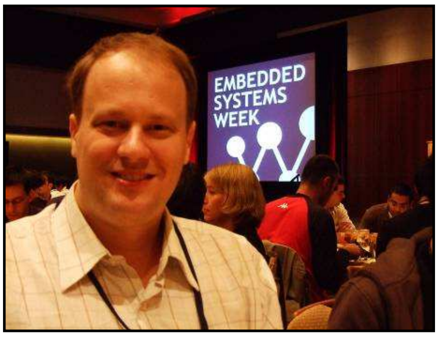
ESWEEK 2008

Student Report by: Kenn Slagter (9662882) *Embedded Systems Week 2008*

Autobiography



Author:	Kenn Slagter
Date of Birth:	15 September 1976
Nationality:	New Zealand/Netherlands
School:	NTHU
Supervisor:	Prof. Yeh-Ching Chung
Department:	Computer Science
Laboratory:	SSLAB

Brief History

I was born and raised in New Zealand, so I received most of my education there. I received my Bachelor Degree in Computer Science from Waikato University, and my Masters in Computer Studies from The University of New England in Australia. I met my future wife at University in New Zealand. Since she is Taiwanese I came to live in Taiwan.

I worked in Taiwan for nearly 7 years working mostly in Hsinchu Industrial Park as a research and development engineer working on embedded system software. I recently became a student again now at the NTHU hoping to complete my tertiary education and receive a PhD qualification. During my time working in the industrial park I have had many experiences with a variety of hardware and software. One area I worked on was in incorporating existing commercial software that implements a Flash Media Manager that handles both NOR and NAND Flash.

While working with this commercial package I had become intrigued by the underlying software that actually manipulates Flash. Since I have joined the SSLAB it now gives me opportunity to actually do study in the field of Flash Memory Technology.

Report

The Embedded Systems Week 2008 was held in October 19-24, 2008. The purpose of ESWEEK is to bring together conferences, tutorials and workshops centered on various aspects of embedded systems research and development. It spans the spectrum from software to hardware as well as issues in, and tools for modeling, specification, verification and validation of complex embedded systems. Three leading conferences in the area CASES, CODES+ISSS, and EMSOFT take place at the same time and location, along with workshops and tutorials.

EMSOFT, the international conference on embedded software which is part of ESWEEK, is important as it the general topic of my research as well as IWSSPS, the International Workshop on Software Support for Portable Storage, which covers many topics about flash memory and related topics.

Seminars Attended

Conference	***Topic***
CASES, tutorial	SSA-Based Register Allocation
EMSOFT	Model-Based Testing and Performance Analysis of Automotive ECU Networks
Opening Session	ACM Turing Award Lecture
CODES+ISSS	Flash Memory Management
EMSOFT	Flash Memory
CASES	Keynote, Power usage in Embedded Systems
CODES+ISSS	You can catch more bugs with transaction level honey
CODES+ISSS	Keynote “Co-design in the Wilderness”
EMSOFT	Virtual Machines, Compilers, Memory Management
PANEL	Will embedded multi-core imply standard architecture and/or single programming model
IWSSPS	International Workshop on Software Support for Portable Storage

SSA – Based Register Allocation

SPEAKERS
starting from the left and going to the right ...

Fernando Pereira,
Fabrice Rastello,
Sebastian Hack,
Philip Brisk,
Jens Palsberg.

STUDENTS'
continuing from left…

Kenn Slagter (me),
Zhi Ying (致穎),
Meng Ju (孟儒)

Sebastian Hack
Email: Hack@cs.uni-sb.de

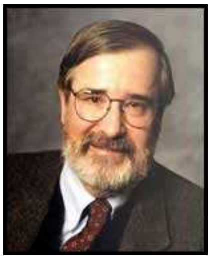
The Speakers above presented an architecture for register allocation based on the SSA-form. They showed how the properties of SSA-form programs and their interference graphs can be exploited and showed new methods they developed for spilling, coloring and coalescing.

