SMASH: Smarter Flash Memory Storage in Low-Power Devices

PI: Professor Kevin Fu, Ph.D.
Computer Science Department
University of Massachusetts Amherst
140 Governors Drive, Amherst, MA 01003

kevinfu@cs.umass.edu
http://www.cs.umass.edu/~kevinfu/IP/SMASH.html

Executive Summary: Our smarter flash (SMASH) memory technology reduces the energy consumption of electronic devices by using novel storage algorithms in firmware to exploit the stochastic behavior of embedded flash memory at low voltages. Devices that would benefit include high-volume products such as carbon monoxide detectors, electronic locks, medical devices, TV remote controls, calculators, and wireless computer peripherals.

We demonstrate the advantages of *SMASH* for specific low-power devices by testing our technology on common, low-power microcontrollers found in consumer electronics. Our technology takes the form of a firmware update such that devices already manufactured or deployed can benefit. We have a working prototype that demonstrates the value of our technology for reducing energy consumption and manufacturing costs through the modification of software.

The cost and energy savings of SMASH memory technology results from the novel use of stochastic storage. Flash memory integrated within a microcontroller typically requires the entire chip to operate on a common supply voltage almost double what the CPU portion requires. Our SMASH approach lowers supply voltage so that the CPU may operate in a more energy-efficient manner without diminishing reliability. Efficient coding algorithms then restore reliability to the low-voltage flash memory. Our algorithms ensure reliable, low-power storage on unmodified or minimally modified hardware by exploiting the electrically cumulative nature of half-written data in flash cells. Tested on NOR flash memory, the techniques would also apply to alternative non-volatile memory technologies that exhibit a stochastic behavior at low voltage. In our experiments, we found **energy savings of 34%** for a sensor workload. The savings translate into (1) products that last longer on a battery without disturbing the manufacturing process or supply chain; (2) reduced manufacturing costs with cheaper, lighter, or smaller hardware components (e.g., batteries, voltage regulators); and (3) products that can better operate on alternative energy sources (e.g., energy harvesting) rather than heavy or toxic batteries. SMASH memory technology solves problems in a manner that would otherwise require higher cost and much guesswork to implement effectively in hardware.

I. Description of Technology

Billions of microcontrollers appear in embedded systems ranging from thermostats and utility meters to tollway payment transponders and pacemakers. Modern microcontrollers often include 128 bytes to 10 Kbytes of on-chip flash memory. While reliability, low cost, and high storage density of flash memory make it a natural choice for embedded systems, its *high voltage requirement* introduces challenges for energy-efficient designs aiming to maximize the system's effective lifetime on a finite source of energy. Writing to flash memory is an energy-intensive operation because of the fundamental behavior of floating gate transistors. An ideal microcontroller would consume energy that varies linearly with the number of flash writes in a workload. **State of the art microcontrollers of this variety require the entire chip to operate on a supply voltage that is significantly higher than what the CPU portion of the microcontroller requires.** This large gap results in wasted energy with diminishing returns on reliability. Present-day technology such as low-power hardware or fast error correction codes do not address this wasted energy. The wasted energy is a result of a mismatch between the reliability needed by software and the reliability provided by the hardware. That is, software could reliably run on significantly less energy by dynamically tuning the storage algorithms. Microcontroller hardware presently over-provisions with wasted energy because the instruction set architectures provide no flexibility for firmware to set reliability parameters to match the precision of data for storage (e.g., low-precision sensor data). SMASH allows application-specific firmware in microcontroller-based products to **automatically set ideal parameters for storage reliability and energy consumption.**

Our SMASH approach reduces the operating voltage of the microcontroller to a point at which the resulting energy savings of the CPU portion of the workload exceeds the energy cost of the algorithms for ensuring reliable writes. In practice, we find **energy savings of 34% not unusual**. The technique requires minimal or no hardware modification and also allows for RFID-scale devices to better exploit capacitors as power supplies. The capacitor provides finite energy and therefore the voltage decays exponentially. The long tail of the curve provides insufficient voltage for conventional writes to flash memory, but is sufficient for reliable storage with our techniques.

