Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON), Santa Clara, CA, September 2005.

RISE – Co-S : High Performance Sensor Storage and Co-Processing Architecture

A. Banerjee, A. Mitra, W. Najjar, D. Zeinalipour-Yazti, V. Kalogeraki[1] and D. Gunopulos[1]
Department of Computer Science, University of California, Riverside
(anirban, amitra, najjar, csyiazti, vana, dg) @cs.ucr.edu

***Abstract*—Low power, high performance sensor designs are of particular importance nowadays considering the plethora of emerging applications in a wide range of fields. Applications such as audio capture, FFT (Fast Fourier Transform), image sensing, motion detection, feature extraction and cepstrum calculations generate raw data in much larger quantity than more mundane applications as temperature and humidity sensing. In-situ processing of data is an effective way to reduce the amount of information needed to be transmitted over the wireless links. The Co-S platform which we develop addresses this very crucial aspect of sensor network design. The Co-S integrated with a System on Chip (SoC) based host platform and a gigabyte scale energy efficient data storage system, simultaneously satisfies the constraints of low power consumption and a small form factor. As opposed to the Sense-and-Send approach which simply entails transmission of raw sensory data from the nodes, the proposed Sense-and-Store paradigm, builds upon the availability of the low power flash memories for storage of sensed data along with key query results (max, min, average, extracted harmonics) for transmission through the network. This approach not only reduces energy consumption (typically two orders of magnitude), of the sensing node in question, but also intuitively reduces network traffic load vis-à-vis complete deployment scenario.**

***Index Terms*–Coprocessors, Performance, Flash Storage**

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are deployments of low power devices [28] which are equipped with a multitude of features such as a processor, a radio, flash memory, and several environmental monitoring sensors. WSNs are expected to help researchers in monitoring environmental conditions at a high fidelity. Deployments of WSNs have already emerged in environmental and habitant monitoring [22], [35], seismic and structural monitoring [36], factory and process automation and a large array of other applications [1], [4]. These nodes may be used for sensing and reporting events as and when they occur, or sensing and logging on a regular time schedule. WSNs coagulate data from the sensing nodes throughout the network at the sink. Sensor platforms designed to live up to the expectations of the deployment specifications need not only to be power efficient [28] but also must incorporate the ability to reduce the amount of information transferred via wireless links over the network in order to avoid depleting the complete setup of precious battery resources. Popular embedded sensor network architectures like Mica [4], Wisenet [5], Rene [6], Telos [7] and iBadge[8], employing power aware computing methodologies have been deployed successfully for a wide range of applications such as temperature, pressure, luminosity measurements [1], [2], [3] . Applications aimed at this domain are no longer restricted to the mundane, temperature and humidity sensing. In fact researchers working on sensor networks strive to conjure up diverse [18] applications such as intrusion detection systems, voice based reporting systems, ecological monitoring [20], [35] structural monitoring [36] and many more. Applications such as these demand an ever greater level of processing capability from sensor nodes which are deployed in the field. Cepstrum calculation for resonance analysis, FFT, harmonic filtering and extraction, image masking and processing all require a significant level of processing capability to be demonstrated by the Micro Controller Unit (MCU) on the small power efficient nodes. Current implementations of such platforms, mostly utilize 8-bit MCU's which are unsuitable for the job. They simply do not possess enough processing capability to satisfy the compute intensive applications which are now being hived onto this domain. Through extensive experimentation and benchmarking we have developed a sensor co-processing (Co-S) architecture integrated with an System on Chip based, RISE (RIverside Sensors) [30] host platform for higher performance sensing needs which simultaneously satisfies the constraints of low power consumption, high computational capability, high capacity onboard storage and a small form factor.

Sensing and reporting architectures developed using traditional sensor devices have been built along the lines of the "Sense and Send" paradigm of transmitting data generated by events as and when they are detected, and networks based on these architectures have been in use for a while now [15], [16], [17], [19], [21]. Quite obviously the architecture of this class of sensor nodes [33] can only reflect the capabilities, which suffice for such a model. During normal conditions, the sensory data remains predictable with gradual changes. Therefore percolation of each and every event through the network, as and when it is sensed is expensive in terms of energy depletion not only at the node sensing the event but also at the nodes, which ferry this information through the network. The "Sense and Store" paradigm which pivots on storage of sensed events, unless absolutely necessary to transmit, necessitates the case for a high capacity power efficient on-board storage architecture, to be employed for logging sensed events continuously. It is in this regard that we exploit advances in low power, ultra high capacity non-volatile storage devices, thus paving the way for integration of widely available, cheap and power efficient flash memory

[1] Research supported by NSF ITR grant #0330481.

storage devices, namely the SD-Cards, Compact Flash, XD-Cards [9],[10],[11],[26],[27].

The remainder of the paper is outlined as follows. Section II, specifies the motivation for this research effort, while section III details out the background work. Section IV lays the foundation of "Sense and Store" mechanism by detailing out low power flash memory technologies followed by Section V, on query processing and local access methods. Section VI and VII lays out the hardware details associated with the RISE platform architecture, and the Co-S architecture. This is followed up with a nuts and bolts description of the final integrated platform in section VIII. Experimental results which prove the efficacy of our integrated architecture are described in section IX. Finally rounded up with sections X, XI and XII are the implications of our "Sense and Store" approach, conclusion and future work.

II. MOTIVATION

Our work is motivated by the requirements of the Bio-Complexity and the James Reserve Projects at the Center of Conservation Biology (CCB) at UC Riverside. CCB is working towards the conservation and restoration of species and ecosystems by collecting and evaluating scientific information. The bio-complexity project is designed to develop the kinds of instruments that can monitor the soil environment directly in environments where factors like high humidity and precipitation will be a challenge for the sensors, rather than in laboratory recreations. One of the goals is to improve understanding of the spatial and temporal processes that control soil carbon sequestration in a tropical seasonal forest and the role of soil micro-organisms. The objectives in particular are to study soil carbon in a fire chronosequence to evaluate on ecological restoration experiment in terms of carbon and to integrate spatially and temporally the information from the sensor arrays with ecosystem scale measurements (e.g. root biomass, litter, soil carbon). Additionally voice signature based recognition mechanisms need to be implemented on the sensor platforms for habitat monitoring, enabling identification of species of birds using certain distinguishing features possible with frequency domain analysis of their native call patterns (songs).