Overall this tutorial was rather interesting and at times it was even amusing. The main points of interest in here were how they attacked the problems of SSA Based Register Allocation. Primarily when they encountered a problem instead of trying to solve that problem as it stood often instead they modified the parameters of the problem into a new one that was solvable.

An example of this was there solution to Graph Coloring. In graph theory, **graph coloring** is a special case of graph labeling; it is an assignment of labels traditionally called "colors" to elements of a graph subject to certain constraints. In its simplest form, it is a way of coloring the vertices of a graph such that no two adjacent vertices share the same color; this is called a **vertex coloring**. Similarly, an **edge coloring** assigns a color to each edge so that no two adjacent edges share the same color, and a **face coloring** of a planar graph assigns a color to each face or region so that no two faces that share a boundary have the same color [WIKIPEDIA].

Now according to the presenters of this paper graph coloring as a solution for register allocation is not a good solution. At least as is not as efficient at allocating registers as it could be. In order to fix this they simply changed the problem. The problem with graph coloring is that it works with complicated graphs. So they made sure that during previous phases things are changed so that the final graph they work with is chordal. That is the graph has no holes in it that are larger than 3 vertices. This simplified the process of finding a perfect elimination order, which allows for optimal coloring. I feel that this is an elegant solution.

ACM Turing Award Lecture



Edmund M. Clarke
School of Computer Science, Carnegie Mellon University

Allen E. Emerson
Department of Computer Sciences, the University of Texas at Austin

Joseph Sifakis
CNRS researcher and the founder of Verimag laboratory, France

These three were the winners of the 2007 Turing Award. They founded in 1981 what has become the highly successful field of Model Checking. Model Checking is a verification technology that provides an algorithmic means of determining whether an abstract model--representing, for example, a hardware or software design--satisfies a formal specification expressed as a temporal logic formula. Great strides have been made on this problem over the past 27 years by what is now a very large international research community. As a result many major hardware and software companies are now using Model Checking in practice. Examples of its use include the verification of VLSI circuits, communication protocols, software device drivers, real-time embedded systems, and security algorithms.

There is little wonder then that the speech delivered by these three gentlemen was on the topic of Model Checking. Overall I found their speech informative and found the men to be quite impressive figures. The odd thing about this speech is the one point made that sticks most in my mind most is not on the topic of Model Checking but is instead that Allen Emerson would have Dijkstra over to his house for dinner and that he seemed to chat with him often.

Some Presentations from Company Attendees Of The Conference

Bill Athas from **Apple** presented his keynote speech "**power on demand for mobile computing devices**". His point of view is that energy use is a leading concern in any portable device. It determines how long the battery life for a device is and whether a given device is going to be actually a successful product. Often consumers tend to think the battery itself maybe the problem when in fact it's the device as a whole that affects power consumption. Each hardware component in the device along with software plays a key role in determining energy consumption. One key point he made during his speech about power on demand, was that better performance was *not* about draining the battery faster. A quite plausible phenomenon one might encounter as feature creep occurs in a product. The problem is exacerbated when one involves the processor-centric problem of wanting the CPU to appear fast.

"**The OS vendor's perspective on system-level diagnostics"** is a presentation given in the special session **"You can catch more bugs with transaction-level honey" (CODES+ISSS)** given by **Joel Greenbaum** who comes from a company called **GreenHillsSoftware Inc** (http://www.ghs.com). His company develops an RTOS. The speech presented by him was humorous, as it describes the fustrations he himself has with customers. One of his main gripes is that with out on chip diagnostics or adequate tools there is little one can do to prove to customers, in this case these customers are companies who are developing new products, what is going wrong with their system. The problem he faces is that after a customer purchases his companies RTOS if anything goes wrong during development then the first thing customers tend to blame is the RTOS. I.E. "This RTOS sucks, the filesystem is so slow" or the other myriad other complaints customers tend to voice whenever their own project hits its various bugs, trials and tribulations.

