Low energy design techniques for digital circuits are examined to determine their suitability for use in a digital logic controller for wireless sensor network nodes. Transistor level simulations are used to evaluate the techniques and those demonstrating an energy reduction are used to implement a digital logic controller. The digital controller for the wireless sensor node, fabricated in a 0.18µm CMOS process, operates at 350mV while consuming 336fJ per clock cycle with a 250kbps data rate. Lab measurements show a 98% reduction in energy consumption compared to an implementation that utilizes standard design techniques, making it the lowest energy digital controller for wireless sensor nodes to date.
Ultra-Low Energy Digital Logic Controller Design for Wireless Sensor Networks

by

Stephen W. Meliza

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented March 27, 2009
Commencement June 2009
Master of Science thesis of Stephen W. Meliza presented on March 27, 2009

APPROVED:

_________________________
Co-Major Professor, representing Electrical and Computer Engineering

_________________________
Co-Major Professor, representing Electrical and Computer Engineering

_________________________
Director of the School of Electrical Engineering and Computer Science

_________________________
Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

_________________________
Stephen W. Meliza, Author
ACKNOWLEDGEMENTS

Academic

I would like to thank both of my major professors, Dr. Terri Fiez and Dr. Kartikeya Mayaram, for advising me and giving me the opportunity to complete this work. I thank my other committee members, Patrick Chiang and Dr. Ethan Minot, for their time and support. I thank my professors, especially Roger Traylor, for teaching and guiding me both in and out of class. I would also like to thank Robbert Batten, Triet Lee, Thomas Brown, James Ayers, Napong Panitantum, Adam Heiberg, Hector Oporta, Farhad Farahbakhshian, and Chris Lindsley of Oregon State University for their help during the course of this project. This work was supported by NSF grant DBI-0529223 and partial fabrication support was provided by Jazz Semiconductor.

Personal

I thank my parents who first taught me an eagerness to learn and perseverance to do my best in all of my work. I thank all of my friends and family for their love and support over the years. And most of all, I thank God, our Lord and Savior, from whom all blessings flow. Post Tenebras Lux.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>4.3</td>
</tr>
<tr>
<td>4.4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>5.1</td>
</tr>
<tr>
<td>5.2</td>
</tr>
<tr>
<td>5.3</td>
</tr>
<tr>
<td>5.4</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>(a) Example tree network of WSN nodes and central hub. (b) WSN temperature sensing node block diagram. (c) Available energy vs. distance for wireless energy harvesting [1].</td>
<td>2</td>
</tr>
<tr>
<td>2.1</td>
<td>Typical CMOS inverter.</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>Leakage current paths of a MOS transistor [2].</td>
<td>7</td>
</tr>
<tr>
<td>2.3</td>
<td>(a) The common bus has a high capacitance and wastes energy by not maintaining locality of reference. (b) Dedicated buses have minimal capacitance and save energy by maintaining locality of reference.</td>
<td>10</td>
</tr>
<tr>
<td>3.1</td>
<td>Simplified block diagram for WSN node digital logic controller.</td>
<td>13</td>
</tr>
<tr>
<td>3.2</td>
<td>Optimal $V_{DD}$ for minimum energy consumption [3].</td>
<td>16</td>
</tr>
<tr>
<td>3.3</td>
<td>Logic level converter.</td>
<td>18</td>
</tr>
<tr>
<td>3.4</td>
<td>Comparison of subthreshold energy consumption per clock cycle for different gate activity levels.</td>
<td>18</td>
</tr>
<tr>
<td>3.5</td>
<td>Power switch.</td>
<td>20</td>
</tr>
<tr>
<td>3.6</td>
<td>Comparison of BCD to Gray coded 5-bit address bus energy use per clock cycle. The Gray code uses 7% less energy with a loading of 28 gates but 5% more energy with 7 gates.</td>
<td>23</td>
</tr>
<tr>
<td>3.7</td>
<td>Comparison of energy per clock cycle for each component in the WSN node. Also shown is the energy for the logic circuits before and after optimization for low energy consumption.</td>
<td>25</td>
</tr>
<tr>
<td>4.1</td>
<td>Simplified command flow diagram for the digital logic controller.</td>
<td>27</td>
</tr>
<tr>
<td>5.1</td>
<td>Chip microphotograph.</td>
<td>30</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Energy per clock cycle for a 16 byte memory block with and without power gating under typical usage scenarios</td>
</tr>
<tr>
<td>3.2</td>
<td>Gray code and equivalent binary coded decimal value</td>
</tr>
<tr>
<td>5.1</td>
<td>Measured energy per clock cycle for various data rates and supply voltages</td>
</tr>
<tr>
<td>5.2</td>
<td>Comparison of energy reduction per clock cycle by implementing sub-threshold supply voltage, power gating, and Gray coding.</td>
</tr>
<tr>
<td>5.3</td>
<td>WSN node controller compared to other recent WSN node controller designs</td>
</tr>
<tr>
<td>5.4</td>
<td>Summary of Measured Results</td>
</tr>
</tbody>
</table>
ULTRA-LOW ENERGY DIGITAL LOGIC CONTROLLER DESIGN
FOR WIRELESS SENSOR NETWORKS
1 INTRODUCTION

Recent advances in electronics have enabled a new field of low power devices for sensing applications in a wireless sensor network (WSN). Many of these devices harvest energy from their environment, allowing long periods of autonomous, battery-free operation. An example of a WSN topology is shown in Fig. 1.1(a). This network depends on a central hub to provide wireless power to the nodes and control the network, including gathering data from the sensor nodes. The network shown in Fig. 1.1(a) is a tree topology, but it is capable of a fully mesh topology where any node can receive and pass on the data from any other node within its communication range.

![Figure 1.1](image-url)

**FIGURE 1.1:** (a) Example tree network of WSN nodes and central hub. (b) WSN temperature sensing node block diagram. (c) Available energy vs. distance for wireless energy harvesting [1].

