av}$, the designer can use the commercial sensor and determine the SoC’s power budget with $P_{c_av} < P_{h_av} - P_{s_av}$.

4. Energy Harvesting Profiles

After determining the target application, designers must look into the environment in which their SoC will be deployed to answer the following questions:

1. What are the available harvesting sources?
2. How much power can be harvested from each source?

Ambient energy in the environment can be harvested to provide power for an SoC. Thus, it is important to analyze the available sources in order to design an energy harvesting unit that maximizes the extracted energy. Ambient sources can be divided into four categories: mechanical, thermal, radiant, and biochemical. Table 4 shows the range of available energy from each of these categories. Once the available ambient sources have been determined, an analysis of the amount of energy they can provide in the applications’ environment should be performed. This step is important for three reasons: (1) it helps to avoid designing an energy harvesting unit for a source that provides little to no power in the environment of the sensor, (2) it helps to determine if energy must be harvested from multiple sources (simultaneously or otherwise) to meet the anticipated power budget of the system, and (3) it helps to build the power profile available to the application throughout a period of time to inform the design of the digital power manager (responsible for avoiding power loss).

Table 4. Ambient harvesting sources along with their estimated power density [13,14].

Category	Source	Estimated Power Density
Mechanical	Motion (human)	1–10 $\mu\text{W}/\text{cm}^2$
	Vibration (machines)	10–100 $\mu\text{W}/\text{cm}^2$
Thermal	Body heat	10–30 $\mu\text{W}/\text{cm}^2$
	Heat (industry)	1–10 mW/cm^2
Biochemical	Bio-fuel	0.1–100 $\mu\text{W}/\text{cm}^2$
Radiant	Sunlight	1–100 mW/cm^2
	Indoor light	10–100 $\mu\text{W}/\text{cm}^2$
	Wi-Fi	1 $\mu\text{W}/\text{cm}^2$
	GSM	0.1 $\mu\text{W}/\text{cm}^2$

Referring back to the three classes of IoT applications from Section 3, there are differences between the environments in which the devices must operate. A health monitoring device, in most cases, is a wearable device that passes through a complex harvesting environment throughout the day depending on the activity of the user. On the other hand, industrial and environmental monitoring devices are deployed in one location, and thus, their harvesting environment is fixed and more predictable. While the health monitoring device might harvest from body heat and indoor as well as outdoor light, an indoor machine monitoring device might only have access to indoor light and the light source might be intermittent (only on during business hours). Thus, even though the two devices harvest from light sources, the energy available to each could still be significantly different.

In [8], the authors presented a platform for measuring the amount of harvested energy from indoor light and body heat for health monitoring IoT devices. The setup includes ambient sensors, energy transducers, boost converters, current monitors, storage elements, and data loggers. In addition to logging the harvested energy, the platform also keeps track of the available light intensity, the ambient temperature, the skin temperature, and the activity of the user. It is used to generate an energy profile based on two days of data that models the harvesting environment to help predict the power available to the system. Similarly, in [9], the authors presented a one-year study of indoor light energy harvesting suitable for environmental monitoring and access control. This study highlighted the changes in the harvesting profiles between different seasons of the year to better inform the design of the system.

5. Designing the SoC

After analyzing the application and its environment, the design process for the SoC can begin. Figure 2 shows a template for a battery-less SoC. SoCs are divided into five major components: sensing interfaces, communication interfaces, data processing, clocking, and power management. In this section, we will focus on the sensing interfaces (particularly the digital sensing interfaces), the data processing, and the power management.

