BodyLAN™: A Low-Power Communications System

by

Thomas J. Barber Jr.

B.S. Electrical Engineering
Michigan Technological University, 1994

Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degree of

Master of Science

at the
Massachusetts Institute of Technology
February 1996

Author .....
Department of Electrical Engineering and Computer Science
30 January 1996

Certified by .....
Anantha Chandrakasan
Analog Devices, Inc. Career Development Professor
Thesis Supervisor

Accepted by .....
F. R. Morgenthaler
Chairman, Department Committee on Graduate Students

BodyLAN™: A Low-Power Communications System

by

Thomas J. Barber Jr.

Submitted to the
Department of Electrical Engineering and Computer Science

30 January 1996

In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

In the last few years the wireless communications market has exploded with products such as pagers and cellular phones becoming common items. The trend in this field has been to move toward communications over a smaller area using less power. This thesis presents the Body-LAN™ system which is a communications network designed to gather data from within the sphere of the body and provide that information to the user through wireless links. To make the system marketable the battery life of the component must be maximized. This presents a challenge to provide wireless communications at microwatt power levels. This goal is accomplished through the use of a custom integrated circuit controller which dissipates under 50 μW.

Thesis Supervisor: Anantha Chandrakasan
Title: Analog Devices, Inc. Career Development Professor

Acknowledgments

I would like to thank my thesis advisor Professor Anantha Chandrakasan who provided me with all the support, encouragement and motivation I needed to complete this project. I would also like to thank Phil Carvey of Bolt Barenek and Newman for defining the project and letting me run with it. Also, for his invaluable advice and feedback. Thanks to Professor Mitch Trott who took time out of his schedule to review some of the communications theory in the project.

The members of our research group especially Raj Amirtharajah who took the time to read this entire thesis. Also, Vadim Gutnik, for being my general Unix guru. Mike Perrott and Don Hitko for helping me with the analog side of the house. As well as Duke Xanthopoulos, Jim Goodman and Tom Simon for sharing their experience with in circuit design.

Finally, I would like to thank those who helped in non-technical ways. My roommates Matt Greenman, Randy Berry and Chris May for helping keep me sane (and walking the dog). Jill Wilkens for getting me out into the real world occasionally (and feeding me). Mark Gustafson and Nate Thern for letting me know what it was like on the outside. Master Jon Henkel without whom I would haven't have had the courage to attend MIT. Finally, Thomas and Mary Ann Barber without whom I would be here at all.

From the very bottom of my heart, thank you all. Goodnight...

For Mom and Dad,

Table of Contents

List of Tables

List of Figures

Chapter 1: Overview of BodyLAN™ System

The BodyLAN™ system is a low-power wireless communications system designed to operate within the sphere of the body. The wireless communications industry has been increasingly moving toward networks that communicate over smaller areas with lower power levels (see Table 1). Until recently most wireless communications was in the form of satellite transmissions. In the 90s, the mobile cellular phone industry has exploded and work has begun on short range wireless networks to communicate data (e.g. Xerox's DigitalDesk [1] and Berkeley's InfoPad Project [2]). BodyLAN™ proposes to take this progression one step further, by lowering the power consumption by more than an order of magnitude.

System	Range	Typical Bit Rate	Transmit Power
Satellite [14]	23,000 miles	155 Mb/s	10 - 100 W
Mobile Cellular Phone [5]	500-1000 feet	700 kbit/s	.1 - 1 W
Wireless LAN [5]	15-25 feet	2 Mb/s	.1 - 1 mW
BodyLAN™	6-10 feet	200 kbit/s	1-10 μW

Table 1: Comparison for Various Wireless Networks

The BodyLAN™ system is intended to augment human senses through the use of specialized sensors and a computer interface. These sensors will provide the user with more detailed data than human senses alone. For example, where a user could determine approximate air temperature (hot, cool, etc.) with a BodyLAN™ sensor the user could be provided with the exact air temperature. The applications of this technology range from medical applications, such as non-invasive patient monitoring, to sports and fitness training, such as automated training logs.

1.1 BodyLAN™ Network Configuration

There are three major network configurations: Star, Ring, and Complete[3]. In the Star configuration, all communication occurs through a single central node, referred to as the Hub. In a Ring configuration messages are passed from node to node until the message reaches the desired node. In a Complete configuration all communication proceeds directly to the desired node.

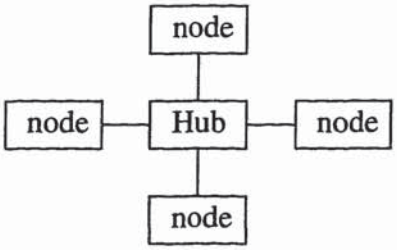

Figure 1a: Star Configuration

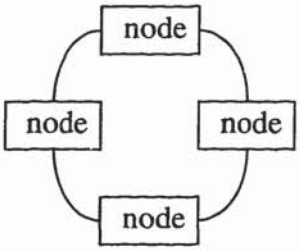

Figure 1b: Ring Configuration

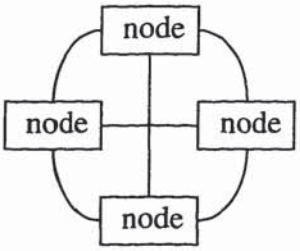

Figure 1c: Complete Configuration

Each node in a network can either be a complete system or a slave to another node. If the node is a complete system it has the ability to independently send data to and receive data from other nodes as well as accept input and deliver output to a user. If a node is a slave to another node the functionality can be reduced. For example, in a Star configuration system the Hub can conveniently control all the communications, therefore the functionality of the other nodes can be reduced.

The Star configuration is preferable for low-power applications because of the distribution of the power consumption. For the BodyLAN™ system reducing the power consumption is important because reducing the power consumption is equivalent to reducing the weight. In a Star configuration it is possible to partition computation such that most of the functionality resides in a single location, the Hub. Concentrating the functionality in one location translates to concentrating the power dissipation (and the weight) in a single node while each of the other nodes can be implemented with the minimum functionality and weight. This translates to a system with a single centralized weight source, which can be worn where the effect of the weight is minimized (i.e., on the waist or back) and a large number of extremely small light functional blocks. If a Ring or Complete configuration is used, all the nodes in the system are identical since the functionality of each of the nodes must be the same due to the fact each node has an identical connection to the network. Therefore, a functional block worn on the wrist would need to be the same size and weight as a block worn on the waist, which is not reasonable.

The Star configuration should also dissipate the minimum amount of power for the BodyLAN™ application. The purpose of the BodyLAN™ network is to gather information and provide the information to a user. If the user is assumed to be in a single location the information can be sent to the user in a single transmission using either the Star or Complete configurations. Using

a Ring configuration it would take an average of n/4 transmissions where n is the number of nodes in the system. In the Complete configuration each node has a channel to all of the other nodes, however, if these channels are not necessary (as in the BodyLAN™ system) providing those channels dissipates excess energy. Therefore, the Star network configuration is used for the BodyLAN™ application because the weight distribution of the Star configuration is optimal and the type of communications necessary (i.e., sensors gathering information for a single user) favors the Star configuration.

1.2 BodyLAN™ Communications Protocol

There are many methods of dividing access to a communications channel, three of which are: by time, by frequency and by code. When a channel is divided by time, Time Division Multiple Access (TDMA), each node has a scheduled time to use the channel and is inactive during all other times. When a channel is divided by frequency, Frequency Division Multiple Access (FDMA), each node communicates using a different frequency range and all nodes can be active all the time. When a channel is divided by code, CDMA, all the nodes share time and frequency as in Direct Sequence Spread Spectrum (DSSS) where the data is modulated by a high speed code such that a receiver with the same code receives the data with little additional noise from other channels.

The Time Division Multiple Access protocol is the natural choice for low-power applications since there is an inherent low-power (inactive) mode. Depending on the data rate the system can be in the low-power mode for a significate portion of the time. The duty cycle for a TDMA system will be defined as the data rate the system is being used at divided by the maximum data rate of the system. For example if a TDMA system where the data is transmitted at 200 kbits/s is being used at 2 kbit/s, the duty cycle is 1:100. The average power dissipated in a TDMA system is given by Eq. 1, where *d* is the duty cycle for the system.

$$P_{TDMA} = dP_{active} + (1-d)P_{inactive} \qquad \text{Eq. 1}$$

The Frequency Division Multiple Access protocol is unsuitable for low-power applications. By definition, FDMA protocol involves establishing a continuous link between each node and the Hub (in a Star configuration) at different frequencies [4]. Maintaining these continuous links while there is no data transmission causes the power dissipation to be higher.

The Code Division Multiple Access protocol has been used for low power communica-

tions[5]. A CDMA link can have the same inherent low-power mode (and duty cycle) as a TDMA system. The addition of the code to the signal causes excess power dissipation as in DSSS where the data must be transmitted at a much higher rate, the chip rate, which is typically orders of magnitude larger than the data rate [4].

For the BodyLAN™ system the TDMA communications protocol is preferred over the CMDA protocol. To implement the BodyLAN™ system at the lowest possible power level it is critical to power-down the RF hardware as much as possible, since the RF hardware can consume up to 5 mW while active continuously. Since the RF section is completely powered down during the interval between communications bursts, the receiver must re-synchronize for each burst. Synchronizing a TDMA system consists of synchronizing the receiver symbol clock to the transmitter symbol clock. Using a voltage controlled crystal oscillator the frequency will not vary much and synchronization can be maintained during power down by storing the control voltage on a capacitor. If the voltage is stored on a large capacitor through low-leakage components (i.e., high quality diodes) the control voltage degradation should be minimized. Synchronizing a CDMA system involves synchronizing the receiver and transmitter code generators as well as symbol clocks. Again, using a voltage controlled crystal oscillator the code lock can be maintained during power down by running the code generator continuously. If the code generator is not run continuously, the code must be reacquired before every transmission, greatly increasing the burst length necessary. For an extremely low power system neither of these options is acceptable. A code generator running continuously at even 10 times the data rate would consume more power than the rest of the system. Increasing the active period of the RF sections would greatly increase the power consumed and decrease the effective data rate (due to the time spent synchronizing the code). Therefore, the BodyLAN™ system is designed to use a TDMA communications protocol.

1.3 BodyLAN™ Architecture

The BodyLAN™ architecture consists of two components a central Hub and many individual nodes, called Personal Electronic Assistants (PEAs). The Hub is responsible for interfacing with the user, controlling the overall TDMA schedule and gathering and processing the data from the PEAs. The PEAs are responsible for initializing the connection to the network, gathering data and transmitting the data to the Hub.

The Hub system consists of two major components: the central Hub computer and the BodyLAN™ controller/modem interface card. The central Hub computer is responsible for pre-

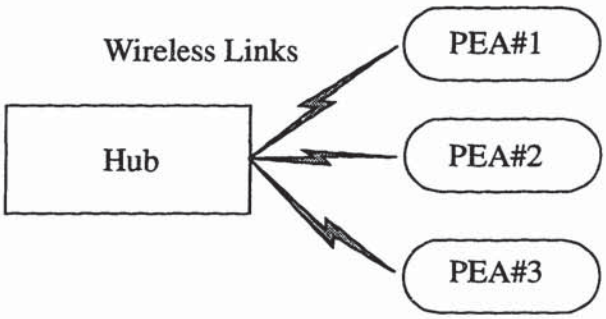

Figure 2: BodyLAN™ Architecture

senting the data gathered from the PEAs in a format that is acceptable to the user. The BodyLAN™ controller/modem interface card is a PCM/CIA card that is responsible for the TDMA scheduling and gathering and processing data from the PEAs.

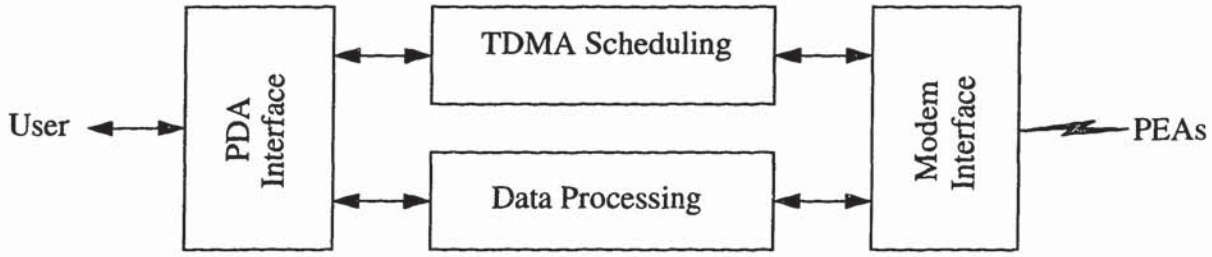

Figure 3: PCM/CIA Interface Architecture

The Personal Electronic Assistants (PEAs) are the heart of the BodyLAN™ system. The PEAs provide all the specialized functionality for the system. A typical PEA will consist of four major components: one or more sensors, a radio modem, a microprocessor and a custom controller integrated circuit. The sensor(s) are responsible for collecting data, which can be provided to the user through the Hub computer. The radio modem provides a wireless link between the PEA and the Hub computer. The microprocessor handles the higher level data processing functions that need to be performed locally (e.g. computing the amount of phase correction for the symbol clock) and provides flexibility due to it's programmable nature. Once the system functionality has been proven and the need for flexibility has decreased the microprocessor functionality will be moved onto the controller integrated circuit. The custom controller integrated circuit provides the control of the entire sensor system.

The BodyLAN™ controller integrated circuit is responsible for synchronizing and controlling the interaction between the PEA and the Hub computer. The custom integrated circuit is

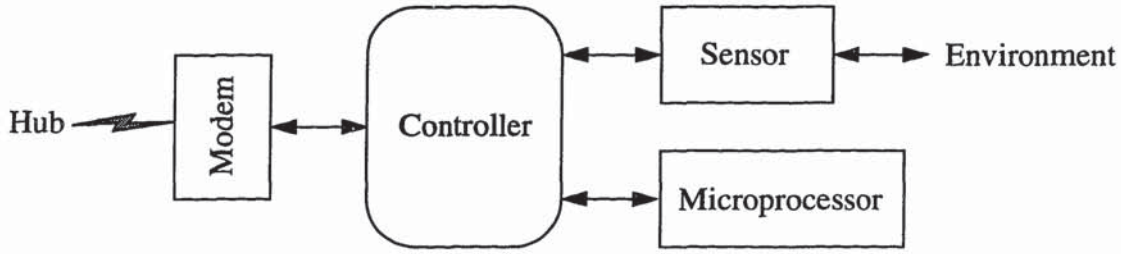

Figure 4: PEA Architecture

responsible for determining if there is a Hub present and initializing the communications link with that Hub. The custom integrated circuit is responsible for providing the information needed to phase-lock the local symbol clock to the Hub symbol clock. The custom integrated circuit is responsible for controlling the interaction of the Hub and the PEA, including maintaining the phase-lock, extracting data from the sensors and transmitting it to the Hub, providing a channel for communications between the microprocessor and the Hub and controlling the Command and Control Channel (CCC) interaction. The architecture of the custom integrated circuit is derived

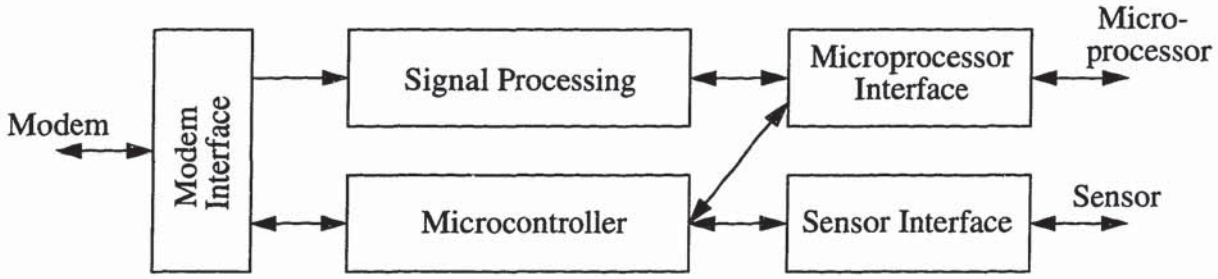

Figure 5: Custom Controller Integrated Circuit Architecture

from the dual functionality of handling the synchronization and communications. There is a signal processing block which handles attaching the PEA to the BodyLAN™ network and a microcontroller block which handles the TDMA plan execution. The rest of the blocks interface to the rest of the system.

1.4 BodyLAN™ System Power Consumption

The power consumption of the BodyLAN™ system must be minimized to provide a system that is convenient from a user's perspective. In normal use, there will be a single Hub and many PEAs. While changing the battery in a single centralized Hub with regularity is acceptable, much in the same way replacing the batteries in notebook computers is acceptable, changing the batteries in a PEA with regularity is not acceptable, much in the same way changing a watch bat-

tery with regularity is not acceptable. Therefore, the power dissipation in the PEA must be minimized.

The PEA can be broken down into four components: the radio modem, the microprocessor, the sensor and the controller. The radio modem has been designed at Bolt Beranek and Newman to comply with FCC regulation. The microprocessor was chosen after comparing the power consumption of a number of microprocessors. The design of low power sensors is an area for further research and the design of the controller is the focus of this thesis.

The BodyLAN™ radio modem has been designed to comply with the regulations of the Federal Communications Commission (FCC). The FCC Part 15 Specification allows for unregulated transmission in the 320-400 Mhz band as long as the field strength is below 200 μV/m at 3 meters [15]. The BodyLAN™ radio modem consists of two independent sections, one for transmission and one for reception. The transmission section consumes 1 mA at 3 V when active and the receiver consumes 1.6 mA at 3 V when active. Including the overhead time needed to power-up the oscillators transmission can be accomplished at 2 nJ/bit and reception can be accomplished at 3 nJ/bit, using 4-bit bursts. This translates to a power dissipation of 5 μW per kbit/s of communications bandwidth.

The PIC16C64 microprocessor manufactured by MicroChip Corporation was selected for use in the BodyLAN™ system. The PIC16C64 exhibited the lowest power consumption per instruction of any microprocessor examined. The PIC16C64 can execute 4 million instructions while dissipating 2mA at 3 V. This gives a cost of 1.5nJ per instruction. The goal of this project was to design the digital controller such that the magnitude of the power consumed by the controller was equal to or less than the power consumed by the rest of the system. Both the radio modem and the microprocessor function in the microwatt power range, so the power consumption of the controller should be in the microwatt range.

Chapter 2: Review of Low-Power Design Principles

The purpose of this chapter is to provide a quick review of low-power design principles. The sources of power consumption will be reviewed first.

2.1 Power Consumption Basics

There are four major components of power consumption in CMOS digital circuits: switching, short-circuit, leakage, and static. Switching power is the energy used to charge and discharge the parasitic capacitances associated with mosfets during transitions. Switching power is the largest component of power dissipation for CMOS circuits, accounting for 90% of the power consumed[2]. The switching power consumed by a circuit can be calculated as:

$$P = \alpha_{0\rightarrow 1} C_L V_{DD}^2 f \qquad \text{Eq. 2}$$

Where V_{DD} is the source voltage, f is the frequency of the switching, C_L is the amount of physical capacitance in the circuit, and $\alpha_{0->1}$ is the activity factor. The activity ($\alpha_{0\rightarrow 1}$) is the probability that the capacitance C_L will be charged during a given cycle.

Short-circuit power is also dissipated during switching. During a transition both the p-network and the n-network are on while the output voltage is between $V_{DD} - |V_{Tp}|$ and V_{Tn}. This creates a direct path between supply and ground. The energy lost due to current flowing directly between supply and ground during a transition is the short-circuit component of power consumption. If the power supply voltage is less than the sum of the magnitudes of the threshold voltages for the n-channel and p-channel devices (i.e. $V_{DD} < |V_{Tp}| + V_{Tn}$), there will be no short circuit current since both devices cannot be active at the same time [2]. However, leakage current will still flow due to sub-threshold effects.

The third component of power consumption is leakage power. Leakage power consumption arises from the fact that MOSFETs are not ideal switches and therefore some current will flow even when the devices are off. There are two components of leakage power consumption, reverse-bias junction leakage and subthreshold conduction.

In a MOSFET the drain/substrate and source/substrate junctions form junction diodes,

which have the following current-voltage relationship:

$$I_D = I_{DSS}\left(e^{\frac{V_D}{kT/q}} - 1 \right) \qquad \text{Eq. 3}$$

If $V_D >> V_T$, this reduces to:

$$I_D \cong I_{DSS} \qquad \text{Eq. 4}$$

Therefore, when a drain or source to substrate pn-junction is reverse biased, current I_{DSS} will flow.