Each WSN node in Fig. 1.1(b) consists of an energy harvesting circuit, a wireless transceiver, at least one sensor, a digital controller, and a circuit to wake-up the node. The digital controller must contain logic that dictates the behavior of the node as well as interface to and control all of the other components in the node. This is especially true for the receiver interface as it must also synchronize the data stream. Each component
of the WSN node must consume minimal power since there is approximately 10 \( \mu \text{W} \) of power available at 8 meters from the power source, Fig. 1.1(c) [1]. This power level is significantly less than the power available from a battery, a common power source for traditional WSN designs, prompting the design of a low energy WSN node controller.

Traditionally, programmable microcontroller architectures have been used to implement the digital controller in WSN nodes. To date, several designs [4–8] have demonstrated low energy operation, but have sacrificed reduction of energy use for the sake of flexibility and performance. All of these designs did not focus on low energy operation as the most important criteria. In this work the focus is on achieving minimal energy consumption, at the expense of flexibility and high performance.

This paper describes the design of a low energy digital controller for WSN nodes. Several traditional approaches for reducing power consumption are explored in Section 2 to determine the ones that result in a savings in energy in this application. Techniques demonstrating energy savings are presented in detail in Section 3. Section 4 describes the operation and design of the low energy digital controller. The measured results for a fabricated test chip are reported in Section 5, and finally Section 6 concludes this work.
2 LOW ENERGY DIGITAL DESIGN

Low power and low energy digital circuit design is a field that has benefited significantly from previous research. Before a digital logic circuit for a WSN sensor node is designed, it is imperative that the fundamental sources of the power consumption and common techniques to reduce them be reviewed and understood. First, the sources of power consumption will be reviewed followed by common techniques used to reduce power consumption in digital circuits. Each of the techniques is evaluated by simulating an example circuit in Cadence Virtuoso Spectre circuit simulator using transistor level circuits with BSIM4 models. The results of the simulations are used to evaluate the suitability of each technique.

2.1 Sources of Power Consumption

There are three primary sources of power consumption: charging and discharging of signal lines, shoot through of current through a gate from supply to ground, and leakage [9]. The first two are sources of dynamic power consumption and are the topics of existing research [10]. Conversely, the leakage is a static power consumption and was considered insignificant in the past but is becoming significant as process sizes decrease.

2.1.1 Dynamic Power Consumption

The first component of dynamic power consumption, charging and discharging of signal lines, is useful as it is consumed in processing information through the circuit. Fig. 2.1 demonstrates a CMOS inverter, showing these current paths and the capacitive load. The power consumed is modeled by [11]:

\[ P_D = (C_{PD} + C_L) \cdot V_{DD}^2 \cdot f, \]  \hspace{1cm} (2.1)
FIGURE 2.1: Typical CMOS inverter.

where $P_D$ is the dissipated power, $C_{PD}$ is the power-dissipation capacitance of the gate (an equivalent capacitance of switching the gate with no load attached, including the gate-source capacitance, gate-drain capacitance, and the shoot through current) [12], $C_L$ is the capacitance of the load, $V_{DD}$ is the power-supply voltage, and $f$ is the transition frequency of the output signal. Instead of the transition frequency $f$, it is more useful to use the activity level “A” since the gate may not transition on every clock edge [9].

The power consumption due to current shoot through is wasted and is modeled by [9]:

$$P_{ST} = V_{DD} \cdot I_{ST},$$  \hspace{1cm} (2.2)

where $I_{ST}$ is the average shoot through current from the supply rail to ground. From (2.1), $P_D \propto V_{DD}^2$, and from (2.2), $P_{ST} \propto V_{DD}$, which indicates that reducing $V_{DD}$ is very important in reducing the dynamic power consumption. There is also a linear dependence
on $C_P, C_L, A$, and $I_ST$ so it is also important to look for ways to reduce these components as well.

### 2.1.2 Static Power Consumption

Static power consumption is wasted as it does not contribute to the functionality of a circuit. The MOS transistor has several leakage current mechanisms as shown in Fig. 2.2. The reverse bias pn junction leakage, $I_1$, results from reverse biased pn junctions between the drain diffusion and substrate that causes reverse leakage current from the drain to the substrate. Subthreshold leakage current, $I_2$, is leakage from the drain to source that results from operating in weak inversion when the gate voltage is below the threshold voltage. Gate oxide tunneling, $I_3$, is the tunneling of electrons through the gate oxide layer to the substrate as the result of an electric field between the gate and substrate. Additionally, there can also be a gate to substrate current due to hot-carrier injection, $I_4$. This occurs in short channel transistors which have a high electric field between the substrate and the gate oxide. Gate induced drain leakage, $I_5$, is caused by a high field effect in the drain junction of a MOS transistor. This high field causes a narrowing of the depletion layer and an accumulation of minority carriers that flow to the substrate. The punchthrough current, $I_6$, exists in short channel devices where an increase in $V_{DS}$ can cause current conduction from the drain to the source through the substrate region [2].

Static power consumption is then [2],

$$P_{static} = V_{DD} \cdot I_{leakage},$$

(2.3)

where $I_{leakage}$ is the sum of the six leakage currents.

Retrograde and halo doping can be used at the process level to reduce all of the leakage mechanisms, but circuit design techniques can only be used to reduce subthreshold leakage, gate leakage, and pn junction reverse bias leakage. Circuit design techniques to combat static leakage include the following: transistor stacking, multiple threshold volt-
FIGURE 2.2: Leakage current paths of a MOS transistor [2].

ages, dynamic threshold voltages, and supply voltage scaling [2]. Supply voltage scaling is particularly effective in reducing static power as pn junction leakage current, subthreshold leakage current, and gate leakage current are proportional to the supply voltage. Short channel devices with a high supply voltage have a greater subthreshold leakage due to a lower threshold voltage resulting from a lowering of the energy threshold between the drain and source (drain-induced barrier lowering). At higher supply voltages the increased gate voltage results in an increased electrical field across the gate oxide which causes an increase in electron flow from the substrate to the gate. Also, the pn junction leakage current is greater at higher supply voltages due to an increased reverse bias voltage. Therefore, in order to reduce the static power consumed, it is very important to lower \( V_{DD} \).