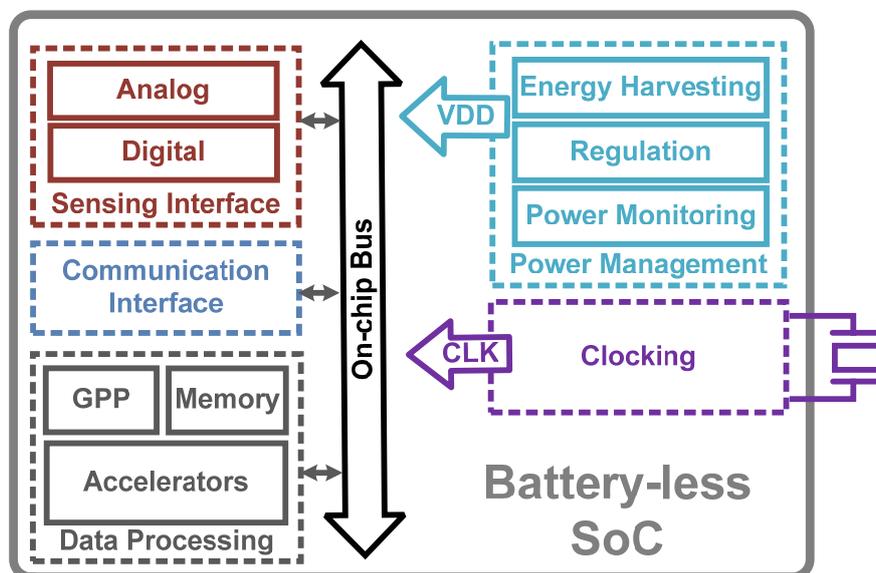


Figure 2. Template of a battery-less systems-on-chip (SoC).

5.1. Sensing Interfaces

The sensing interfaces are defined by the target application, as discussed in Section 3. Based on the power profiles available to the system, the designer must choose between using a commercial sensor or developing one. Commercial sensors have either analog or digital interfaces. Some commercial sensors provide an analog signal, and thus, require an on-chip ADC to measure the voltage they produce.

Other sensors provide a digital output with a standard interface to SoCs. The designer could also choose to design the sensor to integrate within the SoC as part of the analog sensing interfaces or as a stand-alone sensor with a digital interface to the SoC.

Two common digital SoC interfaces are SPI and I2C. Their goal is to enable serial communication through a simple protocol. The main differences in these standards that affect the design of ultra-low power systems are the pin count and energy efficiency. I2C requires fewer pins than SPI but uses open-drain pins with on-board pull-up resistors instead of CMOS pins. This causes short circuit currents when the SoC or sensor tries to pull down an I2C line, needlessly draining the storage element. This becomes especially problematic when I2C is operated at low frequencies. SPI, on the other hand, has no static current, enabling energy efficient communication at low frequencies.

As the system power reduces, I/O is still a dominating component in the power budget [6]. Thus, if a stand-alone sensor is being developed, the digital interface can be customized to reduce power. Reducing the operating voltage of the digital interface between chips on a board reduces the power and energy dramatically at the cost of throughput [15]. This voltage can be tuned for the required sensor throughput enabling the application to reach its minimum energy and power consumption. If the sensor and SoC need to be physically separated over longer distances, the transmission can be made differential to improve the reliability [16]. One such application is a wearable heart rate monitor, where the sensor is placed within a shirt close to the chest, while the SoC is placed on the sleeve to allow maximum exposure to skin and light. With these optimizations, improving off-chip communication to sensing interfaces, as shown in reference [6], can reduce their contributions to the system budget by over 94% for the proposed shipping integrity application, reducing the average system power to below 260 nW.

5.2. Digital Sub-System

The sensing interfaces communicate with the rest of the SoC through the system bus. Different bus architectures are available in the literature. The most common of these are the wishbone and the advanced microcontroller bus architecture (AMBA). The wishbone bus is an open source architecture that enables flexible interfacing between the controllers and different components. This architecture is highly customizable to meet different application needs. On the other hand, AMBA includes different targeted bus protocols. The two bus protocols mostly used in low power systems are the advanced high-performance bus (AHB) and the advanced peripheral bus (APB). The AHB bus is a pipelined bus architecture used to provide a high speed interface between the controller(s) and the different memories in the system. The APB bus, on the other hand, is a serial bus with a single master designed to interface between the controller and low speed peripherals. Generally, digital sensing interfaces are placed on the APB bus. The choice of bus architecture impacts the rate at which the system can run, as we will discuss later in this section.