Our objectives from the ground up are to reduce power consumption, maintain software compatibility vis-à-vis TinyOS [31] and simultaneously broaden the spectrum of applications of compact sensor systems. In keeping with our goals, our host architecture employs a monolithic SoC device viz. ChipCon CC1010, [23] which includes a power optimized 8051 core, radio, 3 ADCs (Analog to Digital Converter), 2KB SRAM (Static Random Access Memory), 32 KB on-chip-flash, 2 UARTs (Universal Asynchronous Receiver Transmitter), SPI (Serial Peripheral Interface) bus, all onto a single SoC architecture running TinyOS networking stack and an interface layer with the Co-S. On one hand this simplifies hardware design due to integration of all the components onto a single chip, eliminating a complex interface, while on the other hand overall system power consumption is reduced due to tightly integrated peripherals on the chip. Since off chip

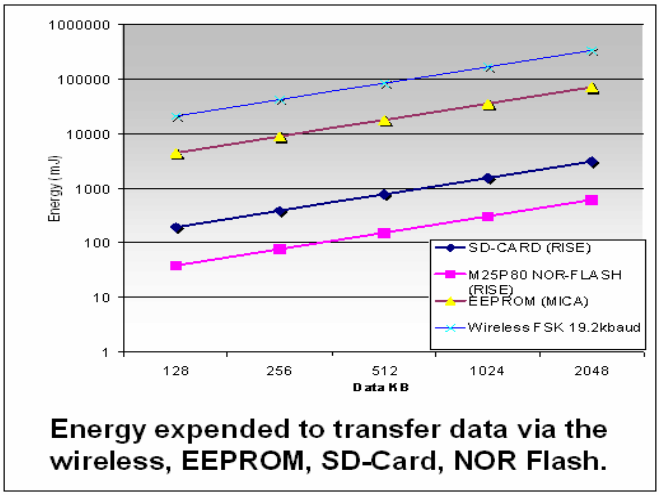

Figure 1. A comparison of the amount of energy expended to transfer data vs local storage. It makes sense to only the relevant, non-redundant sensory information.

peripherals entail individual *Printed Circuit Board (PCB)* area, voltage drop through the longer PCB traces, leakage and quiescent operation currents, our single chip host architecture is effectively power advantaged over discrete systems. Our architecture differs from existing platforms not only in terms of its computing power, flexibility, level of on-chip component integration but also, and quite significantly in the amount of on-board storage memory that it can provide, and an added software paradigm of "Sense and Store" which manages humongous amounts of raw data from the sensing hardware in-situ, before transmitting relevant parts of it efficiently to the base station. The extent of potential savings in energy brought about by employing the "Sense and Store" paradigm versus the "Sense and Send" is illustrated in Figure 1. Our "Sense and Store" paradigm stores the data onto the on-board memory, instead of naively passing on each and every piece of raw data through the hierarchical structure of a sensor network. This new approach has been demonstrated to be a substantial improvement over the Sense and Send architecture [30].

The moot question that needs to be answered now is whether the integration of components on chip, along with employing latest hardware and software paradigm is truly beneficial, and how could the underlying benefits be quantified. Thus to answer this question we have quantitatively compared our platform with various other sensing platforms including the crossbow MICA [4].

A quick calculation of the power consumption of our platform reveals orders of enhancement in power efficiency, which is obtained when a SD-Card is integrated with the sensor platform instead of EEPROMs (MICA). Assuming that 100KB worth of data needs to be gathered by the sensor during a particular time interval, a realistic figure for temperature and CO_2 sensors. In order to store the data on the SD-Card, we measured, in real-time, the overall energy consumed to be a miniscule 245.6mJ, while storing the same amount of data on the EEPROM available on the MICA would entail consumption of 2450mJ [24], [25], and transmitting it via the wireless interface, assuming no errors,

consumes 16,473mJ. The major contributing factor towards the lower energy consumption of the SD-Card is the faster data transfer rate on the SPI Bus (80KB/s) [26], [27], with respect to the EEPROM (1.6KB/s), or the wireless transmitter (1.92KB/s). This simple experiment highlights the advantages of utilizing better storage solutions along with intelligent data management techniques which is one of the compelling motivations discussed in this paper, vis-à-vis the design of sensor architectures that can handle copious amounts of data, store and process them for mining intelligent patterns within. Therefore newer generation of sensors, can thus afford the luxury of storing vast amounts of data, (Gigabyte scale), on board, and in employing these devices lies the crux of the "Sense and Store" paradigm.

Another quick comparison of the bill of materials of the RISE and Co-S platform with MICA highlights the benefits of tighter integration in our architecture. As can be seen in Table I, the CC1010 with similar capabilities as MICA, uses just one integrated SoC, while the MICA utilizes eight IC (Integrated Circuit) devices to obtain the same functionality. Similarly the number of discrete components is more than twice that of the RISE platform, thus resulting in direct improvement in power efficiency, as well as allowing for smaller and simpler form factors, all in all, a simpler system with reduced developmental effort. Even from the support point of view, a single chip manufacturer is involved as compared to a handful of them when a tightly integrated solution is compared to a loosely integrated one. In the same vein, our Co-S platform comprises of a single tightly integrated MCU chip with all necessary functionality available on the same silicon. A detailed feature wise comparison of several commercially available sensor platforms is presented in Table II.

TABLE I. COMPARISON OF ESSENTIAL INTEGRATED COMPONENTS

Integrated Component	**RISE (host)**	**MICA**	**CO-S Board**
SoC	1	0	1
Processor	SoC	2	SoC
Radio	SoC	1	Host
High Capacity Flash Memory	1	0	Host
Buffer	SoC	1	SoC
Onboard Sensor	1	1	0
Total	**3**	**5**	1

We provide a concise view of the significant amount of research efforts directed towards this area, followed by a detailed description of the RISE and Co-S platform and various comparisons with other existing architectures. Our experimental results demonstrate how a synergy between tightly integrated hardware Chipcon (CC1010 SoC), Renesas (M16C/28) along with efficient data management ("Sense and Store") can lead to massive savings in terms of energy for each sensor node, and development time and effort, thus bolstering the motivation for this novel perspective.

We exploit the Renesas M16C/28 platform [32] by offloading onto it computationally intensive task of frequency domain analysis, i.e. calculation of Fast Fourier Transform (FFT). This endows the RISE – Co-S platform with a dual pronged advantage, the first being, higher throughput along with low power consumption, given the use of an optimized M16C/28 architecture for a compute intensive task. The second being the unique ability to maintain network connectivity in the face of severe power depletion of the Co-S module, which can be shut down separately on-the-fly by the host controller. The host module can continue to communicate with the rest of the sensor network even in the face of this extreme situation. This unique ability to allow the Co-S to shut down and yet maintain network connectivity is critical from the point of view of routing updates which would lead to a flood of update messages in the network if a particular node were to withdraw operation due to power depletion at the sensor module. This architecture provides an inimitable warranty, which assures retention of data in the face of complete node failure, this is achieved by logging sensed events onto the SD-Card hooked onto the Co-S. Even though a sensor module may fail, data logged onto the SD-Card can still be queried and mined for required patterns, providing a significant layer of reliability and data retention in the network as a whole.

TABLE II. COMPARISON OF VARIOUS SENSOR ARCHITECTURES

Platform	**(MCU) MHz**	**MCU, Active Current (mA)**	**On Chip RAM (KB)**	**On Chip Flash (KB)**	**Aux. Memory**
RISE	(8051 core) 24	14.8	2	32	Upto 1 GB
M16C	(M16C) 20	16	8	96	Upto 1 GB
MICA	(ATMega 128L) 16	8	4	128	512 KB
TELOS	(TI MSP430) 8	0. 56	2	60	512 KB
EcoNode	(8051 core) 16	3	4	32 EEPROM	NA
iBADGE	(ATMega 103L) 4	5. 5	4	128	Optional 64KB
RENE	(AT90s8535) 4	6.4	0. 512	128	NA

III. BACKGROUND WORK

Wireless senor networks are perhaps one of the behooved technologies, which enable a truly ubiquitous view [1], [12], [13], [22] of the environment we live in, seamlessly integrating themselves with the ambience surrounding them. Power constraints placed on these small sensors lends a new dimension towards development and management of such small factor systems [19]. Sensor nodes with on-board memories need to optimally communicate and respond to queries, directed at them. In this respect the concept of SRTs (Semantic Routing Trees) [14], [15] is critical. It allows every node to determine whether a particular query over a chosen attribute should be forwarded to any of its children or not. Methodologies which deal with "pushing" this data through the sensor network [17], [19] consuming least amounts of processing and transmission power are critical. Development of these mechanisms has been a favorite muse for researchers in this area for some time now. Specific sensor architectures, such as Mica, iBadge, Rene and others as mentioned previously, have been well detailed on their various research group pages. Listing out their hardware parameters as well as the programming methodology used to configure them.