2.2 Common Energy Reduction Techniques

One or more of the power consumption factors \( (V_{DD}, C_{PD}, C_L, A, I_{ST}, \text{ and } i_s) \) need to be reduced in order to obtain a reduction in dynamic and static power. There are several
techniques that can be used to reduce these factors: reduce the supply voltage [3], optimize number encodings [13], eliminate glitches [14], use asynchronous logic [15], partition the architecture and layout [9], vary the threshold voltage [3], use local voltage regulation [10], use adiabatic logic [16], optimize the system algorithm [9], and use a low energy logic family [17].

2.2.1 Reduce Supply Voltage

An effective technique to reduce energy consumption is to reduce the power supply voltage, as dynamic power is a function of $V_{DD}^2$ (2.1). From (2.2) and (2.3) we also know that shoot through and static power are also dependent on $V_{DD}$. From these three equations it is clear that reducing $V_{DD}$ will be key in reducing the power consumption. However, lower power consumption does not necessarily equate to lower energy consumption. In fact, as $V_{DD}$ is reduced static leakage current does reduce, but the slower operation of the circuit increases the time to complete a task, which results in an increase in the leakage energy. Despite the increase in static energy consumption at lower $V_{DD}$ in current CMOS processes, the reduction in dynamic energy dominates. Therefore, it is best to operate with the lowest practical $V_{DD}$ [3].

Subthreshold logic circuits operate with a $V_{DD}$ that is lower than the transistor threshold voltage in order to ensure operation in the subthreshold region. By operating in the subthreshold region, the subthreshold leakage current may be used as the drive current for the gate [18]. Circuits implementing subthreshold supply voltages average an energy savings of 66% over the same circuit using only power gating of idle circuits [3].

2.2.2 Number Encoding

Traditionally binary coded decimal numbers (BCD) are used in digital machines to encode state machines and memory addresses. The problem with BCD from an energy standpoint is that an increment in the value of a number often requires changing two or
more bits. Gray coding can reduce energy consumption by 20–80% since it only requires a single bit transition to increment the value of a number [13].

2.2.3 Elimination of Glitches

A glitch is an unintended and unnecessary transition of the output of a gate due to a skew in the arrival times of the signals at the inputs to the gate. This glitch then propagates to the input of the next gate which could result in an additional glitch at that gate. In circuits with a large fanout, a single glitch can propagate across a large number of gates consuming large quantities of energy in useless transitions. The glitch often manifests as an output voltage that is in the midrange which causes subsequent gates to experience shoot through [9, 14]. Simulations show that removing or preventing a glitch only saves energy in cases of large fanout and propagation of the glitch. Therefore, glitch prevention and removal must be considered on a case by case basis.

2.2.4 Asynchronous Logic

Traditional synchronous circuits supply a clock signal to state machines and on every rising clock edge the inputs are evaluated and the outputs asserted after a brief propagation delay. The driving of a low skew, low jitter clock signal across the entire chip results in a high capacitance net with power hungry clock drivers. Rather than gate the clock to reduce gate activity, asynchronous logic has a property of having no clock and only activating gates and modules as they are needed [15]. However, asynchronous circuits require time and effort to design and are not certain to result in reduced energy consumption.

2.2.5 Architectural Partitioning and Layout

The partitioning of the design into logical blocks can result in lower capacitance and lower energy use [9]. Designs that have been partitioned into blocks can be organized in a manner that allows for effective layout techniques that utilize locality of reference.
Locality of reference is a philosophy of generating and using a signal in locations that are physically close to each other in order to minimize the capacitance of the signal line [9]. An example of this is the use of multiple memory and control blocks rather than a central processor and single memory block. Another example is to not use global buses that connect multiple blocks since these will be large. Instead dedicated signal lines are used between blocks so that the minimum bus length is used from the signal source to the destination as shown in Fig. 2.3. If one block is the control for another block these two blocks should be placed adjacent to each other on the die to minimize the capacitance of a bus with high activity. Likewise, a bus with low activity may be lengthened to accommodate the shortening of critical buses.

![Diagram of circuit blocks](image)

**FIGURE 2.3:** (a) The common bus has a high capacitance and wastes energy by not maintaining locality of reference. (b) Dedicated buses have minimal capacitance and save energy by maintaining locality of reference.

Idle circuit blocks still consume energy even when not being used. An effective technique to eliminate this wasted energy is to power down blocks that are not in use (power gating). Power gating requires additional logic circuits to generate the control signals to power down unused blocks and restore them as needed [10, 17].
2.2.6 Threshold Voltage

Logic gates using transistors with a low threshold voltage have increased static power consumption but result in a lower dynamic power consumption as compared to logic gates using transistors with a high threshold voltage. Gates with a high level of activity should have a low threshold voltage while gates with little activity or needing low static power consumption should have a high threshold voltage [10, 19]. Additionally, there are techniques to adjust transistor threshold voltage to meet the current gate activity and performance requirements. One uses dynamic body bias of the MOS transistors and another uses the floating-body effect of partially depleted SOI transistors to lower threshold voltage when $V_{DD}$ is high and vice versa [19]. To obtain a low static power the threshold voltage is raised and to obtain a high performance it is lowered.

2.2.7 Local Voltage Regulation

Low $V_{DD}$ circuits draw more current than a comparable circuit operating at a higher $V_{DD}$ for the same power consumption. This higher current is penalized with resistive losses in the power distribution grid and suffers from increased voltage droop. A technique to combat these losses is to use a high voltage power distribution grid with local voltage regulation for blocks in the chip [10].