Next, the main controller must be designed/implemented. Many low power SoCs [2,3,7,17] use the ARM's low power Cortex M0/M0+ due to its reduced gate count, its flexible instruction set, and its software and tool support. However, the Cortex M0/M0+ follows the von Neumann architecture, which limits the performance of the system by using the system bus for instruction fetches. When running the system at a low clock rate, the number of cycles consumed by the instruction fetches become especially limiting. Thus, battery-less SoCs, such as those described in references [5,6], use a custom core with a dedicated memory bus to decouple instruction fetches from data transfer. However, designing a custom core with a custom ISA is a challenge. Recently, RISC-V has gained momentum by providing an open source instruction set architecture (ISA) that is easily extensible with software and tool support [18].

Once the core, bus, and sensing interface have been chosen, the users can start developing the software necessary to read information out of the sensor and process it. This is important for two main reasons: (1) it helps to determine the minimum size of the instruction and data memories needed, and (2) it provides the opportunity to optimize the architecture for energy efficiency by highlighting

the functions/operations that take the longest time and/or the largest amount of code to execute. Memories are one of the always-on components within IoT SoCs and contribute significantly to the power budget. Thus, reducing the amount of on-chip memory is one way to reduce a system's power consumption. However, the choice of memory size can only be made if the target application(s) software is developed before the chip is designed. The programming code helps to determine the size of the instruction memory. To determine the size of the data memory, the designer must consider whether the raw data from the sensor must be communicated or whether a processed version is enough. For the former, the designer must account for the size of the raw data needed as well as any data that must be processed and saved. The programming code will also help to determine other potential data that must be saved/held in the data memory.

In IoT applications, data transfer and processing are two main functions that the system performs. Data processing can be optimized by implementing hardware accelerators to process the data efficiently and to reduce the load on the core. A traditional way to off-load data transfer from the core is to include a direct memory access (DMA) on the bus, as was done in reference [5]. However, the core must still configure the DMA before data transfer can start, and the DMA still relies on the bus for data transfer. Thus, the authors in reference [6] proposed a data flow architecture where data is transferred between different components in the system through a dedicated data-flow path that bypasses the bus. Data is moved from the sensing interface directly into the accelerator by processing it through a dedicated path. This architecture can only be made possible if the application space of the SoC is determined prior to the design time. However, such an architecture could offer a significant improvement in processor idle time, reductions in the instruction and data memory sizes, and improvements in energy efficiency.

Affecting all of the choices in the digital system architecture is the required sensing rate of the application. The required sensing rate imposes a lower bound on the system's operating frequency, and the operating frequency linearly impacts the power consumption of the system. In the time between samples, the system must read the output of the sensor, move it to a storage location, or process it through an accelerator or the main core and potentially, react to the outcome. Thus, the system frequency must be fast enough to handle all of these operations in the time between samples to avoid losing important data. The time required for each of these operations depends on the sensing interface used and the system architecture. If an integrated ADC is used, the sensing rate, the number of cycles required to perform reads and write to peripherals on the bus, and the data processing time play important roles in determining the minimum clock rate. Equation (1) illustrates the lower bound on the clock rate imposed by these factors:

$$T_{CLK}[N_C \times N_{BW} + (N_{BW} + N_{BR}) \times N_D + N_P] \leq T_S, \quad (1)$$

where T_{CLK} is the clock period, T_S is the time between samples, N_C is the number of configuration words needed to configure the ADC, N_B is the number of data words expected from the ADC, N_{BW} is the number of cycles needed to write a register on the system bus, N_{BR} is the number of cycles needed to read a register on the system bus, and N_P is the number of cycles needed by the software to manage the data transfer and process the data. To capture a sample from the ADC, the core must first configure (N_C) the ADC and then read its output (N_D), process it, and move it to the data memory.