Subthreshold conduction occurs in mosfets when the magnitude of the gate to source voltage is less than the magnitude of the threshold voltage, but greater than or equal to zero. The current-voltage relationship for MOS devices has been shown to be [6].

$$I_D = SI_{DO} e^{\frac{V_G}{nV_{TH}}} \left(e^{-\frac{V_S}{V_{TH}}} - e^{-\frac{V_D}{V_{TH}}} \right) \qquad \text{Eq. 5}$$

Where S is the shape factor (typically W/L), I_{D0} is a process parameter, V_G is the applied gate voltage, n is the slope factor, V_{TH} is the thermal voltage (kT/q), V_S is the source voltage and V_D is the drain voltage, with respect to substrate. However, for normal operation in digital logic circuit (V_G=0, V_S=0, $V_D >> V_{TH}$) this equation reduces to:

$$I_D = SI_{DO} \qquad \text{Eq. 6}$$

Therefore, the total leakage power consumption for CMOS digital circuits is approximately:

$$P_{leakage} = V_{DD} \times (I_{DSS} + SI_{DO}) \qquad \text{Eq. 7}$$

The final component of power consumption is static power dissipation. Static power is lost due to current flow between supply and ground while a logic circuit is not switching. This only occurs in CMOS logic circuits when one of the devices is not fully turned off (i.e. $V_{GS} \neq 0$ V). Normally this will not occur in CMOS digital logic circuits, therefore static power dissipation in CMOS digital logic circuits is virtually non-existent.

The total power consumed in a digital circuit can be calculated as:

$$P = \alpha C_L V^2 f + I_{SC} V + (I_{DSS} + SI_{DO})\, V + I_{STATIC} V \qquad \text{Eq. 8}$$

2.2 Low Power Design Overview

There are two major principles used in low-power digital circuit design voltage scaling and minimizing effective switched capacitance per cycle. Voltage scaling involves examining the delay in the critical path and lowering the supply voltage until the delay in that path matches the period of the clock. Minimizing effective switched capacitance per cycle involves lowering the amount of physical capacitance switched per cycle (i.e. reducing the number of transition per cycle) and designing circuits so they can be clocked as slowly and as infrequently as possible and still maintain functionality.

2.2.1 Voltage Scaling

Voltage scaling is the most effective method of power reduction in CMOS digital circuits. Switching power, which is the major component of power dissipation in CMOS digital circuits, is proportional to the square of supply voltage. Switching power dissipation is a linearly proportional to effective switched capacitance per cycle. Therefore, since reducing the power supply produces a squared effect while reducing the effective switched capacitance produces a linear effect, reducing the supply voltage as much as possible must be the primary goal in low-power digital design.

In CMOS digital circuits, when the supply voltage is reduced the time delay of all gates increases. The increase in gate delay is due to a decrease in the current drive of the transistors. The increase in the propagation time of a digital circuit has been shown to be [2]:

$$t_p \propto \frac{V_{DD}}{(V_{DD} - V_T)^2} \qquad \text{Eq. 9}$$

Therefore, as the supply voltage is reduced gate delay increases as shown in Figure 6 for a minimum sized inverter $\{(W/L)_n = 2.4/1.2\ \mu m$ and $(W/L)_p = 5.4/1.2\ \mu m\}$ simulated using Hspice.

The increase in gate delay as supply voltage is decreased limits the amount of voltage scaling that is possible. The supply voltage can only be reduced until the time delay of the slowest path, the critical path, of the design is slightly less than the period of the clock. Therefore, to obtain the maximum benefit from voltage scaling the delay of the critical path must be minimized and the clock must be as slow as possible.

Minimizing the delay in the critical path involves using the minimum number of logic lev-

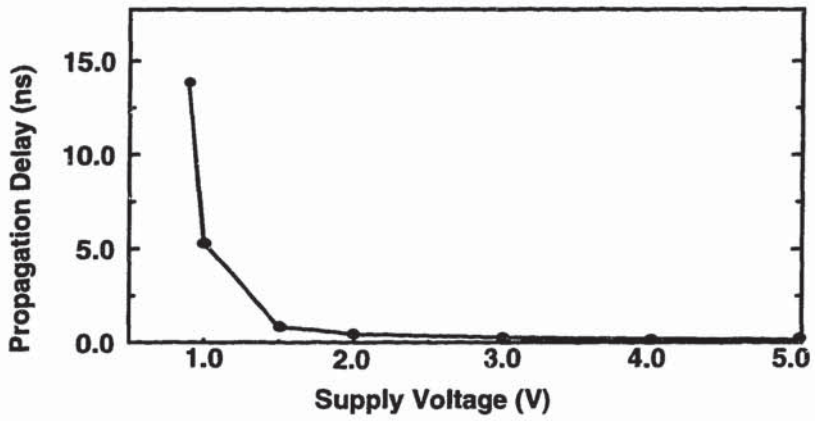

Figure 6: Propagation Delay vs. Supply Voltage for a Single Inverter

els necessary and pipelining. By using the minimum number of logic levels necessary, even at the expense of using more hardware, the number of gate delays between input and output is reduced leading to a shorter critical path. Pipelining can also be used to reduce the effective critical path in systems where throughput rather than actual speed is important (i.e., the system requires an output every cycle, but a delay of a few cycles between input and output is acceptable.) [2], [7]. An example of pipelining a complex path to reduce the critical path is shown in Figure 7. Pipelining can also be used to enable voltage scaling. In systems where throughput is the constraint the delay of the critical path can be reduced below the delay needed to maintain throughtput, the critical rate, then increased back to the critical rate by lowering the supply voltage.

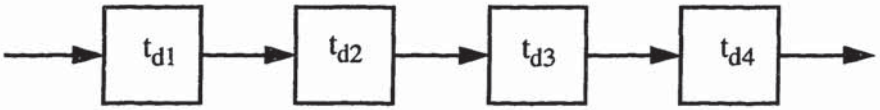

Figure 7a: Non-Pipelined System with $t_d = t_{d1} + t_{d2} + t_{d3} + t_{d4}$

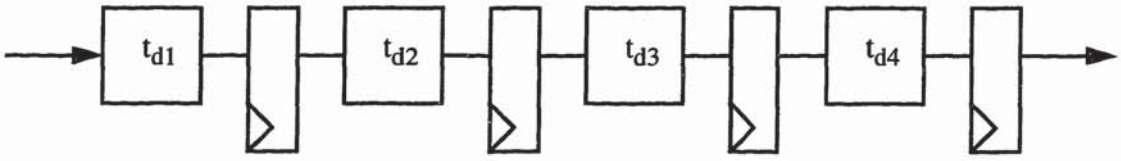

Figure 7b: Pipelined System with $t_d = \max(t_{d1}, t_{d2}, t_{d3}, t_{d4})$

2.2.2 Minimizing Effective Switched Capacitance per Cycle

Minimizing the average amount of capacitance that is switched each cycle, the effective switched capacitance per cycle, is the next method of reducing power consumption in CMOS logic circuits. The switching power dissipated by a CMOS circuit depends linearly on the amount

of effective capacitance switched per cycle.The effective switched capacitance per cycle can be reduced by choice of algorithm and architecture, logic design style, layout style and duty ratio control.

$$P_{switching} = \alpha C_L V^2 f \propto \alpha C_L = C_{eff} \qquad \text{Eq. 10}$$

Proper choice of algorithm and architecture are essential in low-power design. Choosing a proper algorithm involves examining methods of computation and using the method that requires the least number of operations to complete the computation, since this will lead to a minimum amount of switching. When choosing an algorithm, simple operations are favored over complex operations due to the fact complex operations usually require more power and have longer delays. The architecture in low-power digital systems must be designed to minimize the possibility of losing power through unnecessary transitions. The amount of unnecessary transitions can be minimized by powering down systems that are not in use and designing architectures where delays are matched too, as shown in Figure 8. In Figure 8a, the output can go through three values before the output is valid. Using the architecture in Figure 8b, the output will change only once before it is valid.

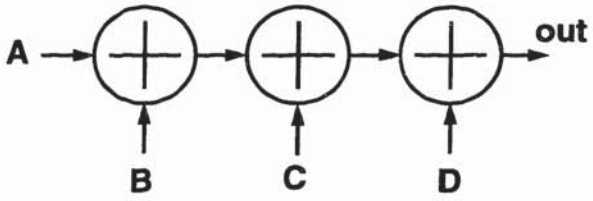

Figure 8a: Chain Architecture for an Addition

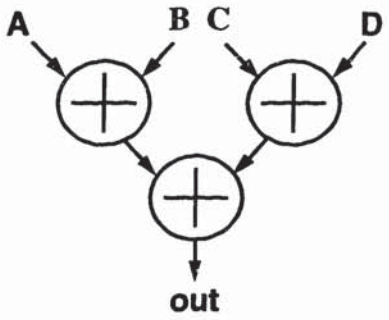

Figure 8b: Tree Architecture for an Addition

Logic design can also be used to reduce the effective amount of capacitance switched per cycle. The basic choice of logic style can be effective in reducing the activity factor of digital circuits. There are many styles of logic to chose from and no clear method of determining which style gives the lowest power. Static CMOS has the advantage of zero static power dissipation, however because of the complementary structure the area and therefore the capacitance are high. Dynamic logic gates tend to have higher activity factors due to pre-charging the circuit every cycle, however due to the smaller size (less capacitance) and higher speed (allowing a greater

voltage scaling) dynamic circuits can use less power. Other logic styles such as pseudo-nmos or DCVSL tend to have disadvantages that outweigh their advantages and are not used commonly in low power design.

The final approach to lower effective switched capacitance per cycle is to lower the effective clock frequency as much as possible. Lowering the effective clock frequency decreases the activity of the circuit and can be used to lower the supply voltage. Switching power consumption is linearly related to the frequency of operation, giving a linear decrease in power consumption with frequency, but decreasing the clock frequency also decreases the throughput. If the frequency is decreased over the critical path, the increased delay can be used to decrease the supply voltage. The primary method of reducing the effective frequency at which a circuit is clocked is through increased parallelism.

Parallelism involves using repeated computational structures clocked at a lower frequency to maintain throughput. A shift register can be used as a good example of a parallel structure. As shown in Figure 9, an n-bit shift register (Figure 9a) can be broken down to two n/2-bit shift registers at half the frequency (Figure 9b). Ideally, the power consumption in a parallel system, with-

Figure 9a: Shift Register without Parallization

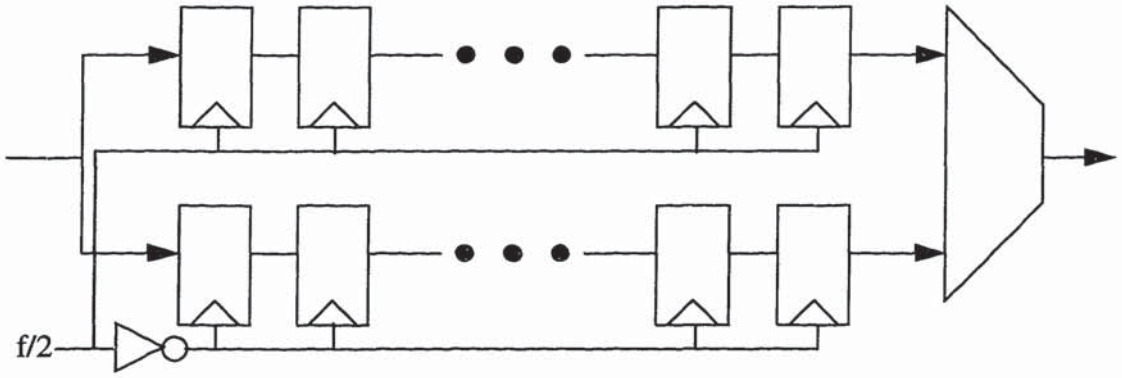

Figure 9b: Parallel Shift Register with n = 2

out voltage scaling, is reduced by 1/n, where n is the degree of parallelism. However, there is an added overhead cost to implement the parallel structure, such as the mux and the hardware to produce the multiple clocks in the shift register case. Due to this overhead, there are diminishing

returns for increasing degrees of parallelism, as shown in Figure 10, where the normalized power consumption vs. degree of parallelism is shown for shift registers of various lengths, using a fixed supply voltage.

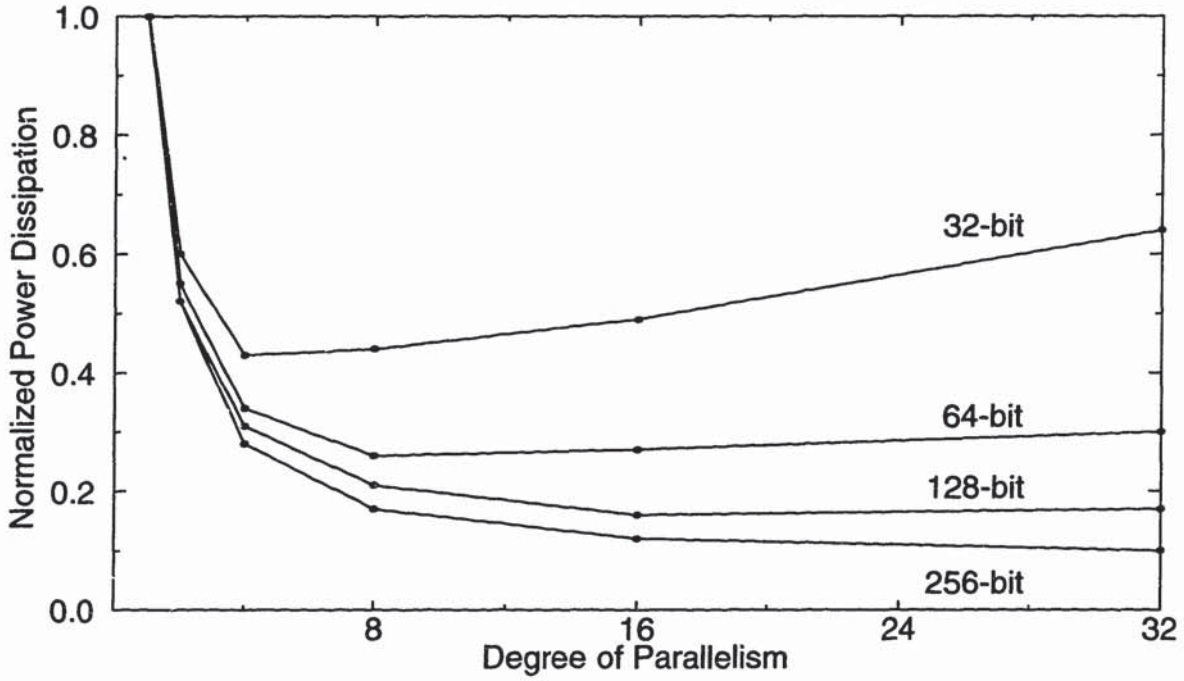

Figure 10: Power Consumption vs. Degree of Parallism for a Shift Register

Parallelism also enables other power reduction techniques, such as voltage scaling. The delay of each cell in the parallel system can be increased n times, where n is the degree of parallelism. If the parallel structure is in the critical path the increased delay can be used to reduce the supply voltage of the system further. If the parallel structure is not in the critical path the increased delay allows the use of slower lower power elements, such as flip-flops with smaller gate areas in the case of the shift register.

2.2.3 Gated Clocks

Another method of reducing the effective switched capacitance per cycle of a circuit is to eliminate the clock signal to that system when the system is not in use. This is commonly referred to as gating the clock. In a gated clock system the power consumption can be expressed as the weighted sum of the power consumption when the system is active and the power consumption when the circuit is inactive (Eq. 11). The weighting factor is the fraction of time the system is active or the duty ratio (d). The power consumption of a system that uses a gated clock can be

expressed as:

$$P_{eff} = P_{inactive}(1-d) + dP_{active} \qquad \text{Eq. 11}$$

Since the supply voltage is constant, the effective switched capacitance is a valid metric for the power consumption of these circuits and is expressed by Eq. 12.

$$C_{eff} = C_{inactive}(1-d) + dC_{active} \qquad \text{Eq. 12}$$

An important result can be derived from looking at Eq. 12. If the effective switched capacitance per cycle while active, C_{active}, satisfies Eq. 13 the power consumption will be dominated by the

$$C_{active} \ll \frac{(1-d)}{d} C_{inactive} \qquad \text{Eq. 13}$$

capacitance switched while inactive. When designing logic circuits this provides interesting results. Assuming the capacitance switched per cycle is two mosfet gates while the circuit is inactive, the minimum for standard CMOS design, if the duty ratio is very high complex circuits can be designed that appear to dissipate the same amount of power a single inverter.

Chapter 3: BodyLAN™ Controller Integrated Circuit Operation

3.0 Introduction

The BodyLAN™ controller custom integrated circuit is responsible for controlling all aspects of the operation of a Personal Electronic Assistant (PEA). This includes controlling the communications link with the Hub, the operation of the microprocessor and the sensor(s).

3.1 System Architecture

The architecture of the BodyLAN™ controller custom integrated circuit is shown in Figure 11. The modem interface is responsible for activating the RF transmitter and giving it data to transmit, activating the RF receiver and routing received data to the proper areas and activating the RF local oscillators and pre-scalers to enable phase locking. The signal processing block is responsible for determining if a Hub is present and receiving the initial TDMA plan. The microcontroller is responsible for generating the control signals for the rest of the PEA once the initial TDMA plan has been received and the local symbol clock is phase-locked to the Hub symbol clock. The microprocessor interface is responsible for gating all signals (including the clock) to the microprocessor and evaluating and acting on the microprocessor outputs. The sensor interface extracts data from the sensor and prepares it to be transmitted as well as gating the clock to the sensor.

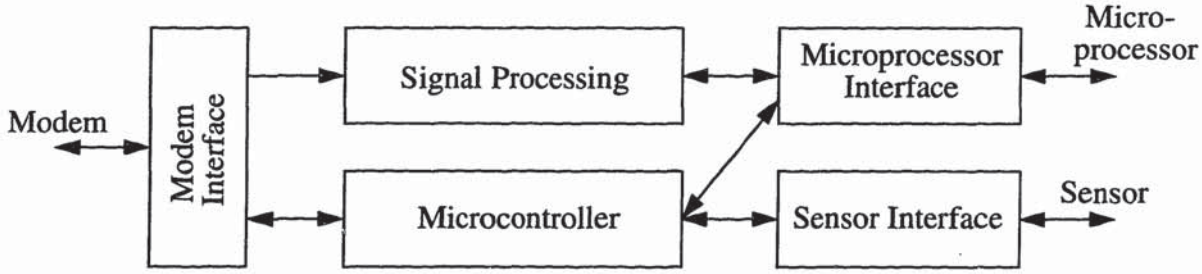

Figure 11: BodyLAN™ Controller Custom Integrated Circuit Architecture

3.2 BodyLAN™ Controller Chip Operation

The BodyLAN™ controller chip has three major modes of operation: attachment, synchronization and TDMA. The control flow for these modes of operation is shown in Figure 12. The controller chip enters the attachment mode when first activated or after Reset. The goal of the

attachment mode is to search for a Hub transmitting valid attachment beacons and receive the initial TDMA plan. After the initial TDMA plan has been received the controller moves to the synchronization mode. In this mode the controller executes the initial TDMA plan and phase-locks the local symbol clock to the Hub symbol clock. The complete TDMA plan is also received during the synchronization mode. Once the local symbol clock is phase-locked to the Hub the controller moves to the TDMA mode of operation. During the TDMA mode the controller executes the complete TDMA plan. The TDMA mode is the "normal" mode of operation for the controller and the controller will only leave the TDMA mode if phase synchronization is lost or the PEA is Reset.

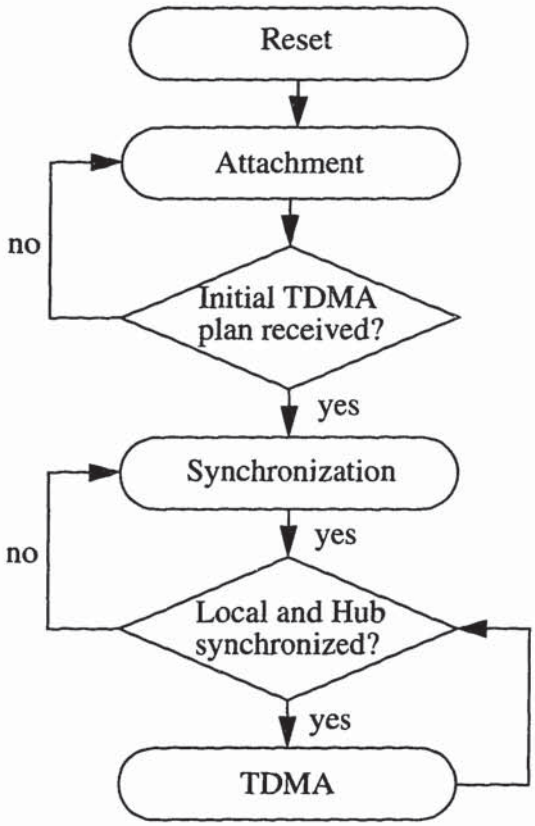

Figure 12: Control Flow of the Controller Chip

3.2.1 Attachment Mode

During the attachment mode the system is searching for a Hub with which to establish a communications link. The existence of a Hub is verified by the detection of attachment beacons. These beacons are transmitted by all Hubs under user command to allow new PEAs to enter the system seamlessly. The attachment beacons and the attachment beacon transmission pattern are

universal for all BodyLAN™ systems, so they can be hard wired into the system.