2.2.8 Adiabatic Logic

Adiabatic circuits are systems in which the total energy remains constant. This is accomplished by having a charge recovery circuit in addition to the logic circuit. This charge recovery circuit may require multiple clocks or varying supply rail voltages depending on which technique is used [16]. The advantage of adiabatic logic is the low energy consumption of the logic circuits. However, this saving may be more than offset by the charge recovery block which may have to generate as many as four clock signals as is the case with adiabatic dynamic logic [20]. Therefore, it is not practical to use adiabatic
logic unless the energy saved is sufficient to overcome the energy consumed by the charge recovery circuit.

2.2.9 System Algorithm Design

The energy efficiency of a circuit is only as good as the underlying algorithm that determines the behavior of a circuit. A poorly designed algorithm will consume excessive energy no matter how much improvement is obtained by the choice of fabrication technology, logic design, layout, and so on.

2.2.10 Logic Family

The choice of a logic family also plays a role in the power consumption. The second component of the dynamic power in (2.1) is capacitance. Some logic families such as complementary pass-transistor logic use only n-type pass transistors which result in fewer gates and, therefore, lower capacitance [18, 21]. Subthreshold circuits work well with minimum sized devices [22] which also reduces the capacitance. The fabrication feature size determines the minimum device size and associated capacitance. Therefore, a lower dynamic energy consumption will result from fabrication processes with smaller feature sizes. For example, a buffer constructed of minimum sized devices may require a slower clock speed, and will use less dynamic energy due to lower capacitance.

The designer must weigh the advantages and disadvantages of specialized logic families with the standard CMOS cell libraries provided with the fabrication technology. The energy savings from a specialized cell library must be sufficiently large to warrant the time required to develop it. Instead, one could focus on optimizing the energy consumption in other parts of the design.
A simplified block diagram for the WSN node digital logic controller is shown in Fig. 3.1. It consists of a central digital logic block with its own small system memory block, memory for data storage, power control switches, input from the sensor and receiver, and output to the transmitter via logic level converters.

![Simplified block diagram for WSN node digital logic controller.](image)

FIGURE 3.1: Simplified block diagram for WSN node digital logic controller.

Before implementing any energy reduction technique, each technique was investigated in simulation to determine if it is suitable for implementation in the WSN node controller shown in Fig. 3.1. Simulations were performed using the Cadence Virtuoso Spectre circuit simulator using transistor level circuits with BSIM4 models. Two example circuits were created, one using standard design methodologies and the other using one of the techniques described in Section 2.2. The total energy used by each circuit to complete
its task was compared to determine if the technique saved energy. The techniques that demonstrated an energy savings are as follows: reduce the supply voltage, power gating, Gray coding, and system algorithm optimization.

3.1 Reducing Supply Voltage

The simplest and most effective energy reduction technique for all of the circuit blocks in Fig. 3.1 is to reduce the supply voltage so that the transistors operate in the subthreshold region. The reduction in supply voltage dictates a corresponding decrease in clock speed, but in a wireless sensor node a high bit rate is not required. It has also been shown that at higher frequencies the power consumed varies linearly with frequency and at lower frequencies the power consumption becomes independent of frequency due to static leakage. Therefore, for a given frequency the lowest possible supply voltage should be used [18].

In the application of a wireless sensor node the energy consumed by the transmitter and receiver dominate and thus determine the data rate which in turn determines the operational frequency of the digital logic circuits. In the target wireless sensor node the receiver [23] and transmitter [24] both are most energy efficient at their highest rates of 1 MHz which corresponds to a data rate of 500 kbps. However, a data rate of 250 kbps is chosen for the wireless sensor network in order to allow for better receiver sensitivity and lower the bit error rate. This data rate sets the digital logic clock frequency to 250 kHz which requires a supply voltage high enough to support this frequency.

Lowering $V_{DD}$ as low as possible may not be the best approach since as the supply voltage is lowered the leakage current is reduced, but the circuit delay time increases. The increase in circuit delay requires a lower clock speed which results in an increase in leakage energy per clock cycle. The increase in static energy and the lowering of dynamic energy result in a certain $V_{DD}$ that consumes the lowest total energy. An analytical model to
determine the optimal supply voltage is [22]:

\[ V_{\text{min}} = \left[ 1.587 \ln \left( \eta \cdot \frac{n}{\alpha} \right) - 2.355 \right] \cdot mV_T, \]  

(3.1)

where \( \eta \) is the delay factor from a non-step input, \( n \) is the number of stages in the inverter chain, \( \alpha \) is the activity factor, \( m \) is the subthreshold swing parameter, and \( V_T \) is the thermal voltage. It was found in [22] that a \( V_{DD} \) of 200 mV provided the optimal energy consumption for a 0.18 \( \mu m \) CMOS process. Another analytical model for the energy optimal supply voltage is [3]:

\[ V_{D_{\text{Dopt}}} = mV_T \left( 2 - \text{lambertW}(\beta) \right), \]  

(3.2)

where \text{lambertW} is the Lambert W function, and \( \beta \) is defined as:

\[ \beta = \frac{-2C_{\text{eff}}}{W_{\text{eff}}L_{PD}KC_g} e^2 > e^{-1}, \]  

(3.3)

where \( C_{\text{eff}} \) is the average effective switched capacitance, \( W_{\text{eff}} \) is the average total width, \( L_{PD} \) is number of gates in the longest path from the input to the output, \( K \) is a delay fitting parameter, and \( C_g \) is the output capacitance. It was found in [3] that 250 mV was the optimal supply voltage for energy efficiency of an 8-bit, 8-tap FIR filter operating at 30 kHz in a 0.18 \( \mu m \) CMOS process. Based upon these two models, represented in Fig. 3.2, it was estimated that the optimal \( V_{DD} \) for lowest energy consumption of the WSN node controller will be approximately 250 mV.
3.2 Logic Level Conversion