On the other hand, if a digital sensing interface is used, the overhead of configuring and reading the sensor serially can often become the bottleneck in ultra-low power and self-powered systems. Many sensors dictate a communication rate for a given sensing rate, forcing the designer to run the digital sensing interface at that rate. For example, the ADXL345 accelerometer recommends a minimum SPI clock frequency of 100 KHz when the output sensing rate is 200 Hz. Thus, the SPI master interface on the SoC must be designed to run at a minimum clock rate of 200 KHz to interface with this sensor. Here, the designer can choose to run the system clock at the same rate as the SPI master

interface, or to decouple the system clock from the SPI master clock. If the designer chooses the former, the system clock rate must respect Equation (2):

$$T_{CLK} \leq T_{SPI}/2, \quad (2)$$

where T_{CLK} is the clock period, and T_{SPI} is the SPI clock period. Otherwise, the designer must have a clock source that is capable of running at the minimum speed imposed by Equation (2) to feed the sensing interface. This clock source can be gated when the sensing interface is not in use. A second clock is also needed to drive the rest of the system. Equation (3) below shows the relationship between this clock and the sensing rate, assuming an SPI sensing interface is used:

$$T_{CLK}[(N_C + N_{SC}) \times N_{BW} + (N_{BW} + N_{BR}) \times N_{SR} + N_P \times T_{CLK}] \leq (N_{SC} + N_{SR}) \times 8 \times T_{SPI}, \quad (3)$$

where T_{CLK} is the clock period, T_{SPI} is the SPI clock period, T_S is the time between samples, N_C is the number of configuration bytes needed to start and configure the SPI master on the SoC, N_{SC} is the number of command bytes needed to start a read operation from the sensor, N_{SR} is the number of read bytes expected from the sensor, N_{BW} is the number of cycles needed to write a register on the system bus, N_{BR} is the number of cycles needed to read a register on the system bus, and N_P is the number of cycles needed by the software to manage the data transfer and process the data.

To start the data transfer between the SoC and sensor through SPI, the core must first configure (N_C) the SPI interface and then load the commands (N_{SC}) needed to start a read operation from the sensor. To do this, the core uses the bus to write the data to the SPI master peripheral. This phase requires $(N_C + N_{SC}) \times N_{BW} \times T_{CLK}$. Next, the SPI master begins the transfer of command (N_{SC}) and data (N_{SR}) bytes. The SPI master transfers a single bit through the bus every T_{SPI} ; thus, the completion of the transfer requires $(N_{SC} + N_{SR}) \times 8 \times T_{SPI}$. Assuming the SPI core does not have a buffer to hold the received data, the core must transfer the data from the SPI core to the data memory or to the accelerator processing the data. To do this operation, the core performs a bus read, followed by a bus write operation. Thus, moving all the bytes out of the SPI core requires $(N_{BW} + N_{BR}) \times N_{SR} \times T_{CLK}$. Finally, the time required to process the data depends on the application and requires $N_P \times T_{CLK}$. Summing all these times gives Equation (3) which relates the clock rate to the sensing and communication rates. Equation (3) assumes a Harvard architecture where the instruction fetches are decoupled from data transfers.

As shown in Equation (3), bus transfers play an important role in determining the minimum clock rate. For applications with very slow sensing rates, the overhead of bus transfers could be an acceptable trade-off for a simplified design architecture. However, for applications with faster sensing rates, the overhead might not be acceptable. Thus, reducing the number of bus transfers becomes a must. Designers can choose to implement a Harvard architecture to overcome the Von Neumann bottleneck. A pipelined bus, such as the AHB, is another way to reduce the number of cycles required to perform consecutive read/writes on the bus. The data flow architecture presented in reference [6] is a third way to completely eliminate the required bus transfers. Hardware accelerators can also help by processing the data in an efficient manner.