Furthermore, the "Sense and Store" paradigm has been ably described in [30]. Some of the major sensors, which have been made part of the RISE platform, are the Vaisala GMT 220 CO_2 sensor [3], temperature sensors, audio sensors, and in the work are the humidity and Carbon Monoxide sensors, thus entailing true-outdoor sensing in the field instead of simulated laboratory conditions. The requirements for this deployment environment inherently imply significant amounts of data to be logged for processing and analysis thereby necessitating the presence of a significant amount of storage memory on the sensor itself.

IV. Flash Memories and RISE Storage Board

A. *Flash Memories*

Flash Memory is the most prevalent storage media used in current sensor systems because of its many advantages including:

a) Non-volatile storage memory.

b) Fast read access and power efficiency.

c) Simple cell architecture, which allows for easy and economical production.

d) Commercial off the shelf availbaility in ruggedized packaging.

These characteristics establish flash memory as an ideal storage media for mobile and wireless devices. Flash memory allows multiple memory locations to be erased or written in one programming operation and holds its content without maintaining a power supply. There are two different types of flash memory, NOR and NAND, which are named according to the logic gate of their respective storage cell. The number of write cycles at a particular flash memory cell is typically 100,000 after which the cell may not reliably store data, while infinite amount of read cycles are supported.

NOR Flash memories are byte addressable memories and were the first kind to be developed in 1988. Their size ranges from 1MB upto 16MB. Due to certain beneficial features these memories are most suitable for code and parameter storage. NOR flash has faster access time (i.e. 200ns) than NAND (50-80µs) but lacks in all other characteristics such as density, power efficiency, erase speed and others. Reading and Writing to NOR Flash memories is simple because a random byte may be read or written. In order to reprogram or erase a byte in the NOR flash memory, the sector storing the byte need to be erased and re-programmed, thus erasing or rewriting is a complicated operation as far as NOR flash memories are concerned. Certain newer generation of NOR flash memories (M45PE80) provide page re-programmability thus leading to smaller granularity erase domains. We have utilised the ST microelectronics M45PE80 NOR flash memory. The memory chip offers 1 Megabyte of NOR Flash space, with a erase page size of 256 bytes, most suitable for storing index data structures and certain pre-defined query results.

SD-Cards [26], [27], (Secure Digital Cards) are postage stamp sized (24mmX32mmX2.1mm) *COTS (Commercial Off The Shelf)* non-volatile flash memory storage devices featuring upto 2 Gigabyte of storage space. SD-Cards utilize the NAND flash memory, which has some distinct characteristics summarized as follows:

a) The minimum quanta of write is a block viz. 512 bytes.

b) Writing to a block requires that the block is already deleted.

c) An erase domain is generally a sector 16KB to 32KB.

d) Memory technology driver in the form of in-built controllers take care of the NAND flash memory management.

e) They consume minute amounts of energy while storing and retrieving data (1.5uJ, 0.05uJ per byte respectively), thus making them highly suitable for integration with sensor platforms.

The humongous amount of onboard flash storage is most suitable for long term data retention, as well as data sampled at fast sampling rates. Storing data generated from such sensors as audio circuits, necessitate high-capacity storage on the sensor platform. The energy required for the transmitting one byte is roughly equivalent to executing 688 CPU instructions, and the cost of writing to the flash is less than 10% of the energy required to transmit the same amount of data [29], thus making local storage and processing highly desirable. The Sense-and-Store paradigm pivots on this very observation. Since the wireless interface is unable to keep up with the high sampling rate of the sensor it is but logical to store the data onto onboard storage, and calculate required features in-situ. To illustrate the storage capabilities of the SD-Card we may store more three years worth of sensed data, continuously sampled at a rate of ten bytes per second, which more than suffices the demands of a wide spectrum of sensor applications. A detailed schematic of the SD-Card and the NOR flash memory is presented in Figure 2.

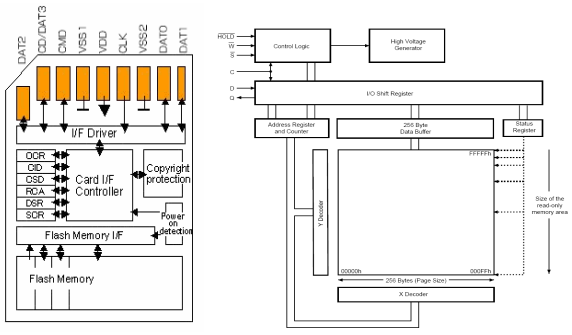

Figure 2. a) A view of the internal architecture of the SD-Card (http://www.sdcard.com) and b) the NOR flash chip.

The slim and compact design of SD-Cards make them an ideal removable storage solution for designs ranging from digital cameras, PDAs, cellular phones, and sensor platforms. A plethora of standards for such cost-efficient and easily available commercially available off-the-shelf devices range

from Compact Flash, to the SD, XD, MM Cards and others. Even though these devices may serve the same purpose and to the unsuspecting user may seem no different from each other, there are thorny issues bristling under the skin when it comes to matching one standard against the other. For example Compact Flash cards communicate through a parallel bus, unsuitable for simple microcontrollers while the XD card is devoid of an internal controller. SD-Cards however support the popular SPI bus interface, which it inherits from an earlier generation of Multi-Media Cards. This feature is a definite thumbs up in terms of the man-hours expended for interfacing the card with the RISE / Co-S platform. The choice of the SD-Card as the on-board storage device is made amenable by its cost efficiency of 6-10 cents per MB, making it an attractive proposition.

B. *RISE Storage Board*

The RISE Storage board (see Figure 3) integrates a SD-Card and a NOR Flash memory chip onto a circuit board. The prime motive of supporting two different variants of flash memories is as follows. NAND flash is most suitable for bulk storage of data. Hence sensed data is stored in the NAND flash memory, which includes periodic temperature Carbon-dioxide and audio samples. After due processing, the final query result consumes a fraction of the amount of the raw data. This data can be easily integrated with the index, for fast query retrieval. Thus the NOR flash memory serves as the storage media for the index and pre-defined query results such as the maximum, minimum, average, harmonics (audio data), and certain top-k values. Since the NOR Flash allows for random read or writes and smaller granularity erase we can easily modify the index and hashing structures in this memory, as and when needed.

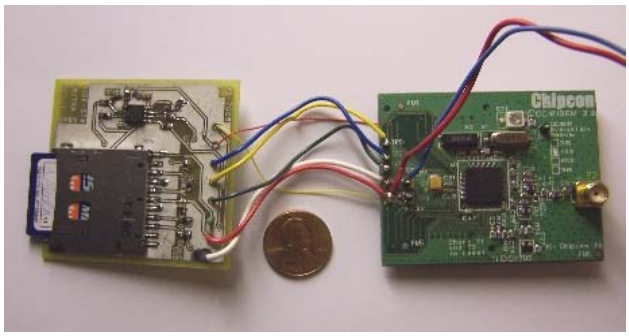

Figure 3. The RISE sensor node interfaced to the Gigabyte Scale Storage Board with indexing capabilities.