The system enters the attachment mode when the system is initially powered up or when the system is Reset. The system first begins to periodically search for attachment beacons. The matched filter system is powered on for periods of 8 ms, which is longer than the maximum time between beacons, to search for attachment beacons. If no beacon is found in 8 ms, the system powers down for 8 seconds. If a beacon is discovered the attachment process begins.

Once the first attachment beacon has been discovered the system begins to accumulate data from the beacons. Each beacon transmitted has a single bit of data encoded. This data and the arrival time of the beacon are stored in the microprocessor. After each successful beacon detection the system powers down for the interval between beacons. After the first beacon is detected the system can only shut down for the minimum time between beacons. However, after the second beacon is detected the system can determine, using the inter-arrival times of the beacons, where in the attachment beacon sequence the system is and when the next beacon will arrive, since the inter-arrival times of the beacons are known a priori. The system will continue in the mode until the entire attachment packet has been received.

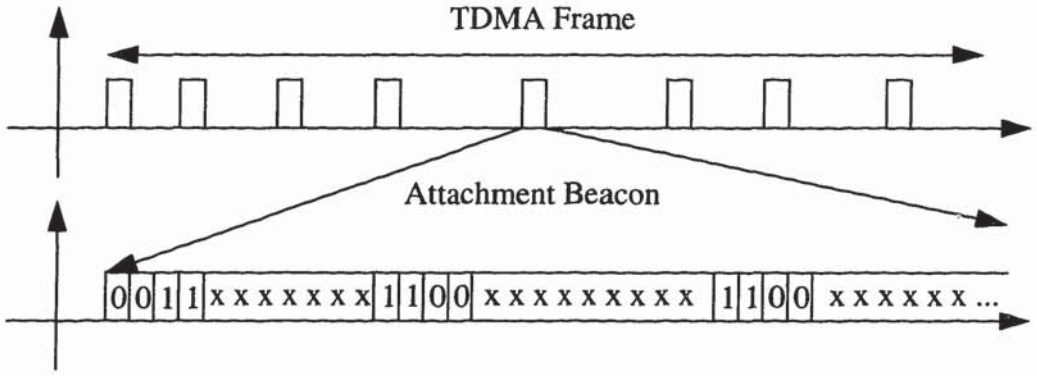

Figure 13: Sample Attachment Beacon Placement within TDMA Frame and a Section of an Attachment Beacon

3.2.2 Synchronization Mode

Once the attachment process has been completed (i.e., the initial TDMA plan has been received) the PEA moves to the synchronization mode. During the synchronization mode the PEA attempts to align the local symbol clock with the Hub symbol clock. The information necessary for this alignment is transmitted to the PEA through synchronization beacons. The PEA controller processes the information supplied by the synchronization beacons and produces phase information used by the microprocessor to adjust the local symbol clock. This allows the use of

multiple Hubs in a limited spatial area.

The synchronization beacon transmitted by the Hub are similar to the attachment beacons. Both beacons consist of 32 bits embedded in a 520 bit sequence. The first 3 four tuples of both consist of a single bit repeated twice and it's complement repeated twice. However, the synchronization beacons are based on the code word transmitted to the PEA during the attachment mode, which is a code specific to that Hub rather than a universal code.

The synchronization system is designed to control the process of phase synchronization. This synchronization is accomplished through the use of an all-digital phase-locked loop. The components of a typical digital phase-locked loop [11] are shown in Figure 14. The phase detector is implemented partially in the PEA controller and partially in the microprocessor. The digital filter consists of the microprocessor and charge pumps. The microprocessor is used as the digital filter because it allows the maximum flexibility in testing and refining the phase-locking algorithm. The local symbol clock oscillator will be a standard low-power CMOS oscillator of the type described in [6]. The oscillator is designed such that the frequency has a tendency to decrease over time, due to leakage from the varactor control capacitor. Therefore, since the oscillator frequency will decrease naturally over time, the system is designed to increase the frequency of the oscillator only.

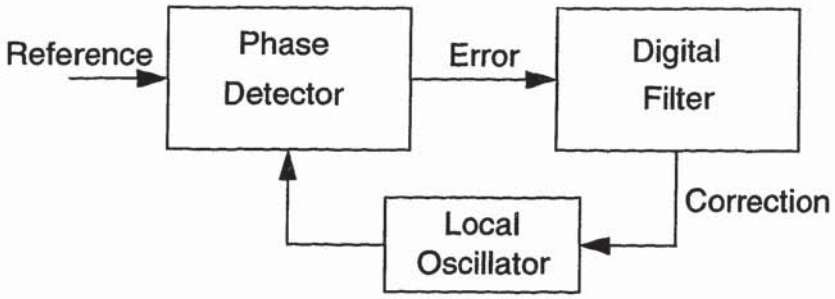

Figure 14: Digital Phase-Locked Loop Architecture

The phase detector has two stages: coarse phase adjustment and fine phase adjustment. During coarse phase adjustment, the PEA controller searches for synchronization beacons. When a synchronization beacon is detected, the arrival time is transmitted to the microprocessor. The microprocessor then computes the inter-arrival time of the current beacon and the previous one. This inter-arrival time is compared with the expected inter-arrival time, which was supplied by the Hub as part of the initial TDMA plan, to determine coarse phase error. If the beacon is late the

local symbol clock frequency will be increased. If the beacon is early, there is no correction, since the local symbol clock frequency will naturally decrease. Once the phase error is reduced below a certain threshold, the system moves to the fine phase adjustment.

During the fine phase adjustment stage the first three four tuples of the synchronization beacons are used to determine phase error. Each of these bit sequences has only one transition, after the second bit. These three 4-bit sequences are first transformed so they are all rising edge sequences (i.e., 0011). The phase error is then determined by sampling the value of the sequence on the third clock edge. If the sampled value is zero, as shown in Figure 15a, the symbol clock frequency is high and no correction is made. However, if the sampled data is a one, as shown in Figure 15b, the symbol clock frequency is too low an needs to be increased. This information is transmitted to the microprocessor, which determines the amount of phase correction necessary.

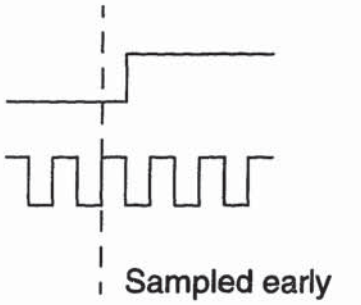

Figure 15a: Synchronization Beacon Sampled Early

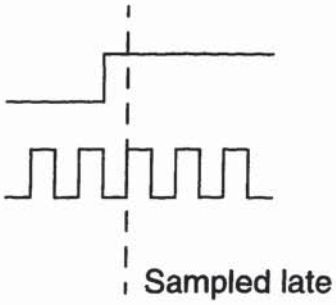

Figure 15b: Synchronization Beacon Sampled Late

The RF transmitter and receiver oscillators must also be phase-locked to the Hub. Once the local symbol clock is phase-locked to the Hub, the RF transmitter and receiver oscillators can be phase-locked to a multiple of the local symbol clock since the Hub transmitter and receiver oscillators are locked to a multiple of the Hub symbol clock. During this process the PEA controller chip acts as a controller for an analog phase detector. The analog phase detector is of the form shown in Figure 16. The comparator used in the analog phase detector does not have extremely

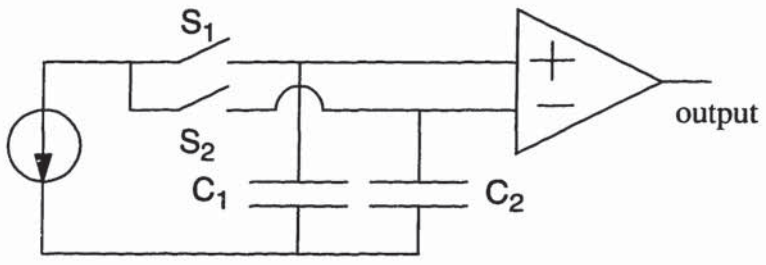

Figure 16: Analog Phase Detector

high gain, therefore an output near zero is possible for an extremely small difference in the inputs. During this stage, both the oscillator-under-test, either the RF receiver or transmitter oscillator, and the local symbol clock are used as clocks on separate counters. These counters are designed to be active for the same amount of time. For example, with 4 Mhz reference and a 6 Mhz oscillator-under-test, the local symbol clock counter would count to 40 and the oscillator-under-test counter would count to 60, so both would be active for 10 μs. The counter connected to the oscillator-under-test is started a fixed amount of time earlier than the local symbol clock counter. During the period when the oscillator-under-test is on and the local symbol clock counter is off, S_1 is closed, charging C_1. Once the oscillator-under-test counter signals completion S_2 closes, charging C_2. The switch S_2 opens when the local symbol clock counter signals completion. Ideally, if both counters are active for the same amount of time, the output will be near zero. If the oscillator-under-test frequency is high, S_2 will be closed for less time than switch 1, causing the voltage across C_1 to be greater than that across C_2 and the output to be high. If the oscillator under-test-frequency is low, S_1 will be closed for longer than S_2, causing the voltage across C_2 to be greater than the voltage across C_1 and the output to be low. Therefore, a high output indicates the oscillator-under-test frequency needs to be increased, while a low output indicates no change.

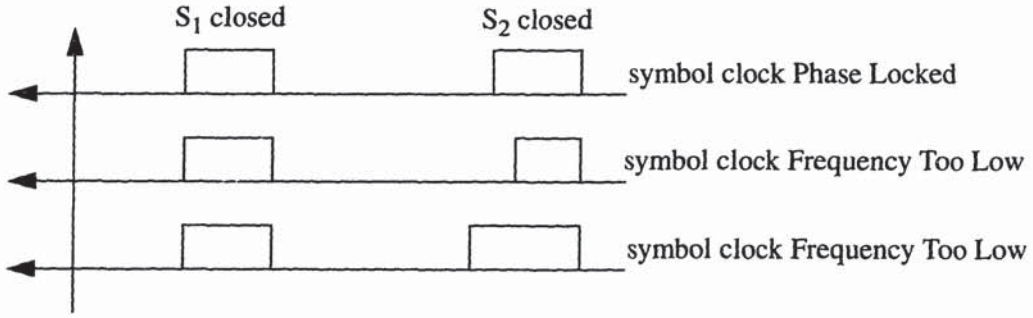

Figure 17: Analog Phase Detector Control Signals for Different Cases

The microprocessor, using an algorithm developed at Bolt, Beranek and Newman, gathers all the phase error information from the three phase detection modes and determines the amount of correction necessary. The amount of correction needed is given as the number of times to pulse a charge pump adding charge to the symbol clock's control varactor.

3.2.3 Time Division Multiple Access Mode

The TDMA mode is the "normal" mode of operation for the system. During the TDMA

mode the PEA and the Hub have an established communications link and are both transmitting and receiving regularly according to the TDMA plan stored in the memory. The TDMA plan is implemented by the microcontroller shown in Figure 18 and described in detail in Chapter 4. The system will remain in the TDMA mode until the system is Reset or loses the phase-lock with the Hub. If the system is reset it moves to the attachment mode. If the system loses phase-lock, it moves to the synchronization mode.

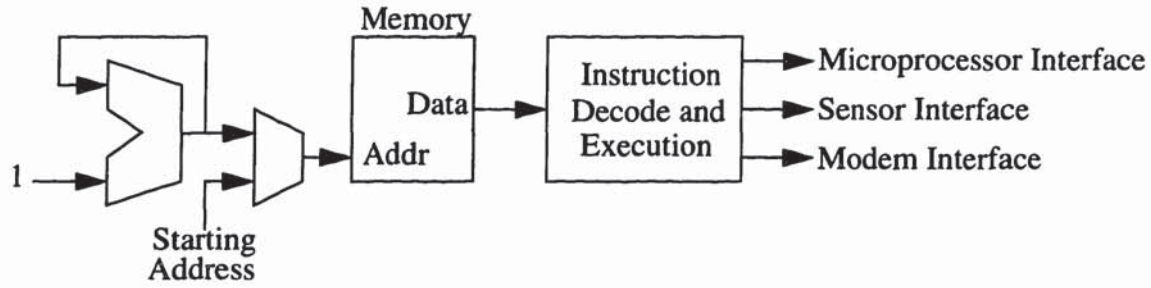

Figure 18: BodyLAN™ Microcontroller Architecture

Chapter 4: Low-Power Microcontroller Design

4.0 Introduction

The BodyLAN™ system needs to operate at an extremely low power level to be considered a reasonable product. In order to operate the system at an ultra-low power level the control circuit must be able to take advantage of the inherent duty ratio of the TDMA protocol. Ideally, while the communications channel is inactive the controller chip should have zero power dissipation, since the controller is not doing any work for the system. However, a certain amount of energy must be expended continuously to maintain timing and synchronization. Minimizing the energy that is expended while the system is inactive is the challenge.

While the system is operating in the TDMA mode the controller chip acts as a simple microcontroller, reading instructions (which are the TDMA plan entries) from an on-chip SRAM. Unlike a "normal" microcontroller the instructions are not executed as fast as possible, but are executed at a time set by the TDMA plan. Since the instructions are executed only a set times, there are gaps between instruction execution where the system can be in a low-power state, which is typically large in the BodyLAN™ system. The problem becomes how does one implement a microcontroller in such a way that the maximum benefit of the duty ratio is achieved. In this chapter a design style which uses a small centralized core, which implements all the functions that need to be performed each cycle (such as instruction decoding) and a large number of digital sequencers which are triggered only when needed to implement the specialized functionality of the individual instructions.

4.1 Microcontroller Core

The design of the microcontroller core involves choosing which functions should be implemented in the core and which functions should be implemented outside the core. By implementing all possible functions outside the core, the amount of power used in the core, which is running continuously, will be minimized. However, if the maximum amount of functionality is outside the core the amount of routing, especially the instruction routing, is maximized. If more functionality is moved to the microcontroller core, the core uses more power, but the routing of the control signals can be significantly reduced lowering the overall power consumption.

With these factors in mind, it was decided to do all instruction decoding in the microcontroller core so the instruction bus needs to be routed to only one place and move as much of the rest of the functionality as possible outside the core. All of the instruction decoding must be computed every cycle whether or not the decoding is in the microcontroller core so moving all of that functionality to the core is reasonable. All other functions are dependant on the opcode, so moving those functions to outside the core to blocks which will only use power when triggered is also reasonable.

4.2 Digital Sequencers

Digital sequencers have been chosen to implement the functionality outside of the core. A digital sequencer is a circuit that will supply a series of signals at the proper time based on an asynchronous activation signal. For example, in a system where the memory latency is 10 cycles, a digital sequencer can be used to gate the address to the memory and the data from the memory at the proper times. Digital sequencers represent a method of standardizing the generation of signals that previously would have been produced using "random" logic. By standardizing the production of these signals a low power method of generating the signals may be determined. With that goal in mind a low power digital sequencer architecture was developed. This architecture was tested over a range of parameters to determine the lowest power implementation of the architecture.

All the digital sequencers investigated in this chapter have the same basic architecture. There is an activation signal, a completion signal, a counter and a decoder, as shown in Figure 19. The activation signal is produced by an outside source and the completion signal is produced by the decoder. Once the activation signal is given, the digital sequencer will be active until the self-generated completion signal. By using a gated clock the power dissipated in the sequencer is min-

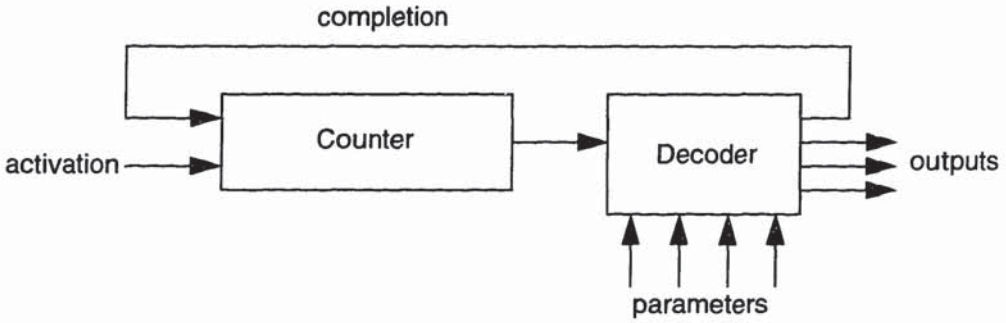

Figure 19: Architecture for all Digital Sequencers Investigated

imized while the sequencer is inactive. Each of the digital sequencer styles uses a different type of counter, which defines the type of digital sequencer. The decoding hardware is responsible for combining the counter output and the input parameters to produce the necessary outputs at the correct time.

It is important to realize that the primary goal of low-power digital sequencer design is to reduce the effective switched capacitance per cycle. The effective switched capacitance per cycle includes the capacitance switched per cycle while the digital sequencer is both active and inactive. The effective switched capacitance for any clocked circuit with a duty ratio can be calculated as:

$$C_{eff} = C_{inactive}(1 - d) + dC_{active} \qquad \text{Eq. 14}$$

Where $C_{inactive}$ is the capacitance switched per cycle while inactive, d is the duty ratio of the circuit and C_{active} is the capacitance switched per cycle of the circuit when active. In systems with small duty ratios, such as the systems on the BodyLAN™ controller chip. the effective capacitance can be dominated by the capacitance switched while the circuit is inactive. Therefore, the capacitance switched while the circuit is inactive needs to be minimized. An architecture, shown in Figure 20, that has only two gate capacitances switching every cycle, the minimum using standard CMOS design, is used in the digital sequencers for the BodyLAN™ controller integrated circuit.Using this architecture, the gated clock is high while the circuit is inactive. When the circuit is activated the first positive edge of the gated-clock will be synchronized with the system clock (clock).

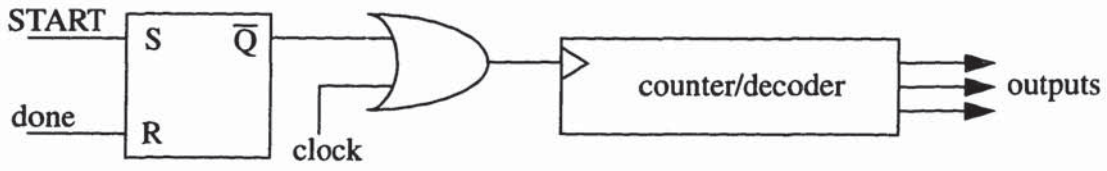

Figure 20: Digital Sequencer Architecture to Minimize Capacitance Switched While Inactive

Virtually all of the power dissipation in a digital sequencer will be depend on the counter, since the counter originates all the transitions in the sequencer. The counter power dissipation includes the power necessary to maintain the count as well as the power needed to charge the

inputs to the decoder. For digital sequencers with a large number of outputs the power needed to charge the inputs to the decoder will be critical. Therefore, designing a counter where the outputs have a low activity factor is important. Four types of counter structures: delay-line, asynchronous, synchronous and pseudo-grey, were tested to determine the lowest power implementation.

4.2.1 Delay-Line Digital Sequencer

The delay-line digital sequencer is based on using a tapped delay line to produce the correct output at the correct time. The tapped delay line is easy to design, a simple shift register, as shown in Figure 21, will suffice and is simple to decode, just connect to the taps that are needed. Using a tapped delay-line a large amount of hardware is necessary to implement a counter, 2^N register cells for an N-bit counter.

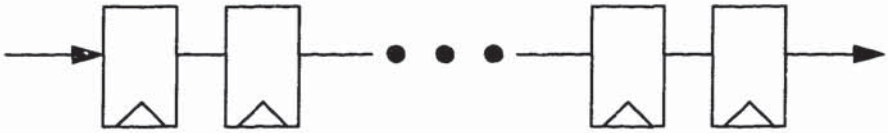

Figure 21: Delay-Line Counter

4.2.2 Asynchronous Digital Sequencer

The asynchronous digital sequencer style is based on an asynchronous or "ripple" counter. The basic asynchronous counter cell is shown in Figure 22. An asynchronous counter uses the minimum amount of hardware to complete the task of counting, a single register cell. The delay in the switching times of the counter bits is the main disadvantage of using an asynchronous counter. The delay in the switching times of the counter bits causes excess transitions in the decoder and raises the activity factor (α) of the decoder.

