High Performance, Low Power Sensor Platforms Featuring Gigabyte Scale Storage

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

Abstract

Sensor applications have now touched onto the realms of real-time data processing involving algorithms as sophisticated as Fast Fourier Transform (FFT), Finite Impulse Response (FIR) filtering and Cepstrum. Moreover since typical sensor networks employ devices based on a simplistic microcontroller, it would be rather inefficient vis-à-vis utilization of energy resources to employ it for applications other than the most mundane "Sense and Send". Our research provides a vista to envision a powerful yet energy efficient sensor node architecture. To this end, we have developed a sensor co-processing (Co-S) architecture integrated with an System on Chip (SoC) based host platform for higher performance sensing needs, coupled with an interface integrating up to gigabyte scale energy efficient data storage system, which simultaneously satisfies the constraints of low power consumption and a small form factor. Our Co-S architecture consumes 24 times less energy than other prevalent uni-processor sensor node architectures while computing FFT. We also demonstrate significant energy savings up to 70 times, via our in-situ data storage and query evaluation mechanism viz. the "Sense and Store" approach.

1. Introduction

Embedded wireless sensor networks comprise of nodes that can process and communicate information to perform the tasks of sensing, and transmitting the sensed data to other nodes in the network. These sensors measure ambient conditions in the environment and then transform these measurements into signals that can be processed to reveal some characteristics about phenomena located in the region of interest.

A constant demand imposed on these platforms is support for broader sensing applications along with low power consumption, rapid deployment and small form factor [1]. 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].

Sensor applications [13] have now touched onto the realms of real-time data processing such as digital filtering, and algorithms as sophisticated as FFT and Cepstrum, required for measurement of resonance for stress analysis of rigid bodies and harmonics extraction from sound / voice samples. Such applications demand real time storage, filtering, frequency domain analysis, which are out of bounds of simple 8-bit Micro Controller Units (MCUs) due to their limited computational and storage capabilities. Therefore, there is a definite need to employ a higher performance computing architecture with large storage capacity, which can cater to such sensor applications. Thus after due experimentation and benchmarking we have developed a sensor co-processing (Co-S) architecture integrated with an System on Chip (SoC) based, RISE (RIverside Sensors) [17] 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], [19]. Quite obviously the architecture of this class of sensor nodes [11] can only reflect the capabilities, which suffice for such a model.

During normal conditions, the sensory data remains predictable with gradual changes, and hence is not of particular importance. 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 storage devices, namely the SD-Cards, Compact Flash, XD-Cards [12] [21] [23].

The remainder of the paper is outlined as follows. Section 2, specifies the motivation for this research effort. Section 3 details out the RISE platform which is based on the Chipcon CC1010. A complete description of the Co-S architecture and the SD-Card is laid out in sections 4 and 5. This is followed up with an integration scheme in section 6. Experimental results which prove the efficacy of our integrated architecture are described in section 7. Finally rounded up with sections 8 and 9 are the future work and the conclusions.

2. Motivation

Our work is motivated by the requirements of the Bio-Complexity and the James Reserve Projects at the Center of Conservation Biology[a] (CCB) at UC Riverside CCB is working towards the conservation and restoration of species and eco-systems 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 understandings 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 eco-system 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.

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, [14] which includes an power optimized 8051 core, radio, 3 ADCs, 2KB SRAM, 32 KB on-chip-flash, 2 UARTs, SPI 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 peripherals entail individual Printed Circuit Board (PCB) area, voltage drop through the longer PCB traces, leakage / 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.

2.1 Sense and Store

Various long-epoch applications involve long time interval between consecutive queries, (e.g. weekly or monthly), although the sensor still acquires data from its surrounding environment frequently (e.g. every second). The user might then ask: "Find the time instance on which we had the highest average temperature in the last month?". To address these needs, our "Sense and Store" paradigm stores the data onto the on-board memory first and instead of naively passing on each and every piece of raw data through the hierarchical structure of a sensor network, it first calculates the queried information on the node and then transmits only the relevant information.