The problem is its usually not the RTOS and sometimes what is obvious to the vendor is not to the customer. It may even be that the customer may wish to keep their "head in the sand" rather than deal with a problem. They will continue to work on the project even if it seems obvious to the vendor it is doomed to fail or hit serious problems later, simply because the customer does not want to listen, or because there is no way to pove to the customer that something is wrong with their design. One example given by Joel Greenbaum was that while he could see that bandwidth problems was the obvious cause of the systems problems, and was causing the slow access to filesystems etc, it was hard to explain to customer why that was. The customer themselves had to spend several weeks using a logic analyzer and manually checking the system in order for them to be convinced and see the error of their ways. If the chip had adequate facilities to do self analysis and appropriate tools on hand this could have been shown in minutes instead of weeks.

"You can catch more bugs with transaction-level honey" (CODES+ISSS) also had other presentations given by other speakers from different companies.

(i) "Using transactions for In-System Silicon Validation and Debug", Miron Abromovici, DAFCA.
(ii) "ESL-Driven Instrumentation Interfaces", Neal Sollen, HDL Dynamics.
(iii) "Communication-centric debug for networks-on-chip", Kees Goosens, NXP.

While these presentations had merit they were hardware-centric and not relevant to my field of research into Flash technologies as far as I could tell.

CODES+ISSS/EMSOFT – Flash Memory Management

CODE+ISSS Papers presented

(i) *A Time Predictable System Initialization Design For Huge Capacity Flash Memory Storage Systems*
(ii) *Deterministic Service Guarantees for NAND Flash using Partial using Partial Block Cleaning*

EMSOFT Papers presented

(i) *µ-FTL: A Memory Efficient Flash Translation Layer Supporting Multiple Mapping Granularities*
(ii) *A PRAM & NAND Flash Hybrid Architecture for High Performance Embedded Storage Subsystems*
(iii) *Workload-Based Configuration of MEMS-Based Storage Devices for Mobile Systems*
(iv) *Lightweight Time shift Flash Translation Layer for Flash Memory based Embedded Storage*

ABSTRACTS

A Time Predictable System Initialization Design for Huge Capacity Flash Memory Storage Systems

The capacity of flash-memory storage systems grows at a speed similar to many other storage systems. In order to properly manage the product cost, vendors face serious challenges in system designs. How to provide an expected system initialization time for huge-capacity flash-memory storage systems has become an important research topic. In this speech, a time-predictable system initialization design is proposed for huge-capacity flash-memory storage systems. The objective of the design is to provide an expected system initialization time based on a coarse-grained flash translation layer. The time-predictable analysis of the design is provided to discuss the relation between the size of main memory and the system initialization time. The system initialization time can be also estimated and predicted by the time-predictable analysis.

Deterministic Service Guarantees for NAND Flash using Partial using Partial Block Cleaning

NAND flash idiosyncrasies such as bulk erase and wear leveling results in non-linear and unpredictable read/write access times. In case of application domains such as streaming multimedia and real-time systems, a deterministic read/write access time is desired during design time. Proposed is a novel NAND flash translation layer called GFTL that guarantees fixed upper bounds (i.e., worst case service rates) for reads and writes that are comparable to a theoretical ideal case. Such guarantees are made possible by eliminating sources of non-determinism in GFTL design and using partial block cleaning. GFTL performs garbage collection in partial steps by dividing the garbage collection of a single block into several chunks, thereby interleaving and hiding the garbage collection latency while servicing requests.

Further, GFTL guarantees are independent of flash utilization, size or state. Along with theoretical bounds, benchmark results show the efficacy of our approach. Based on our experiments, GFTL requires an additional 16% of total blocks for flash management. GFTL service guarantees can be calculated from flash specifications. Thus, with GFTL, a designer can determine the service guarantees and size requirements, during design time.

µ-FTL: A Memory Efficient Flash Translation Layer Supporting Multiple Mapping Granularities

FTL (Flash Translation Layer) is one of the most essential software components in NAND flash-based embedded devices as it allows one to use legacy files systems by emulating the traditional block device interface on top of NAND flash memory. In this speech, they proposed a new FTL, called µ-FTL. The main design goal of µ-FTL is to reduce the memory footprint as small as possible, while providing the best performance by supporting multiple mapping granularities based on variable-sized extents. The mapping information is managed by µ-Tree, which offers an efficient index structure for NAND flash memory.