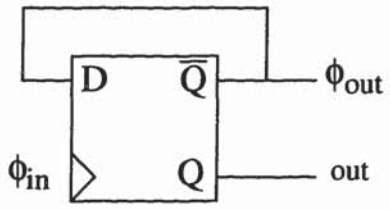

Figure 22: Asynchronous Counter Cell

4.2.3 Synchronous Digital Sequencer

The third design style is the synchronous digital sequencer style. The synchronous style

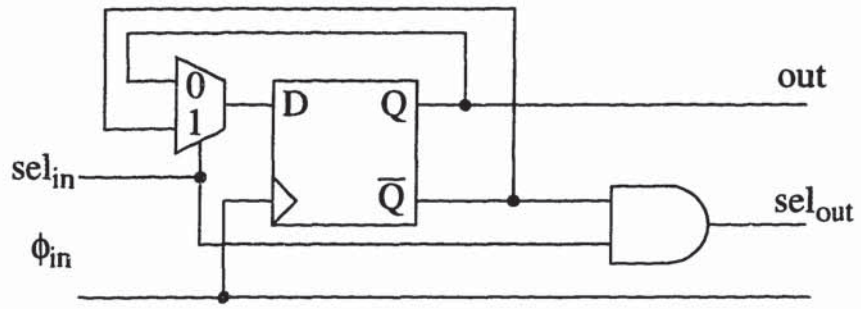

Figure 23: Synchronous Counter Cell

uses the counter cell shown in Figure 23 to implement a synchronous counter. Since the counter is synchronous, designing a glitch-free decoder is now much easier. However, the power consumed by the extra hardware necessary for the synchronous counter can be more costly than the glitching transitions.

4.2.4 Pseudo-Grey Code Digital Sequencer

The pseudo-grey code digital sequencer style is based on pair-wise grey code counter. In a pair-wise grey code counter each pair of bits (01, 23, etc.) counts in grey code. The first pair increments every cycle and all subsequent pairs increment on the cycle after all the lower order pairs are in the 11 state. Table 2 shows a count in decimal, grey code, and pseudo-grey code notation.

Decimal	Grey	Pseudo-Grey	Decimal	Grey Code	Pseudo-Grey
0	0000	0000	8	1100	1100
1	0001	0001	9	1101	1101
2	0011	0011	10	1111	1111
3	0010	0110	11	1110	1010
4	0110	0100	12	1010	1000
5	0111	0101	13	1011	1001
6	0101	0111	14	1001	1011
7	0100	1110	15	1000	0010

Table 2: Decimal, Grey Code and Pseudo-Grey Code

Using pseudo-grey code the activity factor of the counter is reduced, with a minimal increase in counter cell complexity (the counter cell is shown in Figure 24). The activity factor for a synchro-

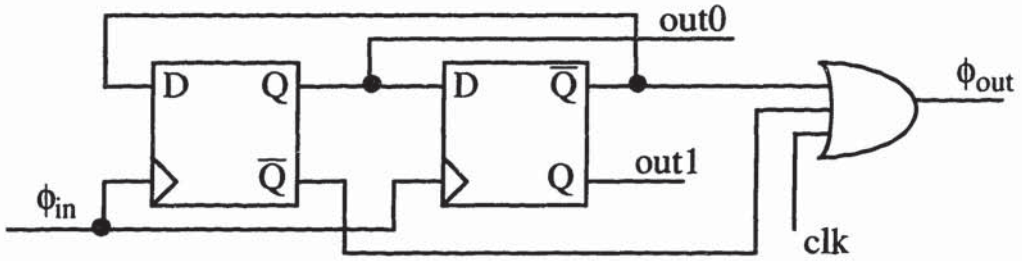

Figure 24: Pseudo-Grey Code Counter Cell

nous counter is 2 [2].The activity factor for a pseudo-grey counter can be calculated by realizing the counter is pair-wise grey-code, therefore, only one bit of each pair will change during a given cycle. The activity factor can then be expressed as follows.

$$\alpha = P_{01} + P_{23} + P_{45} + P_{67} \rightarrow P_{\infty} = \sum_{n=0}^{\infty} \left(\frac{1}{4}\right)^n = \frac{4}{3} \qquad \text{Eq. 15}$$

Which translates to a counter which only uses 2/3 of the power of a standard counter. Regular Grey-Code, which has an activity factor of one, was not used since a Grey-Code counter cell was readily available.

If the power consumption were based strictly on the style of counter, .the asynchronous design style would be preferred since the asynchronous counter consumes the least amount of power in most cases, as shown in Figure 25, which shows the power dissipation for the four counter styles for various counter lengths normalized to the power dissipation of the pseudo-grey code counter for that length, as measured using Irsim-cap from actual layout. With no load the asynchronous counter dissipates the least power since the asynchronous counter cell has only two

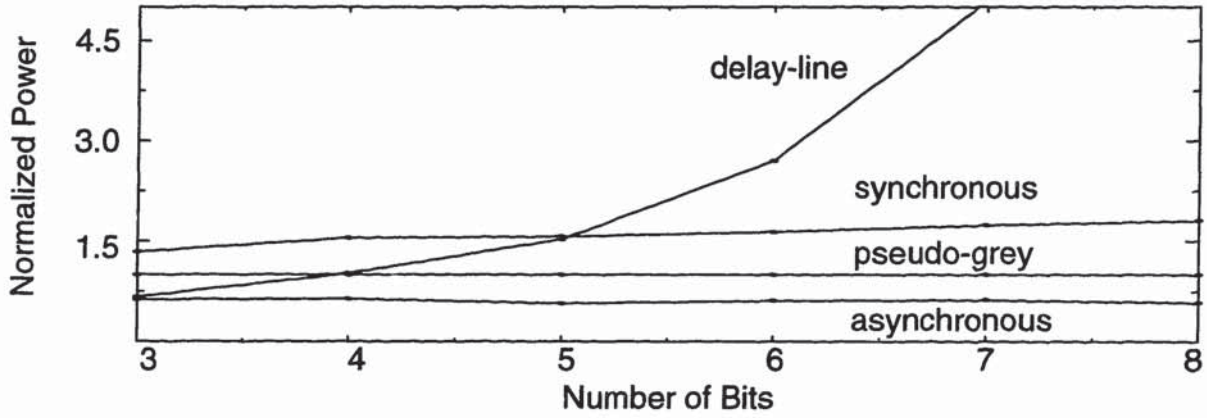

Figure 25: Power Consumption of Counter Normalized to Pseudo-Grey Counter

register cells. The synchronous and pseudo-grey counter cells have approximately the same number of elements, however due to the lower activity factor of the pseudo-grey counter it dissipates only about 2/3 of the power of the synchronous counter. The power dissipated by the delay-line counter is low for small counters however it increases quickly due to the fact a delay-line counter needs 2^n register cells for an n-bit counter, whereas the other counter styles need only a single n register cells for an n-bit counter. However, the power consumed by a digital sequencer in these styles depends not only on the counter, but on the decoding of the signals.

When the decoders and the counters are joined to form a complete digital sequencer, the power dissipation depends on the degree of loading or the number of outputs needed. Each digital sequencer was tested at full load for counter lengths from three to eight. Additionally, three examples of each style of digital sequencer were tested at 3/4, 1/2, and 1/4 loads for counter length from three to eight. The digital sequencers were produced using a C program to generate lager .sdl files and the lager automatic place and route tool to produce the layout. The power dissipation of each of the sequencer styles, normalized to the power dissipation of the pseudo-grey sequencer is shown in Figure 26, for the four degrees of loading. Due to the fact an automatic place and route tool was used to produce the layout, the graphs are not monotonic, however they do provide insight into the general behavior.These figures show that the best choice of digital sequencer style depends on the degree of loading. The pseudo-grey code sequencer style dissipates less power with either a large number of counter bits needed or a high degree of loading. However, this advantage decreases as the number of counter bits and loading decrease.

4.3 BodyLAN™ Microcontroller

The BodyLAN™ microcontroller is used to control the communications during the synchronization and TDMA modes of operation. The instruction set of the microcontroller consists of all possible communications instances (TDMA plan entries). The microcontroller is broken down into a core and a series of digital sequencers. The core is responsible for decoding the microcontroller instructions and triggering the correct digital sequencer(s). There are four digital sequencers (burst, synchronization, duration controller, SRAM controller) which handle all the details of implementing the instructions.

4.3.1 Instruction Set

The BodyLAN™ microcontroller instruction set covers all instructions needed to handle

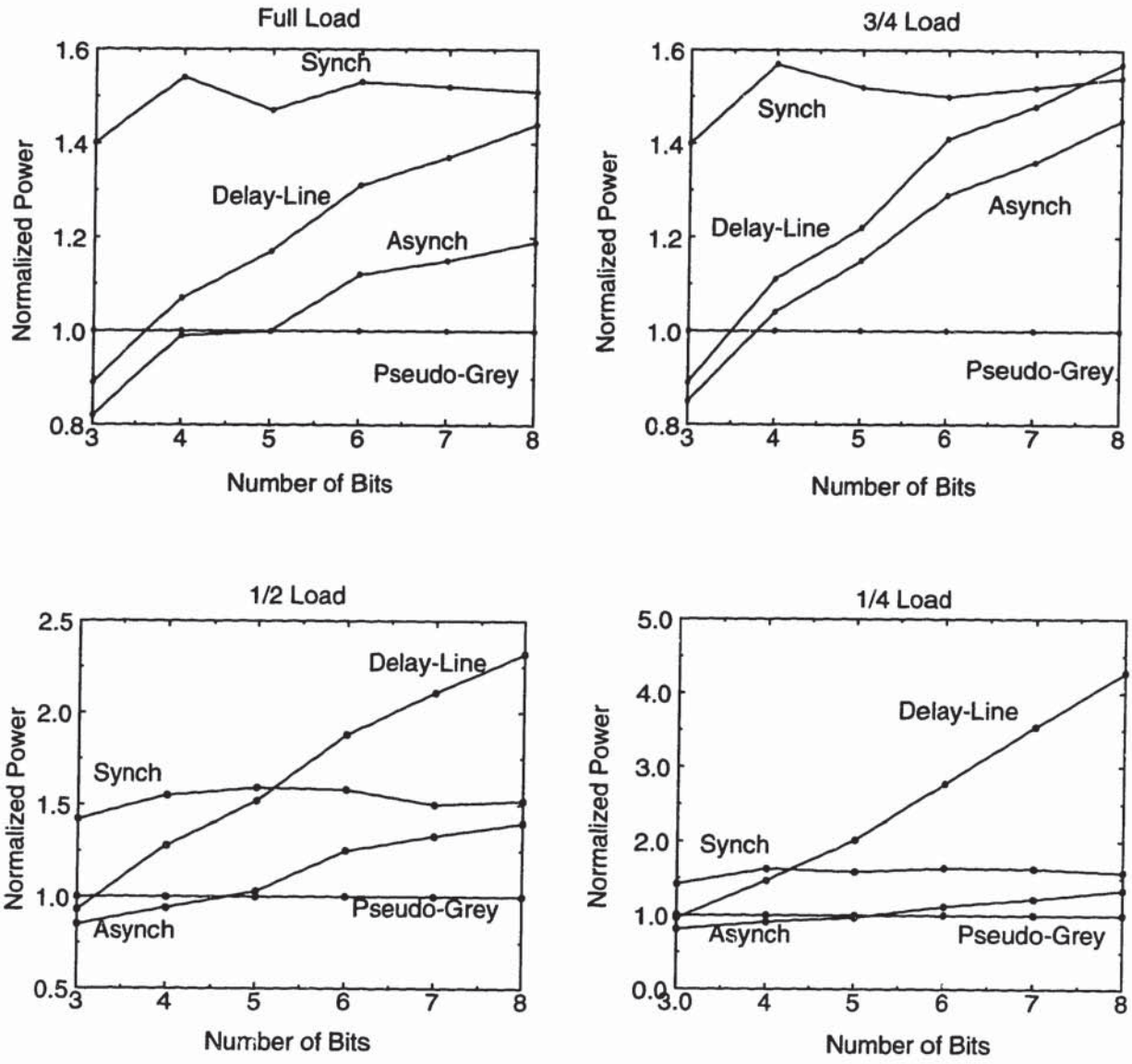

Figure 26: Digital Sequencer Power Dissipation vs. Degree of Loading

the TDMA plan as well as maintain phase-lock and interface with the microprocessor. Each instruction contains a 4-bit operation code, a 14-bit execution time and two 6-bit subinterval times. The operation code specifies what communications event will occur next. The execution time indicates what time, within the frame, the instruction execution should begin. Most of the communications events occur three times, due to the nature of the communications algorithm. The sub-interval times indicate the time delay between the first and second and the second and third executions of the instruction. The complete instruction set is shown in Table 3. The first four instructions (00xx) are all synchronization beacon processing instructions. The next eight instructions (01xx and 10xx) are all Hub to PEA and PEA to Hub communication instructions. The final

four instruction opcodes (11xx) all indicate an unconditional microprocessor wake-up.

Table 3: Microcontroller Instruction Set

Opcode	Instruction
0000	First Synchronization Beacon of the Frame
0001	Synchronization Beacon
0010	Synchronization Beacon with Rx symbol clock Adjustment
0011	Synchronization Beacon with Rx symbol clock Adjustment
0100	Hub to PEA microprocessor information block
0101	PEA to Hub microprocessor information block
0110	Hub to PEA user information block
0111	PEA to Hub user information block
1000	Hub to PEA CCC opcode
1001	PEA to Hub status return
1010	Hub to PEA CCC opcode extension
1011	Hub to PEA or PEA to Hub Command and Control Channel data transfer
11xx	Unconditional microprocessor wake-up

4.3.2 Microcontroller Core

The core for the BodyLAN™ microcontroller is responsible for decoding the instruction opcodes and triggering the digital sequencers at the proper time. Due to the repeated nature of the instructions the microcontroller core has been broken down into three subsections, as shown in Figure 27. The instruction is decoded and the first execution is triggered in phase 1. In phase 2 the second and third executions are triggered. Any operations necessary for completing the instruction are executed in phase 3.

The phase 1 section of the microprocessor core is responsible for decoding the instruction opcode and triggering the first execution of the instruction. The phase 1 section must also determine if the phase 2 and phase 3 sections of the core need to be executed. The structure of the phase 1 section is shown in Figure 28. The START signal is generated by the counter chain (the universal time reference for the chip) when the slot counter matches the execution time given in the instruction. The phase 1 section then begins to activate the digital sequencers necessary to

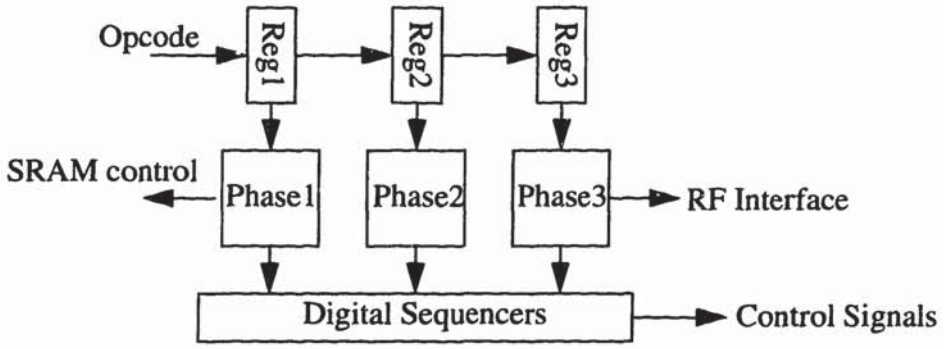

Figure 27: Microcontroller Core

complete the instruction. The phase 1 section of the microcontroller core is responsible for handling all of the instruction decoding because eliminates the need to route an instruction bus with a high activity factor to many locations. The high activity instruction bus is replaced by triggering signals that have a lower activity factor. By performing the translation to a lower activity bus in the first stage the maximum power savings is achieved.

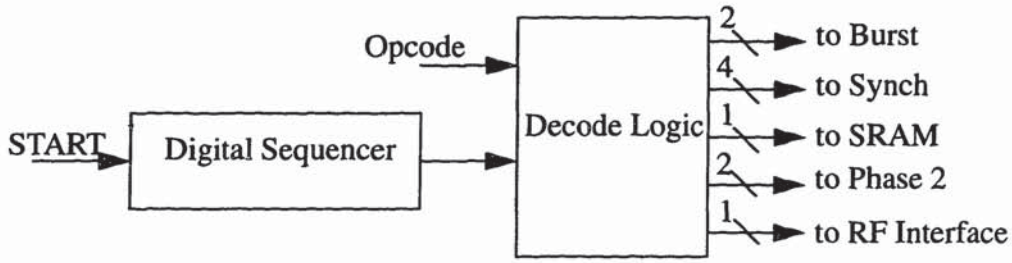

Figure 28: Microcontroller Phase 1

The phase 2 section of the microprocessor core is responsible for triggering the second and third executions of the instruction, if needed. The structure of the phase 2 section is shown in Figure 29. The instruction subintervals are loaded into registers 1 and 2. After phase 1 is com-

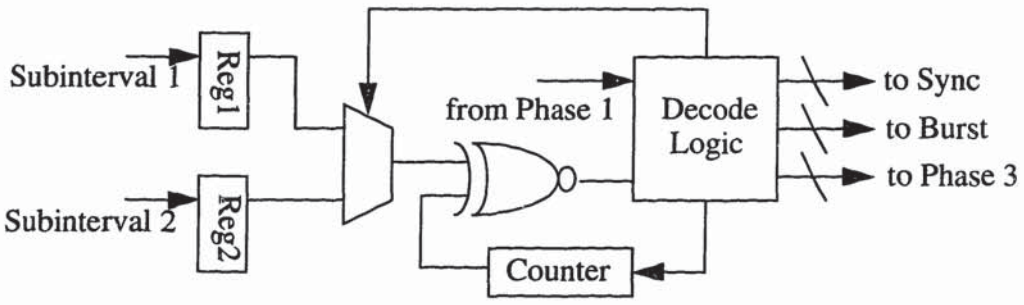

Figure 29: Microcontroller Core Phase 2

plete, the counter begins incrementing. When the counter bits all match the value in register 1, the instruction is executed for the second time and the counter is reset. After the counter bits match the value in register 2, the instruction is executed for the third time and phase 3 is triggered.

The phase 3 section of the microprocessor core is responsible for finishing the instruction. Finishing the instruction involves latching received data to the proper register, sending data to the sensor, or waking up the microprocessor. The structure of the phase 3 section is shown in Figure 30. All of the data routing instructions are handled by the RF interface. The microprocessor wake-up section is used to access the microprocessor.

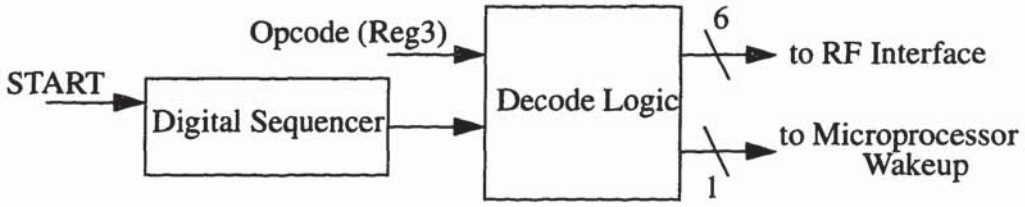

Figure 30: Microcontroller Core Phase 3

4.3.3 Burst Execution Hardware

The burst execution hardware is responsible for controlling the RF transmission and reception of data. The architecture of the burst execution hardware is shown in Figure 31When triggered the burst execution hardware powers up the appropriate RF system (transmit or receive). Once the proper RF system is initialized, the data is either clocked into (receive) or out of (transmit) the RF interface using the RF_data_clk signal.