[a] Center for Conservation Biology,http://www.ccb.ucr.edu

This new approach has been demonstrated to be a substantial improvement over the Sense and Send architecture [17] [24].

Table1. Comparison of number 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**

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. 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 [16], [20], and transmitting it via the wireless interface, assuming no errors, consumes 16,473 mJoules [18]. 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) [12] [21] [23], 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 afford the luxury of storing vast amounts of data [10], (Gigabyte scale), on board, and intelligently and efficiently process queries in-situ, and in employing these devices lies the crux of the "Sense and Store" paradigm.

2.2 Silicon Integration

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]. 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 1, the CC1010 with similar capabilities as MICA, uses just one integrated SoC, while the MICA utilizes five IC (Integrated Circuit) devices to obtain the same functionality. Similarly the number of discrete components is nearly 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.

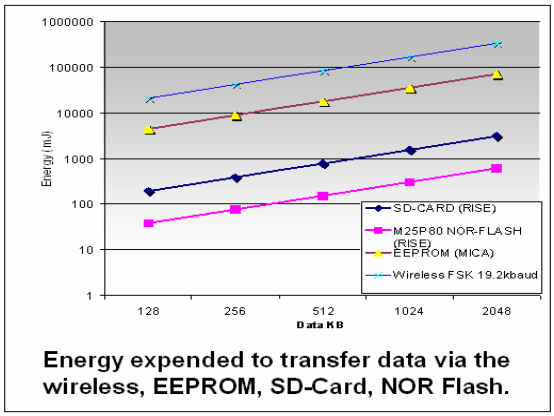

Figure1. A comparison of the amount of energy expended to transfer data via the wireless (pink trace) interface Vs Storing it on the on-chip EEPROM (yellow trace) and the on-board SD-Card (Blue trace). Clearly it makes sense to store data on-node and transmit only the relevant, non-redundant sensory information

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.

A detailed tabular comparison detailing how the RISE and the Co-S stack up against the other popular sensor node architectures is presented in Table 2.

2.3 Co-Processing for sensors

We exploit the Renesas M16C/28 platform [22] 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 also 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 non-volatile SD-Card can still be extracted off-line for required patterns, providing a significant layer of reliability and data retention in the network as a whole.

3. 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 CC1010 SoC, a compact 12mm by 12mm and only 1.2mm wide, is a feature packed device making it an ideal candidate for use in low power wireless embedded device applications. The CC 1010 is a true single chip UHF 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 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. It employs Manchester codec in hardware and RSSI 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 [14]. 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. A brief overview of the integrated components, which the RISE platform sports are presented in Table 2.

Some of the major sensors, which have been integrated with the RISE platform, are the CO_2, temperature, audio, 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.

The RISE platform in action, interfaced with the Carbon Dioxide, temperature sensors is displayed in Figure 2. The latest stable version of TinyOS, i.e., tinyos-1.1, as also the NesC compiler (nesc v1.2alpha1) were ported on to the RISE platform. The starting point of the port was the Wisenet project [5],

which had ported the older versions of the TinyOS and NesC. The newer versions of TinyOS and nesc now include support for clock synchronization, which is essential in indexing and storing the data on the flash.

The C language file was produced by *nesc1.exe*. The script *nesc-compile* was modified to pass source code to the custom post-processor for RISE, *sdccppp*. This script extracts the 8051 specific parameters that were passed through the nesc compiler and invokes the sdcc compiler and the *packihx* tools with the relevant flags to generate the hex file that can then be stored on the flash program memory.

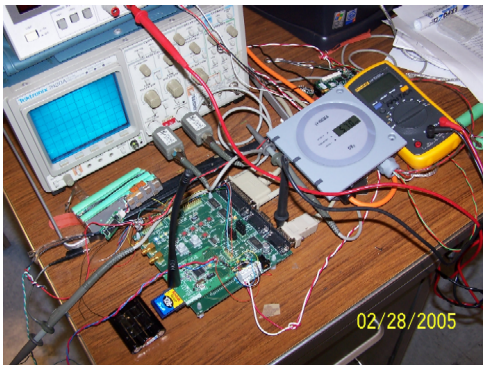

Figure 2. The RISE node interfaced with the sensors and the evaluation testbed.