The data transfer mechanism to and from the Co-S platform is the SPI bus. Dedicated I/O pins from the Co-S platform (Clock, Data IN, Data OUT, Clock Select) are used for the memory interface. The flash memories are powered off the Co-S platform. The microcontroller transfers data using the SPI protocol. Each write transaction to the SD-Card involves writing a 512 byte block of data, while reads may be arbitrarily sized up to a maximum of 512 bytes. One fine detail to consider while writing and subsequently reading logged data from the sensor is the following, in some applications each triggered event may not generate enough information to fill up the 512 byte block completely, zero padding must be employed to take care of this situation. However, this would entail energy being consumed for pushing in useless information into the storage device, thereby to alleviate this malady we buffer readings in a buffer allocated in the Co-S SRAM. A full buffer initiates a data flush from the buffer on to the SD-Card. Figure 4 depicts the number of hours for which a sensor having an integrated on-board storage device can keep logging data onto its SD-Card.

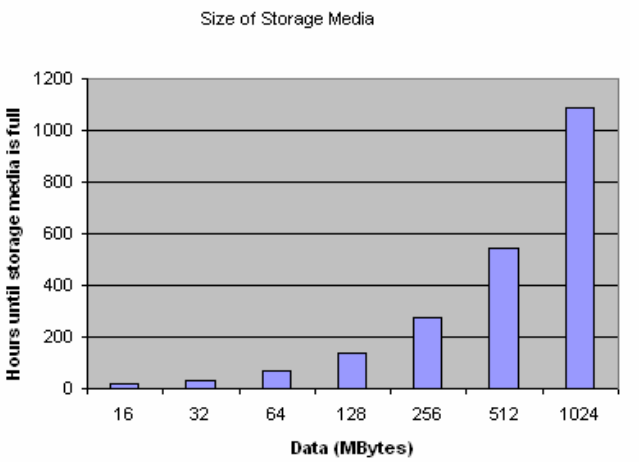

Figure 4. The amount of time for which a RISE-CoS node can continuously log data onto a SD-Card at 10 Bytes/Sec sampling rate. The maximum amount of time for a 1 GB SD-Card is more than 1000 Hours at a stretch

V. QUERY PROCESSING AND LOCAL ACCESS METHODS

In our framework it is important to access data records stored locally at the flash card with a minimum overhead. Consider for example the following query: ``*Find the time intervals when the temperature was 95F*''. Evaluating such a query without an index would require each sensor to perform a sequential scan over all pages stored on the external flash cards. Although reading pages on flash memory is a cheap operation, an external flash card can feature a very large number of pages. For example a 256MB SD-Card features 500,000 pages. Therefore we propose the deployment of indexes (access methods) directly on the sensor node. Specifically, while a node senses data from its environment (i.e. data records), it also creates index entries that point to the respective data stored on the flash card. When a node needs to evaluate some query, it uses the index records to quickly locate the desired data. Since the number of index records might be potentially very large, these are stored on the external flash card as well. Although indexing over magnetic disks and RAM is a well studied problem in the database community the low energy budget of sensor nodes along with the unique read, write, delete and wear constraints of flash memory introduce many new challenges.

We have designed and implemented MicroHash, which is an efficient access method that provides *random* and *sorted* access to records stored on the flash medium. MicroHash serves as a primitive function for the efficient execution of a wide spectrum of queries. Pages on the flash card are organized as a *heap file*, which is naturally ordered by time. Note that a flash card can only hold up to certain number of

pages and hence the available memory is organized as a circular array, in which the newest record replaces the oldest record if the memory becomes full. The two primitive operations we consider in our design are:

a*) Random Access by Value:* An example of such operation is to locally load the records that have a temperature of 95F. In order to fetch records by their value we have implemented the *MicroHash* index (see Figure 5). Our index uses a swap directory that gracefully keeps in memory the directory buckets with the highest hit ratio. We use a static index as opposed to a dynamic hashing index, such as *extendible* or *linear hashing*, because the latter structures are considerably more power demanding (i.e. due to page splits during insertions).

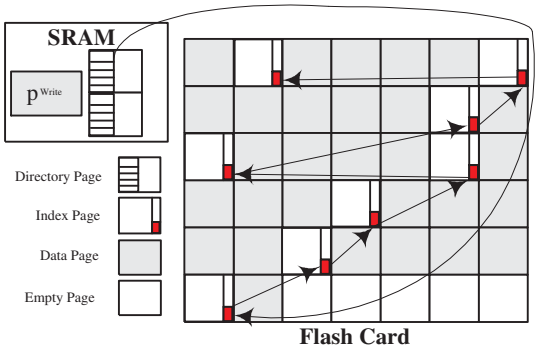

Figure 5. *The MicroHash Index:* While a node senses data from its environment it also creates index entries that point to the respective data stored on the flash card. When a node needs to evaluate some query, it uses the index records to quickly locate the desired data.

b) Sorted Access by Value: An example of such operation is to locally load the records that have a temperature between 94F-96F. An important observation is that sensor readings are numeric readings in a discrete range (for example the temperature is between -40F and 250F).

In order to enable such range queries, we currently use an extension of our random-access index in which we query every discrete value in the requested range. However, we are also developing a simple B+ tree index, which is a minimalistic version of its counterpart found in a real database system. It consists of a small number of non-leaf index pages which provide pointers to the leaf pages. In our current design, we keep a small number of highly used non-leaf index pages (such as the root) in main memory.

Our discussion so far assumes that pages are read from the flash media on a page-by-page basis (usually 512B per page). When pages are not fully occupied, such as index pages, then a lot of empty bytes (padding) is transferred from the flash card to memory. In order to alleviate this burden, we exploit the fact that reading from flash can be performed at any granularity (i.e. as small as a single byte). More specifically, we propose the deployment of a *Two-Phase Page Read* in which the MCU reads a fixed header from flash in the first phase, and then reads the exact amount of bytes in the next phase.

We experimentally evaluated the performance of two-phase reads versus single phase reads using the ABC sensor node. Note that in order to initiate a read over the flash card we need to initiate a 9-byte SD-Transaction (1B sync, 6B SD Instruction, 1B sync, 1B). Figure 6 shows that the two-phase reading is almost always superior to its single phase counterpart with an exception of pages which are adequately full (i.e. >90%). Minimizing the page reading time significantly minimizes energy consumption.