A. Why software? It's more end-to-end.

Our software approach allows the low-power device to operate at a lower voltage than required by flash memory, which results in saved energy in cases where the workload involves more computations than flash writes. In addition, our methods self-tune to any microcontroller so long as it contains on-chip flash memory.

Today's devices are limited by battery longevity or exclude flash memory access to reduce the power consumption. It would be difficult to implement SMASH-equivalent techniques in hardware because of the need to redesign major interfaces such as the Instruction Set Architecture (ISA). Our techniques require little or no modification of hardware, and yet results in significantly reduced energy consumption. Energy savings are as simple as a software update to microcontroller firmware. Moreover, the update can work on binary object code. Source code is helpful, but not required to apply our techniques.

B. Advantages over existing technologies

The main commercial uses of our *SMASH* technology are to (1) extend the effective lifetime of low-power devices by lowering their operating voltage and to (2) allow removal of unnecessary circuitry that is redundant with our more efficient software.

SMASH can improve battery life of a product even in the absence of source code. Because our technique requires only a small number of changes to a subroutine, SMASH consumes only a few dozen bytes of space. Essentially, calls to instructions that write to flash are replaced with subroutine function calls that preserve the size of the main program.

SMASH makes expensive hardware unnecessary. Today, devices add expensive voltage regulators that have excessive quiescent current draw in order to satisfy manufacturer specifications for reliable flash memory. Some devices will add additional circuitry (a few dozen components) to implement discrete voltage scaling. By allowing flash memory to work reliably at low-voltage, these voltage conditioning circuits can be either removed entirely or replaced with cheaper components. The removal leads to both cost savings of manufacturing and increased battery lifetime for the consumer experience.

II. How to incorporate SMASH memory into products

After profiling the power consumption of a product to determine certain parameters, a manufacturer would apply a software update to their firmware. While source code from the product would be helpful, binary splicing would be feasible depending on various programming models (e.g., use of reentrant code). A typical embedded system would then choose a more efficient voltage regulator at a lower voltage. One could also use a less precise regulator to further reduce costs. Some platforms (e.g., batteryless or energy harvesting platforms) may be able to remove additional circuitry that is redundant with the software that can more efficiently provide tunable, end-to-end reliability.

Our preliminary study on a popular microcontroller shows that increasing the operating voltage from 1.8V to 2.2V causes the CPU to draw about 50% more power (without commensurate gain in clock speed) because of the voltage squaring effect. The drawback of lowering voltage below flash memory requirements in order to save power is the reduction of flash memory reliability. Fortunately, our methods mask the deficiencies of flash memory to enable reliable writes while coping with low voltage.

III. Products envisioned

We envision the SMASH technology will have most impact on wireless consumer electronics, battery-powered devices, low-power devices, and energy-harvesting devices. Key to the products are a sensitivity to battery life (a carbon monoxide battery that drains quickly can contribute to tragedies) and a sensitivity to manufacturing cost (high volume markets may extremely tight profit margins per unit).

Battery powered devices: TV remote controls, video game controllers, carbon monoxide detectors, smoke detectors, digital picture frames, pacemakers, thermostats, power/gas/water meters, remote entry car keys, watches/clocks, hotel room locks, handheld credit card terminals, touchscreens, keyboard controllers, wireless keyboards, wireless mice, wireless touchpads, medical monitoring systems.

Energy-harvesting devices powered from capacitors: energy harvesting for recharging consumer electronics, programmable RFIDs, contactless credit cards, electronic passports, tollway payment transponders (e.g., EZPass), in-store anti-theft tags that squawk.

IV. Profile of typical licensee

Thousands of companies develop microcontroller-based products that are cost and energy sensitive. We are currently evaluating SMASH memory on a product at the request of a consumer electronics company. The product presently runs at high voltage because of occasional writes to flash memory embedded in a microcontroller, but the flash memory is rarely used—the precise condition that makes SMASH memory of highest value to a manufacturer. SMASH memory allows the energy consumption to better match the CPU performance rather than the much worse flash memory performance. Moreover, SMASH memory reduces the cost, size, weight, and quantity of components necessary to achieve reliable operation.

V. Intellectual property

This technology is patent pending.