4. CO-S Platform (Renesas M16C)

Our co-processing system design consists of a tightly integrated high performance 16bit MCU (MicroController Unit) (Renesas M16C/30280AFHP) [22] 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 128-point FFT benchmark.

Table 2. A brief listing of the multitude of features present 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
Typical SD-Card Specifications	
Current Consumption (Write, Read, Sleep)	80 mA, 60mA, 500µA
Access Time	200µsec (max)
Read or Write on 256-512 MB card	10MB/sec for both
The SPI bus	
Datarate	Programmable upto 3 MHz
The data block length	512 bytes

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 5.

The Renesas based Co-S, integrated with the RISE platform and sporting a high capacity SD-Card is displayed in Figure 3, additionally a listing of a subset of the capabilities of the Renesas M16C based Co-S is provided in Table 3.

5. Secure Digital Card

SD-Cards [12] [21] [23], (Secure Digital Cards) are postage stamp sized (24mmX32mmX2.1mm) COTS non-volatile flash memory storage devices featuring upto 1 Gigabyte of storage space. SD-Cards utilize the NAND Flash memory, which has some distinct characteristics summarized as follows: a) Every block (512 bytes) can only be written a finite number of times (typically 100,000) b) Writing to a block

requires that the block is already deleted. These cards have in-built controllers, which take care of the NAND flash memory management. They consume minute amounts of energy while storing and retrieving data thus making them highly suitable for integration with sensor platforms.

Table 3. A listing of the features available on the M16C Renesas based 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
Packaging	64 pin/80 pin QFP
Serial I/O	
2 channels (UART0, UART1)	I^2C, SPI, clock synchrounous, UART

The huge amount of onboard flash storage is most suitable for long term storage, as well as data sampled at fast sampling rates. Storing data generated from such sensors 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, 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. Their slim and compact design makes them an ideal removable storage solution for designs ranging from digital cameras, PDAs, cellular phones, and sensor platforms.

Nonvolatile flash memory standards for off-the-shelf memory cards range from Compact Flash, to the SD, XD, MMC and others. Compact Flash cards communicate through a parallel bus, unsuitable for simple microcontrollers while the XD card is devoid of an intelligent internal controller. SD-Cards however support the popular SPI bus interface, which it inherits from an earlier generation of MMC cards. 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. Dedicating four I/O pins from the Co-S platform to the SD-Card prove sufficient (Clock, Data IN, Data OUT, Clock Select), along with the power supply.

The microcontroller transfers data using the SPI protocol. Each write transaction to the 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.

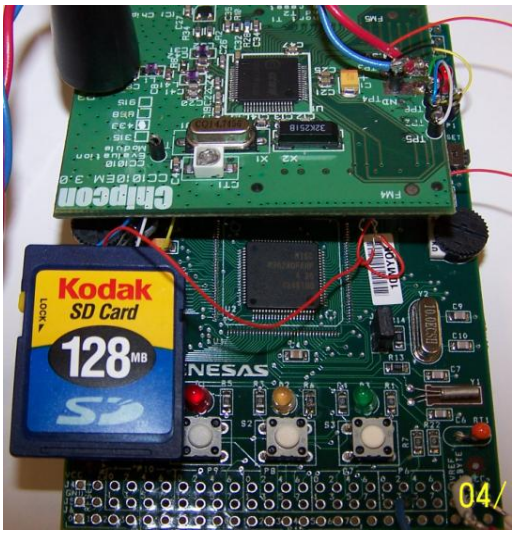

Figure 3. The Renesas M16C 16-bit MCU, Co-S platform, integrated with the Chipcon RISE platform and sporting an SD-Card

6. Co-S enhanced RISE 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. 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. The RISE platform interfaced with Co-S is depicted in Figure 4.