A PRAM & NAND Flash Hybrid Architecture for High Performance Embedded Storage Subsystems

NAND flash-based storage is widely used in embedded systems due to its numerous benefits: low cost, high density, small form factor and so on. However, NAND flash-based storage is still suffering from serious performance degradation for random or small size write access. This degradation mainly comes from the physical constraints of NAND flash: erase-before-program and different unit size of erase and program operations. To overcome these constraints, in this speech it is proposed to use PRAM (Phase-change RAM) which supports advanced features: fast byte access capability and no requirement for erase-before-program.

This speech focuses on developing a high-performance NAND flash-based storage system by maximally exploiting the advanced feature of PRAM, in terms of performance and wearing out. To do this, they first proposed a new hybrid storage architecture which consists of PRAM and NAND flash. Second, they devised two software schemes for the architecture; FSMS (File System Metadata Separation) and hFTL (hybrid Flash Translation Layer). Finally, they demonstrated that their hybrid architecture increased the performance up to 290% and doubled the lifespan compared to existing NAND flash only storage systems.

Workload-Based Configuration of MEMS-Based Storage Devices for Mobile Systems

Because of its small form factor, high capacity, and expected low cost, MEMS-based storage is a suitable storage technology for mobile systems. However, as memory may outperform MEMS-based storage in terms of performance, and energy efficiency. The problem is that MEMS-based storage devices have a large number (i.e., thousands) of heads, and to deliver peak performance, all heads must be deployed simultaneously to access each single sector. Since these devices are mechanical and thus some housekeeping information is needed for each head, this results in a huge capacity loss and increases the energy consumption of MEMS-based storage with respect to flash.

According to this speech the speaker attempts to solve this problem by proposing new techniques to lay out data in MEMS-based storage devices. Data layouts represent optimizations in a design space spanned by three parameters: the number of active heads, sector parallelism, and sector size. We explore this design space and show that by exploiting knowledge of the expected workload, MEMS based devices can employ all heads, thus delivering peak performance, while decreasing the energy consumption and compromising only a little on the capacity. Our exploration shows that MEMS-based storage is competitive with flash in most cases, and outperforms flash in a few cases.

Lightweight Time shift Flash Translation Layer for Flash Memory based Embedded Storage

Flash memory storage has been widely used in various embedded systems such as digital cameras, MP3 players, cellular phones, and DMB devices and now it applies to PCs as a form of SSDs. Characteristics of Flash memory necessitate a software layer called FTL (Flash Translation Layer) that directs modified data to new places in Flash memory and maintains a mapping between a logical sector number to a physical page. Noticed is that this out of-place update scheme of the FTL allows a low-overhead time shifting between multiple versions of storage state. From this observation, the speaker proposes LTFTL (Lightweight Time-shift FTL) that provides not only multiple versions of storage state but also an open-ended interface to traverse them. This open-ended interface can be used to support fault-resilience schemes, transactions of various granularities, and user-friendly roll-back services. Experimental results from a prototype implementation show that the proposed LTFTL can (1) provide a low-overhead time-shift capability at the user level by maintaining multiple storage states and (2) enhance the reliability/survivability of Flash memory by allowing to roll back to a previous consistent storage state at the storage system level.

Comments

The topics presented were in of themselves quite educational, which is not a surprise since this is the nature of the whole conference. One attendee sitting in at the presentation of the paper **“A PRAM & NAND Flash Hybrid Architecture for High Performance Embedded Storage Subsystems”** inquired into whether there was anything new they discovered since their research seemed similar to a paper that he presented in a conference in the previous year. The speaker presenting the papers name was “Jin Kyu Kim” (not the South Korean football player). Since the speaker was in fact Korean, had a touch of the flu, was a student and since English was his second language he had trouble either understanding or answering the question.