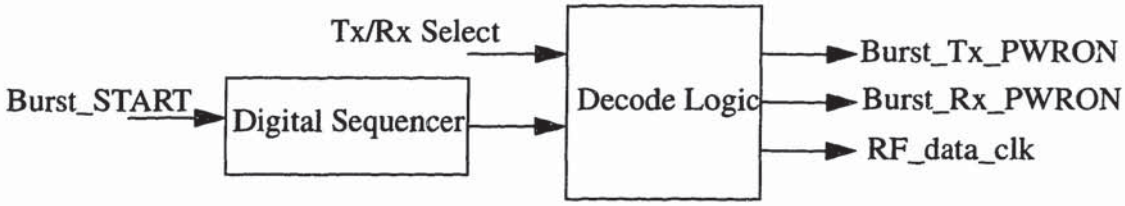

Figure 31: Burst Execution Hardware

4.3.4 Synchronization Beacon Processing Hardware

The synchronization beacon processing hardware is used to extract phase information from synchronization beacons. The architecture of the synchronization beacon processing system is shown in Figure 32. Two digital sequencers are used. The first sequencer is used for the case when two synchronization beacons arrive back-to-back (i.e. eight straight bits). The second digital

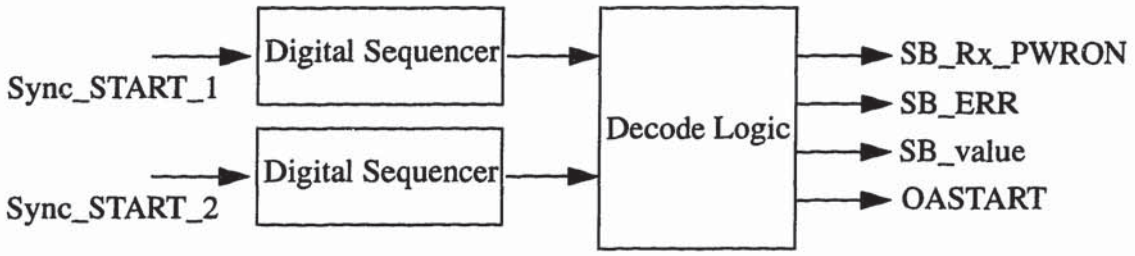

Figure 32: Synchronization Beacon Processing Hardware

sequencer is used when there is only one synchronization beacon. Once triggered the synchronization beacon hardware powers up the RF receiver and checks the synchronization beacons for phase information (described in detail in Chapter 6). The SB_value is the phase information extracted. If the synchronization beacon is not received properly the SB_ERR signal is given. The OASTART signal is used to trigger the fine phase detection hardware (the duration controller hardware).

4.3.5 Duration Comparator Hardware

The duration comparator hardware controls the analog phase detector used to phase-lock the RF receiver and transmitter oscillators to the local symbol clock. The control signals needed for the RF phase detection were defined in Section 3.2.2. The architecture of the duration controller system is shown in Figure 33. The digital sequencer initializes the oscillator-under-test and it is fed to a counter. The digital sequencer then triggers the switch S_1 in the analog phase detector to be closed. After a short period the reference symbol clock is fed to the second counter and the switch S_1 is opened. When the value of the first counter matches the value stored in the register

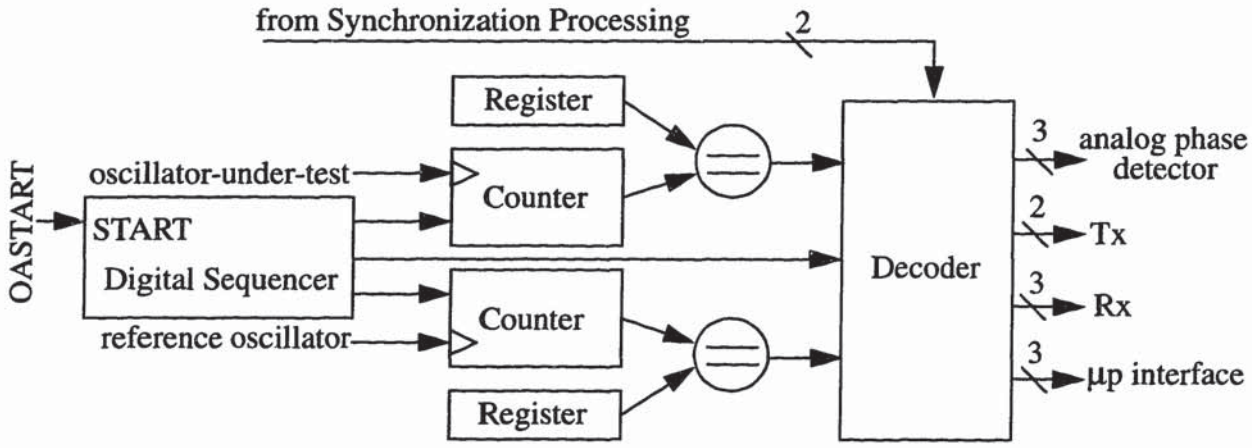

Figure 33: Duration Controller Architecture

the switch S_2 is closed. When the value of the second counter matches the value stored in its register the switch S_2 is opened and the phase error can be determined.

The duration comparator has eleven outputs, three each to the analog phase detector, Rx section (receiver) and the microprocessor interface and two to the Tx section (transmitter). The outputs to the analog phase detector enable the detector, control switch S_1 and switch S_2. The outputs to the Rx section enable the Rx symbol clock, the Rx oscillator and the Rx pre-scaler. The outputs to the microprocessor interface include the phase error detected and two error signals, one indicates the oscillator-under-test was not present and the second indicates the phase error of the oscillator is too large to use the fine phase detection and the synchronization must move back to coarse phase detection.

Chapter 5: Low Power Matched Filter Design

5.0 Introduction

The signal processing subsystem contains the hardware necessary for detecting attachment and synchronization beacons, setting up the initial TDMA plan and beginning frequency synchronization. Attachment beacons are transmitted by the Hub at the user's command and provide information necessary for initializing the TDMA plan. The data extracted from the attachment beacons is transmitted to the microprocessor which develops the TDMA plan.

5.1 Low Power Matched Filters

A matched filter is a digital filter designed to locate a binary code embedded in a bit stream. The basic architecture of a matched filter is shown in Figure 34, consists of a delay line and a comparison circuit. The delay line is responsible for selecting the proper bits from the incoming data stream. The comparison circuit checks those bits against a code to determine if there is a match. Usually in real systems the comparison circuit allows for some errors.

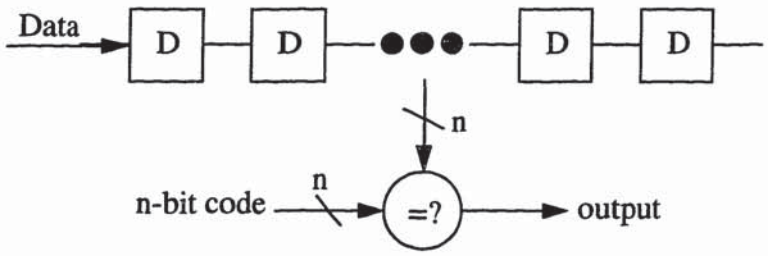

Figure 34: Basic Matched Filter Architecture

5.1.1 Algorithm Design

Designing a low power matched filter involves examining the algorithm, architecture, logical design and physical layout of a matched filter. The filter algorithm is fairly simple, it consists of selecting n-bits out of an m-bit stream and comparing them to a given code word. The major computational task involves processing the outputs of the bit-wise comparison into a single output in a single cycle (assuming real-time operation is desired).

The bit-wise output of the comparison must be processed to determine if a beacon has

been detected. If there are no errors allowed, this process involves ANDing all of the bits to determine if all the bits are ones. If errors are allowed, the simplest method of processing the bit-wise output of the comparison is to simply sum the outputs using an adder tree structure as shown in Figure 35 and determine if enough of the bits matched. The number of adders needed is shown in

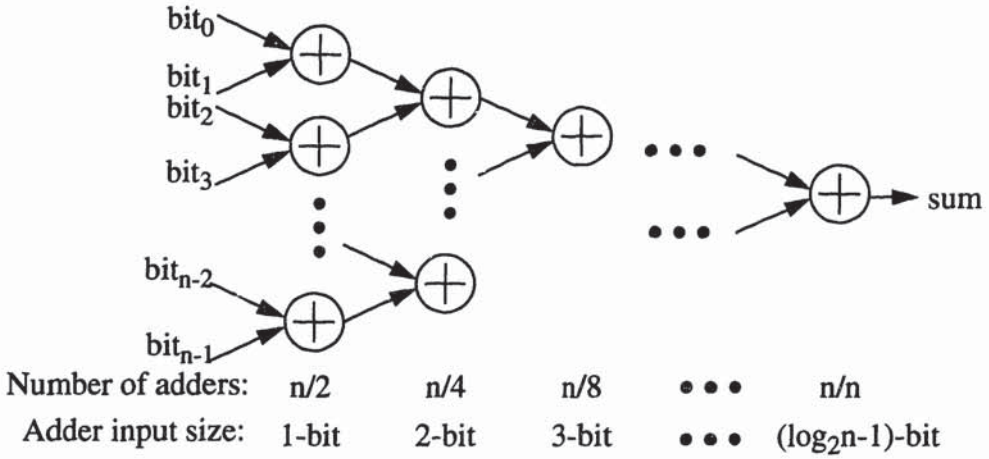

Figure 35: Tree Adder Structure

Eq. 16 below.

$$n \sum_{i=1}^{\log_2 n} \left(\frac{1}{2}\right)^i \qquad \text{Eq. 16}$$

Assuming the use of ripple adders, the delay through the entire adder tree should be equal to $\log_2 n + 1$ times the delay through a single adder cell. Since the delay is of order log n, it should be fairly good for voltage scaling.

The other solutions to the comparison problem use combinational logic or lookup tables. Using combinational logic works well if the output desired is a simple indication of whether or not all the bits matched. If the system allows for errors the number of cases increases and the combinational logic approach becomes unfeasible, as shown in Table 4 for a 32-bit system. Using

number of errors allowed	combinations
0	1
1	33
2	529
3	5489

Table 4: Number of Combinations Needed for a 32-bit System

number of errors allowed	combinations
4	41449

Table 4: Number of Combinations Needed for a 32-bit System

a lookup table to produce the correct output has the same problem, the number of table entries needed increases rapidly as the number of allowable errors increases.

The final solution involves using a hybrid approach merging the combinational logic approach and the tree adder approach. The combinational logic approach works well for low order results while the tree adder approach works well for high order systems, so a combination of approaches is reasonable. The first two levels of adders in the tree structure can be replaced with a combinational logic coder that combines four of the inputs and gives a three bit output. The rest of the tree has the same structure, as shown in Figure 36 The number of adders needed for the hybrid

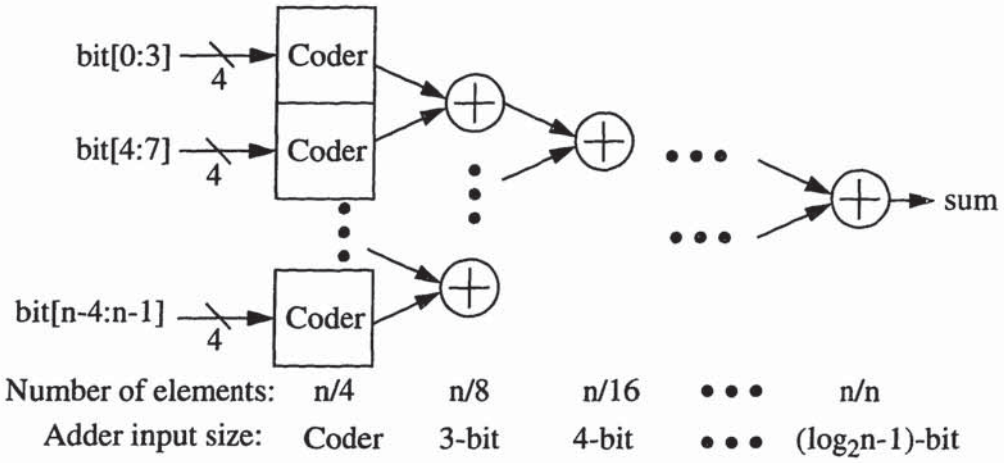

Figure 36: Hybrid Comparison Algorithm

approach can be determined from Eq. 17.

$$n \sum_{i=3}^{\log_2 n} \left(\frac{1}{2}\right)^i \qquad \text{Eq. 17}$$

The hybrid approach uses fewer adders than the full tree structure which results in a savings in area and interconnect capacitance. If the coders are implemented in such a way that the delay is equivalent to the delay through a 3-bit adder, there are no voltage scaling problems, since the delay through an adder tree is equal to the delay of a single adder with width one bit larger than the final adder tree level.

Both the hybrid and the full adder tree structure were simulated from layouts produced using the Lager (ref?) toolset and simulated for 100 random inputs using Irsim-cap. The power consumption of the tree structure was found to be slightly less than the power consumption of the hybrid-structure for all number of inputs tested.

5.1.2 Matched Filter Architecture

The design of the matched filter architecture involves examining the structure of the system and determining how power consumption can be reduced. An approach is presented which uses parallelism to decrease the effective capacitance switched per cycle of the delay line. Complications arise, however, due to the need to have fixed taps on the delay line since each cell in a parallelized delay line represents a different bit for each clock phase. For example, consider the small parallel shift register shown in Figure 37. During one phase of the clock the value in reg A is d(n-2) and the value in reg B is d(n-3). During the other phase of the clock, the value in reg A is d(n-3) and the value in reg B is d(n-2). Therefore, if the filter taps are to represent a fixed delay, the filter taps must be able to adjust for the phase of the clocks.

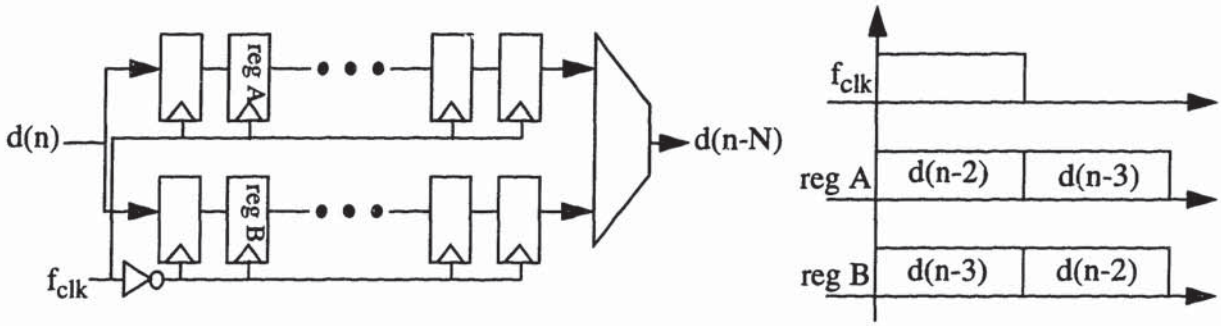

Figure 37: Example Parallel Shift Register

There are two main methods used of gating different inputs to a single output, muxes and tri-state buffers. The tri-state buffer method was chosen for use in this system. A tri-state buffer is

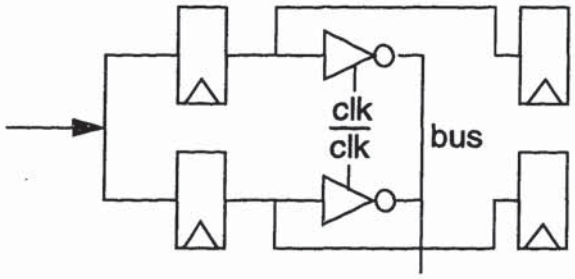

Figure 38: Tri-State Buffer Tapping Architecture

connected to a bus placed after each register cell that could possibly be the output necessary. These buffers are then gated such that the correct output is always on the bus line. Using this style of filter tapping, a 520 bit delay line with 32 taps was tested using increasing degrees of parallelism. The results, shown in Figure 39, show that for this case 2-level parallelism the effective

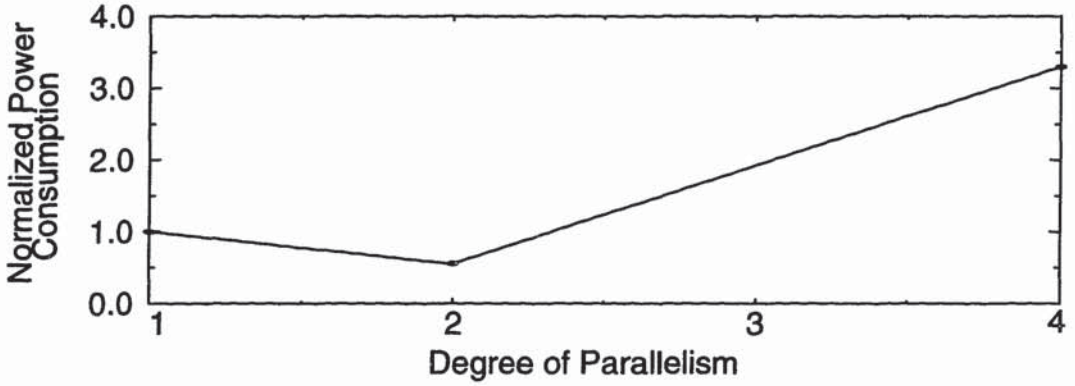

Figure 39: Tapped Delay Line Parallelism

switched capacitance decreases by half, but for 4-level parallelism, the routing capacitance is too great and the effective switched capacitance increases.

5.1.3 Matched Filter Logic Design

The matched filter logic design involved not only designing the logic to avoid unnecessary power consumption, but also choosing logic elements that dissipate the minimum power. The main logic element used in the matched filter is a register cell, so it is important to use a register cell with the minimum power consumption. Four low-power shift register cells were designed and analyzed: a C^2MOS register, a true single-phase clock register (TSPCR), a low-power frequency divider featured in [8] and another register with low clock load taken from [9]. Each register was designed (layed out) using minimum sized transistors whenever possible. Using minimum size transistors in almost all cases means that the n and p networks for most of the registers are not balanced. Therefore, the rise and fall times of the registers will not be equal. While this is not ideal, it poses no real problems as long as both the rise time and fall time meet the delay constraint. (*note: all transistors are 2.4/1.2 μm unless specified otherwise).

The C^2MOS register consists of two tri-state buffers connected in series and activated using clocks with opposite phase. The minimum number of transistors are used in this design, eight, which limits gate capacitance and there are only two paths between source and ground,

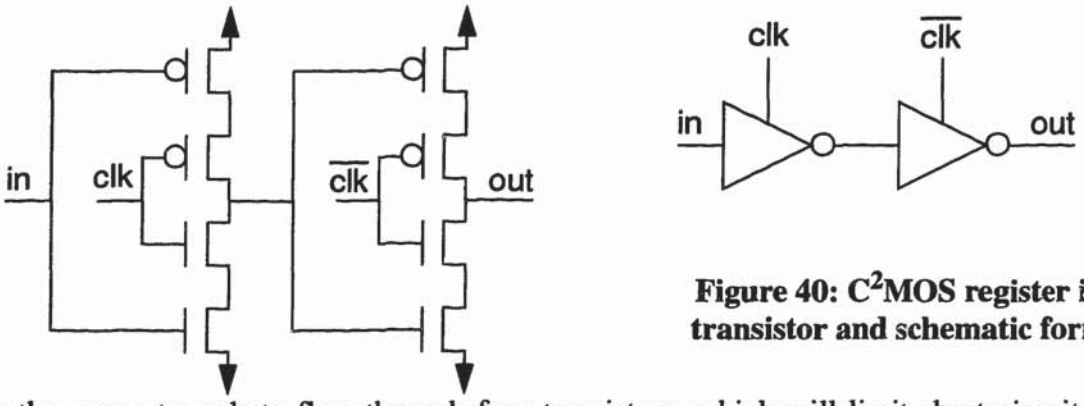

Figure 40: C^2MOS register in transistor and schematic form

where the current needs to flow through four transistors, which will limit short circuit current. Unfortunately, this register requires both clock phases for each register cell.

The true single-phase clock register is taken from the Berkeley Low-Power Cell Library [10]. The true single-phase clock register uses only one clock phase which is a huge benefit when the clock routing includes long or heavily loaded paths, however that single clock is still connected to four transistor gates. Other disadvantages of using the true single-phase clock register are the number of transistors, eleven, and the fact not all of the transistors can be minimum sized, some must be larger to insure proper circuit operation.