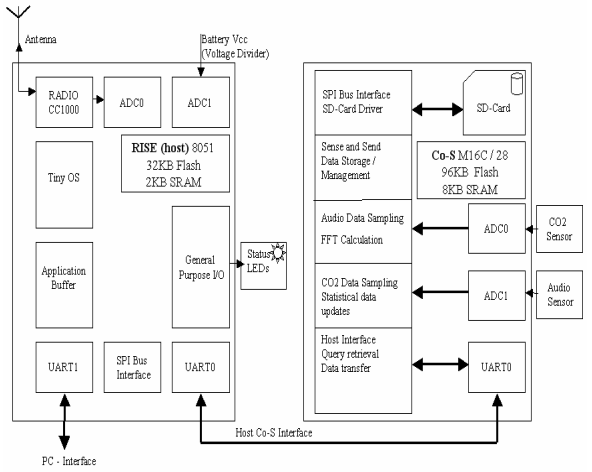

Figure 4. The RISE platform interfaced with the Co-S

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 between ten to sixty samples per minute. At the elapse of a minute, these samples are averaged, time-stamped, thereafter stored onto the on-chip buffer. A full on-chip buffer triggers a flush to the SD-Card. A twenty-four hour indexing scheme is used to index these samples. The index is stored in an SRAM buffer (2KB) and is committed to the SD-Card past elapse of twenty-four hours. We also investigated a simple hashing mechanism, with buckets implemented on the SD-Card.

The sound samples are acquired through the ADCs at a rate of 8Ksamples/Sec. A 128-point FFT operation on 64 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 4. Sensors interfaced with the RISE, Co-S

Sensor Type	Sensed Parameter	Sampling Rate
Vaisala Carbocap GMT 220	Carbon Dioxide	10-60 Samples / min.
National Semi. LM 61	Temperature	10-60 Samples / min.
Microphone Ckt.	Audio	8KHz

7. Evaluation

The hardware setup revolves around the RISE and the Co-S system, with which is interfaced a 128 Megabyte SD-Card. The Vaisala Carbocap GMT 220 (www.vaisala.com) carbon dioxide sensor, the LM61 temperature sensor and a microphone interface circuit were employed to obtain real-time sensor data for analysis. 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 multi-meter 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.

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. Table 5 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 execute the operation ten-thousand times and report the total time divided by the number of iterations.

It is clear from Table 5 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.

Table 6 depicts the power consumption of a SD-Card when interfaced to the RISE platform using the SPI Bus. Pseudo Random data was generated on the platform and then written onto the SD-Card.

Table 5. depicting the power consumption and time taken for evaluation a 128-point FFT on various microcontrollers. Ref to [16], [20]

Platform	Processor	Clk MHz	Time ms	Energy uJ
MICA[9] (3.0V)	AVR Mega 128L	8	14.5	934
Stargate[9] (3.3V)	Intel PXA 255	400	0.095	45.8
RISE (host) (3.0v/20mA)	Chipcon CC1010	14	11	660
RISE (Co-S) (2.7V/15mA)	Renesas M16C/28	20	0.892	36

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.

Table 6. Detailed power consumption of SD-Card interfaced with the SPI-Bus on RISE-Co-S platform

Operat-ion	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 [2] Write 8 blocks	1MB	13.5	6896	1915.65	1.92
Erase [3]	100 MB	14.5	74	2727.45	0.027

[1]Erase is handled by the internal controller of the card, and is not managed by the host. [2]Read / Write is managed by the host. Hence Read / Write speed is limited by the wire-speed connecting the host to the SD-CARD until the actual read / write speed of the SD-CARD is reached. [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.

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 Co-St of re-writing to a single block involves the Co-St of 1) erasing the sector 2) updating all the blocks other than the one being programmed 3) re-writing the given block. 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 loaded 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.

The graph in Figure 1 contrasts the energy consumption while communicating data using the wireless channel Vs storage of the same on the SD-Card and EEPROM. This pictorial representation clearly implies the significant penalty that a sensor, following the "Sense and Send" approach, has to incur for each and every transmission, when it could have logged the data on to the on-board storage for analysis.