Embedded Systems Week
www.esweek.org
A PRAM and NAND Flash Hybrid Architecture for High-Performance Embedded Storage Subsystems
Jin Kyu Kim, Hyung Gy
Kyoung Il Bahng
HYATT

Jin Kyu Kim presenting **“A PRAM & NAND Flash Hybrid Architecture for High Performance Embedded Storage Subsystems”**

Another topic presented “***Workload-Based Configuration of MEMS-Based Storage Devices for Mobile Systems”*** I found to be off the topic of this particular session. The speaker presented what seems to be a different technology to contrast against Flash. Since current prototypes were not available he had to rely on diagrams making it a little harder to understand what he was trying to convey.

The presenter of "**Lightweight Time shift Flash Translation Layer for Flash Memory based Embedded Storage"** was also Korean. His name was **"Kyoungmoon Sun"**, A masters student from Dankook University. He was there with his professor **"Jongmoo Choi"** and his fellow student **"Myoung Sub Shim"** who presented his paper to the conference on the last day in the IWSSPS workshop. At the end of his presentation he was very flushed, and his face had become quite red as he was obviously still trying to calm down having been quite nervous speaking in front of so many academics and professionals in his field of research.

On the day of IWSSPS I spent a lot of time with Kyoungmoon Sun and Myoung Sub Shim discussing various things, including the difficulties they had faced during their research. Since the number of people at the IWSSPS was about 20 people I got to interact with them quite well and we had lunch together at the HYATT hotel where the conference was held.

Kyoungmoon Sun, email: msg2me@dankook.ac.kr

Myoung Sub Shim, email: sms1222@mail.hongik.ac.kr

IWSSPS –The International Workshop on Software Support for Portable Storage

IWSSPS Keynotes

(i) Design Space Surrounding Flash Memory
(Bumsoo Kim, President and CEO, Indilinx Co., Ltd.)

IWSSPS Papers presented

(ii) Web Browser Cache Management Technique for Full Browsing on NAND-based Mobile Devices
(iii) PT-R Practical Paging Stategy For Real-Time Code Extension of Media Player on NAND Flash*
(iv) Implementation study on a Unified SCRAM Manager for Storage Class Random Access Memory
(v) Buffer-Aware Garbage Collection for NAND Flash Memory-Based Storage Systems
(vi) Cost and Performance Analysis of NAND Mapping Algorithms in a Shared-Bus Multichip
(vii) The Effect of Absorbing Hot Write References on FTLs for Flash Supporting High Data Integrity
(viii) System support for Byte Addressable Non-Volatile Memory in Portable Storage

ABSTRACTS

Web Browser Cache Management Technique for Full Browsing on NAND-based Mobile Devices

As wireless communication and mobile device technologies improve,mobile full browsing is emerging as a killer application for mobile devices. In order to use mobile full browsing efficiently in terms of performance, energy, and download cost, mobile Web browsers adopt Web browser cache. However, the I/O access patterns of a Web browser cache make it difficult for these mobile devices to use a NAND-based secondary storage efficiently. In this presentation, we propose Web browser cache management techniques for mobile full browsing on NAND-based mobile devices. A proposed filtering technique reduces useless write operations in NAND flash memory by selectively caching frequently reused Web pages only. A proposed NAND-aware packaging technique reduces garbage collection overheads by combining small files into a single NAND page. Our experimental results show that the proposedWeb browser cache management techniques reduce the total response time of a Web browser cache on average 28% over a conventional Web browser cache management technique.

PT-R* Practical Paging Stategy For Real-Time Code Extension of Media Player on NAND Flash

*NAND flash memory is widely used as a secondary storage of embedded devices due to its attractive features such as small size, fast access speed, shock resistance, and low power consumption. Although it is extensively used as a non-volatile storage for storing data, NAND flash memory requires a large amount of SRAM for executing program codes in it since it does not support byte-level random access. Program code pages should be loaded into SRAM from NAND flash before executing. RT-PLRU has demonstrated a promising possibility of using NAND flash memory as a code storage for real-time programs. RT-PLRU strategy combines pinning and LRU to minimize the SRAM size required to provide the pre-negotiated deadline meet probability. However, to implement RT-PLRU in a real system, it gives a challenging issue of implementing pure LRU in operating system. Since it is not possible to implement pure LRU in software, this presentation studies alternative algorithms that can be implemented in a real system. This presentation also presents extensive experimental results using various page replacement algorithms in place of LRU part in RT-PLRU strategy. Through this experimental study, we show that an implementable variation of LRU algorithm, called iLRU, can be used as a base page replacement policy for RT-P*strategy.*