Once the architecture and clock rate are chosen, the designer must estimate the power consumption of the system to ensure the design remains within the power budget. Synthesis tools provide an initial estimate of the power consumption of the design based on the RTL description and the characterized standard cell libraries. These tools also allow the designer to explore the benefits of using different low power features, such as sub-threshold design, multi-threshold design, voltage scaling, power gating, and clock gating, to reduce the power consumption further. With the introduction of a new test methodology for sub-threshold design [19], these low power features become feasible in battery-less SoC products. The designer can also build a digital power manager to control the different low power features depending on the current harvesting environment. The digital power

manager can combine insights from the energy harvesting unit with models of the harvesting profile of the system's environment to anticipate changes to the harvesting conditions and react accordingly.

When the harvesting conditions are poor, the battery-less system can lose power completely. To aid in the recovery of critical data, designers can choose to implement a non-volatile processor with non-volatile memory (NVM) or a processor with off-chip NVM for backup and recovery. Recently, nonvolatile processing research has resulted in lower power, non-volatile processors designed for use in ambient harvesting. These processors offer complete recovery from loss of power by saving the state of the entire processor before power loss [20]. The trade-off between a nonvolatile processor and a low power processor with partial backup NVM [6] is a function of the active and leakage power of each processor, backup and recovery costs, and probability of failure for a given harvesting mechanism. NVPs generally have a higher active power than low power processors with off-chip NVM, but shorter backup and recovery times. NVPs can also be limited by the manufacturing technology, since not all process technologies support non-volatile elements. NVPs also require more design time compared to the off-chip NVM, especially when a commercial NVM is used. However, designers can implement low power NVM features to reduce their power consumption compared to commercial NVMs. To assist in the design choice, Equation (4) breaks down the power budget of the system into the active (P_{ACT}), leakage (P_{LEAK}), backup (P_{BACKUP}), and recovery power ($P_{RECOVERY}$). The active power is only consumed during the active duty cycle (D_{ACT}) which is impacted by the sampling scheme. Backup and recovery power are only incurred when the system loses power. The probability of power failure (Pr_{FAIL}) can be derived from the energy harvesting profile described in Section 4. The NVP usually has higher P_{ACT} and P_{LEAK} than low power processors with off-chip NVM that are completely shut off in normal mode. However, their backup and recovery times might exceed those of the NVP due to the requirement of the serial interface to move data from the processor to the off-chip NVM. Thus, the designer can use Equation (4) to compare the two architectures and choose the optimal solution for the target application:

$$P_{ACT} \times D_{ACT} + P_{LEAK} \times (1 - D_{ACT}) + [P_{RECOVERY} + P_{BACKUP}] \times Pr_{FAIL} = P_{AVG} \quad (4)$$

5.3. Energy Harvesting Power Management Unit

The application's environment dictates the type of harvester to use and the energy harvesting unit to implement. To harvest energy from heat or temperature fluctuations, a thermoelectric generator (TEG) or pyroelectric device [21] can be used. Photovoltaic (PV) cells produce energy from light. Piezoelectric harvesters or triboelectric nanogenerators [22] produce energy from vibrations. Each of these harvesters imposes different constraints on the energy harvesting unit. A TEG produces very low DC voltages, as low as 10 mVs. A PV cell also produces a DC voltage output but in a higher range. A piezoelectric harvester, on the other hand, produces an AC voltage that must then be converted to DC before it can be stored.

A number of energy harvesting units that extract power from these harvesters have been presented in the literature [4–7,23]. Most energy harvesting units have a Maximum Power Point Tracking (MPPT) circuit designed to extract the maximum power from the harvester and into the system. The maximum power point of a TEG occurs at 50% of its open circuit voltage, whereas that of a PV cell occurs at 76% of its open circuit voltage [5]. In addition to the MPPT circuit, the energy harvesting unit generally requires a boost converter with an off-chip inductor to boost the input voltage with high efficiency. This is especially true for TEG harvesting since the input voltage could be as low as 10 mV. A PV cell, on the other hand, produces a much higher input voltage (starting from ~600 mV) and thus, could benefit from a switched capacitor voltage doubler circuit [24] to boost its input voltage without the need for an off-chip inductor.