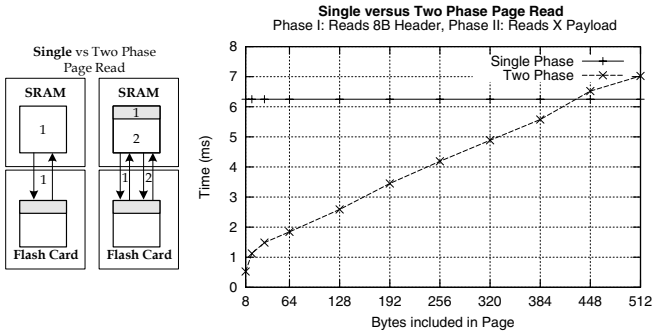

Figure 6. *a) Illustration of the Single Phase and Two Phase Page Read strategies. b) Performance comparison of the strategies on RISE.*

VI. RISE Platform (CC1010 SoC)

The RISE platform entails the use of commercial off-the-shelf components and is designed from the bottom up in a modular fashion. It entails the use of a National Semiconductors (www.national.com) LM61 temperature sensor, a Vaisala GMT 220 Carbon Dioxide sensor [3] to sense environmental data. The RISE platform based on the Chipcon CC 1010, a compact 12mm by 12mm and only 1.2mm wide is a feature packed SoC making it an ideal candidate for use in low power wireless embedded device applications. The CC 1010 is a true single chip UHF (Ultra High Frequency) transceiver with an integrated high performance 8051 microcontroller with 32 KB of flash programmable memory. The CC 1010 unlike other microcontroller and sensor nodes needs hardly any external integration to make it an effective sensor node. The RISE platform in effect has the benefit of being built upon a high-performance and energy-optimized 8051-core microcontroller that typically gives 2.5 times the performance of a standard 8051. Idle and sleep modes for reduced power consumption are fully supported. The system can wake up from an interrupt or when the ADC (Analog to Digital Converter) input exceeds a particular defined value. In addition to this it has a low current consuming fully integrated UHF RF (Radio Frequency) transceiver with programmable frequency and output power and low current consumption. It also supports frequency hopping protocols by virtue of a fast settling time of the PLL (Phase Locked Loop). It employs Manchester codec in hardware and RSSI (Received Signal Strength Indicator) output, which can be sampled by an on-chip ADC. Also it wields 32KB of nonvolatile flash memory with programmable read and write locks for software security along with a 2k+128byte block of SRAM. Peripheral features include three channel, 10 bit ADCs, programmable watchdog timers, real time clock with 32KHz crystal oscillator, two programmable serial UARTS, master SPI interface, two counters and pulse

width modulators, 26 configurable general purpose I/O pins and random bit generators along with DES encryption and decryption in hardware [23]. Since the ADCs on the Co-S platform is utilized for sampling sensed data, the ADCs on the RISE platform are used for measurement of diagnostics data, viz. battery voltage level, and radio signal strength. The battery voltage level is a useful gauge of the remaining system lifetime, while the Radio Signal strength is utilized to detect other transmitting nodes in the vicinity of the sensor, useful for collision avoidance in the wireless network. Figure 7 depicts the RISE node interfaced with the Vaisala GMT 220 CO_2 sensor and Table III provides a brief overview of the integrated components of the RISE platform.

VII. Co-S Platform (Renesas M16C/30280AFHP)

Our co-processing system design consists of a tightly integrated high performance 16bit MCU (Renesas M16C/30280AFHP) with most functionality available on-chip, such as integer multipliers, ADCs (Analog to Digital Converter), USARTs (Universal Synchronous/Asynchronous Receiver Transmitter). Many Lookup Table (LUT) based operations such as FFT and FIR filtering benefit from the large amount of available on-chip memory (8KB SRAM, 96KB Flash) of the M16C/28 platform. Communication between the host platform and the Co-S module is achieved through high speed USARTs.

The M16C/28 platform is employed for execution of offloaded tasks, viz. data sensing, storage, management and computationally intensive operation of calculating FFT. The suitability of employing the Renesas M16C/30280AFHP Co-S platform is demonstrated by its efficient architecture, which is indicted by the performance metrics obtained by running the 128point FFT benchmark. The higher throughput, of the optimized M16C/28 architecture enables faster data processing upto 24 times that of the Atmel AVR architecture (MICA) or the host platform (CC1010). Moreover the low power consumption of the Co-S module results in savings upto 900uJ per 128 point FFT operation when compared to the AVR / 8051 based designs, as demonstrated in Table VI. A brief overview of the features is provided in Table IV.

VIII. Integrated Platform

We exploit the M16C/28 platform, by offloading onto it sensing, data storage, and computationally intensive tasks, i.e. FFT. Since sense-and-store entails small local processing on sensed data for future retrieval, thus updates of local minimum, maximum, average, and bookkeeping of sorted lists and indexes, are calculated periodically by the Co-S and stored in the SD-Card. Queries are received by the host platform over the wireless link, and are transferred to the Co-S over the UART (Figure 8). The Co-S evaluates the query, which entails copying relevant data from the SD-Card to the onboard buffer and streaming it back to the host platform over the UART. The host sends back the retrieved data over the Tiny-OS networking stack, relaying it via the wireless interface.

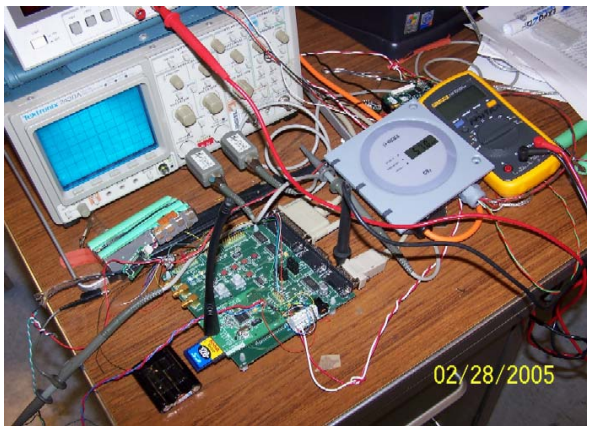

Figure 7. The RISE node interfaced with the Vaisala GMT 220 CO_2 sensor.

TABLE III. Features on the RISE Platform

Component	*Capability*
The MCU	
Processor	24 MHz 8051 core
On-chip flash	32 KB
Current Consumptions (Active, Idle, Power Down) at 14.7456 MHz	14.8 mA, 8.2 mA, 0.2 µA
The Radio	
On-chip Radio	300-1000MHz low power RF transceiver
RF transmission rate	76.8 kbits/s
Range	Upto 250m at 868/915 MHz
Current Consumptions for RF transceiver (Receive, Transmit at 10dBm)	11.9 mA, 26.6 mA
The SPI bus	
Datarate	Programmable upto 3 MHz
The data block length	512 bytes

TABLE IV. Features on the Co-S Platform

Component	*Capability*
The MCU	
Processor	20MhZ M16C
On-Chip flash	96 KB
Current Consumption (Active @ 20MHz, Power Down @ 32KHz)	16 mA, 0.7 µA
SRAM	8 KB
ADC (10 bits)	24 channels
Serial I/O	
2 channels (UART0, UART1)	I^2C, SPI, clock synchrounous, UART

Our current sensing application utilizes the 10 bit ADCs on the Co-S for sampling temperature, Carbon-Dioxide, and audio samples. The sampling rate of temperature and Carbon-Dioxide is ten samples per minute. At the elapse of a minute, these samples are averaged, time-stamped, thereafter stored onto the on-chip buffer. A twenty-four hour indexing scheme is used to index this sample. The index is stored in an SRAM buffer (2KB) and is committed to the SD-Card past elapse of twenty-four hours. The sound samples are acquired through the ADCs at a rate of 10KHz, for 5 seconds, once every minute. An FFT operation on 128 samples accumulated in the buffer of the Co-S takes approximately 1ms. Thereby iterative calculation of 128 point FFT on five seconds worth sampled data takes less than one second of processing time. The FFT as computed is stored on the SD-Card along with timestamps. Since the data remains stored in the non-volatile memory of the flash card, it is also possible to retrieve all the stored data for in-depth data mining operations.