Operating at a low supply voltage is beneficial in lowering the energy consumption but complicates interfacing with surrounding circuits. The wireless transceiver and sensor circuits in the wireless sensor node operate at approximately 1.1 V which can drive the inputs of the subthreshold logic but the inputs of the transmitter cannot be driven by the subthreshold circuits. Therefore, the logic level converter shown in Fig. 3.3 is used at the outputs of the subthreshold logic circuits as shown in Fig. 3.1. The logic level converter is similar to a standard buffer except that the first inverter, (M1 and M2), has a diode tied PMOS device, M5, to reduce the threshold voltage of the first inverter low enough to be activated by subthreshold logic signals, but keep the output voltage high enough to meet the threshold voltage requirements of the second inverter. All devices are minimum width to minimize capacitance, but the gate lengths vary in order to achieve the desired...
switching and output characteristics. The gate of M5 is relatively long in order to produce a significant source-drain voltage drop and help limit shoot through current. M3 also has a relatively long channel length in order to limit the shoot through current. Shortening the gate length of M3 from 1 $\mu$m to 0.5 $\mu$m does not significantly improve the logic converter’s performance and results in a 33% increase in energy use. Conversely, lengthening the gate of M3 beyond 1 $\mu$m does little to reduce the energy consumption and begins to negatively impact the rise time of the output.

A simulation was performed with a pair of 32-bit FIFOs clocked at 250 kHz, one powered from the normal supply voltage of 1.1 V and the other in subthreshold with a 350 mV supply and a logic level converter at its output. The simulation was performed first with the FIFO input changing on every clock cycle. This results in each flop-flop changing state on every clock cycle. Next, the data input changed only on every other clock cycle which results in a 50% level of activity. The energy for 32 clock cycles was measured then averaged to get an estimated energy per clock cycle. This data is shown in Fig. 3.4. Note that the energy for the subthreshold FIFO includes the logic level converter. It is unlikely that the final implementation would have so few gates per logic level converter, but even so the subthreshold circuit with logic level converter uses at least an order of magnitude less energy than the standard implementation.
FIGURE 3.3: Logic level converter.

FIGURE 3.4: Comparison of subthreshold energy consumption per clock cycle for different gate activity levels.
3.3 Power Gating

Another energy reduction technique that is effective in a wireless sensor network node is to power down unused circuit blocks. Even when a circuit is not being utilized it still consumes static power and dynamic power [17] which can be eliminated by powering down the circuit block. Gating the clock can eliminate dynamic power consumption but does nothing to reduce the static power consumption [10], which is becoming a significant factor as the feature sizes decrease. An issue with power gating is that the wake-up time for the circuit can be lengthy [10] which requires a delay in addition to the overhead of the power down management circuit.

The local data memory and data storage memory blocks in Fig. 3.1 are large circuits that may go unused for great lengths of time, making them ideal candidates for power gating. A typical 16 byte memory block was simulated with and without power gating. As can be seen from Table 3.1, the energy to read from and write to the memory block is not affected by the power switch. However, if the memory block is not in use and the address and data buses are idle then the power gated memory block uses 359 times less energy per clock cycle than the un-gated memory block. If the address and data buses are active, as is the case when an adjacent memory block is in use, the power gated memory block uses 434 times less energy per clock cycle. It is apparent from these numbers that a significant energy loss due to leakage currents can effectively and significantly be reduced by power gating the unused circuit blocks. However, in order to realize a net energy savings, the power gated blocks must have significant leakage current and the power gate control logic must use little energy.

The power switch shown in Fig. 3.5 is a simple device that places a PMOS device between the memory block’s supply pin and $V_{DD}$ and an NMOS device between the memory block’s return pin and ground. The enable signal drives the NMOS device directly and the PMOS device via a standard inverter. What is unique about the switch is that
TABLE 3.1: Energy per clock cycle for a 16 byte memory block with and without power gating under typical usage scenarios

<table>
<thead>
<tr>
<th></th>
<th>Buses Idle</th>
<th>Buses Active</th>
<th>Read</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>263 fJ</td>
<td>319 fJ</td>
<td>305 fJ</td>
<td>336 fJ</td>
</tr>
<tr>
<td>Gated</td>
<td>0.733 fJ</td>
<td>0.735 fJ</td>
<td>304 fJ</td>
<td>335 fJ</td>
</tr>
</tbody>
</table>

the device bulk connections are tied to the gates, this increases the threshold voltage when the switch is off, which minimizes the leakage current [2], and lowers the threshold voltage when the switch is being turned on. However, this requires the use of a triple-well process to isolate the NMOS bulk which makes this technique suitable only for select devices such as the power switches due to the added size of the device well.

FIGURE 3.5: Power switch.
3.4 Gray Coding

Gray coding is a number scheme by which a sequential increase in the value of a number results in a single bit transition. Table 3.2 shows an example Gray code scheme compared to the binary coded decimal (BCD) value and the decimal equivalent. When used to encode an address bus, Gray code can reduce energy consumption by reducing the number of bit transitions when accessing sequential memory locations. However, Gray coding only saves energy for sequential incrementing and decrementing, a random access can result in the Gray code having more bit transitions than BCD [13]. Therefore, Gray coding should only be used where most or all of the transitions are sequential. Gray coding can also be used to encode state machines since it reduces state register and combinational logic activity for sequential state transitions. Various techniques have been introduced to develop hybrid Gray code state machines to minimize the Hamming distance between expected state transitions based on statistical models of the state machine input [25, 26]. However, in a wireless sensor node some of the state machines transition in a sequential and known manner in which it is most effective to simply use Gray codes to encode the entire state machine. By combining Gray coded address buses and state machines that have a known sequential access the gate activity can be reduced, further reducing unnecessary energy consumption.