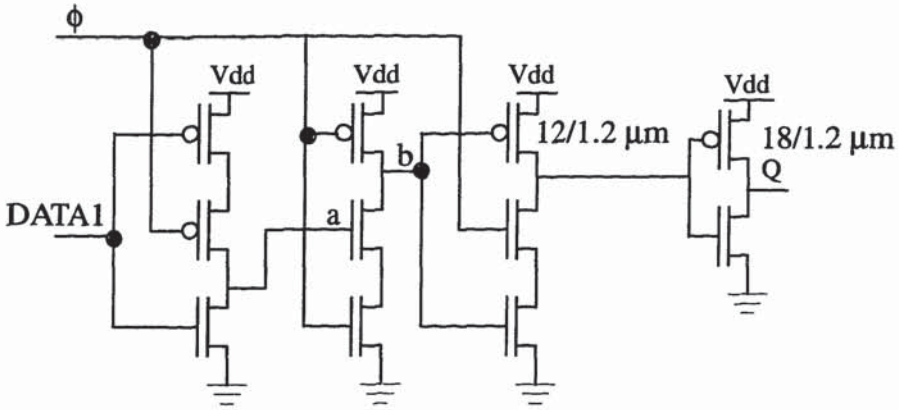

Figure 41: True Single Phase Clock Register

The low-power frequency divider, presented in [8], is basically a standard CMOS register with a number of redundant transistors removed. The register consists of two D-latches clocked on opposite phase as shown in Figure 42. The problem with this register is the number of transistors that are needed, sixteen, and the fact that both input and it's complement are needed as well as both clock phases, which leads to a lot of excess routing capacitance. The one small advantage is the fact both Q and $\overline{Q}$ are produced.

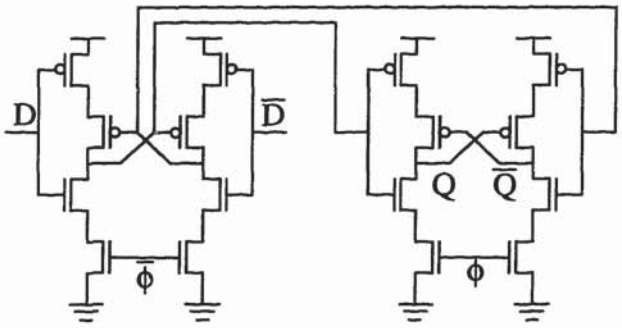

Figure 42: Low-Power Frequency Divider Based Register Cell

The final register design, the low clock load register, was developed from a D-latch presented in [9]. Only four transistors are connected to the clock in this design and all of the transistors in the design are minimum-sized. Also, all the p networks consist of one transistor and all the n networks consist of two transistors in series, the rise and fall times should be about equal. Unfortunately, this design uses twelve transistors and both clock phases.

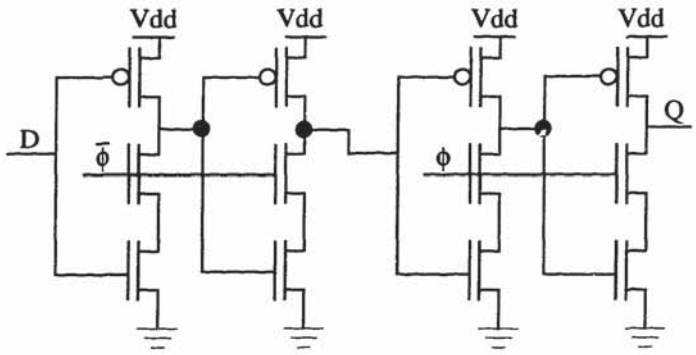

Figure 43: Low Clock Load Register Design

Each of the registers was layed out in Magic and extracted to Hspice for simulation. The average switched capacitances for a rising (Data =1) and falling (Data = 0) transitions, driving another register of the same type, were then measured for a range of supply voltages. The results, shown in Figure 44, show the C^2MOS register consumes the least power.

The increase in capacitance as supply voltage is increased can be described qualitatively by looking at the dependance of the gate capacitance on supply voltage. The C-V characteristic for a MOS capacitor is shown in Figure 45[12]. Determining if the device is operating in the low frequency region or the high frequency region depends on the availability of carriers, for a MOS capacitor. For a mosfet device there is a large supply of available carriers in both the drain and the

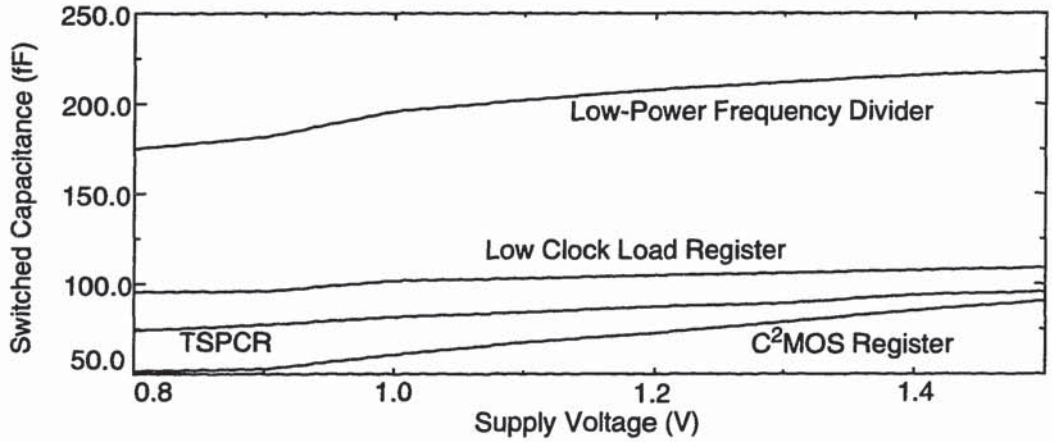

Figure 44: Register Cell Power Consumption vs. Supply Voltage

source, so the capacitance is always on the low frequency curve. The rise in capacitance as voltage is increased is the easily explained. As the gate charges the capacitance varies according the low frequency curve. As the supply voltage is increased the capacitance is at the maximum for a greater and greater amount of the voltage swing causing an apparent increase in the capacitance.

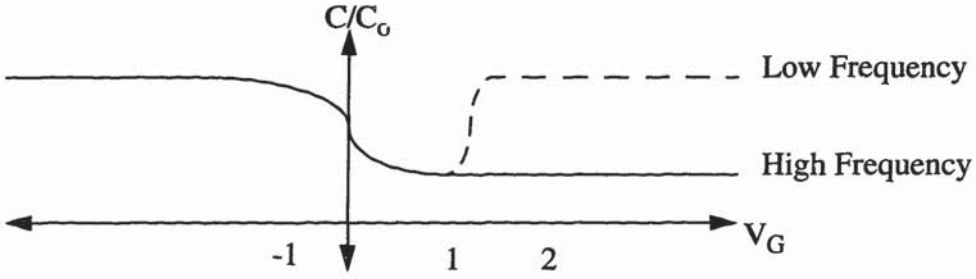

Figure 45: C-V Characteristic for a MOS Capacitor

5.2 BodyLAN™ Attachment System

During the attachment phase of operation the sensor searches for a Hub to connect with and begins to phase lock the local symbol clock with the Hub symbol clock. The BodyLAN™ attachment system contains two sections, the matched filter section and the beacon detection section. The matched filter section searches the incoming data stream for attachment beacons, while the beacon detect section prepares the data for the microprocessor.

5.2.1 Attachment Beacons

Each attachment beacon is a series of 32 bits transmitted during a 520 bit period. The attachment beacons are the same for all PEAs and consist of a 23 bit code word expanded to 32

bits, as shown in Table 5. A single bit of data is encoded in each beacon. If the complement of the

Interval Bit	Transmitted Bit	Interval Bit	Transmitted Bit
0:1	code[22]	168:169	code[20]
2:3	$\overline{\text{code[22]}}$	170:171	$\overline{\text{code[20]}}$
3:4	code[21]	256:263	code[19:12]
5:6	$\overline{\text{code[21]}}$	420:423	code[11:8]
		511:519	code[7:0]

Table 5: Code Bits and the Interval Location Where They are Transmitted

code word is transmitted the data bit is a zero and if the code word is transmitted, the data bit is a one. These bits are transmitted to the microprocessor, which collects the data and forms the initial TDMA plan. The initial TDMA plan is used during the synchronization phase.

A set of 106 attachment beacons contains the entire sequence of bits necessary to initialized the TDMA plan. The 106 bits are broken down as follows: 23 bits for a new, PEA specific code word, 42 bits define seven 6-bit time intervals for beacon arrival, 6 bits for a PEA identification number, 6 bits for frequency selection, an 8 bit initialization flag and a 16 bit checksum. Since there are eight beacons per frame and about 30 frames per second, the entire attachment packet is repeated about every 440 ms.

5.2.2 BodyLAN™ Matched Filter

The BodyLAN™ matched filter is responsible for searching the incoming data stream for beacons transmitted by the Hub. The matched filter section consists of two full matched filters, working on opposite phases of the symbol clock. Two matched filters are necessary for the phase-locking algorithm. Each matched filter consists of a tapped delay line, a binary comparison circuit, an adder and another pair of comparison circuits. The tapped delay line produces the necessary bits from an input string. The first comparison circuit takes the 32 bits from the delay line and bit-wise compares them to the code. The 32 bit output of the first comparison circuit is then summed to determine the total number of bits which matched. The 6 bit output of the adder is then compared against two thresholds using the second two comparison circuits to determine if the output is a hit. Up to three errors are allowed, so any value less than 4 is an anti-match and any value greater than 28 is a match.

A single matched filter was simulated using Irsim-cap to determine the effective capacitance switched for a representative test vector. This value can then be divided by the number of cycles to determine the effective switched capacitance per cycle. The effective switched capacitance was determined for both the condition the filter was active and the condition the filter was inactive. While inactive the filter has a data input which is random data at 2 Mhz and a single clock input which is also at 2 Mhz. The filter was simulated while active by using random data with valid codes inserted periodically. The effective switched capacitance per cycle (1 Mhz) of the filter and power consumption are shown in Table 6. However, since there are only 8 beacons

Mode	Cap. Switched per cycle	Power Dissipation
Active	80 pF	180 μW
Inactive	0.17 pF	.36 μW

Table 6: Effective Switched Capacitance per Cycle for the Matched Filter

(length 520 bits each) in each frame (length 65536 bits) and assuming 10% overhead needed to detect a beacon (i.e., active for 572 bits) the matched filter has a maximum duty cycle of 6.4%. Therefore, the maximum power dissipated by the matched filter should be approximately 23.9μW.

5.2.3 BodyLAN™ Beacon Detection Processing

The beacon detect section process the outputs of the matched filter section for the microprocessor. The system architecture is shown in Figure 47. Before the expected arrival time of a beacon the START signal is given. At this point the shift register begins to fill with data. The digital sequencer enables the decoder after the shift register is filled with valid data. The decoder then examines the outputs of the matched filters to determine if a beacon has been detected by one or

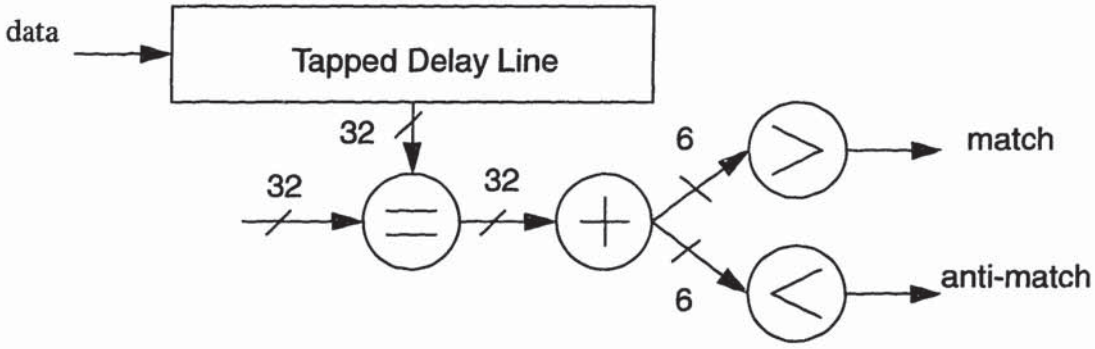

Figure 46: BodyLAN™ Matched Filter Architecture

both of the matched filters and what data bit has been encoded on the data bit (beacon_value). In addition the counter chain (described in Chapter 6) is sampled to determine the arrival time of the beacon. The three least significant bits of the counter chain are represented by SCC[2:0]. These values are sampled in the Beacon Detection Processing hardware to eliminate the effects of propagation delay of the counter-chain sampling signal.

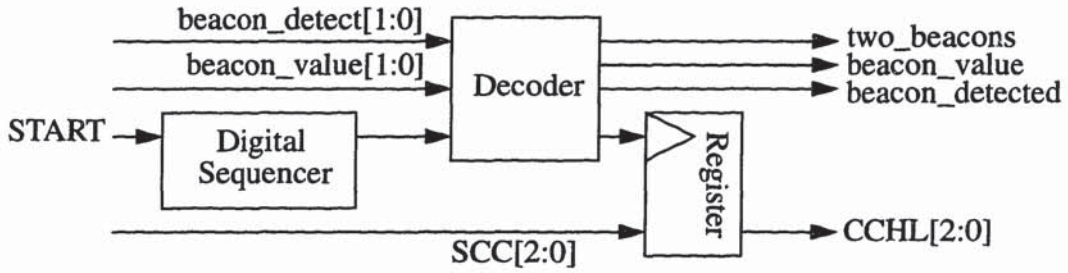

Figure 47: Beacon Detect System Architecture

The Beacon Detection Processing hardware was simulated using representative data streams and the effective switched capacitances while active and inactive are shown in Table 7.

Mode	Cap. Switched per cycle	Power Dissipation
Inactive	0.95 pF	0.13 μW
Active	1.06 pF	0.14 μW

Table 7: Effective Capacitance Switched for the Beacon Detection Hardware

Therefore, while the BodyLAN™ controller chip is in the attachment mode of operation the power dissipation should be 20-25 μW.

Chapter 6: BodyLAN™ Controller Chip

6.0 Introduction

In this chapter the implementation of the BodyLAN™ controller integrated circuit will be explained. Each of the blocks of the controller hardware architecture will be expanded and explained. The power dissipation for each block will be evaluated as well as the overall power dissipation for the controller.

6.1 BodyLAN™ Controller Hardware

A generalized picture of the BodyLAN™ controller hardware was shown in Chapter 1 and is repeated in Figure 48. This simple diagram has been expanded into a more detailed block diagram and is shown in Figure 49. The complete system consists of the microcontroller, the matched filter, the modem interface, the microprocessor interface and the counter chain. The microcontroller is responsible for generating the control signals necessary for implementing the TDMA plan. The matched filter is responsible for searching for and extracting data from beacons. The RF interface is responsible for routing data to and from the RF analog hardware. The sensor interface is responsible for extracting data from and feeding instructions to the sensors. The counter chain is responsible for the universal timing of the controller as well as determining the execution time for instructions. The microcontroller (Chapter 4) and matched filter (Chapter 5) sections have been described previously.

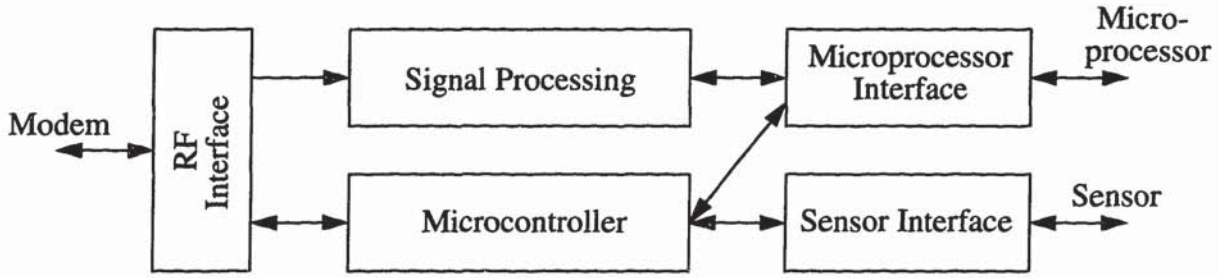

Figure 48: Custom Integrated Circuit Architecture

6.1.1 RF Interface

The modem interface is responsible for routing the proper data and control signals to the RF analog hardware and routing the output of the RF analog hardware (the received data) to the proper location. The modem interface system architecture is shown in Figure 50. The modem

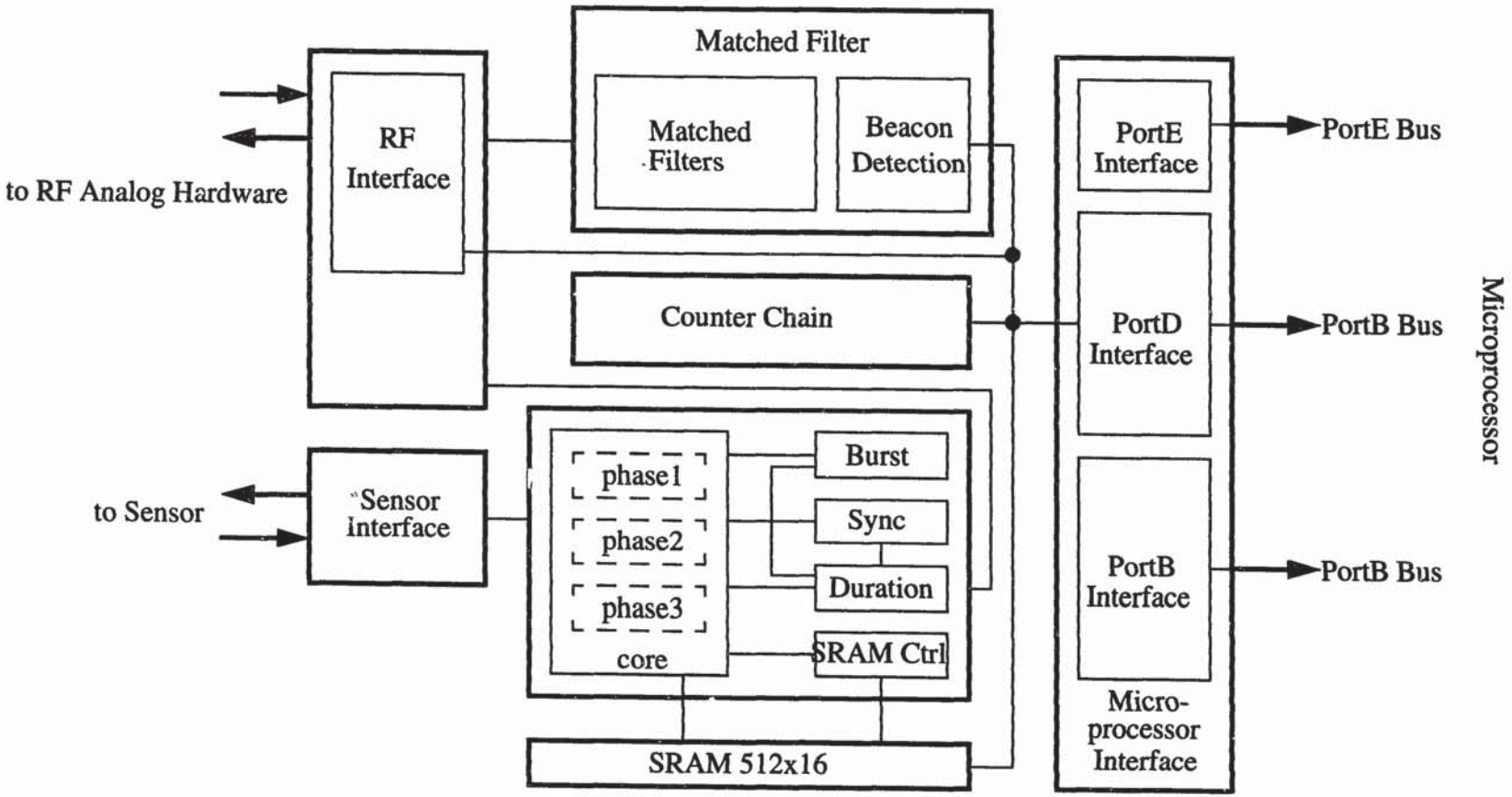

Figure 49: Expanded View of the BodyLAN™ Controller Architecture

interface has two modes of operation receive and transmit. The primary responsibility while in transmit mode is to route data to the output. The primary responsibility while in receive mode is to route data from the incoming data stream to the proper registers.

When the system is in the transmit mode the 12-bit loadable shift register is first loaded with the appropriate data, which can be PEA to Hub microprocessor (up) information, PEA to Hub CCC data, the PEA status, or information from the sensor. Once the data is loaded, it is shifted out through the RF_transmit line at the appropriate time. A 12-bit shift register is necessary since each 4-bit data block is transmitted three times, because of the error detection and correction scheme.

When the system is in receive mode, the 12-bit loadable shift register is always in shift register mode. After all twelve bits of data have been clocked into the shift register error correction and detection are performed on all twelve bits in parallel. If the data has been received without error, it is then routed to the appropriate location. If the data has been received in error the microprocessor may be awakened to evaluate the problem. All the control signals for the muxes as well as the clock for the loadable shift register are supplied by the microcontroller.