TABLE V. SENSORS INTERFACED WITH RISE-CO-S

Sensor Type	Sensed Parameter	Sampling Rate
Vaisala Carbocap GMT 220	Carbon Dioxide	10 Samples / min.
National Semi. LM 61	Temperature	10 Samples / min.
Microphone Ckt.	Audio	10KHz

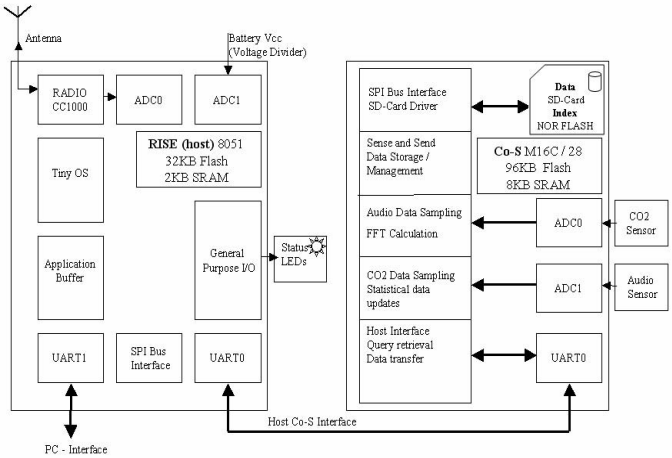

Figure 8. The RISE platform interfaced with the Co-S

IX. EVALUATION

The hardware setup revolves around the RISE and the Co-S system, with which is interfaced a 256 Megabyte SD-Card. The Vaisala Carbocap GMT 220 (www.vaisala.com) carbon dioxide sensor, the LM61 temperature sensor and a micro-phone interface circuit were employed to obtain real-time sensor data for analysis (see Table V). The setup was deployed on the premises of the RISE lab. We utilized an HP E36308 precision power supply to regulate supply voltage and FLUKE DM112 True RMS digital multimeter for accurate current measurements. The operating voltage of CC1010 was set at 3.0V while the M16C/28 was running at 2.7V. Tiny OS was loaded onto the RISE platform (CC1010) while native C code was compiled using the Renesas *HEW (High Performance Embedded Workshop)* environment for sensing, processing, and data communication with the RISE.

TABLE VI. 128 POINT FFT BENCHMARK RESULTS

Platform	Micro Controller	Clock MHz	Time ms	Total Energy uJ[a]
MICA [34] (3.0V)	Atmel AVR Mega 128L	8	14.5	934
Stargate [34] (3.3V)	Intel PXA 255	400	0.095	45.8
RISE (host) (3.0 V) / 20mA	Chipcon CC1010	14	11	660
RISE (CoS) (2.7V) / 15mA	Renesas M16C/28	20	0.892	36

aWirelessly Transmitting 256 Bytes consumes 41000 uJ at 19.2KBaud FSK

A. Fast Fourier Transform

Table VI illustrates the performance of three popular sensing platforms along with the Co-S while a 128-point FFT benchmark loop with 16 bit data and 32-bit output. Since it may not be possible to measure system parameters such as the current drawn, for a small interval of time i.e. a few milliseconds, thus we iteratively executed the operation ten-thousand times and report the total time divided by the number of iterations. It is clear from Table VI that the M16C/28 architecture is extremely energy efficient, consuming 18 times less energy than the CC1010 (8051), and about 6 uJ less per 128 point operation than the Stargate (INTEL PXA 255 XScale) platform. Thus utilizing the highly efficient Co-S for FFT calculation, we may achieve improved power efficiency in sensor systems. Another interesting point to be noted is the 29 percent improved power efficiency of the CC1010 SoC vis-à-vis Atmel AVR for FFT calculations.

Figure 9, displays the energy expended in order to sample, process (FFT), store audio streams and transmit the occurrence of specific harmonics in the given samples. The number of samples processed until the actual transmission vary from 64 to more than 65536. The result of the query is a packet of size 10 bytes for each 64 sample points (refer to Figure 10). Thus for each occurrence of the harmonic, amount of data that needs to be sent across the network, is only 10 bytes (for e.g. the values 1852Hz and 3704Hz vis-à-vis the bird call depicted in Figure 10), while the raw FFT data measures 512 Bytes for a 128-point FFT operation. Since ecological audio samples such as bird songs generally demonstrate certain harmonics, the existence of these harmonics suffices as the query, rather than the raw audio data. Moreover during periods of silence the sensed data do not exhibit these harmonics and percolation of such data across the network would result in un-necessary consumption of energy resources.

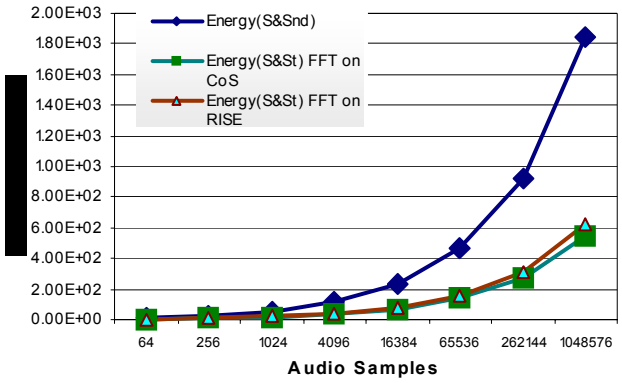

Figure 9. Graph comparing the energy expended for audio sampling, FFT, Storage, and Transmission of a query (particular harmonics) to sending raw audio data over the wireless link.

B. *Flash Memory*

Table VII depicts the power consumption of the SD-Card and the NOR flash memory when interfaced to the RISE platform using the SPI Bus.

1) *SD-Card*

Pseudo Random data was generated on the platform and then written onto the SD-Card.

TABLE VII. POWER CONSUMPTION OF SD-CARD AND NOR FLASH

Operation (SD-Card)	**Data Size**	**Time sec.**	**Data Rate KB/s**	**Total Energy mJoules**	**Energy / Byte uJ/B**
Read[1]	1MB	13.0	76.9	50.19	0.05
Write[1]	1MB	12.5	80.0	1513.9	1.51
Stride[3] Write (8 blocks)	1MB	13.5	6896	1915.65	1.92
Erase[2]	100MB	14.5	74	2727.45	0.027

Operation (M45PE80) (M25P80)	**Data Size**	**Time Sec**	**Data Rate KB/s**	**Total Energy mJoules**	**Energy / Byte uJ/B**
Read[1]	1MB	19	55.2	62.7	0.06
Write[1]	1MB	23	45.6	113.85	0.108
Erase[2]	1MB	10.2	102.8	218.78	0.21

[1]Read / Write are managed by the host. Hence Read / Write speed is limited by the wire-speed connecting the host to the flash memory until the actual read / write speed of the given memory is reached [2]Erase is handled by the internal controller of the memory, and is not managed by the host. [3]Stride-Write utilizes 8*512 = 4096 byte strides to emulate random writes onto the card, since each sector of an SD-CARD consists of 32 block = 16KB. Observed fact is that using Stride-Writes, the contents of the sector are preserved even though data in individual block is re-written.