The memory address bus connecting the digital logic block to the data storage blocks in Fig. 3.1 is lengthy, has a relatively large capacitive loading, and is only incremented sequentially, making it a good candidate for Gray coding. A simulation of the memory address bus was performed to determine the feasibility of Gray coding for energy savings. The simulation modeled the 5-bit address bus to each of the 7 data memory blocks with each memory block having 4 gates on each bit line for a total of 28 gates on each bit line. To further increase accuracy, the metal to substrate capacitance for a typical signal line was added to each bit of the bus. As shown in Fig. 3.6, Gray coding provides a slight
<table>
<thead>
<tr>
<th>Gray</th>
<th>BCD</th>
<th>Decimal</th>
<th>Gray</th>
<th>BCD</th>
<th>Decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0000</td>
<td>0</td>
<td>1100</td>
<td>1000</td>
<td>8</td>
</tr>
<tr>
<td>0001</td>
<td>0001</td>
<td>1</td>
<td>1101</td>
<td>1001</td>
<td>9</td>
</tr>
<tr>
<td>0011</td>
<td>0010</td>
<td>2</td>
<td>1111</td>
<td>1010</td>
<td>10</td>
</tr>
<tr>
<td>0010</td>
<td>0011</td>
<td>3</td>
<td>1110</td>
<td>1011</td>
<td>11</td>
</tr>
<tr>
<td>0110</td>
<td>0100</td>
<td>4</td>
<td>1010</td>
<td>1100</td>
<td>12</td>
</tr>
<tr>
<td>0111</td>
<td>0101</td>
<td>5</td>
<td>1011</td>
<td>1101</td>
<td>13</td>
</tr>
<tr>
<td>0101</td>
<td>0110</td>
<td>6</td>
<td>1001</td>
<td>1110</td>
<td>14</td>
</tr>
<tr>
<td>0100</td>
<td>0111</td>
<td>7</td>
<td>1000</td>
<td>1111</td>
<td>15</td>
</tr>
</tbody>
</table>

Energy savings when used on this memory address bus, but if the bus load is reduced to 7 gates (one quarter of the loading of the previous simulation), the Gray coding scheme actually consumes more energy due to the energy overhead of converting from BCD to Gray code. Therefore, Gray coding a bus will save energy if it is only accessed in a sequential manner and has significant capacitive loading.
FIGURE 3.6: Comparison of BCD to Gray coded 5-bit address bus energy use per clock cycle. The Gray code uses 7% less energy with a loading of 28 gates but 5% more energy with 7 gates.

3.5 Algorithm Design

One facet of energy reduction that can be easily overlooked is the optimization of the algorithms used in the digital logic block in Fig. 3.1. Optimizing the algorithm to use as few steps as possible will naturally reduce the energy used [9]. Other techniques involve exploiting symmetry and approximations that allow for simple arithmetic such as a multiplier consisting of a shift register and two adders [15]. The WSN should have as few tasks as possible and as much arithmetic operations as possible passed on to post-processing of the data by systems with a limitless energy supply. Those simple arithmetic operations that may be required, such as addition and subtraction, should be implemented with a design that uses the least amount of energy. There is also a need for networking protocols and data encoding to minimize the energy consumed by the radio transceiver.
Data encoding can be somewhat problematic as reducing the number of data bits transmitted is very important to energy conservation, but complex data compression can also consume vast quantities of energy. The typical networking protocol consists of a complex arrangement of routing tables, bandwidth controls, handshaking, and error checking. Implementing any of these aspects of a typical network protocol in a wireless sensor node is impractical from an energy consumption perspective [27]. Therefore, an energy efficient wireless sensor network consists of simple nodes with a network entirely controlled by the central hub whose job it is to dictate node activity, network routing, and perform error checks on the data.

As can be seen in Fig. 3.7, the transmitter consumes a majority of the energy used by the node so it is imperative that the number of bits to be transmitted is minimized. One way to minimize the quantity of data being transmitted is to encode the sensor data as an 8-bit reference followed by 4-bit two’s complement values that represent the amount of change since the 8-bit reference value was taken. If every sample is 8 bits the transmitting and receiving of 31 samples and a 7-byte header requires 9.81 clock cycles per sample, which equates to 90.2 nJ per sample. If a single 8-bit measurement is taken followed by 60 4-bit samples then the transmitting and receiving of these 61 samples and a 7-byte header requires 4.98 cycles per sample, or 45.8 nJ per sample. In order for a data encoding scheme to be viable, the energy consumed by the data encoder must be less than 44.4 nJ per sample. Any energy savings, however slight, will be beneficial to the WSN in Fig. 1.1(a) as the number of samples to be transmitted and the energy used grows exponentially for nodes closer to the hub as they have the added burden of passing on data from more remote nodes.
FIGURE 3.7: Comparison of energy per clock cycle for each component in the WSN node. Also shown is the energy for the logic circuits before and after optimization for low energy consumption.

A simulation of a data encoder was performed to measure the energy used to take a single 8-bit sample followed by 60 4-bit encoded differential values. The average energy per sample is 28.3 fJ which is far less than the maximum allowed 44.4 nJ. The data encoding requires a memory block to store the 8-bit reference value, the previous results do not account for this energy, which when implemented in a SRAM currently consumes approximately 16 fJ per cycle. This demonstration of the advantage of reducing the number of bits transmitted is also proof that using a network protocol that requires the nodes to communicate and arbitrate between themselves is not practical and the best solution is to have a central hub control every aspect of the network.
3.6 Scaling Trends