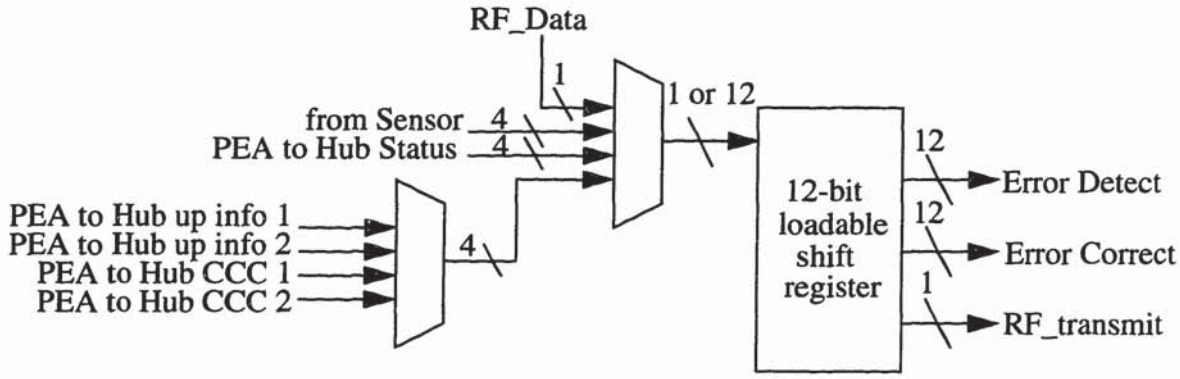

Figure 50: Modem Interface Architecture

6.1.2 Microprocessor Interface

The microprocessor interface provides the link between the BodyLAN™ controller integrated circuit and the microprocessor. The microprocessor and the BodyLAN™ controller integrated circuit are linked by three main buses. The B-bus is a 6-bit bus used mainly as an address bus. The D-bus is an 8-bit bus used mainly as the data bus. The E-bus is a 2-bit bus used to control the microprocessor's D-bus latch, which operates in the slave mode. The microprocessor interface consists of a series of registers on the BodyLAN™ controller integrated circuit that are writable

by the microprocessor as well as a few registers that are readable by the microprocessor. In addition, the microprocessor has access to the SRAM on the BodyLAN™ controller integrated circuit.

There are a total of twenty-one different addresses that can be accessed by the microprocessor. The address, register size, type (read-only, write-only), and register content are shown in Table 8. On a write access, the data is first latched on the D-bus, then the address is asserted on the

Table 8: Register Allocation for BodyLAN™ System

Address	Size (bytes)	Type	Register Content
000000	0	N/A	Null Register
000001	1	Write-Only	Charge Pump Control Register
000010	1	Write-Only	TDMA Starting Address
000011	1	Write-Only	Chip Control Register
000100	1	Write-Only	Matched Filter Code[22..16]
000101	1	Write-Only	Sub-Burst Interval Match Register
000110	1	Write-Only	Unused
000111	1	Write-Only	Duration Comparator Control Register
0010xx	1	Write-Only	Unused
001100	2	Write-Only	Counter Chain Match Registers
001101	2	Write-Only	Pre-scaler Divisor Registers
00111x	2	Write-Only	Matched Filter Code [15..0]
01xxxx	1	Write-Only	Enter Sleep Mode
10x0xa	3	SRAM Write	Initiate SRAM write access (a is msb of address)
10x1xa	2	SRAM Read	Initiate SRAM read access (a is msb of address)
1100x0	1	Read-Only	Oscillator Status Register
1100x1	1	Read-Only	Received Data Info Block (from Hub)
1101xx	2	Read-Only	Counter Chain Match Register
111x00	1	Write-Only	CCC data out register high nibble
111x01	1	Write-Only	CCC data out register low nibble
111x10	1	Write-Only	microprocessor data out register high nibble

Table 8: Register Allocation for BodyLAN™ System

Address	Size (bytes)	Type	Register Content
111x11	1	Write-Only	microprocessor data out register low nibble

B-bus. When the BodyLAN™ controller integrated circuit detects a change in the B-bus it will route the data to the correct register and trigger the load signal. On a read access, the address is asserted on the B-bus and the data is latched to the D-bus as soon as it is available.

The PortB interface consists of a small amount of hardware to determine if the PortB bits have changed and a lot of decoding hardware. A change in any of the PortB bits indicates a register access request. So, the PortB bits must constantly be monitored for changes. This monitoring is accomplished using the circuit shown in Figure 51. Once it has been determined that the PortB bits have changed, the proper registers need to be triggered. If the data to be transferred is a single byte the transfer is trivial, the D-latch on the microprocessor is triggered using PortE and the data can be immediately loaded to the correct register.

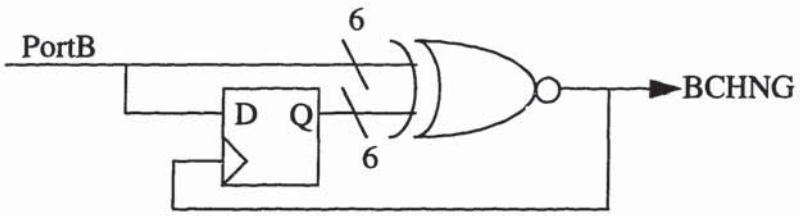

Figure 51: Hardware to Determine Change in Port B

Transfers of more than one byte are not trivial to implement. To determine proper triggering for a double byte access, the microprocessor's I/O method must be examined. For the PIC16C64 microprocessor an I/O operation take two instructions, one to prepare the data and one to latch the data to the output.. The PIC16C64 instructions require four clock cycles to complete.

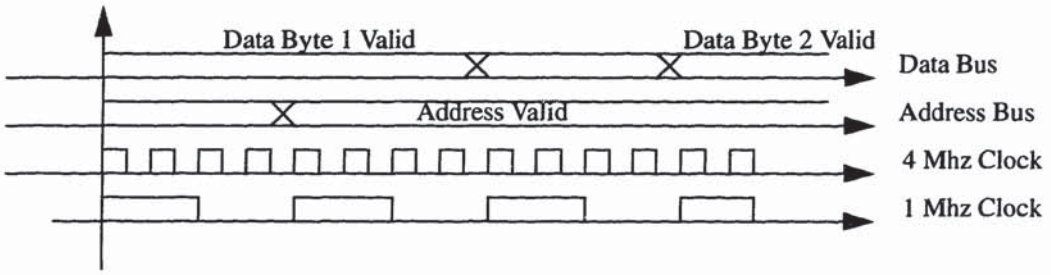

Figure 52: Timing of a Double-Byte Access

Since the first byte of data is valid as the address changes, the first byte can be latched asynchronously as soon as the transfer is decoded. The second byte must be latched two instructions (8 cycles) later as shown Figure 52

There are also two special accesses that are handled outside the PortB interface, the SRAM accesses and the microprocessor sleep access. If PortB indicates an SRAM access the portB interface triggers the SRAM controller. When a microprocessor sleep access is requested the three lsb of the address are needed to determine the length of time the microprocessor is disabled.

The PortD interface primarily multiplexes data so it can be accessed using the single 8-bit D-bus. The information accessible by the microprocessor includes: SRAM reads, counter chain values, the oscillator status register, the information block register and the TDMA wake-up information (both TDMA and non-TDMA mode). The structure of the PortD interface is shown in Figure 53. The control for the PortD bus is provided by the microcontroller.

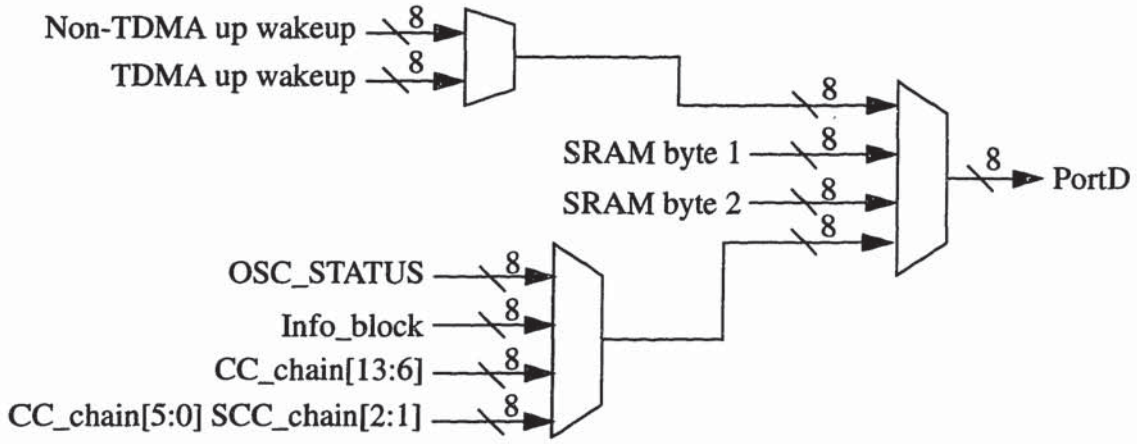

Figure 53: Port D Interface

6.1.3 Sensor Interface

The sensor interface consists of a serial to parallel converter used to prepare the serial data stream from the sensor for transmission by the PEA and a parallel to serial converter used to prepare instructions from the PEA for transmission on the parallel stream. The input and output of the sensor are operated in slave more so clocks for the input and output latches must be provided by the PEA.

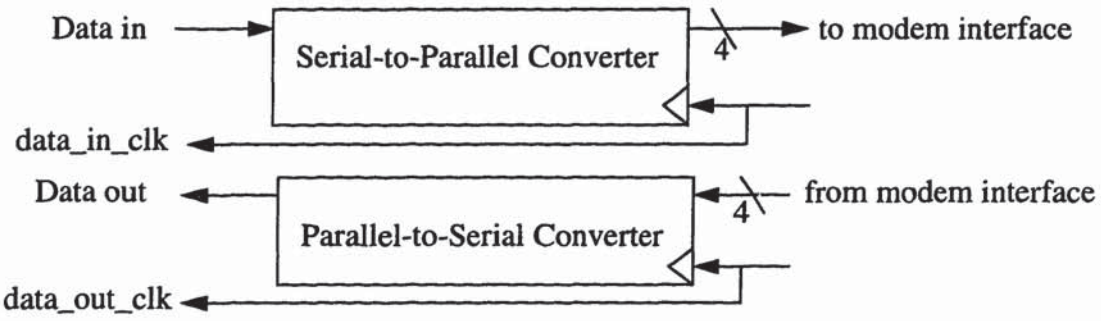

Figure 54: Sensor Interface Architecture

6.1.4 Clock Division Chain

The clock division chain provides the universal time reference for the entire BodyLAN™ controller integrated circuit. The clock division chain also produces the START signal (CC_match) for the microcontroller instructions. The design of the clock division chain is critical since it is one of the few systems on the chip that is always powered on. The design for the clock division chain is shown in Figure 55.

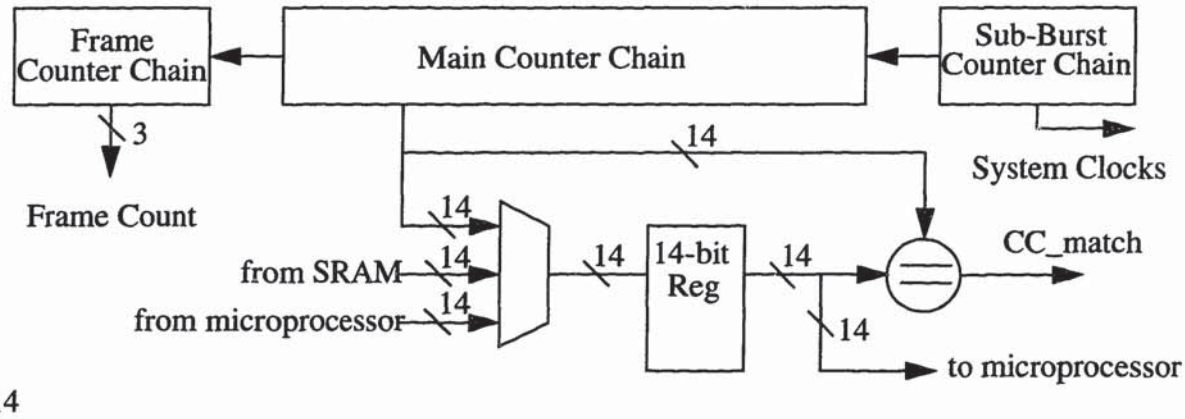

Figure 55: Clock Division Chain Design

The three counter chains provide the universal timing for the entire BodyLAN™ controller integrated circuit. The Sub-Burst Counter Chain is a three bit counter that increments at 4 Mhz. This counter provides the 2 Mhz, 1 Mhz and 500 kHz clocks used by the rest of the system. The Main Counter Chain is a 14-bit counter which increments at 500 kHz. Since the data rate is 2 Mhz and all data is packed into 4-tuples the maximum burst rate is 500 kHz, the Main Counter Chain indicates how many bursts have occurred in the current frame. The Frame Counter Chain is another 3-bit counter which operates at the frame rate (30.51 Hz). The Frame Count is used to indicate for which PEA a CCC communication is intended, since a CCC transmission is only valid when the PEA identification number, which is assigned by the Hub as part of the attachment

packet, matches the Frame Count.

The output of the counters are used in various ways depending on the mode of operation of the circuit. During the course phase detection phase the arrival times of the beacons are necessary to determine actual inter-arrival times. When a beacon is detected during course phase adjustment the value in the Main Counter Chain is recorded in the 14-bit register which is accessible by the microprocessor. When the TDMA plan is initiated, the counter chain needs to be reset at the correct moment to frame synchronize the PEA to the Hub. This timing of this event is calculated by the microprocessor which can then write that information into the 14-bit shift register. When the Main Counter Chain matches the value stored in that register the counter chain will reset, frame synchronizing the PEA to the Hub. Finally, during normal operation the execution time of an instruction needs to be compared to the Main Counter Chain to provide proper timing of the instruction execution. During TDMA operation the 14-bit register is loaded with the 13-bit execution time from each instruction. The CC_match signal indicates the microcontroller should begin execution of the current instruction.

6.1.5 Microprocessor Wake-up Circuitry

The microprocessor wake-up circuitry is responsible for enabling the microprocessor when it is needed and enabling the wake-up data to the microprocessor. The microprocessor can be enabled by three different events. During the attachment phase the microprocessor is enabled by the successful reception of an attachment beacon. During the TDMA mode the microprocessor can be awakened to handle a complex instruction. Finally, the microprocessor will always be enabled after a set period of time to do a check on the system, this is referred to as a time-out wake-up. The signals for the first two cases are produced by the beacon detection and microcontroller systems respectively. The time-out wake-up is implemented in the wake-up circuitry.

The time-out wake-up is a backup to prevent the system from entering a locked state. Communications event are constantly scheduled, so if one of the communications events is missing, something may be wrong with the system. Therefore, the time-out wake-up can be scheduled to occur a period of time after the next communications event. If the microprocessor is enabled by the time-out circuitry, the microprocessor can make corrections, such as moving back to the synchronization or attachment modes. The three least significant bits of an 01xxxx address determine

the length of the time-out as shown in Table 9.

Code	Time	Code	Time
000	256 µs	100	1 sec
001	512 µs	101	2 sec
010	1024 µs	110	4 sec
011	2048 µs	111	8 sec

Table 9: Microprocessor Sleep Times

6.1.6 SRAM Controller

The SRAM controller is responsible for controlling the access to the SRAM. Both the microprocessor and the microcontroller have access to the SRAM. The microprocessor has both read and write access to the SRAM, while the microcontroller has read access only. The microprocessor accesses the SRAM through the Port B bus. Once an access is requested the microprocessor provides the address (and data if a write access) on the Port D bus. The SRAM controller latches the address and the data, completes the SRAM access and latches the data back on the Port D bus (for a write access). A microcontroller access consists of reading a single instruction from memory, which is two words of SRAM data. The first word, containing the opcode and sub-interval times is latched to an instruction register. The second word, which is the execution time, is latched in the counter chain so it can trigger the microcontroller to begin execution at the proper time. The SRAM controller architecture is shown in Figure 56.

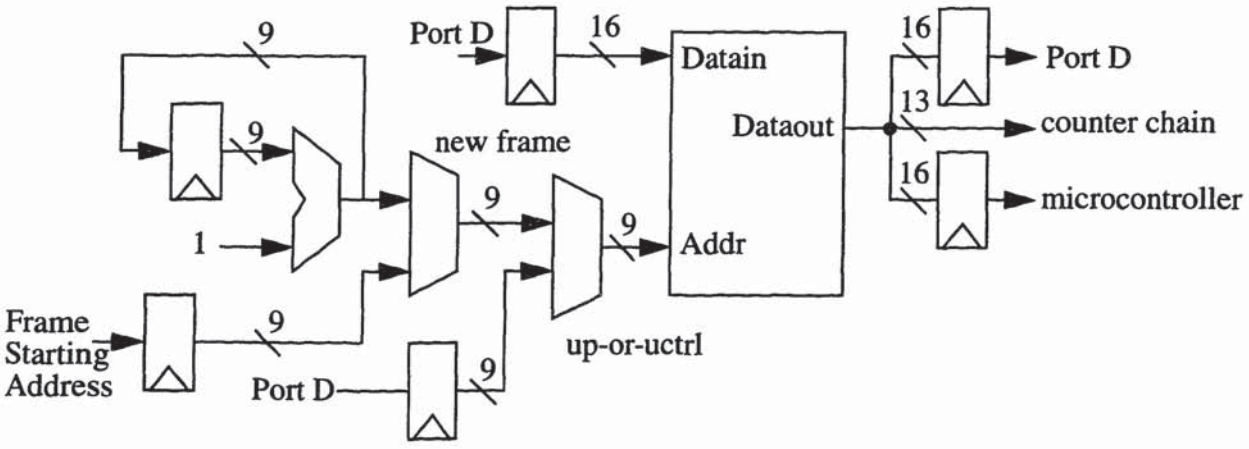

Figure 56: SRAM Controller Architecture

6.2 BodyLAN™ Controller Power Measurements

The BodyLAN™ controller integrated circuit has been designed to operate at a minimum power level. To determine an estimate of the power dissipation the minimum supply voltage must be determined. The minimum supply voltage is dependent on the critical path of the circuit. For the BodyLAN™ controller the critical path is the addition tree used in the matched filter. The addition tree is used at 2 Mhz, so the maximum delay through the tree is 500 ns. Using estimates from the CMOS Standard Cell 2_3lp Library Documentation [13] the delay should be about 100 ns at 1.5 V. Figure 57 shows the delay through the adder as a function of the supply voltage [13]. From this data it is obvious the system should have no problem running at 1.5 V..

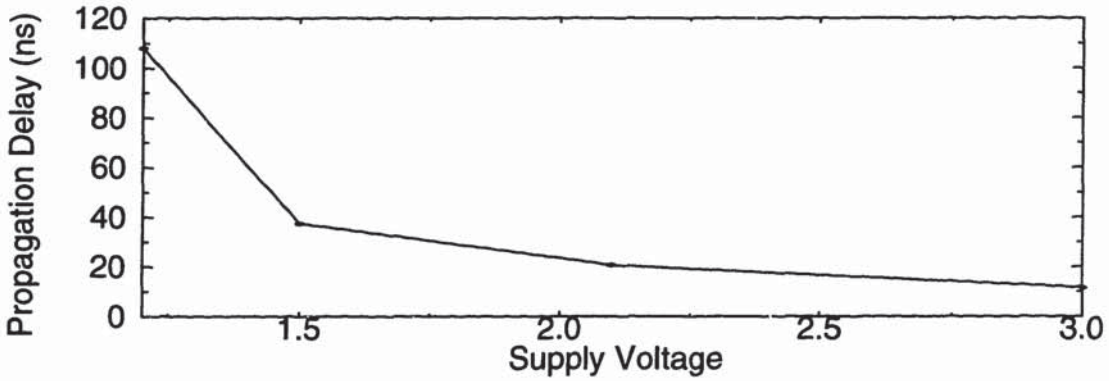

Figure 57: Delay vs. Supply Voltage for the Adder Tree

All sections of the BodyLAN™ controller have been simulated from layout using irsim-cap and representative data. For the purpose of simulating the power consumption the BodyLAN™ controller chip has been broken into seven components: microcontroller, microprocessor interface, RF interface, SRAM, counter-chain, charge pump and microprocessor wakeup.