This data was then read / erased from the SD-Card. We first sequentially filled the complete SD-Card in order to ensure that all blocks contained data. Thereafter a Read operation of 1 Megabyte was executed, on sequential blocks. Since the data throughput of an SD-Card is much higher than that offered by the SPI Bus, therefore the speed of random as well as sequential reads remain the same while reading onto the RISE platform. Apparently the intelligent controller on the SD-Card executes a caching / buffering scheme which leads to very low power consumption while reading at a rate slower than the native speed of the flash memory device. While writing a similar amount to the SD-Card, we need to expend almost thirty times more energy. Moreover, since the underlying NAND architecture of the flash memory entails a complete erasure of a sector (32 blocks) before re-writing a block (512 bytes), the cost of re-writing to a single block involves the cost of:

a) Erasing the sector

b) Updating all the blocks other than the one being programmed

c) Re-writing the given block.

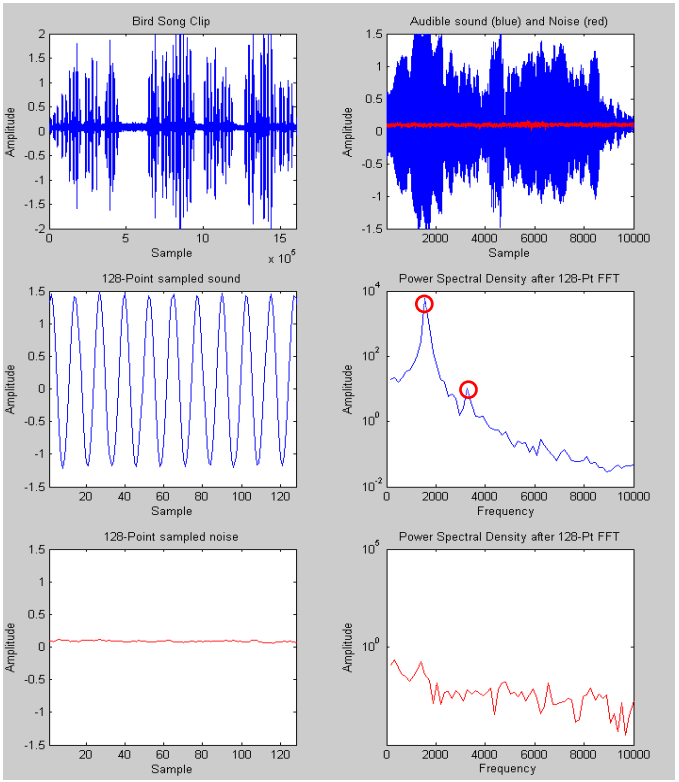

Figure 10. Graphical demonstration of our harmonic calculation mechanism. The occurrence of specific harmonics (red circles) are used for addressing queries and indexing, instead of the raw audio data.

This complicated architecture of NAND flash memory is handled completely by the SD-Cards' intelligent controller and remains invisible externally. This is aptly indicated by our stride write benchmark, which involves a stride of 4096 bytes between writes and consumes 27% more energy than sequential writes. Thus for power constrained systems, sequential writes to the SD-Card is the most efficient method. The SD-Card's internal controller handles erasing, once the starting and the ending block addresses are programmed onto it. An ACK signal indicates complete erasure of the given segment of the SD-Card. Since bulk erase is executed

internally at a very high speed, it is by far the most energy efficient of all given SD-Card operations.

2) *NOR Flash Memories M25P80, M45PE80*

We obtained two variants of available NOR flash memory viz. M25P80 and M45PE80, with a capacity of 8 Megabits. Although these memories support the random read and write characteristics of the NOR flash memories, their primary difference lies in their erasing and re-programming granularity. The M45PE80 offers a page level granularity of 256 bytes while reprogramming or erasing while the M25P80 offers only sector level granularity (64KB). The graph in Figure 11 depicts the difference in energies while erasing and reprogramming the two NOR flash memories as the size of the data increases. The M45PE80, due to it page level granularity offers lower power erase than the M25P80, until the erase data size reaches the sector size. Therefore the M45PE80 makes the most sense for deployment as the flash memory of choice where frequent smaller granularity modification or erase operation would be required while consuming the least amount of energy such as for storing index data structures. When the amount of data is an integral multiple of the sector size, both of these memories would entail equal amounts of power for erasure and re-programming while for all other cases the M45PE80 would be more energy efficient than its counterpart.

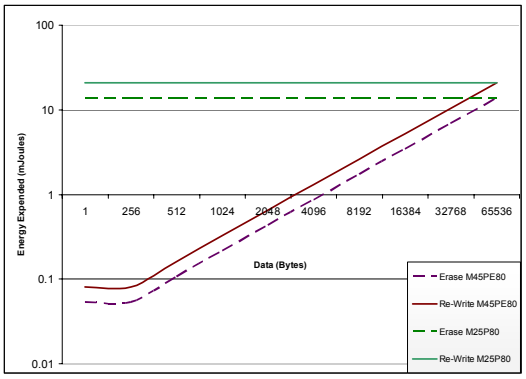

Figure 11. Graph depicting energy consumption when erasing / re-programming two variants of NOR flash memory. M45PE80 offers lower granularity programming and is most suitable for storing hashing and indexing data structures which require frequent modifications.

X. DISCUSSION

One of the first successful experiments involving environmental and habitat monitoring [10] using sensor nodes was conducted by University of California, Berkeley at the Great Duck Island, Maine. Various sensor motes fitted with light, temperature, barometric and acoustic sensors were utilized for the task. The sense and send paradigm although largely successful, resulted in a lifetime of almost seven months for the experiment.

Our sense and store paradigm can be implemented in such situations to advantage, and to increase the longevity of similar experiments by virtue of saving power and cutting down on communication costs, as most sensor data in such specific experiments is a slowly varying, redundant time series [10] [12]. Moreover if the types of queries are similar to the k-maximum / minimum, average, etc such queries could be most effectively handled by sense and store as the nodes may continuously calculate and store these values in their local memories. Exemplifying the data received at a typical daylight sensor, that for most parts of the day, data across the light sensor increases gradually until midday and starts decreasing slowly thereafter, and remains zero at night [12]. Hence this data may be tightly compressed and stored during the day, while also calculating certain statistically important values such as maximum, minimum, average, etc, that would be transmitted to the base station at night, effectively reducing communication traffic during the day. Similar variations can be expected for various other geo-climatic variations such as the temperature and pressure at a location. Similarly, the measurement of event longevity, such as infrared monitoring of animal dwellings [10] [12] involves long intervals of near constant data, followed by the end of the event, and is largely a binary phenomenon, i.e. either the event persists or has lapsed. Sense and send may not be justifiable, due to the requirement of communicating the data throughout the lifetime of the event, and there seldom lies any information for the duration of the event. On the other hand sense and store can locally store and process the complete details of the event while communicating only relevant details for its reconstruction (such as start, stop time). Data fidelity and power consumption are two opposing criteria for sensor systems and most power constrained sensor networks tend to balance between both for data transmission and storage.

Many sensed data such as photographic, acoustic, etc need to be down-sampled before being transmitted over the wireless channel, thus sacrificing the quality of the sensed data, possibly hindering future data mining. Local storage at the sensors thus helps to preserve the original data, which can then be manually retrieved and correlated at the end of the experiment. Another problem afflicting sensor networks is the inability of modeling the actual behavior and lifetime of the nodes in the field, particularly due to lack of original operational data / post mortem data [10]. Due to the availability of large local storage, strategic vital statistics and diagnostic data vis-à-vis the motes may be stored for post mortem analysis thus aiding our understanding of actual deployment. Tiered sensor networks have been proposed for various environmental monitoring deployments such as the habitat-sensing array for biocomplexity mapping, as proposed for James Reserve [6]. The platforms in the tiered network constituting the highest level of the hierarchy are relatively high performance computing devices communicating with the next in hierarchy, the tags (sensor nodes similar in capabilities to the RISE). Sense and Store enable the tags to locally store and manage data, as broadcasted from minute memory-less sensors (devoid of any receive circuitry), which just sense and send, owing to their utterly simplistic and constrained designs.