Due to the low input voltage of the TEG harvester, a cold start circuit is required to kickstart the system. Some systems rely on an Radio Frequency (RF) power [4], and thus use an RF harvester as a cold start circuit that rectifies the input and stores the extracted energy onto the super-capacitor.

Another approach relies on a Ring Oscillator (RO) with a clock doubler circuit that can start the system at an input voltage of 220 mV [23]. The choice of cold start circuit depends on the environment in which the sensor is deployed. If an RF transmitter is unavailable to kickstart the system, the RO cold start circuit can be used instead.

In addition to the energy harvesting unit, regulation circuits are needed to produce a stable supply for the rest of the system. The designers must carefully choose the supply voltages and output powers of each of the rails produced by the regulation unit. The sensing interface required by the application determines the characteristics of at least one of the rails. If an off-chip sensor with a digital interface is to be used with the system, the minimum supply voltage and expected current drawn must be taken into account in the design of the regulation circuit. Next, the power analysis from the synthesis tools (refer to Section 5.2) can be used to determine the supply voltage and expected current drawn from the digital power rail. Once the characteristics of the power rails have been determined, the designer can explore different regulation schemes to determine the scheme that has the highest power conversion efficiency for the target load. Here, the designer must investigate the available power from the harvesting source (information from the harvesting profile can be used here) and the power consumption of the load circuits (the rest of the SoC and any off-chip components). These two numbers will impose a minimum power conversion efficiency that the power management unit must meet in order for the system to operate reliably. Another factor that could play a role in the choice of regulation scheme is the form factor of the system. A single-inductor-multiple output (SIMO) regulator could be used to produce the different power rails of the system at the cost of an additional inductor in the system [23]. To eliminate the extra inductor needed, the authors in reference [25] relied on a switched capacitor regulator to produce the different power rails, while the authors in reference [24] developed a scheme to share an inductor between the regulation and harvesting units. Choosing the optimal harvesting unit and regulators for a system's application ensures that the system is taking full advantage of the available ambient sources to improve its lifetime and reliability.

6. Conclusions

This paper presented a top-down methodology for building battery-less systems-on-chips for different classes of IoT applications. The first step in the proposed methodology is the identification of the target application class and the extraction of three main features of the application: the sensing interface, the sensing rate, and the sampling scheme. Next, the environmental conditions in which the system will be deployed are studied to determine the type of harvester needed and the amount of energy available. The final step is to take these five features and translate them into design decisions to build the SoC. The sensing interfaces affect the blocks implemented on the system and the supply voltages produced by the power management unit. The sensing rate and sampling scheme affect the power budget, the minimum clock rate, and the digital architecture. The type of harvester impacts the implementation of the energy harvesting unit, and the harvesting profile affects the power states of the system. Transitioning to a top-down methodology will result in an optimized system that consumes a lower average power and has higher harvesting, sensing, communication, and processing efficiencies.

Author Contributions: Farah and Chris defined the methodology proposed with input and guidance from Benton. Farah and Chris wrote different sections of the paper. Benton reviewed the paper.