The microcontroller power dissipation has been quantified as the energy needed to perform each instruction. This measurement when combined with the number of times each instruction is performed per frame and the power dissipation while the microcontroller is inactive yields the power dissipation of the microcontroller for a given TDMA plan. The power dissipated while the circuit is inactive was measured by running the microcontroller for 100 cycles without triggering it to execute an instruction. The power dissipation while inactive was found to be 3.8 μW. The energy dissipated by an instruction was determined by measuring the amount of capacitance switched to perform that instruction and subtracting the amount of capacitance that would have been switched, due to the clock, in that time period if no instruction had been performed. The instruction subintervals were set to one half of the maximum to provide an average result. The

results are shown in Table 10 as well as an example of computing the power dissipation for a 5 kb/

Opcode	Excess Cap Switched (pF)	Energy (pJ)	Rate (per Frame)	Power (μW)
0000	54.3	122	1	.004
0001	55.4	125	5	.019
0010	219	492	1	.015
0011	226	507	1	.015
0100	61.1	138	41	.170
0101	60.2	135	41	.166
0110	61.2	138	41	.170
0111	60.2	135	41	.166
1000	62.2	140	1	.004
1001	61.2	138	1	.004
1010	62.2	140	1	.004
1011	61.2	138	1	.004
1100	40.0	90	1	.003
Total (including clock)				4.5

Table 10: Microcontroller Power Dissipation

s bi-directional user and command channel links.

The microprocessor interface power dissipation was measured in a similar manner. The energy needed for each type of access was determined. The power dissipation while inactive was

PortB	Excess Capacitance (pF)	Energy (pJ)
000 001	4.9	11
000 010	5.7	13
000 011	4.9	11
000 100	6.3	14
000 101	5.0	11

Table 11: Energy Dissipation for Microprocessor Interface

PortB	Excess Capacitance (pF)	Energy (pJ)
000 111	5.0	11
001 100	10	23
001 101	8.8	19
001 11x	8.5	19
01x xxx	6.2	14
10x 0xx	5.9	14
10x 1xx	6.8	15
110 0x0	5.9	13
110 0x0	11	24
110 0x1	12	28
110 1xx	14	32
111 x00	5.5	12
111 x01	5.6	13
111 x10	5.6	13
111 x11	4.8	11

Table 11: Energy Dissipation for Microprocessor Interface

found to be .33 μW.

The power dissipation of the RF interface depends only on the number of access since there is no clock routed to the RF interface permanently, all of the clock signals are supplied from outside sources. The energy dissipated for each type of RF interface access is shown in Table 12.

Type of Access	Cap. Switched (pF)	Energy Dissipated (pJ)
Hub to User	42	95
Hub to Microprocessor	30	68
User to Hub	74	170
PEA status to Hub	26	59
Microprocessor info to Hub	30	68

Table 12: RF Interface Energy Dissipation

Type of Access	Cap. Switched (pF)	Energy Dissipated (pJ)
CCC info to Hub	30	68

Table 12: RF Interface Energy Dissipation

The power dissipated in an SRAM accesses consists of the power dissipated in the SRAM controller and the SRAM. The power dissipation for an SRAM access for a 512x16 SRAM can be estimated to be about 35 pJ per access [15]. The power dissipated while the SRAM controller is inactive is .47 μW. The energy dissipated by the SRAM controller is shown in Table 13, remember that a microcontroller access consists of reading two words.

Access Type	Excess Cap Switched (pF)	Energy (pJ)	Total Energy (with SRAM) (pJ)
microprocessor read	9.8	22	57
microprocessor write	12	27	62
microcontroller read	25	56	126

Table 13: SRAM Energy Dissipation

The power dissipation of the final three components is shown in Table 14.

Component	Inactive	Active
Counter Chain	N/A	6.3 μW
Wakeup	.8 μW	.8 μW
Charge Pump	.13 μW	.06 μW

Table 14: Power Dissipation of BodyLAN™ Controller Chip Components

The layout of the core of the BodyLAN™ controller chip is shown in Figure 58 and the chip statistics are shown in Table 15

Area	17.9 mm^2
Number of Transistors	91,000
Supply Voltage	1.5 V
Power Dissipation	20 - 50 μW

Table 15: BodyLAN™ Controller Chip Statistics

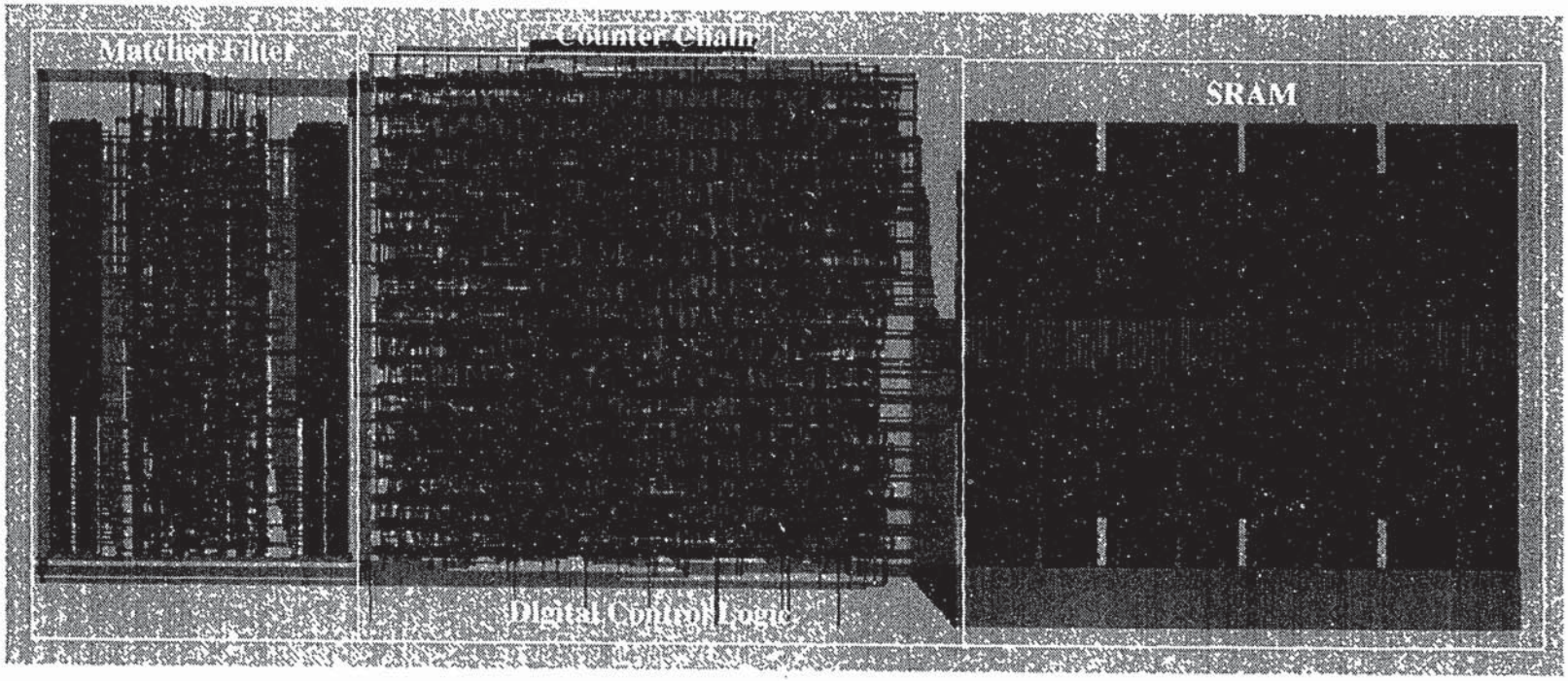

Figure 58: BodyLAN™ Controller Chip

Appendix 1: A Comparison of Switched Capacitance Estimation Methods

Since the design described in this thesis was never fabricated, physical measurement of the power dissipation of the system was impossible. Therefore, extensive simulation was used to approximate the average capacitance switched per cycle. Switched capacitance is a good metric for computing power consumption since it is easily combined with the frequency and the supply voltage, both of which are known, to give an estimate of average power consumed, as shown below.

$$P = CV^2f \qquad \text{Eq. 18}$$

In this appendix three different methods of determining the average switched capacitance of a circuit are presented. The first method is using irsim-cap, which is a switch level simulator that has extensions for computing switched capacitance. The second method uses Hspice to measure the amount of current drawn from the supply. The final method is estimating the switched capacitance by hand calculation.

Irsim-cap is a modified version of the Irsim-9.0 program developed by Stanford University. The modifications, performed by Paul Landman at UC Berkeley, include an expanded number of commands to record the capacitance switched as well as expanding the algorithm to include partial glitching. To measure the switched capacitance for a given simulation, Irsim-cap records the number of times each node transitions and the capacitance of each node. The effective switched capacitance is then calculated as shown below.

$$C_{eff} = \sum_{\text{for all i}} (\#\text{ of transitions})_i \times C_i \qquad \text{Eq. 19}$$

The following method was used to prepare a design to be simulated by Irsim-cap. First, the circuit must is designed in a magic format, either using hand-layout in magic or an automatic layout generation program that interfaces with magic (such as lager). The circuit is then extracted in magic (**:extract all** command). The .ext file produced by magic is then translated to a .sim file, which is readable by Irsim-cap, by using **ext2sim**.

A command file must then be created which specifies the input stimulus for the circuit

under test. To get an accurate measurement of the switched capacitance, all nodes in the circuit must be initialized to either high or low value. The circuit can then be simulated using input stimulus that model the expected stimulus in real use (i.e. keep the activity factors on the inputs as close as possible to the expected activity factors in the final design.)

Irsim-cap uses a .prm file to configure the devices to be modeled. The only variable that needs to modified in this file is the *capga* variable which specifies the gate capacitance per μm^2. The gate capacitance per unit area is computed by:

$$C_g = \frac{\varepsilon_{si}}{t_{ox}} \qquad \text{Eq. 20}$$

Which equals .6011 $fF/\mu m^2$ for the process modelled in this thesis (Mosis 1.2 μm process, t_{ox} = 212 angstroms).

Estimating the switched capacitance using Hspice involves measuring the total power dissipated during a simulation and comparing that with the energy need to charge a capacitor.

$$E = \int v_{DD} i_{DD} dt = C_{eff} \times V_{DD}^{\ 2} \qquad \text{Eq. 21}$$

$$\frac{\int i_{DD} dt}{V_{DD}} = C_{eff} \qquad \text{Eq. 22}$$

Since the supply voltage is constant, the effective switched capacitance is equal the to integral of the supply current divided by the supply voltage.

The integral of the supply current can be measured directly in Hspice by sampling the source current and running the same supply current into a capacitor as shown below.

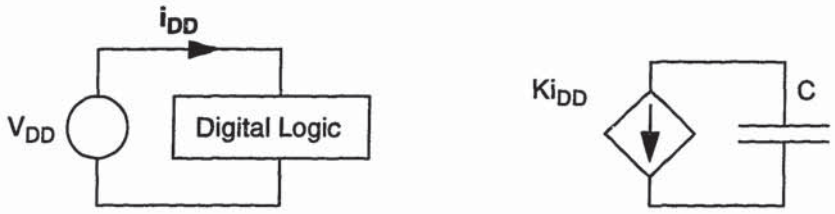

Figure 59: Circuit Used to Measure Power Consumption in Hspice

Using this configuration the supply current is integrated by the capacitor and scaled by the capacitor and the scale factor (K) of the dependant source.

$$v_C = \frac{1}{C}\int i_C dt = \frac{K}{C}\int i_{DD} dt \qquad \text{Eq. 23}$$

When the scale factor (K) is set equal to the capacitance, v_C exactly equals the integral of the supply current. The effective switched capacitance equals the integral of the supply current (or the voltage v_C) divided by the source voltage.

$$C_{eff} = \frac{\int i_{DD} dt}{V_{DD}} \qquad \text{Eq. 24}$$

(note: To get a more accurate measurement of the switched capacitance the current supplied by all the input sources must also be measured)

The final method of power estimation used was hand calculations. To estimate the power consumed using hand-calculations the capacitance of the cells must first be determined. Since the purpose of the hand-calculation method is to provide a quick estimate of the power consumed all cells were characterized by only the input and output capacitances ignoring any internal capacitances. The reasons for this is simple, first of all the activity factors for the inputs and outputs are generally more easily derived than the activity factors for internal nodes which are generally much more complex. Secondly, the number of input and output nodes for a cell is generally small, while the number of internal nodes can be very high.

To determine the input and output capacitances the capacitance per unit area for gate, p-diffusion, and n-diffusion were calculated from Spice parameters provided by Mosis for a 1.2 μm process. The total gate, p-diffusion, and n-diffusion area were determined by examining each cell.

Type	Capacitance
Gate	.6011 fF/λ^2
n-diffusion	.1673 fF/λ
p-diffusion	.1416 fF/λ

Table 16: Area Capacitances for a 1.2u CMOS process

The capacitances per unit area and area measurements were then combined to produce an equivalent capacitance at each input and output nodes. The activity factors (α) for each input and output node were then determined. These activity factors multiplied by the capacitances produced an

estimate of switched capacitance.

$$C_{eff} = \sum_{\text{for all i}} \alpha_i \times C_i \qquad \text{Eq. 25}$$

When determining activity factors, special care must be take so that those activity factors truly represent the expected signal statistics.

Each method of power estimation has its advantages and disadvantages. Hspice is the most accurate of the power estimation techniques since it accounts for all the current drawn from all the sources, including short circuit, subthreshold, and leakage currents. However, simulating large designs in hspice is not feasible because the run time of an hspice simulation is so long. Therefore, hspice simulation should be restricted to those sections that require precision power estimation.

Irsim-cap is the second most accurate power estimation technique since it takes into account the internal capacitances as well as the external (input/output) capacitances. The main advantage to Irsim-cap is the simulation speed. Since Irsim-cap is a switch level simulator on average, it will run much faster than hspice, with results that are relatively accurate. The major disadvantage in using Irsim-cap is the fact a circuit layout must be produced before the simulator can be used. Therefore, irsim-cap is ideal for power consumption estimation of large systems from layout.

Hand-calculations tend to be the least accurate of the power estimation techniques, since it does not account for any internal capacitance switching. Since none of the internal nodes are accounted for hand-calculated power estimates are increasingly low as circuit complexity increases. The advantage of hand-calculations is that sections do not need to be layed out. A power estimate, within an order of magnitude, can be achieved with only a paper design and logic cell characterizations.

All three power estimation methods were used on a set of circuits which exhibited increasing complexity. The circuit set included, in order of increasing complexity: a single inverter, a chain of five inverters, a four bit AND-AND array, a five-bit shift register, and a five bit full adder. Figure 60 shows the output power estimation normalized to the Hspice estimate. For circuits with low complexity, all three of the power estimation techniques give values that are similar. As the systems become increasingly complex, the power estimate using hand analysis is increasingly

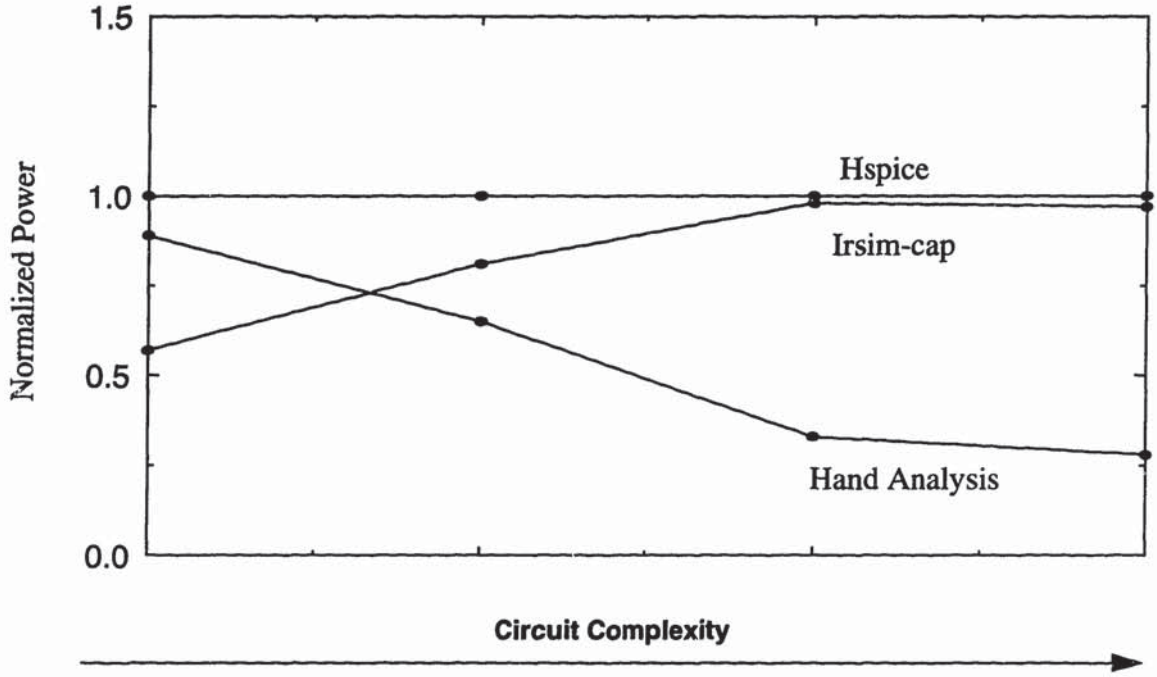

Figure 60: Plot of Power Estimate Normalized to the Hspice Estimate for Increasing Circuit Complexity

low, which the analysis using irsim-cap is still approximately equal to the Hspice analysis. While hand analysis can be used to give a rough estimate of power consumption, within an order of magnitude, it is not reliable for complex circuits. However, irsim-cap power estimations are valid at high levels of complexity.

References

[1] Wellner P., "Interacting with Paper on the DigitalDesk.", Comm. ACM, Vol. 36 no. 7, July 1993, pp. 87-96

[2] Chandrakasan, Anantha, "Low Power Digital CMOS Design", PhD Thesis, UC Berkeley, 1994;UCB/ERL M94/65.

[3] Tanenbaum, Andrew S., Computer Networks, Prentice Hall, 1988.

[4] Hambley, Allen R., An Introduction to Communications Systems, Computer Science Press, 1989.

[5] Samuel Sheng, Randy Allmon, Lapoe Lynn, Ian O'Donnell, Kevin Stone, Robert Brodersen; "A Monolithic CMOS Radio System for Wideband CDMA Communications", Calgary Wireless 94 Conference, July 1994.

[6] E. Vittoz, "CMOS Analog Integrated Circuits Based on Weak Inversion Operation", IEEE JSSC, Vol. SC-12, No. 3, June 1977

[7] C. Piquet, "Ultra Low-Power Digital Design", Low-Power/Low-Voltage IC Design Course, May 9-13, 1994, Monterey, CA, USA.

[8] M. Fujishima, K. Asada, Y. Omura, and K. Izumi, "Low-Power 1/2 Frequency Dividers Using 0.1-mm CMOS Circuits Built with Ultrathin SIMOX Substrates", IEEE JSSC, Vol. 28, No. 4, April 1993.

[9] N. Weste and K. Eshraghian, Principle of CMOS VLSI Design: A Systems Perspective, Addison-Wesley Publishing Co., 1992.

[10] T. Burd, "Low-Power CMOS Library Design Methodology, MS Thesis, UC Beerkeley, 1994; UCB/ERL M94/89.

[11] W. Lindsey and C. Chie, "A Survey of Digital Phase-Locked Loops", Proc. of the IEEE, Vol. 69, No. 4, April 1981.

[12] R. Pierret, Field Effect Devices, Modular Series on Solid State Devices Volume IV, Addison-Wesley, 1990.

[13] R.W. Broderson, et. al., LagerIV Cell Library Documentation, Electronics Research Laboratory, University of California, Berkeley, June, 1988.

[14] Bergamo, M. A, Hoder, D. J., Gigabit Satellite Network Using NASA's Advanced Communications Technology Satellite (ACTS): Features, Capabilities, and Operations, Pacific Telecommunications Conference, January 22-26, 1995

[15] A. Burstein, Low Power SRAM Cell Documentation, Electronics Research Laboratory, University of California, Berkeley, 1994

[16] The Code of Rederal Regulations Title 47, Telecommunications, Part 15, Radio Frequency Devices, Subpart C, Intentional Radiators, Section 15.209, Radiated Emission Limits; General Requirements

THESIS PROCESSING SLIP

FIXED FIELD: ill. _____ name _____

index _____ biblio _____

► COPIES: (Archives) Aero Dewey (Eng) Hum

Lindgren Music Rotch Science

TITLE VARIES: ►☑ "Low Power" is not hyphenated in degree book.

NAME VARIES: ►☑ James

IMPRINT: (COPYRIGHT) _____

►COLLATION: 74 l

►ADD. DEGREE: _____ ►DEPT.: _____

SUPERVISORS: _____

NOTES:

cat'r: date:

page: F44

►DEPT: E.E.

►YEAR: 1996 ►DEGREE: M.S.

►NAME: BARBER, Thomas J.