Implementation study on a Unified SCRAM Manager for Storage Class Random Access Memory

These days, a variety of embedded systems are actively employing Storage Class RAM (SCRAM) such as FeRAM, MRAM and PRAM. SCRAM has not only DRAM-characteristic, that is a random byte-unit access capability but also Disk characteristic, that is a non-volatility. These characteristics enable SCRAM to be used both for main memory and for secondary storage. Presented is a novel SCRAM manager for SCRAM-equipped embedded systems. In conventional operating system, file objects and memory objects are treated as independent objects and managed separately by file systems and memory managers. In the SCRAM manager, however, these objects are treated as a single object under a unified view. This unified management gives an opportunity to apply various performance-improving techniques such as object migration and reduction of data copy. Preliminary experimental results conducted on real embedded board with FeRAM have shown that the proposed SCRAM manager supports seamlessly both file system and memory manager interfaces with overheads comparable to the previous real-time operating systems.

Buffer-Aware Garbage Collection for NAND Flash Memory-Based Storage Systems

With continuing improvements in both the price and the capacity, flash memory-based storage devices are becoming a viable solution for satisfying high-performance storage demands of desktop systems as well as mobile embedded systems. Because of the erase-before-write characteristic of flash memory, a flash memory based storage system requires a garbage collection, which often introduces large performance degradation due to a large number of page migrations and block erase operations. In order to improve the overall I/O performance of the flash-based storage systems, therefore, it is important to support the garbage collection efficiently. In this presentation, we propose a novel garbage collection scheme, called buffer-aware garbage collection (BA-GC), for flash memory-based storage systems. In implementing two main steps of the garbage collection module, the block merge step and the victim block selection step, the proposed BA-GC scheme takes into account the contents of a buffer cache (e.g., a page cache and a disk buffer) which is used to enhance the I/O performance of storage systems. The buffer-aware block merge (BA-BM) scheme reduces the number of unnecessary page migrations by enforcing a dirty page eviction in the buffer cache during the garbage collection. The buffer aware victim block selection (BA-VBS) scheme, on the other hand, selects a victim block so that the overall I/O performance can be maximized. Our experimental results show that the proposed BAGC technique improves the overall I/O performance up to 45% over existing buffer-unaware schemes.

Cost and Performance Analysis of NAND Mapping Algorithms in a Shared-Bus Multichip

NAND flash memory is widely used in mobile embedded devices and PCs due to its versatile features such as non-volatility, low power consumption, and shock resistance despite low access performance. The performance limitation of such memory devices requires flash-based storage systems to adopt an interleaved configuration to increase high aggregated access bandwidth. However, most mapping algorithms used in flash memory systems are designed to accomplish high performance and high utilization in a single chip, and therefore do not pay much attention to flash memory durability. In this paper, we present new mapping algorithms for flash-based storage systems with a multi-chip interleaving configuration. The simulation results show that the proposed algorithm could effectively increase NAND life cycles by up to 40% via the reduction of erase operations without a loss of access performance.

The Effect of Absorbing Hot Write References on FTLs for Flash Supporting High Data Integrity

Flash storages are prevalent as portable storage in computing systems. When we consider the detachability of Flash storage devices, data integrity becomes an important issue. We consider the performance of Flash Translation Layer (FTL) schemes in conjunction with file system behavior that pursue high data integrity. To assure extreme data integrity, file systems synchronously write all file data to storage accompanying hot write references. In this study, 1) we concentrate on the effect of hot write references on Flash storage, and 2) we consider the effect of absorbing the hot write references via nonvolatile write cache on the performance of the FTL schemes in Flash storage. In so doing, one can quantify the performance of typical FTL schemes for workloads that contain hot write references through a wide range of experiments on a real system environment. Results show that for workloads with hot write references FTL performance does not conform with previously reported studies. In conclusion it is shown that the impact of the underlying FTL schemes on the performance of Flash storage is dramatically reduced by absorbing the hot write references via non-volatile write cache.