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank Dave Abdallah for his input.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Van Helleputte, N.; Konijnenburg, M.; Pettine, J.; Jee, D.; Kim, H.; Morgado, A.; van Wegberg, R.; Torfs, T.; Mohan, R.; Breeschoten, A.; et al. 18.3 A multi-parameter signal-acquisition SoC for connected personal health applications. In Proceedings of the 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), San Francisco, CA, USA, 9–13 February 2014; pp. 314–315.
2. Dongsuk, J.; Chen, Y.; Lee, Y.; Kim, Y.; Foo, Z.; Kruger, G.; Oral, H.; Berenfeld, O.; Zhang, Z.; Blaauw, D.; et al. 24.3 An implantable 64 nW ECG-monitoring mixed-signal SoC for arrhythmia diagnosis. In Proceedings of the 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), San Francisco, CA, USA, 9–13 February 2014; pp. 416–417.
3. Kim, G.; Lee, Y.; Foo, Z.; Pannuto, P.; Kuo, Ye.; Kempke, B.; Ghaed, M.H.; Bang, S.; Lee, I.; Kim, Y.; et al. A millimeter-scale wireless imaging system with continuous motion detection and energy harvesting. In Proceedings of the 2014 Symposium on VLSI Circuits Digest of Technical Papers, Honolulu, HI, USA, 10–13 June 2014; pp. 1–2.
4. Zhang, Y.; Shakhsheer, Y.; Silver, J.D.; Klinefelter, A.; Nagaraju, M.; Boley, J.; Pandey, J.; Shrivastava, A.; Carlson, E.J.; Wood, A.; et al. A Batteryless 19 uW MICS/ISM-Band Energy Harvesting Body Sensor Node SoC for ExG Applications. *IEEE J. Solid-State Circuits* **2013**, *48*, 199–213. [CrossRef]
5. Roy, A.; Klinefelter, A.; Yahya, F.B.; Chen, X.; Gonzalez-Guerrero, L.P.; Lukas, C.J.; Kamakshi, D.A.; Boley, J.; Craig, K.; Faisal, M.; et al. A 6.45 μ W Self-Powered SoC With Integrated Energy-Harvesting Power Management and ULP Asymmetric Radios for Portable Biomedical Systems. *IEEE Trans. Biomed. Circuits Syst.* **2015**, *9*, 862–874. [CrossRef] [PubMed]
6. Yahya, F.B.; Lukas, C.J.; Breiholz, J.; Roy, A.; Patel, H.N.; Liu, N.; Chen, X.; Kosari, A.; Li, S.; Akella, D.; et al. A Battery-less 507 nW SoC with Integrated Platform Power Manager and SiP Interfaces. In Proceedings of the 2017 Symposium on VLSI Circuits, Kyoto, Japan, 5–8 June 2017.
7. Lim, W.; Lee, I.; Sylvester, D.; Blaauw, D. 8.2 Batteryless Sub-nW Cortex-M0+ processor with dynamic leakage-suppression logic. In Proceedings of the 2015 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), San Francisco, CA, USA, 22–26 February 2015; pp. 1–3.
8. Fan, D.; Ruiz, L.L.; Gong, J.; Lach, J. Profiling, modeling, and predicting energy harvesting for self-powered body sensor platforms. In Proceedings of the 2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN), San Francisco, CA, USA, 14–17 June 2016; pp. 402–407.
9. Viehweger, C.; Baldauf, M.; Keutel, T.; Kanoun, O. Energy profile analysis by simulation for the design of energy harvesting systems. In Proceedings of the International Multi-Conference on Systems, Signals & Devices, Chemnitz, Germany, 20–23 March 2012; pp. 1–3.
10. Sharma, A.; Lee, S.B.; Polley, A.; Narayanan, S.; Li, W.; Sculley, T.; Ramaswamy, S. Multi-modal smart bio-sensing SoC platform with >80dB SNR 35 μ A PPG RX chain. In Proceedings of the 2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits), Honolulu, HI, USA, 15–17 June 2016; pp. 1–2.
11. Jia, N. Detecting Human Falls with a 3-Axis Digital Accelerometer, Analog Devices. Available online: <http://www.analog.com/en/analog-dialogue/articles/detecting-falls-3-axis-digital-accelerometer.html> (accessed on 13 June 2018).
12. CCS801 Ultra Low Power Analog VOC Sensor for Indoor Air Quality Sensor Datasheet, AMS. Available online: <https://ams.com/ccs801> (accessed on 13 June 2018).
13. Raju, M. Energy Harvesting: ULP Meets Energy Harvesting: A Game Changing Combination for Design Engineers. TI. Available online: http://www.ti.com/corp/docs/landing/cc430/graphics/slyy018_20081031.pdf (accessed on 13 June 2018).
14. Boisseau, S.; Despesse, G.; Seddik, B.A. Electrostatic Conversion for Vibration Energy Harvesting, Small-Scale Energy Harvesting. *arXiv* **2012**, arXiv:1210.5191v1.
15. Lukas, C.J.; Calhoun, B.H. A 0.38 pj/bit 1.24 nW chip-to-chip serial link for ultra-low power systems. In Proceedings of the 2015 IEEE International Symposium on Circuits and Systems (ISCAS), Lisbon, Portugal, 24–27 May 2015; pp. 2860–2863.
16. Lukas, C.J.; Calhoun, B.H.; Bhakta, R.; Jur, J.S. A 3.77 nW, 11.4 fj/b/mm link for reliable wireline communication in ultra-low power on-body sensor networks. In Proceedings of the 2017 IEEE Biomedical Circuits and Systems Conference (BioCAS), Torino, Italy, 19–21 October 2017; pp. 1–4.