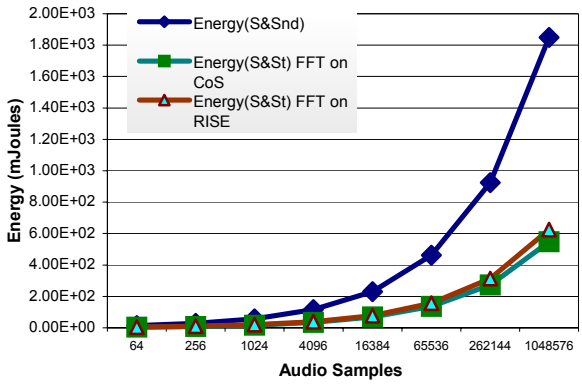

Figure 5. Graph displaying energy expended for audio sampling, FFT, Storage, and Transmission of a query (existence of a particular harmonic).

Figure 5, 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. Thus for each occurrence of the harmonic, amount of data that needs to be sent across the network, is only 10 bytes, while the raw FFT data measures 512 Bytes for a 128-point FFT operation.

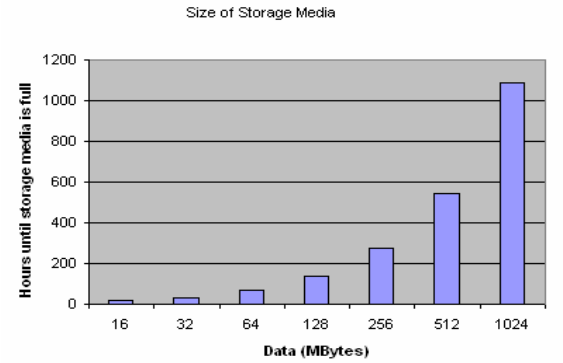

Figure 6. Total time for which a RISE-Co-S node can continuously log data onto a SD-Card at 10 Bytes/Sec.

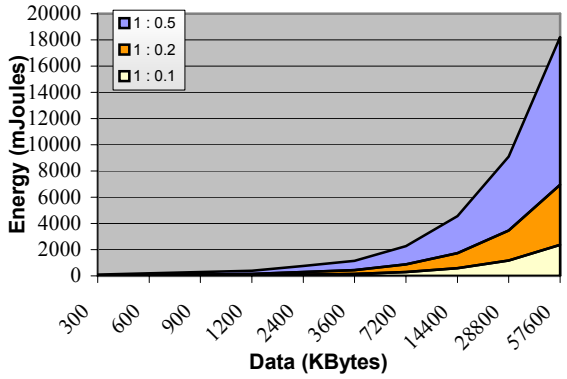

Figure 7. Graph displaying the energy expended for various Store Vs Send ratio on the RISE platform. For example, the first metric, 1:0.1, is energy consumed following a policy of storing 90% of the data and transmitting only 10%.

Figure 6 depicts the number of hours for which a sensor having an integrated on-board storage device can keep logging data onto its SD-Card. thereafter stored onto the on-chip buffer.

For pre-defined queries, being calculated on the fly, the Sense and Store along with an indexing scheme on the SD-Card offers the highest efficiency in terms of energy as depicted in Figure 8.

Also highlighted is that the RISE/Co-S duo performs well even in cases of unanticipated queries being demanded from the archived data, such as reporting Top-K values, which entails a sorting operation on the stored data first. This metric includes the power consumption due to extraction of stored data from the SD-Card, sorting it, and resulting in generation of the top-k values.

A hash based indexing scheme, involves storage and processing of numerous buckets many of which are not guaranteed to be completely occupied, to deal with this, we store occupancy information in the first 8 bytes of a 512-byte bucket. The energy required to probe the first 8 bytes of a bucket and thereafter read the remaining portion of the bucket for various occupancy levels is highlighted in Figure 9. Since quite a few buckets may not be completely occupied, we save on energy being consumed to read the complete bucket, even though we incur a negligible penalty of about 10uJ for accessing the first 8 bytes of the bucket.