As the fabrication technology scales down the effectiveness of these techniques for use in a WSN node controller will change considerably. In order to determine this trend, simulations were performed in a 90 nm CMOS process and compared to the results when implemented in the 0.18 µm CMOS process. The general trend noted was a decrease in dynamic energy consumption but an increase in the static energy consumption. At some point the overall static energy consumption will equal and then surpass the dynamic energy consumption which will shift the focus from lower gate activity to minimizing the static leakage energy. For this reason, in the future, subthreshold voltage supplies will continue to produce an energy savings. However, there will be an increasing need to locate and operate at the minimum energy supply voltage as shown in Fig. 3.2. Some techniques such as Gray coding already provide a minimal return in energy savings and may eventually cease to be an effective tool. It is anticipated that power gating will become very important since the energy cost to implement the control logic will be reduced and the energy saved by it will increase. And of course reducing the number of data bits transmitted will continue to be the most effective method to reduce WSN node energy use. It will likely be advantageous to introduce more complex data compression and encoding schemes as long as these blocks are power gated and turned off when not in use.
4 CIRCUIT DESIGN

The WSN node digital logic controller in Fig. 3.1 is not powered until activated by the central hub, then it proceeds to operate as shown in Fig. 4.1. Once activated, the controller begins operation by listening for commands from the hub or adjacent sensor nodes. The commands from the hub may tell the node to take a sensor measurement, transmit its data, clear its memory, or to power down. Additionally, the sensor node may receive a command from a more remotely located node, directing it to temporarily store the data that it is being sent. As the digital logic controller processes these commands it enables or disables power to memory blocks.

In addition to selectively enabling memory blocks, several other energy reduction techniques have been employed. An 8-bit memory block holds a reference data measurement that is used by the data encoder to reduce the number of bits needing to be stored and transmitted. The address bus to the data storage memory blocks is Gray coded to reduce the energy used to drive the bus. The networking protocol requires little processing.
by the node by requiring the central hub to control the network and post-process data. And finally, proper layout techniques were utilized to minimize parasitic capacitance and maintain locality of reference.

4.1 Subthreshold Supply Voltage

The supply voltage is designed to be in the subthreshold region in order to take advantage of the energy reduction it provides. In order to interface with the remainder of the WSN node, which operates at 1.1 V, logic level converters are required for the output pins. The design of these converters is shown in Fig. 3.3.

4.2 Data Memory

There are two types of memory blocks in Fig. 3.1, those that store the data generated in the node by its sensors and those that temporarily store the data sent from other nodes as it flows through the WSN towards the hub. The former see frequent and long term use while the latter are only used during brief periods when the hub is gathering data from the WSN. There are seven memory blocks of 32 bytes for storing data from other nodes, each of these blocks is independently power gated using the switch shown in Fig. 3.5. When data arrives from another node the next available data storage block is energized and filled with the data. The memory for storing local sensor data measurements is partitioned into two blocks of 16 bytes. Only one of these blocks receives power continuously, the other is power gated and is not used unless 17 or more bytes of memory are needed. The power gated memory blocks will continue to remain powered until the node is directed by the hub to delete its memory contents. By dividing the memory into blocks, power gating is effectively utilized to realize an energy savings.
The address bus to the seven data storage memory blocks has a significant bus length and capacitive loading. Encoding the 5-bit address bus in Gray code saves 265 fJ per clock cycle as was shown in Fig. 3.6. This energy saving is only realizable since the address bus is only incremented sequentially and never randomly which would negate the advantage of the Gray coding.

4.3 Data Compression

To reduce the number of bytes transmitted and the amount of data storage memory required, the data is encoded as 4-bit 2’s complement values of change from a reference value. The data encoder inside of the digital logic block retains this 8-bit reference value in a special memory register and uses a simple and energy efficient ripply-carry adder architecture to compute the change from the reference measurement to the current measurement.

4.4 Communication Protocol

All complexity and burden in implementing the networking protocol have been removed from the nodes and placed into the hub. The task of the node is then very simple, listen for a command then act on it. The hub is responsible for maintaining a routing table, tracking sensor energy use, tracking sensor memory use, waking up the nodes, directing the nodes to perform various tasks, decoding the sensor data, error checking the data, and putting the nodes back to sleep. The nodes have no error checking and no networking protocol overhead that would consume energy needlessly.
5 MEASURED RESULTS

The low energy design techniques that showed the most promise were used to design and implement a WSN sensor node digital logic controller in a 0.18 µm CMOS process. The chip micrograph is shown in Fig. 5.1 and it is 1.4mm × 1.2mm. A testbench simulated data input from the receiver and the current draw was measured under a variety of typical situations to assess the energy consumption.

![Chip microphotograph](image)

FIGURE 5.1: Chip microphotograph.

5.1 $V_{DD}$ and Clock Speed

The power is calculated and then divided by the clock speed in order to obtain a normalized energy consumed per clock cycle. This data is shown in Table 5.1 where we can see that, in general, a lower supply voltage and a higher data rate result in less energy
used to complete the same task. The wireless sensor network is designed to operate at 250 kbps, but the digital controller was found to need a $V_{DD}$ of 350 mV to operate at 250 kbps. This results in a 336 fJ per clock cycle energy consumption. As a comparison, the energy consumption for 125 kbps and 62.5 kbps are also shown, a supply voltage of 300 mV can be used for both of these data rates.

<table>
<thead>
<tr>
<th>$V_{DD}$ (mV)</th>
<th>62.5 kbps</th>
<th>125 kbps</th>
<th>250 kbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>461 fJ</td>
<td>331 fJ</td>
<td>–</td>
</tr>
<tr>
<td>350</td>
<td>601 fJ</td>
<td>437 fJ</td>
<td>336 fJ</td>
</tr>
<tr>
<td>400</td>
<td>779 fJ</td>
<td>568 fJ</td>
<td>425 fJ</td>
</tr>
</tbody>
</table>

### 5.2 Logic Level Conversion

One of the largest and easiest energy savings that can be realized is from subthreshold operation. Every subthreshold circuit must interface with other blocks. In this case, the outputs to the transmitter are nominally 1.1 V which requires a logic level converter as shown in Fig. 3.3. Simulations show that the converter will consume 319 fJ per clock cycle which is insignificant compared to the energy saved. Actual measured data shows each converter consumes 440 fJ per clock cycle. Based on Fig. 3.4, it was expected that operating in the subthreshold region at 350 mV would result in a 92% reduction in energy per clock cycle as compared to operating at 1.1 V. Measured results show that operating at 1.1 V consumes 177 pJ per cycle, whereas at 350 mV, the energy for the controller logic and logic level converters is 3.8 pJ per clock cycle, i.e., a reduction of nearly 98%. This better than expected energy savings is due to a lower average gate level activity
throughout the circuit than the 50% gate activity level that indicated a 92% reduction in energy use.