17. Myers, J.; Savanth, A.; Howard, D.; Gaddh, R.; Prabhat, P.; Flynn, D. 8.1 An 80nW retention 11.7pJ/cycle active subthreshold ARM Cortex-M0+ subsystem in 65 nm CMOS for WSN applications. In Proceedings of the 2015 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), San Francisco, CA, USA, 22–26 February 2015; pp. 1–3.
18. RISC-V. Available online: <https://riscv.org/specifications> (accessed on 13 June 2018).
19. Lukas, C.J.; Yahya, F.B.; Calhoun, B.H. Modeling trans-threshold correlations for reducing functional test time in ultra-low power systems. In Proceedings of the 2017 IEEE International Test Conference (ITC), Fort Worth, TX, USA, 31 October–2 November 2017; pp. 1–10.
20. Liu, Y.; Li, Z.; Li, H.; Wang, Y.; Li, X.; Ma, K.; Li, S.; Chang, Me.; John, S.; Xie, Y.; et al. Ambient energy harvesting nonvolatile processors: From circuit to system. In Proceedings of the Design Automation Conference (DAC), San Francisco, CA, USA, 8–12 June 2015; pp. 1–10.
21. Mane, P.; Xie, J.; Leang, K.K.; Mossi, K. Cyclic energy harvesting from pyroelectric materials. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **2011**, *58*, 10–17. [[CrossRef](#)] [[PubMed](#)]
22. Khushboo; Azad, P. Triboelectric nanogenerator based on vertical contact separation mode for energy harvesting. In Proceedings of the 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 5–6 May 2017; pp. 1499–1502.
23. Shrivastava, A.; Wentzloff, D.; Calhoun, B.H. A 10 mV-input boost converter with inductor peak current control and zero detection for thermoelectric energy harvesting. In Proceedings of the IEEE 2014 Custom Integrated Circuits Conference, San Jose, CA, USA, 15–17 September 2014; pp. 1–4.
24. Amin, S.S.; Mercier, P.P. MISIMO: A multi-input single-inductor multi-output energy harvester employing event-driven MPPT control to achieve 89% peak efficiency and a 60,000× dynamic range in 28 nm FDSOI. In Proceedings of the 2018 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 11–15 February 2018; pp. 144–146.
25. Roy, A.; Calhoun, B.H. A 71% efficient energy harvesting and power management unit for sub- μ W power biomedical applications. In Proceedings of the 2017 IEEE Biomedical Circuits and Systems Conference (BioCAS), Torino, Italy, 19–21 October 2017; pp. 1–4.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).