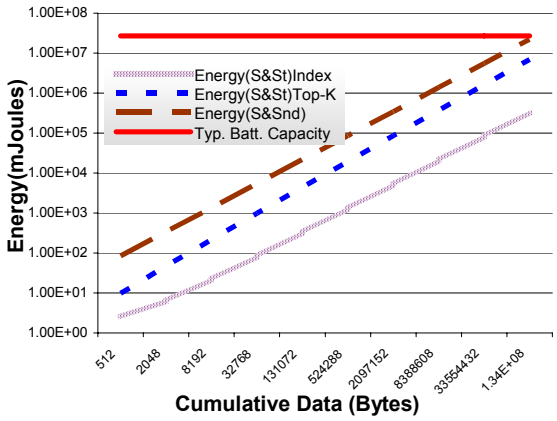

Figure 8. Graph displaying the energy expended for Top K query processing. The index based data extraction consumes the least amount of energy Vs the sequential retrieval from the SD-Card.

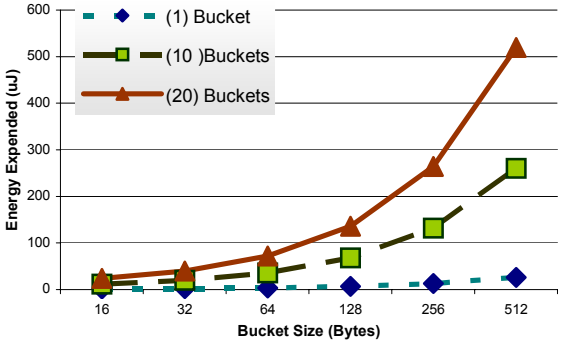

Figure 9. Graph displaying the Energy expended for various occupancy level of each bucket used in the hash based indexing scheme on the SD-Card

8. 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. 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.

Our research effort has also been able to highlight the basis and advantages of the "Sense and Store" methodology. Large capacity SD-Cards hooked onto sensor platform allows accumulation of gigabyte scale sensed data onto the sensor itself, resulting not only in power savings but also in terms of the amount of data available for post-mortem purposes. 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.

9. Future Work

We are already at an advanced stage of developing an integrated storage board, easily interfaced with the Co-S enhanced RISE platform. This storage board features additional flash memory based on a NOR architecture along with the NAND based SD-Card. Flash memory based on a NOR architecture allows read and write on smaller level of granularity, most suitable for hash and tree based indices. Finally, we are also investigating the use of ARM-7 and ARM-9 based microcontrollers for offloading and catering to future higher performance sensing applications.

10. References

[1] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., and Cayirci, E., ``A Survey on Sensor Networks, IEEE Communications Magazine," 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] D. McIntire, "Energy benefits of 32 bit microprocessor wireless sensing system", Sensoria Corp.

[10] P. Zhang, C. Sadler, S. Lyon, and M. Martonosi, "Hardware Design Experiences in ZebraNet", Proceedings of SenSys 2004, November 2004.

[11] J. Polastre, R. Szewczyk, C. Sharp and D. Culler, "The Mote Revolution: Low Power Wireless Sensor Network Devices", Hot Chips 2004.

[12] Secure Digital Memory Card Physical Layer Specification, Version 1.01, SD Group.

[13] J. Elson and D. Estrin, "Wireless Sensor Networks: A bridge to the Physical World", in Wireless Sensor Networks , Kluwer, 2004.

[14] http://www.chipcon.com

[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] http://www.tinyos.net/nest/doc/tutorial/lesson7.html.

[17] Demetrios Zeinalipour-Yazti, Som Neema, Dimitrios Gunopulos, Vana Kalogeraki, Walid Najjar, "Data Acquisition in Sensor Networks with Large Memories", NetDB 2005.

[18] A. Mitra, "Bit error analysis of new generation wireless transceivers", IEEE-ICCS 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] 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.

[21] http://www.sdcard.org

[22] Renesas, M16C/28 Group Hardware Manual-rej09b0047_16c28hm.pdf

[23] Secure Digital Card, Product Manual, Revision 1.7, Sandisk Corporation.

[24] Philippe Bonnet, Allan Beaufour, Mads Bondo Dydensborg and Martin Leopold, "Bluetooth-based sensor networks", ACM SIGMOD, 2003.