System support for Byte Addressable Non-Volatile Memory in Portable Storage

Portable storage was a synonym for block-addressable non-volatile memory like hard disks and NAND flash memories. With the upcoming of fast, wear-free, high-density, and low-power NVRAM technology like PC-RAM and STT-RAM a novel software interface besides the legacy block-interface is worth discussion. Portable NV-RAM makes a file system dispensable but requires system support for address mapping, access control, and consistency maintenance. NV-RAM not only allows execution in place but also facilitates applications with irregular write patterns to operate directly on portable storage without the need to copy data back and forth to/from RAM. For portable computers this results in lower demands for volatile RAM, in an extended standby time, and in an accelerated launch of applications.

Design Space Surrounding Flash Memory

Kim Bumsoo, **President** and **CEO** of **INDILINX**, discussed software and systems design issues for flash memory applications. He is perhaps one of the most **important** speakers at the conference due to his experience in the field of Flash.

Based on his previous work experience in Samsung as flash memory software engineer, he also reviewed some design alternatives of flash memory device drivers, file systems and memory cards, etc. He also presented his thoughts and vision about SSD market and technology. The topic of his presentation was "**Design Space Surrounding Flash Memory**"

Design Space

Design space is the set of possible designs and design parameters that meet a specific product requirement. Exploring design space means evaluating the various design options possible with a given technology and optimizing with respect to specific constraints such as power or cost. [1994 National Technology Roadmap for Semiconductors]

Trends in NAND Flash development

In the last decade the dominant flash technology has changed from NOR to NAND type flash. NAND flash has become the new order in memory technology in the memory industry providing rich utility for a myriad of products. One important point to make is the rapid decrease in price per Megabyte one sees for Flash. In fact current trend for flash is a reduction in cost of 1% a week, that's a reduction of about 50% in cost per year!

NAND Flash is slowly becoming more and more competitive against the traditional hard disk found in computers, already finding its way in some of the laptop products one sees today i.e. EeePC from ASUS. Another important feature one finds of NAND Flash is that Hard Disk cannot reach the speeds one can achieve with NAND Flash due to its mechanical complexity. Since Flash requires less mechanical components and needs less energy to run it in turn is the greener technology.

In the future NAND Flash technology will continue to increase in size and be needed for more and more technologies. NAND Flash will eventually work its way on to desktop PCs and Servers. In 3 years one will expect to see blade-servers using SSD. The reason servers will slowly convert to flash not including previously mentioned reasons is that NAND Flash has a lower failure rate than Hard Disks do. SSD has a failure rate of about 1% per year compared to HD failure rate of about 3% per year. This is a significant cost to server users as most of the costs they incur are not from the purchase of servers but from the costs involved in maintaining them.

As the size of NAND Flash increases it tends to drive it toward more complex usage

– OS Building Blocks and System Software

- boot loaders / device drivers / file systems
- Record & database management systems

– Application Requirements

- Faster booting and application loading
- Faster synchronization between device and host
- Faster browsing of captured files in a single directory

INDILINX(www.indilinx.com), backed by Softbank Ventures Korea, is a fabless semiconductor company specialized in SSD controller and systems, and has been accredited for the best technology by many governmental institutions and the industry after its foundation in 2006. The business operation is located in Milpitas, California and the R&D head is in Seongnam, Korea.

Research Plan

Based on the information I gleaned from ESWEEK related to **flash** technology I have garnered several facts. NAND Flash is going to continue to grow in size and importance in the computer industry. The root cause for the stimulation and rapid growth in flash size appears to be that it is being pushed a lot by consumers who find the technology appealing for its many uses.