Overall the benefits of sense and store are perceptible in sensor networks deployed for monitoring time series data with predictable queries and where post mortem analysis and collection of data would be undertaken.

XI. CONCLUSION

We have developed a highly integrated sensing platform utilizing SoCs thus, resulting in an overall simplified design. Our design integrates a powerful co-processing SoC that effectively broadens the spectrum of applications suitable for small and low powered sensing systems. We have also maintained complete software / network compatibility by using the TinyOS on the host platform. Our research effort has been able to prove conclusively the efficacy of the "Sense and Store" paradigm over the "Sense and Send" methodology being employed by most conventional sensor deployments in today's scenario. We have demonstrated a performance improvement of seventy times in terms of power consumption following the "Sense and Store" approach. Furthermore, our Co-S architecture leads to a significant improvement in terms of computational ability, twenty four times more than the conventional low power sensors in use today. Co-S platform coupled with the on-board gigabyte scale data storage can now enable computation intensive applications such as FIR, FFT, Harmonic extraction, voice signature mapping and a plethora of similar demanding applications to be run on small and low power sensor systems. Large capacity SD-Cards hooked onto our platform allows accumulation of gigabyte scale sensed data onto the sensor itself. And coupled with the "Sense and Store" approach, is a win-win situation for sensor networks not only in terms of power savings but also in terms of the amount of data available for post-mortem purposes. Additionally our design sports the novel ability to maintain network connectivity in the face of power depletion by terminating all sensing operations, by shutting down the sensing module, thus enhancing the longevity of the network.

XII. FUTURE WORK

We are investigating mini-SD-Cards as the primary data storage memory. We are considering various available SRAM memories and 32 bit microcontroller for enabling efficient data management on the nodes. We are also considering implementing a new security paradigm, using on chip encryption schemes implemented within the SD-Card itself.

XIII. REFERENCES

[1] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., and Cayirci, E., ``A Survey on Sensor Networks**,** IEEE Communications Magazine," Vol. 40, No. 8, pp. 102-116, August 2002.
[2] http://www.campbellsci.com/sensors
[3] http://www.vaisala.com/
[4] http://www.xbow.com
[5] http://cegt201.bradley.edu/projects/proj2003/wisenet/
[6] http://ttdp.org/tpg/html/book/x313.htm
[7] http://www.moteiv.com/
[8] http://nesl.ee.ucla.edu/projects/ibadge/concept.htm
[9] http://www.sdcard.org/
[10] http://www.compactflash.org/
[11] http://www.livingroom.org.au/photolog/news/flash_memory_prices_fall.php
[12] Philippe Bonnet, Allan Beaufour, Mads Bondo Dydensborg and Martin Leopold, "Bluetooth-based sensor networks", ACM SIGMOD, 2003.
[13] J. Elson and D. Estrin, "Wireless Sensor Networks: A bridge to the Physical World", in Wireless Sensor Networks , Kluwer, 2004.
[14] Deepak Ganesan, Alberto Cerpa, Yan Yu, Wei Ye, Jerry Zhao, and Deborah Estrin, "Networking Issues in Sensor Networks", Journal of Parallel and Distributed Computing (JPDC), Special Issues on Fronteirs in Distributed Sensor Networks*, 2004.*
[15] Sam Madden, Michael J. Franklin, Joseph M. Hellerstein and Wei Hong, "The Design of an Acquisitional Query Processor for Sensor Networks", ACM SIGMOD *Conf*, San Diego, CA, June 2003.
[16] J. Deng, R. Han and S. Mishra, " A Robust and Light-Weight Routing Mechanism for Wireless Sensor Networks ", *1st Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS)*, 2004.
[17] Geoff Lyon, John Grundbäck, Cyril Brignone and Salil Pradhan, "Directed Information Harvesting"*, HP Labs Palo Alto, USA*
[18] Andrew Howard, Maja J. Mataric´, and Gaurav S. Sukhatme, "An Incremental Self-Deployment Algorithm for Mobile Sensor Networks," In *Autonomous Robots Special Issue on Intelligent Embedded Systems*, Vol. 13, No. 2, pp. 113-126, 2002.
[19] Q. Han, S. Mehrotra and N. Venkatasubramanian, "Energy Efficient Data Collection in Distributed Sensor Environments", The 24th IEEE International Conference on Distributed Computing Systems (ICDCS). March, 2004.
[20] A. Banerjee and D. Saha. " Live Bio-Sensor Nodes for Invisible Pervasive Computing". Advanced Autonomic Workshop HiPC, India, 2003
[21] Jamal N. Al-Karaki and Ahmed E. Kamal, "Routing Techniques in Wireless Sensor Networks", IEEE Wireless Comm., Dec. 2004.
[22] P. Zhang, C. Sadler, S. Lyon, and M. Martonosi, "Hardware Design Experiences in ZebraNet", Proceedings of SenSys 2004, November 2004.
[23] http://www.chipcon.com
[24]http://www.tinyos.net/nest/doc/tutorial/lesson7.html.
[25] Victor Shnayder, Mark Hempstead, Bor rong Chen, Geoff Werner Allen, and Matt Welsh, "Simulating the Power Consumption of Large-Scale Sensor Network Applications", SenSys'04, Baltimore, MD, November 2004.
[26] Secure Digital Memory Card Physical Layer Specification, Version 1.01.
[27] Secure Digital Card, Product Manual, Revision 1.7, Sandisk Corporation.
[28] Roundy, S., Steingart, D., Fréchette, L., Wright, P. K., Rabaey, J., "Power Sources for Wireless Networks", Proc. 1st European Workshop on Wireless Sensor Networks (EWSN '04), Berlin, Germany, 01/2004.
[29] A. Mitra, "Bit error analysis of new generation wireless transceivers", IEEE-ICCS 2002.
[30] Demetrios Zeinalipour-Yazti, Som Neema, Dimitrios Gunopulos, Vana Kalogeraki, Walid Najjar, "Data Acquisition in Sensor Networks with Large Memories", NetDB 2005.
[31] http://www.tinyos.net
[32] Renesas, M16C/28 Group Hardware Manual-rej09b0047_16c28hm.pdf
[33] J. Polastre, R. Szewczyk, C. Sharp and D. Culler, "The Mote Revolution: Low Power Wireless Sensor Network Devices", Hot Chips 2004.
[34] D. McIntire, "Energy benefits of 32 bit microprocessor wireless sensing system", Sensoria Corp., www.sensoria.com/pdf/Energy_Benefits_ of_32-bit_Microprocessor_Wireless_Sensing_Systems.pdf
[35] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, D. Culler, "An Analysis of a Large Scale Habitat Monitoring Application", In SenSys'04, Baltimore, Maryland, Nov 2004.
[36] N. Xu, S. Rangwala, K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan and D. Estrin, "A Wireless Sensor Network for Structural Monitoring", In Sensys'04, Baltimore, MD, 2004,