5.3 Power Gating

The logic blocks in the wireless sensor node are quite small so it is practical only to gate power to the entire WSN node controller when it is not in use and memory circuits that are not in use when the WSN node is active. Based on the simulated data shown in Table 3.1, we would expect to see at least 300 fJ per clock cycle energy reduction by powering down a 16 byte memory block. However, actual measurements showed only an energy savings of 56 fJ per clock cycle. While not as high as expected, this is still more than enough to compensate for the energy used to generate the power control signal.

The sensor node also has 7 memory blocks that contain 32 bytes so we would expect to save approximately 600 fJ per clock cycle per memory block when not in use. Unfortunately, a design error of using a latch instead of a flip-flop resulted in a logic error that prevents these memory blocks from ever turning on so only the leakage energy can be measured. This is found to be 1 fJ per clock cycle per memory block which closely matches the simulated results. Since these block cannot be turned on, we can only extrapolate from the 16 byte memory block measurements that a savings of 108 fJ per clock cycle per 32 byte memory block will actually be realized. A revision of the design is in fabrication that will allow actual measurements of the 32 byte memory blocks to be taken.

5.4 Effectiveness

It is difficult to quantize the energy saved by each method as implementing one method will have an impact on the energy saved by another method. For example, by im-
implementing a subthreshold supply voltage all of the techniques to save energy by reducing
gate activity appear to be less effective than they otherwise would be. It is also difficult to
make a broad statement about the energy saved because some techniques only save energy
under certain circumstances. For example, Gray coding a bus only saves energy when that
bus is being used, and data encoding only saves energy when transmitting data. Despite
these challenges in making an equitable and definitive evaluation of the energy saved by
implementing these techniques, Table 5.2 shows the percent of energy reduction per clock
cycle by each technique compared to a traditional design (assuming $V_{DD}$ is 350mV and a
250 kbps data rate).

TABLE 5.2: Comparison of energy reduction per clock cycle by implementing subthreshold
supply voltage, power gating, and Gray coding.

<table>
<thead>
<tr>
<th></th>
<th>Subthreshold</th>
<th>Power Gating</th>
<th>Gray code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduction in energy</td>
<td>97.3%</td>
<td>0.46%</td>
<td>0.03%</td>
</tr>
</tbody>
</table>

Table 5.3 compares the digital controller to recently published WSN digital con-
trollers and Table 5.4 summarizes the measured results. The design presented here is the
lowest energy digital controller reported.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>$V_{DD}$</strong></td>
<td>1.0 V</td>
<td>1.2 V</td>
<td>0.6 V</td>
<td>0.36 V</td>
<td>1.0 V</td>
<td>0.35 V</td>
</tr>
<tr>
<td>Technology</td>
<td>0.25µm CMOS</td>
<td>0.25µm CMOS</td>
<td>0.18µm CMOS</td>
<td>0.13µm CMOS</td>
<td>0.13µm CMOS</td>
<td>0.18µm CMOS</td>
</tr>
<tr>
<td>Throughput</td>
<td>0.033 MIPS</td>
<td>–</td>
<td>6 MIPS</td>
<td>1.1 MIPS</td>
<td>–</td>
<td>0.25 MIPS</td>
</tr>
<tr>
<td>Clock Speed</td>
<td>500 kHz</td>
<td>100 kHz</td>
<td>46–62.5 MHz</td>
<td>833 kHz</td>
<td>16 MHz</td>
<td>250 kHz</td>
</tr>
<tr>
<td>RF Data Rate</td>
<td>1 Mbps</td>
<td>250 kbps</td>
<td>–</td>
<td>–</td>
<td>50 kbps</td>
<td>250 kbps</td>
</tr>
</tbody>
</table>

**TABLE 5.3**: WSN node controller compared to other recent WSN node controller designs
TABLE 5.4: Summary of Measured Results

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply Voltage</td>
<td>350 mV</td>
</tr>
<tr>
<td>Technology</td>
<td>0.18µm CMOS</td>
</tr>
<tr>
<td>Energy</td>
<td>336 fJ/Inst</td>
</tr>
<tr>
<td>Throughput</td>
<td>0.25 MIPS</td>
</tr>
<tr>
<td>Clock Speed</td>
<td>250 kHz</td>
</tr>
<tr>
<td>RF Data Rate</td>
<td>250 kbps</td>
</tr>
<tr>
<td>Core Area</td>
<td>168 x 175 µm</td>
</tr>
<tr>
<td>Transistor Count</td>
<td>11,942</td>
</tr>
<tr>
<td>Gate Count</td>
<td>1033</td>
</tr>
</tbody>
</table>
6 CONCLUSION

A variety of energy savings techniques have been investigated for their suitability in implementing the digital controller of a wireless sensor network node. The techniques that showed promise were implemented in a 0.18 µm CMOS process and measured results were used to determine the energy savings. Techniques such as Gray coding and locality of reference were not measurable so simulated energy savings is presented. However, subthreshold supply voltage, power gating, and data encoding are measurable and the measured energy savings is presented. The effectiveness of these techniques is evident in Fig. 3.7 where it is shown that the energy per clock cycle of the logic circuits with no optimization is significant when compared to the other circuit components in the node. With select energy reduction techniques that have been implemented, the energy consumption is reduced by 98%. This results in the lowest energy WSN node controller reported to date.
BIBLIOGRAPHY