Despite its reasonable speed a lot of current research extends to properly handling NAND Flash due to the innate faults with the medium. There are still a lot of research done and variations of the File Translation Layer, Garbage Collection, Wear Leveling, and even Application level attempts to reduce unnecessary writing to the flash. Unnecessary writes can be initiated simply due to the way garbage collection works. As the user must erase a whole block of data first before they can write to that block it appears that if the system is taken in a more holistic view it has a more optimal approach toward avoiding unnecessary writing.

Another area of research extends to reliability of the medium and speedier handling especially as the size of the medium increases into larger and larger sizes. It seems then that as the NAND Flash size approaches greater size these issues may become more and more of a problem.

Based on the issues presented above one possible plan for future research work would be to delve deeper into better software solutions and develop a system of techniques that would solve at least one if not all of these problems. At the minimum I should consider their impact on any future research I might make into this area of research.

Schedule

A rough schedule then for publishing a paper for a top conference similar to ESWEEK within a year would then require one to follow several typical research phases…

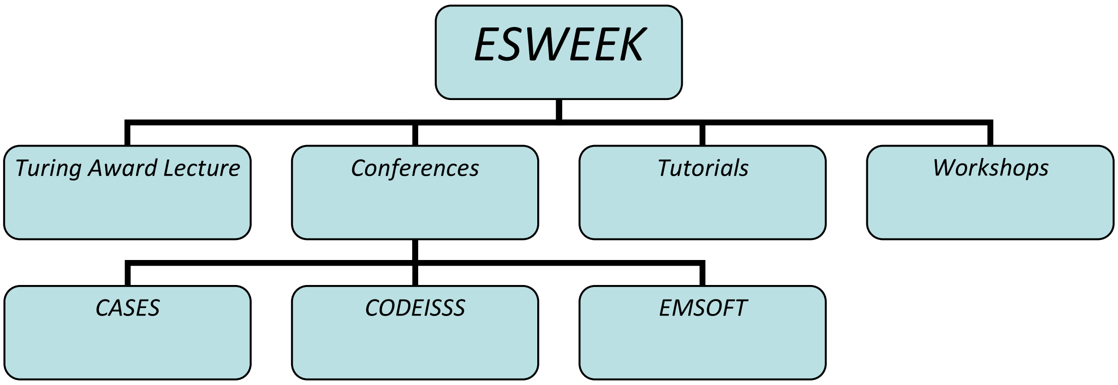
Phase	Description	Time Required
Survey	One would need to go through various published papers about the topic of interest. The point here now is to restrict the subject and then develop a preliminary thesis statement.	1 month
Platform	Obtain the appropriate hardware necessary to test your thesis on. You will then need to be able to run a very simple test program on the platform in order to continue further research.	1 month
Experimentation and Development	Find a way to benchmark the program you wish to create and then begin developing that program. During this period of time first prototype should be developed.	5 months
Obtain Result and commence Tuning	This is a cyclic behavior. The experiment needs further development to improve performance and fix any new problems discovered. This process probably needs to be done again and again until one is satisfied with the final result	2 months
Submit Paper	The paper needs to be written up and submitted to appropriate venue.	1 month
Respond to Feedback	The venue will likely need time to survey the paper give feedback. Several revisions of the paper is likely	2 months

Appendix

Important Acronyms, Abbreviations and Contractions.

Acronym	Definition
CASA	Compiler Assisted SoC Assembly Workshop
CASES	International Conference on Compilers, Architecture, and Synthesis for Embedded Systems
CODEISSS	International Conference on Hardware – Software Co-design and System Synthesis
CPS	Workshop on Cyber Physical Systems
EMSOFT	International Conference on Embedded Software
ESWEEK	Embedded Systems Week
ESTIMEDIA	IEEE Workshop on Embedded Systems for Real-time Media
IWSSPS	International Workshop on Software Support for Portable Storage
WESE	Workshop on Embedded Systems Education
WESS	Workshop on Embedded Systems Security
WFCD	Art Design Workshop on Foundations of Component Design

Organization of ESWEEK



Relevant Websites

http://www.esweek.org
http://www.emsoft.org
http://www.iwssps.org